DISS. ETH NO. XX

Eco-evolutionary processes in ecological and economic systems

Confronting dynamical models and data

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

VICTOR BOUSSANGE

M.Sc. Energy and environmental sciences, Institut National des Sciences Appliquées de Lyon
born January 10th, 1995
citizen of Bordeaux, France

accepted on the recommendation of

Prof. Dr. Loïc Pellissier, (doctoral thesis supervisor), examiner
Prof. Dr. Didier Sornette, co-examiner
Prof. Dr. Arnulf Jentzen, co-examiner
Prof. Dr. Samir Suweis, co-examiner

2022

Eco-evolutionary processes in ecological and economic systems

Confronting dynamical models and data

Victor Boussange

July 26, 2022
Version: 0.0.1

Victor Boussange

Eco-evolutionary processes in ecological and economic systems

Confronting dynamical models and data

July 26, 2022

Reviewers: Didier Sornette Samir Suweisand Arnulf Jentzen

Supervisors: Loïc Pellissier

ETH Zürich

Ecology Landscape and Evolution

Institute of Terrestrial Ecosystems

Department of Environemtal Sciences

Universitätstrasse 18

8055 and Zürich

Summary

- Since life emerged 4 billion years ago, its complexity has evolved.
  - Complex Systems (CS) are generally defined as a category of dynamical systems composed of many individual entities, be they biological, socio-cultural or economic, spatially organised and interacting locally in a nonlinear way. The adjective Adaptive is used to define CS which are subjected to evolutionary mechanisms (Levin [2002]). These include the biosphere, socio–cultural systems and economical systems. The agents adapt to local conditions and are subjected to selection processes acting at a macro level.
  - Recent development
    - in computing power and AI
    - Interdisciplinary science
    - opens up unprecedented scientific pathways to better understand the world surrounding us
- In this thesis, a novel framework for bridging mechanistic models of CAS and data is presented.
  - An eco-evolutionary model of interacting organisms is theoretically investigated
    - to understand the emergence of biodiversity in complex landscapes
  - A set of tools are developped to
  - Those tools are used to investigate the processes that drive the macroscopic dynanmics of economies across countries
- It is shown that bridging theoretical models and data deepens our current understanding of processes and can brings a new perspective.
  - Bridging disciplines provides a remarkably clear understanding of universal mechanisms that have shaped the economics dynamics.

- This thesis moves beyond the dichotomy between theoretical and data science approaches and provides a novel framework for formalizing and exploring multiple hypotheses and reconstructions associated with the processes that drive CAS. Model comparision with empirical data serves as hindcast, which might inform evolutionary trajectories. By advancing our understanding on the processes that dictate the dynamics of CAS, we can better anticipate the radical changes that we will face in the next decades
    - the thesis provides answer to the elusive hypothesis that eco-evolutionary processes can be found in economics

Résumé

- Same as above, but in french

Acknowledgement

I thank my parents, my sister, my friends, and my beloved Flora.

Contents

1 Introduction

> *Le bout du monde et le fond du jardin*
> *contiennent la même quantité de merveilles.*
>
> — **Christian Bobin**
> (French poet)

1.1 Context of modelling

- human curiosity: build models of what surrounds us
  - Nature has been fascinating since the beginning of humankind.
    - Human's curiosity lead us to propose models capturing our belief on how things work. Those were mainly conceptual models.
  - The scientific method started with the Enlightment, during the 18th century (Holmes, 1997).
    - Isaac Newton became the revered founder of modern Mechanics due to his intuition, gathered by empirical evidence, about a possible mathematical formalization for the law of universal gravitation. [**Equations2021**]
    - 20th century: logical empiricism dominated the philosophy of science and sceitnsts searched for fundamental theoretical principles explaing the laws of nature, with physics at the central stage (Okashaa 2002)
    - in biology, natural history mainly (classification of living organisms without further questioning).
      - why is it that we better understand the motion of planets, or the surface of the moon, than e.g. the mechanisms that drive our fingers? (REF)

        - biological world poses obstacles in finding laws: nonlinear processes and complexity of processes and spatial and time scales

    - The mathematicalisation of soft science was driven by inspiration from physics

        * From the 1700s on, nature has been increas-ingly described by mathematical equations,with differential or difference equationsforming the basic framework for describingdynamics. The use of mathematical equationsfor ecological systems came much later, pio-neered by Lotka and Volterra, who showedthat population cycles might be described interms of simple coupled nonlinear differentialequations. It took decades for LotkaVolterra-type models to become established, but thedevelopment of appropriate differential equa-tions is now routine in modeling ecologicaldynamics. There is no question that the in-jection of mathematical equations, by forcingclarity and precision into conjecture(2), hasled to increased understanding of populationand community dynamics. As in sciencein general, in ecology equations are a keymethod of communication and of framinghypotheses. These equations serve as compactrepresentations of an enormous amount ofempirical data and can be analyzed by thepowerful methods of mathematics [0].

    - AI era

    - Scientific revolution of Darwin thinking, by Kuhn's definition (Dawkins 2010)

    - Universal Darwninism

- more than a curiosity : a necessity

    - Approaching a state shift in Earth biosphere: [0]

- Bridge those models with data

1.2 Complex adaptive systems

- Ecological and economic systems are complex adaptive systems (CAS): they are systems that are composed of many entities with heterogeneous characteristics, which interact and experience selection processes. Those processes act at the

individual level, but are key in determining the macroscopic behaviour at the system level, a feature that make those systems unique.

- Complex interconnected systems pose a major challeng to scientific study in ecology and economics [0] (and references therin).
  - the common approach of reducing these systems to linearly independent components overlooks important interactions for the sake of computational tractability
  - statistical frameworks (e.g., PCA, GLM, multivariate autoregressive models), assume that caysal factors do not interact with each other and have independent or additive effects on a response variable,
    - simplification leads to probelms in identifying associations (refs 5-6 of [0])
    - cannot predict out-of sample behaviour
  - complex equation-based modeol explicitly accounting for each interaction have great intuitive appeal
    - but those models suffer from their many parameters to be precisely determined given the available data (curse of dimensionality (ref 9 [0]))
    - problem is amplified because in biological fields the relevant units may not behave according to the fundamental equations.

Biological systems

- Biodiversity results from a hierarchy of processes acting at different scales of time and space. Variations experienced by organisms, their interactions between them and with the environment, and selection pressure acting upon groups of organisms are of particular relevance for explaining differences in species richness at the ecosystem levels.
  - The synthetic theory of evolution (see e.g. Gayon 2003): with genetics (Mendel) and DNA (James Watson and Francis Crick)
  - *Nothing in biology makes sense except in the light of evolution* (Dobhansky 1973)

- explanation for the main principles underlying the emergence of biodiversity: mutliple processes that interact at different scale in space and time
  - allopatric speciation
  - ecological speciation
  - dispersal
  - adaptation
  - those processes interact simulatneously withing the surrounding environment
- Traits: measurable characteristics that reflect and shape evolutionary history (Darwin 1859). Natural selection promotes the evolution of traits thatoptimize species survival under specific environmental conditions..

Economic systems

- The economic trajectory of a country is greatly affected by the ensemble of economic actors and their interactions, that structure its economy. Firms are adaptive entities that respond to the environment in which they operate according to their characteristics, that vary over time. By interacting together and experiencing selection pressure, they determine economic growth at the country level.
  - **Universal Darwinism**

Research questions

- Despite the intrinsic variability of the entities that compose them, and despite the complexity of the processes driving their dynamics, regularities at the macroscopic level emerge in ecological and economic systems. This is the case of large-scale spatial patterns of biodiversity and differences in economic growth across countries, calling for a mechanistic understanding of the essential mechanisms that generate them.
  - Multiple arrangements of parts that result in a complex set of effects in a system are defined as mechanisms (Dawkins 2010)

1.3 Eco evolutionary processes

- Eco-evolutionary processes and analogous economic processes acting upon firms have been proposed to play a major role in the emergence of macroscopic patterns in ecological and economic systems. Nonetheless, a quantitative investigation of their importance is missing.
  - The interplay between ecological processes, the processes that regulate interactions between organisms, and evolutionary processes, the change of the characteristics of biological populations over time, has recently received increasing attention to explain current biodiversity patterns.
  - Analogous economic processes have been proposed to explain differences in economic growth across nations.
- A quantitative investigation of how those patterns can emerge from eco-evolutionary processes is required to improve our current understanding and generate a parsimonious theory with predictive power. This defines the goal of this project, which undertakes this investigation through a unique approach that consists in confronting quantitative eco-evolutionary models to empirical data.

1.4 Models and challenges

- Eco-evolutionary models are complicated and necessitate the use of computers to be simulated and analysed against data. This poses a number of methodological challenges that we adress in the first part of this project.
  - Entities in CAS have distinct quantitative attributes that determine their fitness in a given environment. Accounting for the variety of these characteristics leads to models with high dimensionality, associated to a high if not prohibitive computational cost preventing its simulation.
    * The model zoo
      · Agent Based model: hard to scale up
      · PDE: hard to scale up

        - In particular, partial differential equation (PDE) models, which can encode eco-evolutionary processes acting upon entities defined by many characteristics, are cursed by their dimensionality.

        - Machine learning: scale up

    * To this aim, we develop machine learning algorithms that break down the curse of dimensionality by relying on neural networks to approximate the solution to PDE models.

  – An other difficulty consists in confronting eco-evolutionary models with data, since those models cannot be manipulated by standard statistical techniques.

    * We apply methods commonly employed in the training of neural networks, together with model selection techniques, to infer from candidate models fundamental mechanisms that characterise the patterns under investigation.

- The machine learning approximations that we develop allow for efficient model simulations, that we combine with training techniques and model selection methods to explore the motivated research question.

Here you could add a coneptual figure, similar to Florian Patout (see evernote), that shows the interplay between selection and variation.

1.5 Machine learning : opportuntities

- State of the art machine learning techniques have yielded transformative results across divers scientific disciplines [REF], but rely on a large amount of data [REF], while environmental sciences rely in a small data regime where those techniques are typically not suited [**Raissi2019a**]. Recently, physics informed machine learning has emerged as a tool to constrain fully parametric methods with scientific knowledge, for data efficiency and extrapolation

[**Raissi2019a**]. The key idea is to refine the learning with scientific knowledge by adding additional constraints in the objective function, given by ODEs/ PDEs models.

- [0]

1.6 Learning from models

- we develop quantitative models that embed general eco-evolutionary processes, and test them against data to explore hypotheses on the fundamental mechanisms that drive patterns of biodiversity and economic growth.
  - From one hand, we explore how eco-evolutionary processes, in combination with complex landscape topologies, can explain patterns of species diversity.
  - To this aim, we develop and analyse an eco-evolutionary model on spatial graphs, to understand how the combination of eco-evolutionary processes and complex landscapes might have shaped biodiversity patterns that are found in complex landscapes such as mountain regions.
  - On the other hand, we investigate how eco-evolutionary processes can provide new insights in the understanding of economic dynamics.
  - We proceed by developing a simple eco-evolutionary model which explanatory power we test against long time series that capture the dynamics of asset size of economic sectors.
- Overall, this project is a step towards providing a useful conceptualisation of fundamental eco-evolutionary mechanisms that shape the features of the world that surrounds us.

1.7 Thesis Structure

Part ??

An eco-evolutionary model on spatial graphs

It is not clear how landscape connectivity and habitat heterogeneity influence differentiation in biological populations. To obtain a mechanistic understanding of underlying processes, we construct an individual-based model that accounts

for eco-evolutionary and spatial dynamics over graphs. Individuals possess both neutral and adaptive traits, whose co-evolution results in differentiation at the population level. In agreement with empirical studies, we show that characteristic length, heterogeneity in degree and habitat assortativity drive differentiation. By using analytical tools that permit a macroscopic description of the dynamics, we further link differentiation patterns to the mechanisms that generate them. This part provides support for a mechanistic understanding of how landscape features affect diversification.

Part ??
Scientific machine learning for eco-evolutionary modelling

It is a daunting task to obtain an agreement between mechanistic models and real world systems. In particular, there is a need to account for the dimensionality of the evolutionary and spatial structures over which agents interact and evolve. Furthermore, the calibration of such models is difficult. To adress the difficulties that arise due to the dimensionality of models, we develop two numerical methods to solve high-dimensional non-local nonlinear PDES that arise in eco-evolutionary models. We implement those methods in a software, `HighDimPDE.jl`, that integrates within an open source ecosystem for Scientific Machine Learning in the Julia programming language. We further present a scheme to estimate the parameters of a mechanistic model from empirical data sets. We show with analytical arguments that the use of different shallow time series allows for a better estimation than a unique, possibly deeper time series. This part provides ready-to-use modeling tools to adress the intrinsic complexity of complex adaptive systems.

Part ??
Briding eco-evolutionary models and data

Despite evidences that alike biological systems, economic systems are complex adaptive systems that continuously adapt and experience evolutionary processes, economists have discarded biological models and have rather relied on mechanistic models inspired from physics. Building upon an analogy between economic sectors and biological functional groups, we use a biological model to quantitatively investigate whether eco-evolutionary processes characterise the dynamics of economic sectors. Overall, we find that interactions across economic sectors, evolution of new economic sectors, and international transfers play a major role in the dynamics of economic sectors at the national level. The significance and the strength of such processes strongly vary across countries and correlate with standard macroeconomic indices such as the Economic Complexity Index. We relate such patterns to documented patterns in ecology and evolution. This part provides a new perspective on the understanding of the dynamics of economic systems.

2

Eco-evolutionary model on spatial graphs reveals how habitat structure affects phenotypic differentiation

by Victor Boussange[1,2] and Loïc Pellissier[1,2]

[1] Swiss Federal Research Institute WSL, CH-8903 Birmensdorf, Switzerland

[2] Landscape Ecology, Institute of Terrestrial Ecosystems, Department of Environmental System Science, ETH Zürich, CH-8092 Zürich, Switzerland

Published in Communications Biology (2022)

doi:10.1038/s42003-022-03595-3

Differentiation mechanisms are influenced by the properties of the landscape over which individuals interact, disperse and evolve. Here, we investigate how habitat connectivity and habitat heterogeneity affect phenotypic differentiation by formulating a stochastic eco-evolutionary model where individuals are structured over a spatial graph. We combine analytical insights into the eco-evolutionary dynamics with numerical simulations to understand how the graph topology and the spatial distribution of habitat types affect differentiation. We show that not only low connectivity but also heterogeneity in connectivity promotes neutral differentiation, due to increased competition in highly connected vertices. Habitat assortativity, a measure of habitat spatial auto-correlation in graphs, additionally drives differentiation under habitat-dependent selection. While assortative graphs systematically amplify adaptive differentiation, they can foster or depress neutral differentiation depending on the migration regime. By formalising the eco-evolutionary and spatial dynamics of biological populations on graphs, our study establishes fundamental links between landscape features and phenotypic differentiation.

2.1 Introduction

Biodiversity results from differentiation processes influenced by the features of the landscape over which populations are distributed [30]. The documentation of high levels of species diversity in mountain regions and riverine systems suggests that complex connectivity patterns and habitat heterogeneity foster differentiation [57, 19, 17, 27]. However, hypotheses formulated based on empirical evidence should be complemented by mechanistic models to crystallise a causal understanding between processes and patterns [38]. While the number of simulation studies is growing steadily [10], such studies often lack a mathematical formalism to facilitate the interpretation of the model outcomes by providing an analytical underpinning to the simulation results [39].

Phenotypic differentiation processes emerge as a result of mutation, selection and migration and can be classified as neutral or adaptive [28]. Neutral differentiation is initiated by the stochastic drift of local phenotypes when spatial isolation and limited dispersal create barriers to gene flow, allowing distinct phenotypes to emerge in spatially structured populations [60]. In contrast, adaptive differentiation results from heterogeneous selection, which promotes distinct, locally well-adapted phenotypes in populations occupying patches with different habitat conditions [18]. The evolution of neutral phenotypes and of adaptive phenotypes are not independent, as selective forces can indirectly select for those neutral phenotypes that happen to be linked to the fittest adaptive phenotypes, a mechanism called the hitchhiking effect [31]. Moreover, selection can generate barriers to gene flow between populations in heterogeneous habitat landscapes [50, 67], a phenomenon coined isolation by environment, which can amplify neutral differentiation. How neutral processes, adaptive processes and their interplay are affected by landscape features is difficult to comprehend without a formalised mechanistic model [23].

Models link patterns to processes [38], and the explicit representation of the landscape within an eco-evolutionary model can lead to a causal understanding of how landscape features shape differentiation. Spatial graphs provide a convenient mathematical representation of landscapes, where vertices represent suitable habitats hosting populations, and edges capture the connectivity between habitats [14]. Under ecological dynamics, metapopulation models have been used to study the role of graph topology in the persistence and stability of metapopulation [29, 24, 42, 26] and community diversity [11, 64, 63]. Evolutionary mechanisms are nevertheless fundamental drivers of diversity, and should therefore be explicitly integrated into models [51]. Evolutionary game theory explores how graph topology impacts the fixation probability and the fixation time of a mutated phenotype [65]. However, the framework does not consider the continuous accumulation of mutations, and is therefore not suited to addressing the emergence of phenotypic differentiation. By combining a metapopulation model with a model of neutral evolution, [22, 21] investigated how graph topology affects neutral diversity. Their approach demonstrated the key role of topological properties in shaping diversity, and its predictions could be matched with empirical data from e.g. river basins [47]. Nonetheless, diversity results from the combination of neutral and adaptive processes developing at the population level. A first principles modelling

approach considering spatial graphs, but also building upon the elementary processes of ecological interactions, reproduction, mutation and migration may therefore be promising to investigate the emergence of diversity.

Stochastic models for structured populations, rooted in the microscopic description of individuals, offer a generic framework for modelling eco-evolutionary dynamics [12, 5]. In particular, these models can capture the interplay between population dynamics, spatial dynamics and phenotypic evolution, while providing a rigorous set-up for analytical investigation. By anchoring this modelling paradigm in a mathematical framework, the work of Champagnat et al. [12] generalises models of population genetics [9] (investigating the evolution of the frequencies of alleles) and quantitative genetics [61, 35, 48] (investigating the evolution of phenotypic traits), which stimulated research into the link between spatial population structure and neutral differentiation. The framework embraces density-dependent selection, which could explain the emergence of phenotypic differentiation from competition processes [18], and how spatial segregation can emerge as a byproduct of these adaptive processes along environmental gradients [20]. Related models have addressed the effects of landscape dynamics and habitat heterogeneity on adaptive differentiation, providing mathematical insights into the dynamics [45, 1, 16, 69, 53, 46]. Because it accounts for finite population size, the baseline model of Champagnat et al. [12] can also capture neutral differentiation dynamics and therefore the coupling between neutral and adaptive processes [6, 3]. Nonetheless, the aforementioned studies were not spatially explicit [6, 3] or they assumed regular spatial structures (regular graphs [45, 1, 16, 46] or continuous space [20, 69, 53]), therefore not addressing the role of the spatial complexity of landscapes. A stochastic individual-based model using spatial graphs as a representation of the landscape could help formalise fundamental links between landscape features and phenotypic differentiation.

A key challenge is to understand how individual dynamics result in the emergence of differentiation in complex landscapes [41]. Here, we investigate how complex connectivity patterns and habitat heterogeneity affect both neutral and adaptive phenotypic differentiation by constructing an individual-based model (IBM) that accounts for eco-evolutionary dynamics on spatial graphs. The individuals disperse between habitat patches and possess co-evolving neutral and adaptive traits. The finite size of local populations generates neutral differentiation by inducing a stochastic drift in the neutral trait evolution, while heterogeneous selection gives rise to adaptive differentiation. Macroscopic properties of the model are analytically tractable, and we obtain a deterministic approximation of population size and adaptive trait dynamics which connects the emerging patterns to the graph properties that generate them. However, neutral differentiation is stochastic by nature, which complicates its analytical underpinning. We therefore rely on numerical simulations of the IBM to measure the effect of graph topology on neutral differentiation. In the case where heterogeneous selection is absent, we investigate how graph topology affects neutral differentiation. In the case of heterogeneous selection, we investigate how the graph topology, in combination with the spatial distribution of habitat types, affects levels of (i) adaptive and (ii) neutral differentiation. By combining analytical methods with numerical simulations, we expect to identify graph properties that determine the level of differentiation. Overall, our study establishes

causal links between landscape properties and population differentiation and contributes to a fundamental understanding of how landscape features promote biodiversity.

2.2 Results

2.2.1 Eco-evolutionary model on spatial graphs

We establish an individual-based model (IBM) where individuals are structured over a trait space and a graph representing a landscape. For the sake of simplicity, we consider the case of asexual reproduction and haploid genetics [12]. Individuals die, reproduce, mutate and migrate in a stochastic fashion, which together results in macroscopic properties. The formulation of the stochastic IBM allows an analytical description of the dynamics at the population level, which links emergent properties to the elementary processes that generate them.

The trait space $\mathcal{X} \subseteq \mathbb{R}^d$ is continuous and can be split into a neutral trait space $\mathcal{U}$ and an adaptive trait space $\mathcal{S}$. We refer to neutral traits $u \in \mathcal{U}$ as traits that are not under selection, in contrast to adaptive traits $s \in \mathcal{S}$, which experience selection. The graph denoted by G is composed of a set of vertices $\{v_1, v_2, \ldots, v_M\}$ that correspond to habitat patches (suitable geographical areas), and a set of edges that constrain the movement of individuals between the habitat patches. We use the original measure of genetic differentiation for quantitative traits Q_{ST} (standing for Q-statistics) in the case of haploid populations [36, 68]. We denote the neutral trait value of the k-th individual on v_i as $u_k^{(i)}$, the number of individuals on v_i as $N^{(i)}$, the mean neutral trait on v_i as $\bar{u}^{(i)}$, and the mean neutral trait in the metapopulation as $\bar{u}$. It follows that we quantify neutral differentiation $Q_{ST,u}$ as

$$Q_{ST,u} = {}^{\sigma^2_{B,u}}/_{(\sigma^2_{B,u}+\sigma^2_{W,u})} \tag{2.1}$$

where $\sigma^2_{B,u} = \mathbb{E}\left[\frac{1}{M}\sum_i \left(\bar{u}^{(i)} - \bar{u}\right)^2\right]$ denotes the expected neutral trait variance between the vertices and $\sigma^2_{W,u} = \frac{1}{M}\sum_i^M \mathbb{E}\left[\frac{1}{N^{(i)}}\sum_k \left(u_k^{(i)} - \bar{u}^{(i)}\right)^2\right]$ denotes the average expected neutral trait variance within vertices. We similarly quantify adaptive differentiation $Q_{ST,s}$.

Following the Gillespie update rule [25], individuals with trait $x_k \in \mathcal{X}$ on vertex v_i are randomly selected to give birth at rate $b^{(i)}(x_k)$ and die at rate $d(N^{(i)}) = {}^{N^{(i)}}/_K$, where K is the local carrying capacity. The definition of d therefore captures competition, which is proportional to the number of individuals on a vertex and does not depend on the individuals' traits (we relax this assumption later on). The offspring resulting from a birth event inherits the parental traits, which can independently be affected by mutations with probability μ. A mutated trait differs from the parental trait by a random change that follows a normal distribution with variance σ^2_μ (corresponding to the continuum of alleles model [32]). The offspring can further migrate to neighbouring vertices by executing a simple random walk on G with probability m. A schematic overview of the two different settings

considered is provided in Fig. 2.1. Under the setting with no selection, individuals are only characterised by neutral traits so that $\mathcal{X} = \mathcal{U}$. For individuals on a vertex with trait $x_k \equiv u_k$ we define $b^{(i)}(x_k) \equiv b$, so that the birth rate is constant. This ensures that neutral traits do not provide any selective advantage. Under the setting with heterogeneous selection, each vertex of the graph v_i is labelled by a habitat type with environmental condition Θ_i that specifies the optimal adaptive trait value on v_i. It follows that, for individuals with traits $x_k = (u_k, s_k) \in \mathcal{U} \times \mathcal{S}$ on v_i, we define

$$b^{(i)}(x_k) \equiv b^{(i)}(s_k) = b(1 - p(s_k - \Theta_i)^2) \tag{2.2}$$

where p is the selection strength [46]. This ensures that the maximum birth rate on v_i is attained for $s_k = \Theta_i$, which results in a differential advantage that acts as an evolutionary stabilising force. In the following we consider two habitat types denoted by $\bullet$ and $\bullet$ with symmetric environmental conditions $\theta_\bullet$ and $\theta_\bullet$, so that $\Theta_i \in \{\theta_\bullet, \theta_\bullet\}$ and $\theta_\bullet = -\theta_\bullet = \theta$, where θ can be viewed as the habitat heterogeneity [46].

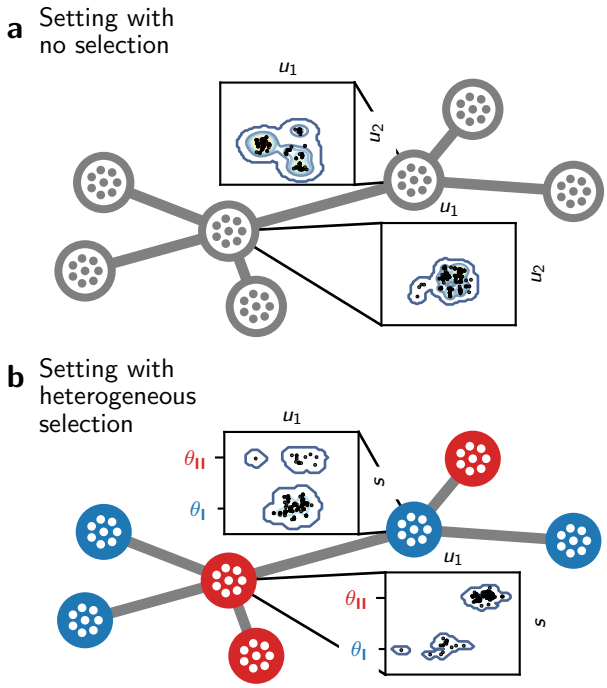

Fig. 2.1: Graphical representation of the structure of individuals in the eco-evolutionary model. (a) Setting with no selection, where individuals are characterised by a set of neutral traits $u \in \mathcal{U}$. The scatter plots represent a projection of the first two components of u for the individuals present on the designated vertices at time $t = 1000$, obtained from one simulation of the IBM.
(b) Setting with heterogeneous selection. In this setting, individuals are additionally characterised by adaptive traits $s \in \mathcal{S}$. Blue vertices favour the optimal adaptive trait value $\theta_{\bullet}$, while red vertices favour $\theta_{\bullet}$. The scatter plots represent a projection of the first component of u and s for the individuals present on the designated vertices at time $t = 1000$, obtained from one simulation. The majority of individuals are locally well-adapted and have an adaptive trait close to the optimal value, but some maladaptive individuals originating from neighbouring vertices are also present. $m = 0.05$.

2.2.2 Deterministic approximation of the population dynamics under no selection

The model can be formulated as a measure-valued point process ([5] and Supplementary Information). Under this formalism, we demonstrate in the Supplementary Information how the population size and the trait dynamics show a deterministic behaviour when a stabilising force dampens the stochastic fluctuations. This makes it possible to express the dynamics of the macroscopic properties with deterministic differential equations, connecting emergent patterns to the processes that generate them. In particular, in the setting of no selection,

competition stabilises the population size fluctuations, and the dynamics can be considered deterministic and expressed as

$$\partial_t N_t^{(i)} = N_t^{(i)} \left[b(1-m) - \frac{N_t^{(i)}}{K} \right] + mb \sum_{j \neq i} \frac{a_{i,j}}{d_j} N_t^{(j)} \tag{2.3}$$

where $A = (a_{i,j})_{1 \leq i,j \leq M}$ is the adjacency matrix of the graph G and $D = (d_1, d_2, \ldots, d_M)$ is a vector containing the degree of each vertex (number of edges incident to the vertex). The first term on the right-hand side corresponds to logistic growth, which accounts for birth and death events of non-migrating individuals. The second term captures the gains due to migrations, which depend on the graph topology. Assuming that all vertices with the same degree have an equivalent position on the graph, corresponding to a mean field approach (see Machine learning framework for ecosystem models), one can obtain a closed-form solution from Eq. (2.3) (see Eq. (2.12)), which shows that the average population size $\bar{N}$ scales with $\langle\sqrt{k}\rangle^2/\langle k\rangle$, where $\langle k\rangle$ is the average vertex degree and $\langle\sqrt{k}\rangle$ is the average square-rooted vertex degree. The quantity $\langle\sqrt{k}\rangle^2/\langle k\rangle$, denoted as h_d, relates to the homogeneity in vertex degree of the graph and can therefore be viewed as a measure negatively associated with heterogeneity in connectivity. Simulations of the IBM illustrate that h_d can explain differences in population size for complex graph topologies with varying migration regimes (Fig. 2.2a for graphs with $M = 7$ vertices and Fig. S1a for $M = 9$). This analytical result is connected to theoretical work on reaction diffusion processes [13] and highlights that irregular graphs (graphs whose vertices do not have the same degree) result in unbalanced migration fluxes that affect the ecological balance between births and deaths. Highly connected vertices present an oversaturated carrying capacity ($N^{(i)} > bK$, see Machine learning framework for ecosystem models), increasing local competition and lowering total population size compared with regular graphs (Fig. 2.2a). Because populations with small sizes experience more drift ([9] and Fig. S2), this result indicates that graph topology affects neutral differentiation not only through population isolation, but also by affecting population dynamics.

Nonetheless, the stochasticity of the processes at the individual level can propagate to the population level and substantially affect the macroscopic properties. In particular, neutral differentiation emerges from the stochastic fluctuations of the populations' neutral trait distribution. These fluctuations complicate an analytical underpinning of the dynamics, and in this case simulations of the IBM offer a straightforward approach to evaluate the level of neutral differentiation.

2.2.3 Effect of graph topology on neutral differentiation under no selection

We study a setting with no selection and investigate the effect of the graph topology on neutral differentiation. When migration is limited, individuals' traits are coherent on each vertex but stochastic drift at the population level generates neutral differentiation between

the vertices. Migration attenuates neutral differentiation because it has a correlative effect on local trait distributions. Following [22, 11, 64], we expect that the intensity of the correlative effect depends on the average path length of the graph $\langle l \rangle$, defined as the average shortest path between all pairs of vertices [7]. For a constant number of vertices, $\langle l \rangle$ is strictly related to the mean betweenness centrality and quantifies the graph connectivity [7]. High $\langle l \rangle$ implies low connectivity and a greater isolation of populations, and hence we expect that graphs with high $\langle l \rangle$ are associated with high differentiation levels. We consider various graphs with an identical number of vertices and run simulations of the IBM to obtain the neutral differentiation level $Q_{ST,u}$ attained after a time long enough to discard transient dynamics (see Machine learning framework for ecosystem models). We then interpret the discrepancies in $Q_{ST,u}$ across the simulations by relating them to the underlying graph topologies.

We observe strong differences in $Q_{ST,u}$ across graphs for varying m, and find that $\langle l \rangle$ explains at least 55% of the variation in $Q_{ST,u}$ across all graphs with $M = 7$ vertices for (Fig. 2.2b). Nonetheless, some specific graphs, such as the star graph, present higher levels of $Q_{ST,u}$ than expected by their average path length. To explain this discrepancy, we explore the effect of homogeneity in vertex degree h_d, as we showed in Eq. (2.12) that it decreases population size, which should in turn increase $Q_{ST,u}$ by intensifying stochastic drift. We find that h_d explains 57% of the variation for low m (Fig. 2.2c). However, the fit remains similar after correcting for differences in population size (see Table S1), indicating that irregular graphs structurally amplify the isolation of populations. Unbalanced migration fluxes lead central vertices to host more individuals than allowed by their carrying capacity. This causes increased competition that results in a higher death rate, so that migrants have a lower probability of further spreading their trait. Highly connected vertices therefore behave as bottlenecks, increasing the isolation of peripheral vertices and consequently amplifying $Q_{ST,u}$.

We then evaluate the concurrent effect of $\langle l \rangle$ and h_d on $Q_{ST,u}$ with a multivariate regression model that we fit independently for low and high migration regimes (Fig. 2.2d). The multivariate regression model explains at least 70% of the variation in $Q_{ST,u}$ for the migration regimes considered and for graphs with $M = 7$ vertices (see Table S2 for details). Moreover, we find that $\langle l \rangle$ and h_d have akin contributions to neutral differentiation for low m, but the effect of $\langle l \rangle$ increases for higher migration regimes while the effect of h_d decreases. To ensure that these conclusions can be generalised to larger graphs, we conduct the same analysis on a subset of graphs with $M = 9$ vertices and find congruent results (Fig. S1). In the absence of selection and with competitive interactions, graphs with a high average path length $\langle l \rangle$ and low homogeneity in vertex degree h_d, or similarly graphs with low connectivity and high heterogeneity in connectivity, show high levels of neutral differentiation.

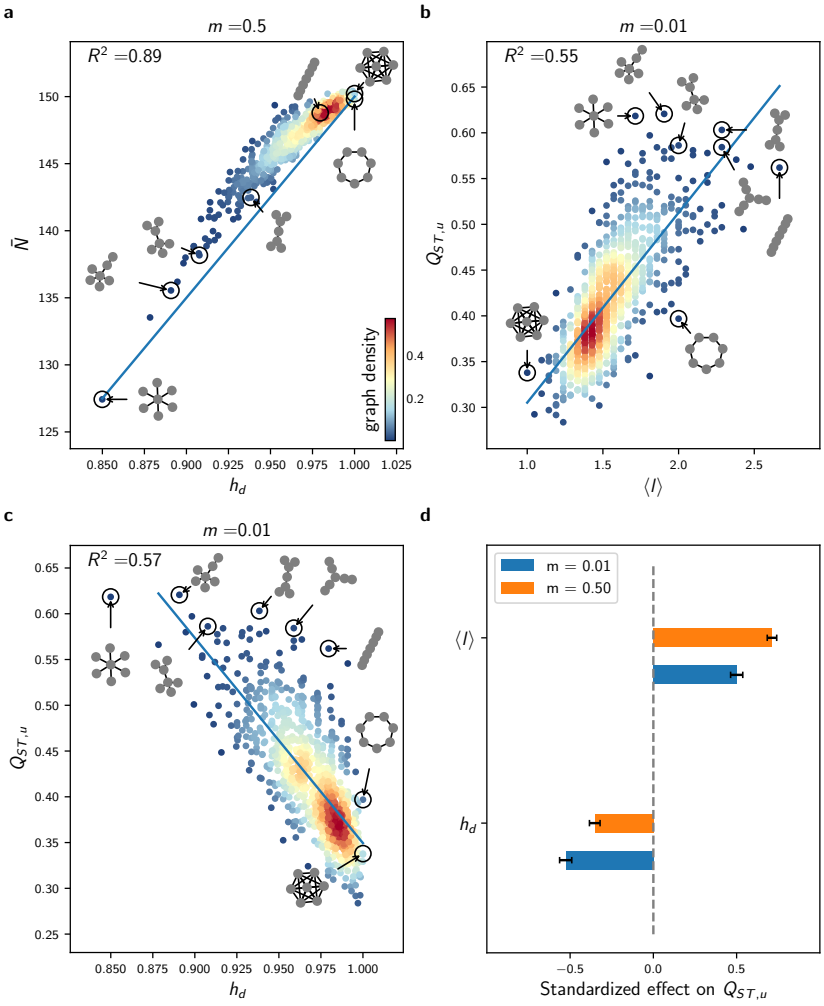

Fig. 2.2: Effect of $\langle l \rangle$ and h_d on average population size $\bar{N}$ and neutral differentiation $Q_{ST,u}$ in the setting with no selection. (a) Response of $\bar{N}$ to homogeneity in degree $h_d = \langle \sqrt{k} \rangle^2 / \langle k \rangle$ for all undirected connected graphs with $M = 7$ vertices and $m = 0.5$. (b) Response of $Q_{ST,u}$ to average path length $\langle l \rangle$ for similar simulations obtained with $m = 0.01$. (c) Response of $Q_{ST,u}$ to homogeneity in degree h_d for the same data. In (a), (b) and (c), each dot represents average results from 5 replicate simulations of the IBM, the colour scale corresponds to the proportion of the graphs with similar x and y axis values (graph density), and the blue line corresponds to a linear fit. (d) Standardized effect of h_d and $\langle l \rangle$ on $Q_{ST,u}$, obtained from multivariate regression models independently fitted on similar data obtained for $m = 0.01$ and $m = 0.5$. The contributions of $\langle l \rangle$ and h_d to $Q_{ST,u}$ are alike for low migration regimes. Error bars show 95% confidence intervals. Analogous results on graphs with $M = 9$ vertices are presented in Fig. S1 and all regression details can be found in Table S2.

2.2.4 Deterministic approximation of the population dynamics and adaptation under heterogeneous selection

We next consider heterogeneous selection and investigate the response of adaptive differentiation to the spatial distribution of habitat types, denoted as the Θ-spatial distribution. Adaptive differentiation emerges from local adaptation, but migration destabilises adaptation as a result of the influx of maladaptive migrants. We expect that higher connectivity between vertices of similar habitat type increases the level of adaptive differentiation, because it increases the proportion of well-adapted migrants. Local adaptation can be investigated by approximating the stochastic dynamics of the trait distribution with a deterministic partial differential equation (PDE). We demonstrate under mean field assumption how the deterministic approximation can be reduced to an equivalent two-habitat model. We analyse the reduced model with the theory of adaptive dynamics [45, 46] and find a critical migration threshold $m^\star$ that determines local adaptation. $m^\star$ depends on a quantity coined the habitat assortativity r_Θ, and we demonstrate with numerical simulations that r_Θ determines the overall adaptive differentiation level $Q_{ST,s}$ reached at steady state in the deterministic approximation.

Heterogeneous selection, captured by the dependence of the birth rate on Θ_i, generates a stabilising force that dampens the stochastic fluctuations of the adaptive trait distribution. The dynamics of the adaptive trait distribution consequently shows a deterministic behavior and we demonstrate in the Supplementary Information and Figs. S3 and S4 that the number of individuals on v_i with traits $s \in \Omega \subset \mathcal{S}$ can be approximated by the quantity $\int_\Omega n^{(i)}(s)ds$, where $n^{(i)}$ is a continuous function solution of the PDE

$$\partial_t n_t^{(i)}(s) = n_t^{(i)}(s)\left[b^{(i)}(s)(1-m) - \frac{1}{K}\int_{\mathcal{S}} n_t^{(i)}(\mathbf{s})d\mathbf{s}\right] + m\sum_{j\neq i} b_j(s)\frac{a_{i,j}}{d_j} n_t^{(j)}(s) + \tfrac{1}{2}\mu\sigma_\mu^2\Delta_s\left[b^{(i)}(s)n_t^{(i)}(s)\right] \tag{2.4}$$

Equation (2.4) is similar to Eq. (2.3), except that it incorporates an additional term corresponding to mutation processes and that the birth rate is trait dependent. We show how Eq. (2.4) can be reduced to an equivalent two-habitat model under mean field assumption. The mean field approach differs slightly from the setting with no selection because vertices are labelled with Θ_i. Here we assume that vertices with similar habitat types have an equivalent position on the graph (see Fig. S5 for a graphical representation), so that all vertices with habitat type $\bullet$ are characterised by the identical adaptive trait distribution that we denote by $\bar{n}^\bullet$, and are associated with the birth rate $b^\bullet(s) = b(1-p(s-\theta_\bullet)^2)$. Let $P(\bullet,\bullet)$ denote the proportion of edges connecting a vertex v_i of type $\bullet$ to a vertex v_j of type $\bullet$, and let $P(\bullet)$ denote the proportion of vertices v_i of type $\bullet$. By further assuming that habitats

are homogeneously distributed on the graph so that $P(\bullet) = P(\bullet) = \frac{1}{2}$, Eq. (2.4) transforms into

$$\begin{aligned}\partial_t \bar{n}_t^{\bullet}(s) = \bar{n}_t^{\bullet}(s)\left[b^{\bullet}(s)(1-m) - \frac{1}{K}\int_{\mathcal{S}} \bar{n}_t^{\bullet}(\mathbf{s})d\mathbf{s}\right] + \tfrac{1}{2}\mu\sigma_{\mu}^2(\Delta_s b^{\bullet}\bar{n}_t^{\bullet})(s) \\ + \frac{m}{2}\left[(1-r_{\Theta})b^{\bullet}(s)\bar{n}_t^{\bullet}(s) + (1+r_{\Theta})b^{\bullet}(s)\bar{n}_t^{\bullet}(t)\right]\end{aligned} \tag{2.5}$$

(see Machine learning framework for ecosystem models), where we define

$$r_{\Theta} = 2\left(P(\bullet,\bullet) - P(\bullet,\bullet)\right) \tag{2.6}$$

as the habitat assortativity of the graph, which ranges from -1 to 1. When $r_{\Theta} = -1$, all edges connect dissimilar habitat types (disassortative graph), while as r_{Θ} tends towards 1 the graph is composed of two clusters of vertices with identical habitat types (assortative graph). Eq. (2.5) can be analysed with the theory of adaptive dynamics [45, 16, 46], a mathematical framework that provides analytical insights by assuming a trait substitution process. Following this assumption, the mutation term in Eq. (2.5) is omitted and the phenotypic distribution results in a collection of discrete individual types that are gradually replaced by others until evolutionary stability is reached (see Machine learning framework for ecosystem models and [45, 16, 46] for details). By applying the theory of adaptive dynamics, we find a critical migration rate $m^{\star}$

$$m^{\star} = \frac{1}{(1-r_{\Theta})}\frac{4p\theta^2}{(1+3p\theta^2)} \tag{2.7}$$

so that when $m > m^{\star}$, a single type of individual exists with adaptive trait $s^* = {}^{(\theta_{\bullet}+\theta_{\bullet})}/_2 = 0$ in the steady state (see Machine learning framework for ecosystem models for the derivation of Eq. (2.7)). In this case, adaptive differentiation $Q_{ST,s}$ is nil and the average population size is given by $\bar{N} = bK(1-p\theta)^2$. In contrast, when $m = 0$ and/or $r_{\Theta} = 1$, all individuals are locally well-adapted with trait Θ_i on v_i, and it follows that the average population size is higher and equal to $\bar{N} = bK$, while adaptive differentiation is maximal and equal to $Q_{ST,s} = {}^{\mathrm{Var}(\Theta)}/_{(\mathrm{Var}(\Theta)+0)} = 1$. When $0 < m < m^{\star}$, the coexistence of two types of individuals on each vertex v_i is predicted but the calculation of the trait values is more subtle. To understand the effect of m and r_{Θ} on the local trait distributions and on $Q_{ST,s}$, we therefore leave behind the adaptive dynamics framework and numerically solve Eq. (2.5) by including the mutation term. When $0 < m < m^{\star}$, the local trait distributions are bimodal with peaks corresponding to the two types of individuals predicted by the adaptive dynamics. The highest peak corresponds to the well-adapted individuals, whose adaptation is destabilised by the influx of maladaptive migrants (Fig. 2.3a). This phenomenon is dampened as r_{Θ} increases, since the proportion of maladaptive migrants is reduced in assortative graphs (Fig. 2.3b). As a consequence, the habitat assortativity r_{Θ} increases the differentiation $Q_{ST,s}$ when $0 < m < m^{\star}$ (Fig. 2.3c). The simulations further confirm that the adaptive dynamics prediction given by Eq. (2.7) is still valid when the continuous accumulation of mutations is considered, so that for $m > m^{\star}$ the local trait distributions obtained from Eq. (2.5) are unimodal and $Q_{ST,s}$ vanishes (Fig. 2.3a,c). Our analysis of the mean field deterministic approximation Eq. (2.5) therefore demonstrates that assortative

graphs present high levels of adaptive differentiation $Q_{ST,s}$. On the other hand, the analysis shows that $Q_{ST,s}$ rapidly declines with increasing m on disassortative graphs, until $Q_{ST,s}$ vanishes when $m > m^\star$.

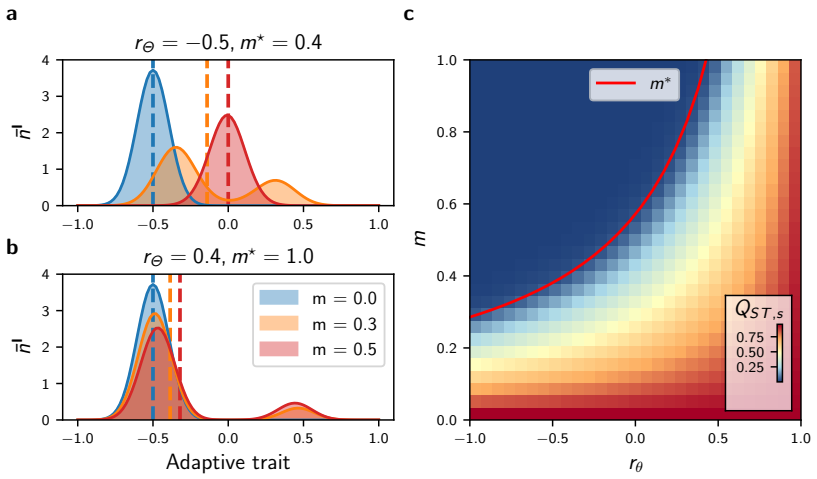

Fig. 2.3: Effect of habitat assortativity r_Θ and migration m on the local adaptive trait distribution $\bar{n}^\bullet$ and on the adaptive differentiation level $Q_{ST,s}$ under the mean field, deterministic approximation Eq. (2.5). (a) Effect of m and r_Θ on $\bar{n}^\bullet$. Migration induces the apparition of maladaptive individuals (centred around $\theta_\bullet = 0.5$), which destabilise local adaptation by displacing the mean value of the well-adapted individuals (centred around $\theta_\bullet = -0.5$). Together with the decrease in local adaptation, migration causes a displacement of the mean value of the local trait distribution (represented by the vertical dashed lines), which decreases local population size and adaptive differentiation $Q_{ST,s}$. (b) Similar data for higher r_Θ. Increasing r_Θ increases population size and $Q_{ST,s}$. (c) Effect of r_Θ on $Q_{ST,s}$. The red line indicates the critical migration threshold $m^\star$ predicted by Eq. (2.7); $Q_{ST,s}$ vanishes when $m > m^\star$.

2.2.5 Effect of graph topology on adaptive differentiation under heterogeneous selection

To generalise the conclusions drawn from the mean field deterministic approximation Eq. (2.5), we generate different Θ-spatial distributions for varying graph topology, and compare outputs of the IBM simulations with those of Eq. (2.5) (see Machine learning framework for ecosystem models for the details of the simulations). For each combination of Θ-spatial distribution and graph, we compute the habitat assortativity r_Θ, since r_Θ can be generalised from Eq. (2.6) to any graph topology following the original definition of [49] as

$$r_\Theta = \frac{\mathrm{Cov}(\Theta_\times, \Theta_\wedge)}{\sigma_{\Theta_\times}\sigma_{\Theta_\wedge}} \tag{2.8}$$

where $\Theta_\times$ and $\Theta_\wedge$ denote the sets of habitats found at the toe and tip of each directed vertex of graph V, and $\langle\Theta_\times\rangle$, $\langle\Theta_\wedge\rangle$ and $\sigma_{\Theta_\times}$, $\sigma_{\Theta_\wedge}$ denote their respective means and standard

deviations (see Supplementary Information). The mean field deterministic approximation Eq. (2.5) is in very good agreement with the IBM simulations for general graph ensembles at low migration regimes, and captures the response of $\bar{N}$ and $Q_{ST,s}$ to r_Θ (Fig. 2.4). Nonetheless, under high migration regimes, higher levels of $Q_{ST,s}$ are observed in the stochastic simulations compared with the mean field deterministic approximation (Fig. S6). We hypothesize that this reinforcement is generated by stochastic drift, which must become the main driver of differentiation when local adaptation is lost for $m > m^\star$, and perform a multivariate regression analysis to investigate the additional effect of $\langle l \rangle$ and h_d on $Q_{ST,s}$. As expected, the analysis highlights that the effect of $\langle l \rangle$ and h_d are substantial and complement the effect of r_θ for high m (Fig. 2.5c for graphs with $M = 7$ vertices and Fig. S7a for $M = 9$), further explaining the discrepancies observed (see Table S3).

We extend our analyses to realistic landscapes with a continuum of habitat types by running simulations on graphs obtained from real spatial habitat datasets and by considering mean annual temperature as a proxy for habitat type (see Fig. S8 and Table S4). We also consider simulations accounting for trait-dependent competition to test whether our results hold under more complex ecological processes (see Supplementary Information for the implementation details and Table S5 for the results). The simulations are congruent and show that the effects of r_Θ, h_d and $\langle l \rangle$ are similar under these alternative settings, underlining the robustness of these metrics and the generality of our conclusions. Taken together, these results indicate that under sufficiently strong selection and sufficiently high habitat heterogeneity, adaptive differentiation $Q_{ST,s}$ is mainly driven by habitat assortativity r_Θ. Nonetheless, local adaptation is lost in disassortative graphs when $m > m^*$, such that $\langle l \rangle$ and h_d become complementary determinants of $Q_{ST,s}$ for high migration regimes.

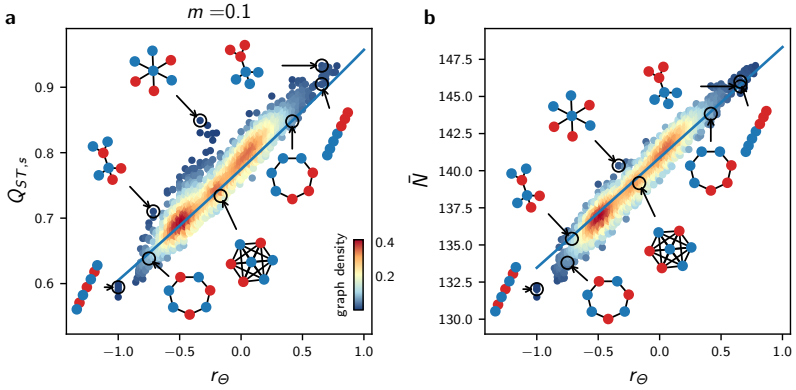

Fig. 2.4: Effect of habitat heterogeneity r_Θ on $Q_{ST,s}$ and average population size $\bar{N}$ for general graph ensembles. (a) Effect of r_Θ on $Q_{ST,s}$ for all undirected connected graphs with $M = 7$ vertices and varying r_Θ, for $m = 0.1$. (b) Effect of r_Θ on average population size $\bar{N}$ for the same simulations. In (a) and (b), each dot represents average results from 5 replicate simulations of the IBM, the colour scale corresponds to the proportion of the graphs with similar x and y axis values (graph density), and the blue lines correspond to results obtained from the mean field approximation Eq. (2.5). Insights from Eq. (2.5) are congruent with the IBM simulations for complex habitat connectivity patterns at low m. Similar results with $m = 0.5$ are presented in Fig. S6.

2.2.6 Effect of habitat assortativity on neutral differentiation under heterogeneous selection

We finally consider a setting with heterogeneous selection where individuals carry both neutral and adaptive traits. With distinct habitat types, selection promotes neutral differentiation by reducing the birth rate of maladaptive migrants, reinforcing the isolation of local populations. We have shown above that adaptive differentiation $Q_{ST,s}$ is driven by habitat assortativity r_Θ, so we expect r_Θ, together with the topological metrics found in the setting with no selection, to influence the level of neutral differentiation $Q_{ST,u}$. We first investigate how the response of $Q_{ST,u}$ to migration compares between the setting with no selection and the setting with heterogeneous selection for graphs with an identical topology. We then examine how the response compares between graphs with an identical topology but different r_Θ. We finally consider simulations on different graphs with varying r_Θ to assess the concurrent effect of $\langle l \rangle$, h_d and r_Θ on $Q_{ST,u}$.

Migration has a fitness cost because maladaptive migrants present lower fitness. Under an equivalent migration regime, migrants therefore have a lower probability of reproduction, increasing the populations' isolation compared with a setting without selection. Simulations with varying m on the complete graph confirm that selection in heterogeneous habitats reinforces $Q_{ST,u}$ compared with a setting without selection (Fig. 2.5a). Nonetheless, previous results show that adaptive differentiation $Q_{ST,s}$ vanishes on a disassortative graph when $m > m^\star$, implying that individuals become equally fit in all habitats. In this case, the isolation effect of heterogeneous selection is lost and $Q_{ST,u}$ reaches a similar level as in the setting with no selection for $m > m^\star$ (Fig. 2.5a), although $Q_{ST,u}$ is slightly higher in the setting with heterogeneous selection due to a lower population size ($\bar{N} = bK(1 - p\theta)$ vs. $\bar{N} = bK$, see section above and Machine learning framework for ecosystem models). This suggests that r_Θ reinforces $Q_{ST,u}$, as assortative graphs sustain higher levels of adaptive differentiation (Figs. 2.3 and 2.4). Simulations on the path graph with varying Θ-spatial distribution support this conclusion for high migration regimes, but show the opposite relationship under low migration regimes, where the habitat assortativity r_Θ decreases $Q_{ST,u}$ (Fig. 2.5b). Assortative graphs are composed of large clusters of vertices with similar habitats, within which migrants can circulate without fitness losses. Local neutral trait distributions become more correlated within these clusters, resulting in a decline in $Q_{ST,u}$ for assortative graphs compared with disassortative graphs. Figure 2.5b therefore highlights the ambivalent effect of r_Θ on $Q_{ST,u}$. r_Θ reinforces $Q_{ST,u}$ by favouring adaptive differentiation, but also decreases $Q_{ST,u}$ by decreasing population isolation within clusters of vertices with the same habitat type.

We compare the effect of r_Θ on $Q_{ST,u}$ to the effect of the topology metrics $\langle l \rangle$ and h_d found in the setting with no selection using a multivariate regression analysis on simulation results obtained for different graphs with varying Θ-spatial distribution (Fig. 2.5d for graphs with $M = 7$ vertices and Fig. S7b for $M = 9$). The multivariate model explains the discrepancies in $Q_{ST,u}$ across the simulations for low and high migration regimes (see Table S3 for details), and we find that r_Θ, $\langle l \rangle$ and h_d contribute similarly to neutral differentiation.

Hence, the effects of r_Θ and the topology metrics $\langle l \rangle$ and h_d add up under heterogeneous selection. A change in sign of the standardized effect of r_Θ on $Q_{ST,s}$ for low and high migration regimes verifies that the ambivalent effect of r_Θ on $Q_{ST,u}$ found on the path graph holds for general graph ensembles. Simulations with trait-dependent competition and simulations on realistic graphs with a continuum of habitat types equally confirm the ambivalent effect of r_Θ and further support the complementary effect of $\langle l \rangle$ and h_d on $Q_{ST,u}$ (see Fig. S8). $\langle l \rangle$ and h_d therefore drive neutral differentiation with and without heterogeneous selection. r_Θ becomes an additional determinant of neutral differentiation under heterogeneous selection. In contrast to the non-ambivalent, positive effect of habitat assortativity on adaptive differentiation, r_Θ can amplify or depress neutral differentiation depending on the migration regime considered.

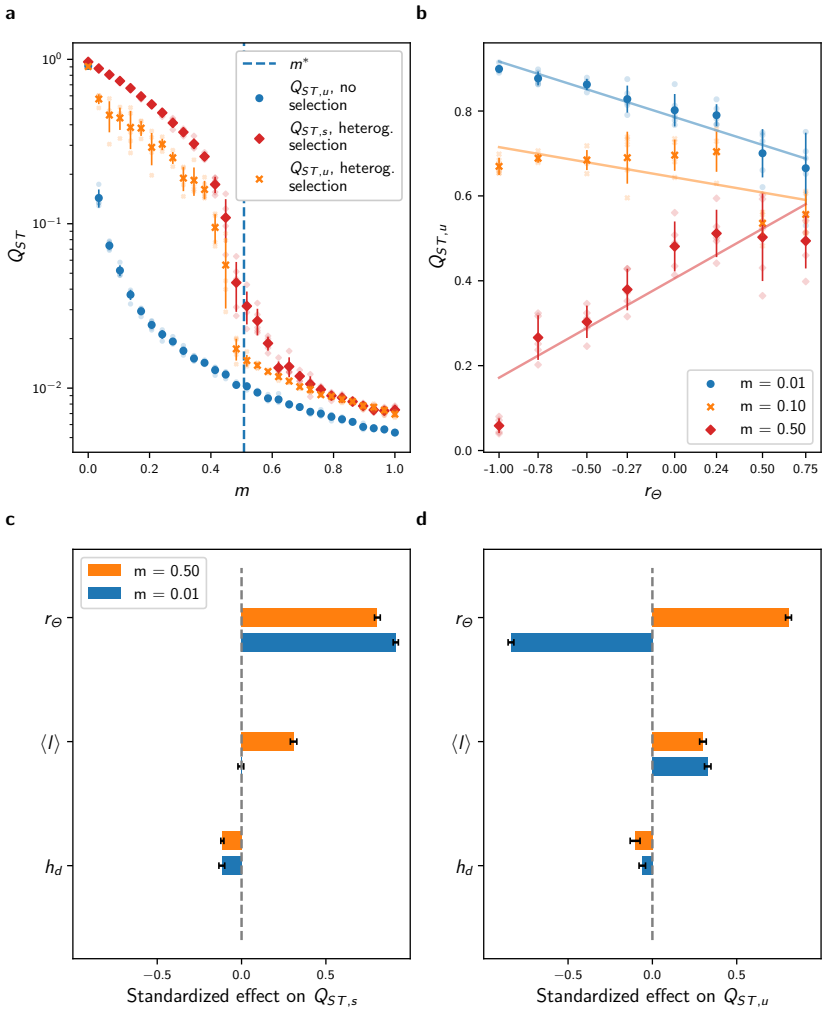

Fig. 2.5: Effect of r_Θ, $\langle l \rangle$ and h_d on $Q_{ST,s}$ and $Q_{ST,u}$ in the setting with heterogeneous selection. (a) Comparison of the response of $Q_{ST,u}$ to migration with the response of $Q_{ST,u}$ in the setting with no selection for the complete graph. The dashed vertical blue line corresponds to the critical migration regime $m^\star$ predicted by Eq. (2.7). Heterogeneous selection increases $Q_{ST,u}$ when $m < m^\star$, but local adaptation is lost when $m > m^\star$, and in this case $Q_{ST,u}$ reaches similar levels as $Q_{ST,u}$ in the setting with no selection. (b) Response of $Q_{ST,u}$ to r_Θ and migration for the path graph. r_Θ correlates positively with $Q_{ST,u}$ for high m, but correlates negatively for low m. In (a–b), each plain dot represents average results from 5 replicate simulations, the bars represent one standard deviation, and each fade dot represents a single replicate value. (c–d) Standardized effect of h_d, $\langle l \rangle$ and r_Θ on $Q_{ST,s}$ and $Q_{ST,u}$ obtained from a multivariate regression model independently fitted for low and high migration regimes on average results from 5 replicate simulations of the IBM on all undirected connected graphs with $M = 7$ vertices and varying r_Θ (see Machine learning framework for ecosystem models). The ambivalence of the effect of r_Θ on $Q_{ST,u}$ found for the path graph holds for general graph ensembles and adds up to that of $\langle l \rangle$ and h_d. Error bars show 95% confidence intervals. Analogous results on graphs with $M = 9$ vertices are presented in Fig. S7 and all regression details can be found in Table S3.

2.3 Discussion

Using analytical tools and simulations, we have built upon a graph representation of landscapes and a stochastic individual-based model to investigate how landscape features drive phenotypic differentiation. Our study is based on a first principles modelling approach [12] describing the stochastic dynamics of individuals and capturing the interplay between population dynamics, phenotypic evolution and spatial dynamics in heterogeneous habitats. In contrast to metacommunity models [29, 24, 42, 26, 11, 64, 63] and evolutionary metacommunity models [22, 21], we have focused on differentiation at the population level. Quantitative genetics and population genetics studies have investigated the effect of topology on differentiation under the assumption of non-overlapping generations, constant population sizes and regular spatial structures [9, 32, 35, 48, 70]. Generalising beyond these assumptions, our modelling framework accounts for population dynamics and includes competition and frequency-dependent selection. The systematic investigation of the effect of topology on differentiation over general graph ensembles and under different ecological settings shows that average path length $\langle l \rangle$, homogeneity in vertex degree h_d and habitat assortativity r_Θ contribute equally to differentiation. These results support correlative studies that have associated population differentiation [41, 44] and species richness [40, 17, 56, 34, 15, 66, 27, 62] with a variety of metrics used as surrogates for connectivity, connectivity heterogeneity and habitat heterogeneity. To further our understanding of the origin of spatial biodiversity patterns, the contribution of landscape properties to discrepancies in population differentiation could be investigated at large scales by (i) using techniques to project real landscapes on graphs (see Fig. S8a–b); (ii) characterising the landscape features with $\langle l \rangle$, h_d and r_Θ; and (iii) relating the obtained metrics maps to observation data. More generally, the proposed eco-evolutionary model on spatial graphs could be combined with approximate bayesian computation to estimate ecological, spatial and evolutionary processes of real populations from observation data, similarly to [37]. This approach might improve current inferential techniques based on models that do not account for competition nor heterogeneous selection (see e.g. [52]). Overall, our results point to topology metrics that can connect spatial biodiversity patterns to the generating eco-evolutionary and spatial processes.

In the absence of selection, neutral differentiation is more pronounced on graphs with a high average path length $\langle l \rangle$, but is also negatively associated with homogeneity in degree h_d (Fig. 2.2c–d). $\langle l \rangle$ generalises the concept of dimensionality in [32, 35, 48], where it is shown that differentiation is lower for two-dimensional grid graphs compared with path graphs. $\langle l \rangle$ also closely relates to the concept of resistance distance shown theoretically and empirically to drive genetic differentiation [43, 44]. At the species level, a similar effect of $\langle l \rangle$ on β-diversity (pairwise differences in species composition) has been reported with the graph metacommunity model of [11] and with the graph eco-evolutionary metacommunity model of [22]. Accounting for population dynamics and specifically including competition processes, we have shown that not only $\langle l \rangle$ but also h_d affects neutral phenotypic differentiation (Fig. 2.2c,d). Our model realistically assumes that population growth is limited by the local carrying capacity. The latter becomes saturated on highly connected vertices in irregular

graphs, an effect that has been experimentally documented in microcosm experiments [2]. As a consequence, central vertices behave as bottlenecks and amplify the isolation of peripheral vertices [50]. The role of h_d cannot be captured with classical metapopulation and quantitative genetics models or with models of evolutionary dynamics in graphs, as they assume constant population size. This behaviour should be prevalent in patchy landscapes where interspecific competition is high because of limiting resources. Our study highlights that heterogeneity in connectivity can reinforce differentiation patterns through the creation of unbalanced migration fluxes which affect ecological equilibrium.

Habitat assortativity r_Θ is a useful indicator for assessing how the spatial distribution of habitat types modulates local adaptation and adaptive differentiation in complex landscapes [58]. While adaptation has been extensively studied along environmental gradients [59, 61, 33, 54, 53, 4, 20], landscapes can be patchy and it is unrealistic to assume regularity [14]. Our model of heterogeneous selection on spatial graphs extends the two-habitat setting investigated in [45, 70, 16, 46] and captures irregularity in connectivity between distinct habitats [14]. Similarly to the aforementioned studies, we have found a critical migration regime $m^\star$ that dictates the possibility of adaptation. Equation (2.7) indicates that $m^\star$ increases with increasing selection strength p and with increasing environmental heterogeneity θ, the latter playing a similar role as the slope of the environmental gradient in [59, 61, 54, 53]. Local adaptation would consequently be sustained under higher migration regimes following an increase in these parameters. Additionally, the critical migration regime $m^\star$ in Eq. (2.7) involves the habitat assortativity r_Θ, which must be regarded as a measure of habitat spatial auto-correlation based on the dispersal range of a species [58]. Our results indicate that for general habitat distributions, r_Θ is the main determinant of adaptive differentiation under sufficiently strong selection p and high habitat heterogeneity θ, irrespective of the graph topology (Fig. 2.5c, Fig. S7a and Fig. S8). As p decreases, however, the effect of stochastic drift on $Q_{ST,s}$ should increase, and in this case the topology metrics $\langle l \rangle$ and h_d should become the most important determinants of $Q_{ST,s}$. Our results predict that in landscapes with heterogeneous habitats and where selection is strong, populations structured over assortative habitats are larger, support higher adaptive differentiation, and can be locally well-adapted even in the case where migration rates are high.

Spatial eco-evolutionary feedbacks in heterogeneous habitats can critically affect differentiation [58]. While most eco-evolutionary studies have investigated diversification by considering a unique adaptive trait [20, 33, 54, 53], distinguishing between neutral and adaptive processes is crucial [28] and our work underlines the distinct responses of neutral and adaptive differentiation to landscape features (Fig. 2.5c vs. Fig. 2.5d). Our study builds upon recent mathematical models that consider the co-evolution of neutral and adaptive traits [6, 3] and extends those works to a spatial context. Our work provides an analytical framework to the concept of isolation by environment (IBE) [50], which has been suggested to be one of the most important mechanisms governing differentiation in nature [67]. Heterogeneous selection leads to more isolation by modifying the fitness of migrants [53], which further reduces gene flow [58] and therefore affects the level of neutral differentiation (Fig. 2.5a) [23]. Our work proposes a mechanism by which habitat assortativity, relative to the migration regime, controls the direction of the effect of habitat heterogeneity on differ-

entiation (Fig. 2.5d). Patchy, heterogeneous habitats can promote neutral differentiation as a result of selection that reduces effective migration [62]. Nonetheless, adaptive differentiation decreases substantially when migration is high relative to the critical migration regime $m^\star$. In this case, neutral differentiation should be higher in landscapes with more aggregated habitats [58]. Our study suggests that habitat assortativity must be considered for a complete understanding of differentiation in complex environments [62].

In conclusion, we have established how differentiation can emerge at the population level from eco-evolutionary feedbacks in complex landscapes by using an analytical description of micro-evolutionary processes explicitly accounting for spatial dynamics over graphs. Our study formalises how differentiation emerges from the interplay between spatial dynamics, the co-evolution of neutral and adaptive traits, and landscape properties. Connectivity and habitat assortativity emerge as core determinants of differentiation in spatial graphs. These results resonate with empirical findings and previous theoretical works. Our study further stresses that habitat assortativity can depress or foster neutral differentiation depending on the migration regime. Additionally, our work highlights that heterogeneity in connectivity is an equally strong determinant of differentiation because highly connected habitats behave as bottlenecks, increasing the isolation of peripheral habitats. The present approach offers a promising framework for studying complex adaptive systems, as it can elucidate how macroscopic properties emerge from microscopic processes acting upon agents structured over complex spatio-evolutionary structures.

2.4 Methods

2.4.1 Mean field approximation

In the setting with no selection, the mean field approach involves the assumption that all vertices having the same degree are equivalent. For this, let $P(k,k')$ denote the proportion of edges that map a vertex with degree k to a vertex with degree k', and consider the average population size $\bar{N}_t^{(k)}$ in each vertex with degree k at time t. An individual has probability $P(k,k')/k'$ to migrate from a vertex with degree k' to a vertex with degree k. Viewing $a_{i,j}/d_j$ as the probability that an individual on v_i chosen for migration moves to v_j, Eq. (2.3) then transforms into

$$\partial_t \bar{N}_t^{(k)} = \bar{N}_t^{(k)} \left[b(1-m) - \frac{\bar{N}_t^{(k)}}{K} \right] + mbk \sum_{k' \in V} \frac{P(k,k')}{k'} \bar{N}_t^{(k')} \tag{2.9}$$

Assuming uncorrelated graphs for which $P(k,k')/k' = P(k')k'/\langle k \rangle$, where $\langle k \rangle$ denotes the average degree of the graph [13], yields

$$\partial_t \bar{N}_t^{(k)} = \bar{N}_t^{(k)} \left[b(1-m) - \frac{\bar{N}_t^{(k)}}{K} \right] + mb \frac{k}{\langle k \rangle} \bar{N}_t \tag{2.10}$$

where

$$\bar{N}_t = \sum_k P(k)\bar{N}_t^{(k)}. \tag{2.11}$$

When solving for the stationary state and setting $m = 1$, one obtains $\bar{N}^{(k)} = \sqrt{bK\frac{k}{\langle k\rangle}\bar{N}}$ from Eq. (2.10). Combining this with Eq. (2.11) yields

$$\bar{N} = bK\langle\sqrt{k}\rangle^2/\langle k\rangle \tag{2.12}$$

In the setting with heterogeneous selection, the mean field approach involves the assumption that all vertices with a similar habitat are equivalent. In this case, an individual from a vertex of habitat type $\bullet$ has the probability $P(\bullet,\bullet)/P(\bullet)$ of migrating to a vertex of type $\bullet$, and therefore Eq. (2.4) transforms into

$$\begin{aligned}\partial_t \bar{n}_t^{\bullet}(s) = \bar{n}_t^{\bullet}(s)\left[b^{\bullet}(s)(1-m) - \frac{1}{K}\int_{\mathcal{S}} \bar{n}_t^{\bullet}(\mathbf{s})d\mathbf{s}\right] + \tfrac{1}{2}\mu\sigma_\mu^2\Delta_s\left[b^{\bullet}(s)\bar{n}_t^{\bullet}(s)\right] \\ + m\sum_{i\in\{\bullet,\bullet\}} b_i(s)\frac{P(\bullet,i)}{P(i)}\bar{n}_t^i(s)\end{aligned} \tag{2.13}$$

Considering that $P(\bullet) = P(\bullet) = \frac{1}{2}$ (habitats are equally distributed), $P(\bullet,\bullet)+P(\bullet,\bullet) = P(\bullet)$ (sum of conditional expectations), and $r_\Theta = 2\left(P(\bullet,\bullet) - P(\bullet,\bullet)\right)$ (Eq. (2.6)), one obtains

$$P(\bullet,\bullet) = \tfrac{1}{4}(1-r_\Theta) \quad \text{and} \quad P(\bullet,\bullet) = \tfrac{1}{4}(1+r_\Theta) \tag{2.14}$$

Combining Eq. (2.14) with Eq. (2.13) yields Eq. (2.5). We show in the Supplementary Information how one can derive Eq. (2.6) from the general definition of assortativity given in Eq. (2.8) and initially introduced in [49].

2.4.2 Adaptive dynamics on graphs

The adaptive dynamics theory considers a monomorphic population that evolves following a trait substitution process [45]. Accordingly, the trait s of the monomorphic metapopulation evolves gradually along the direction given by its fitness gradient, until it reaches a singular strategy s^* for which the fitness gradient vanishes. By omitting the mutation term, Eq. (2.6) can be written in the matrix form

$$\partial_t \bar{\mathbf{n}}_t(s) = M(s,\bar{\mathbf{N}}_t)\,\bar{\mathbf{n}}_t(s) \tag{2.15}$$

where $\bar{\mathbf{n}}_t = (\bar{n}_t^{\bullet},\bar{n}_t^{\bullet})$ and $\bar{\mathbf{N}}_t = (\bar{N}_t^{\bullet},\bar{N}_t^{\bullet})$ are the vectors containing the population densities and the population size on each habitat type, and

$$M(s,\bar{\mathbf{N}}) = \begin{bmatrix} \mathfrak{r}^{\bullet}(s,\bar{N}^{\bullet}) & \frac{m}{2}(1-r_\Theta)b^{\bullet}(s) \\ \frac{m}{2}(1-r_\Theta)b^{\bullet}(s) & \mathfrak{r}^{\bullet}(s,\bar{N}^{\bullet}) \end{bmatrix} \tag{2.16}$$

is the so-called projection matrix [45], with $\mathfrak{r}^\bullet(s,\bar{N}^\bullet) = b^\bullet(s)(1+\frac{m}{2}(r_\Theta-1)) - \bar{N}^\bullet/K$. The overall fitness of individuals with trait s is the leading eigenvalue of M, which we denote with $\lambda(s,\bar{\mathbf{N}})$. We obtain the singular strategy s^* by setting the fitness gradient $\frac{\partial\lambda}{\partial s}(s,\bar{\mathbf{N}}) = 0$, from which we further obtain the demographic equilibrium $\bar{\mathbf{N}}^{s^*}$. Because of symmetries, we must have $\bar{N}^{\bullet,s^*} = \bar{N}^{\bullet,s^*}$ and $s^* = \frac{\theta_\bullet+\theta_\bullet}{2} = 0$, such that $\bar{N}^{\bullet,s^*} = \bar{N}^{\bullet,s^*} = bK(1-p\theta^2)$. s^* is said to be evolutionary stable if no mutants can invade, i.e. if s^* locally maximises the fitness of a mutant with trait y in the resident population with trait s^*, given by $\lambda(y,\bar{\mathbf{N}}^{s^*})$ (see [45] for details). One can show that $\left[\frac{\partial\lambda}{\partial y}(y,\bar{\mathbf{N}}^{s^*})\right]_{y=s^*} = 0$ and the condition for evolutionary stability becomes $\left[\frac{\partial^2\lambda}{\partial y^2}(y,\bar{\mathbf{N}}^{s^*})\right]_{y=s^*} < 0$. We compute and simplify this inequality through computer algebra (see Mathematica notebook provided in the simulation code), which leads to Eq. (2.7).

2.4.3 Numerical simulations

The model was implemented in a multi-purpose Julia package called `EvoId.jl`, available at https://github.com/vboussange/EvoId.jl. For each result presented, $b = 1$, local carrying capacity $K = 150$, selection strength $p = 1$, mutation rate $\mu = 0.1$, mutation range $\sigma_\mu = 5\cdot10^{-2}$, and total time span $t = 1000$. This parameter choice made it possible to discard transient dynamics while obtaining results in a reasonable computational time (see Fig. S9). In settings (1) and (2), we ran simulations on all of the 853 undirected connected graphs with $M = 7$ vertices and on 1126 of the 261,080 undirected connected graphs with $M = 9$ vertices, listed at http://oeis.org/A001349. Graphs with $M = 9$ vertices were selected with a stratified sampling method: we randomly sampled without replacement a maximum of 50 graphs for each class of graphs with an equal number of vertices. For the setting with heterogeneous selection, we generated the labeled graphs by randomly generating Θ-spatial distributions, and by using a stratified sampling strategy to select without replacement at most 3 and 2 Θ-spatial distributions corresponding to the quartiles of the r_θ values obtained, respectively for graphs with $M = 7$ and $M = 9$ vertices. This sampling strategy allowed to obtain a uniform distribution of the topology metrics investigated in the study, and therefore permitted to correctly represent the population of graphs to investigate their effect on differentiation. We then computed $Q_{ST,u}$ and $Q_{ST,s}$, which we further averaged over the last time steps and across the replicates. Since the dynamics of $Q_{ST,u}$ is characterised by large quadratic variations, we simulated individuals with $d = 300$ neutral traits, where each trait can independently be affected by mutations. $Q_{ST,u}$ values presented were then obtained from the average $Q_{ST,u}$ for each trait. This reduced the variance of the numerical simulations and is also biologically meaningful because populations are characterised by many traits, most of which are neutral [28]. As initial conditions, MK individuals were homogeneously distributed over all of the vertices, with traits centred on 0 and with standard deviation σ_μ. Graph metrics used for the meta-analysis were calculated using the **LightGraphs.jl** library [8]. We numerically solved the PDEs with a finite difference scheme using **DifferentialEquations.jl** [55], ensuring that the domain was large enough to avoid border effects.

2.4.4 Statistics and reproducibility

Statistical anyalses were conducted in Julia using **StatsKit.jl**. All simulations can be exactly reproduced from the code available at https://github.com/vboussange/differentiation-in-spatial-graphs.

Data availability

The data underlying our figures is available at https://github.com/vboussange/differentiation-in-spatial-graphs.

Code availability

The simulation code is available at https://github.com/vboussange/differentiation-in-spatial-graphs.

Author contributions

V.B. and L.P. designed research; V.B. performed research; V.B. and L.P. wrote the paper.

Acknowledgements

We thank Thomas Poulet, Sylvian Billiard, Sepideh Mirrahimi, Heike Lischke, Joshua Payne, Conor Waldock, Yaquan Chang, Flora Desmet, Benjamin Flück and Alexander Skeels for helpful discussions and comments on the manuscript. L.P. was supported by the Swiss National Science Foundation grant (Nř 310030_188550). We thank two anonymous reviewers for constructive comments and valuable suggestions on a previous version of this article.

References

[1] R. Aguilée, D. Claessen, and A. Lambert. “Adaptive radiation driven by the interplay of eco-evolutionary and landscape dynamics”. In: *Evolution* 67.5 (2012), no–no. DOI: 10.1111/evo.12008.

[2] F. Altermatt and E. A. Fronhofer. “Dispersal in dendritic networks: Ecological consequences on the spatial distribution of population densities”. In: *Freshwater Biology* 63.1 (2018), pp. 22–32. DOI: 10.1111/fwb.12951.

[3] N. Anceschi et al. “Neutral and niche forces as drivers of species selection”. In: *Journal of Theoretical Biology* 483 (2019), p. 109969. DOI: 10.1016/j.jtbi.2019.07.021.

[4] M. AndradeRestrepo, N. Champagnat, and R. Ferrière. “Local adaptation, dispersal evolution, and the spatial ecoevolutionary dynamics of invasion”. In: *Ecology Letters* 22.5 (2019). Ed. by V. Calcagno, pp. 767–777. DOI: 10.1111/ele.13234.

[5] V. Bansaye and S. Méléard. “Some stochastic models for structured populations : scaling limits and long time behavior”. In: *Stochastic Models for Structured Populations: Scaling Limits and Long Time Behavior* (2015), pp. 1–107. DOI: 10.1007/978-3-319-21711-6. arXiv: 1506.04165.

[6] S. Billiard et al. “Stochastic dynamics of adaptive trait and neutral marker driven by eco-evolutionary feedbacks”. In: *Journal of Mathematical Biology* 71.5 (2015), pp. 1211–1242. DOI: 10.1007/s00285-014-0847-y. arXiv: 1310.6274.

[7] G. Bounova and O. de Weck. “Overview of metrics and their correlation patterns for multiple-metric topology analysis on heterogeneous graph ensembles”. In: *Physical Review E* 85.1 (2012), p. 016117. DOI: 10.1103/PhysRevE.85.016117.

[8] S. Bromberger and other Contributors. “JuliaGraphs/LightGraphs.jl”. In: (2017). DOI: 10.5281/ZENODO.1412141.

[9] R. Bürger. *The mathematical theory of selection, recombination, and mutation*. eng. Wiley series in mathematical and computational biology. Chichester [etc: J. Wiley, 2000.

[10] J. S. Cabral, L. Valente, and F. Hartig. “Mechanistic simulation models in macroecology and biogeography: state-of-art and prospects”. In: *Ecography* 40.2 (2017), pp. 267–280. DOI: 10.1111/ecog.02480.

[11] F. Carrara et al. “Dendritic connectivity controls biodiversity patterns in experimental metacommunities”. In: *Proceedings of the National Academy of Sciences of the United States of America* 109.15 (2012), pp. 5761–5766. DOI: 10.1073/pnas.1119651109.

[12] N. Champagnat, R. Ferrière, and S. Méléard. “Unifying evolutionary dynamics: From individual stochastic processes to macroscopic models”. In: *Theoretical Population Biology* 69.3 (2006), pp. 297–321. DOI: 10.1016/j.tpb.2005.10.004.

[13] V. Colizza, R. Pastor-Satorras, and A. Vespignani. “Reactiondiffusion processes and metapopulation models in heterogeneous networks”. In: *Nature Physics* 3.4 (2007), pp. 276–282. DOI: 10.1038/nphys560. arXiv: 0703129 [cond-mat].

[14] M. R. Dale and M. Fortin. “From graphs to spatial graphs”. In: *Annual Review of Ecology, Evolution, and Systematics* 41 (2010), pp. 21–38. DOI: 10.1146/annurev-ecolsys-102209-144718.

[15] R. G. Davies et al. “Topography, energy and the global distribution of bird species richness”. In: *Proceedings of the Royal Society B: Biological Sciences* 274.1614 (2007), pp. 1189–1197. DOI: 10.1098/rspb.2006.0061.

[16] F. Débarre, O. Ronce, and S. Gandon. “Quantifying the effects of migration and mutation on adaptation and demography in spatially heterogeneous environments”. In: *Journal of Evolutionary Biology* 26.6 (2013), pp. 1185–1202. DOI: 10.1111/jeb.12132.

[17] M. S. Dias et al. “Global imprint of historical connectivity on freshwater fish biodiversity”. In: *Ecology Letters* 17.9 (2014). Ed. by M. Anderson, pp. 1130–1140. DOI: 10.1111/ele.12319.

[18] U. Dieckmann and M. Doebeli. “On the origin of species by sympatric speciation”. In: *Nature* 400.6742 (1999), pp. 354–357. DOI: 10.1038/22521.

[19] W.-N. Ding et al. “Ancient orogenic and monsoon-driven assembly of the world’s richest temperate alpine flora”. In: *Science* 369.6503 (2020), pp. 578–581. DOI: 10.1126/science.abb4484.

[20] M. Doebeli and U. Dieckmann. “Speciation along environmental gradients”. In: *Nature* 421.6920 (2003), pp. 259–264. DOI: 10.1038/nature01274.

[21] E. P. Economo and T. H. Keitt. “Network isolation and local diversity in neutral metacommunities”. In: *Oikos* 119.8 (2010), pp. 1355–1363. DOI: 10.1111/j.1600-0706.2010.18272.x.

[22] E. P. Economo and T. H. Keitt. “Species diversity in neutral metacommunities: a network approach”. In: *Ecology Letters* 11.1 (2007), 071117033013001–??? DOI: 10.1111/j.1461-0248.2007.01126.x.

[23] D. Garant, S. E. Forde, and A. P. Hendry. “The multifarious effects of dispersal and gene flow on contemporary adaptation”. In: *Functional Ecology* 21.3 (2007), pp. 434–443. DOI: 10.1111/j.1365-2435.2006.01228.x.

[24] L. J. Gilarranz and J. Bascompte. “Spatial network structure and metapopulation persistence”. In: *Journal of Theoretical Biology* 297 (2012), pp. 11–16. DOI: 10.1016/j.jtbi.2011.11.027.

[25] D. T. Gillespie. “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions”. In: *Journal of Computational Physics* 22.4 (1976), pp. 403–434. DOI: 10.1016/0021-9991(76)90041-3.

[26] D. Gravel, F. Massol, and M. A. Leibold. “Stability and complexity in model meta-ecosystems”. In: *Nature Communications* 7.1 (2016), p. 12457. DOI: 10.1038/ncomms12457.

[27] J.-F. Guégan, S. Lek, and T. Oberdorff. “Energy availability and habitat heterogeneity predict global riverine fish diversity”. In: *Nature* 391.6665 (1998), pp. 382–384. DOI: 10.1038/34899.

[28] R. Holderegger, U. Kamm, and F. Gugerli. “Adaptive vs. neutral genetic diversity: implications for landscape genetics”. In: *Landscape Ecology* 21.6 (2006), pp. 797–807. DOI: 10.1007/s10980-005-5245-9.

[29] M. D. Holland and A. Hastings. “Strong effect of dispersal network structure on ecological dynamics”. In: *Nature* 456.7223 (2008), pp. 792–794. DOI: 10.1038/nature07395.

[30] S. P. Hubbell. *The unified neutral theory of biodiversity and biogeography*. Monographs in Population Biology 32. Princeton [etc: Princeton University Press, 2001.

[31] N. L. Kaplan, R. Hudson, and C. H. Langley. “The hitchhiking effect revisited.” In: *Genetics* 123.4 (1989), pp. 887–899. DOI: 10.1093/genetics/123.4.887.

[32] M Kimura and G. H. Weiss. “The stepping stone model of population structure and the decrease of genetic correlation with distance.” In: *Genetics* 49.4 (1964), pp. 561–76. DOI: 10.1093/oxfordjournals.molbev.a025590.

[33] M. Kirkpatrick and N. H. Barton. “Evolution of a Species’ Range”. In: *The American Naturalist* 150.1 (1997), pp. 1–23. DOI: 10.1086/286054.

[34] H. Kreft and W. Jetz. “Global patterns and determinants of vascular plant diversity”. In: *Proceedings of the National Academy of Sciences of the United States of America* 104.14 (2007), pp. 5925–5930. DOI: 10.1073/pnas.0608361104.

[35] R. Lande. “Isolation by distance in a quantitative trait”. In: *Genetics* 128.2 (1991), pp. 443–452.

[36] R. Lande. “NEUTRAL THEORY OF QUANTITATIVE GENETIC VARIANCE IN AN ISLAND MODEL WITH LOCAL EXTINCTION AND COLONIZATION”. In: *Evolution* 46.2 (1992), pp. 381–389. DOI: 10.1111/j.1558-5646.1992.tb02046.x.

[37] C. Lepers et al. “Inference with selection, varying population size, and evolving population structure: application of ABC to a forwardbackward coalescent process with interactions”. In: *Heredity* 126.2 (2021), pp. 335–350. DOI: 10.1038/s41437-020-00381-x. arXiv: 1910.10201.

[38] S. A. Levin. “Complex adaptive systems: Exploring the known, the unknown and the unknowable”. In: *Bulletin of the American Mathematical Society* 40.01 (2002), pp. 3–20. DOI: 10.1090/S0273-0979-02-00965-5.

[39] S. Lion. “Moment equations in spatial evolutionary ecology”. In: *Journal of Theoretical Biology* 405 (2016), pp. 46–57. DOI: 10.1016/j.jtbi.2015.10.014.

[40] C. Liu et al. “Mountain metacommunities: climate and spatial connectivity shape ant diversity in a complex landscape”. In: *Ecography* 41.1 (2018), pp. 101–112. DOI: 10.1111/ecog.03067.

[41] S. Manel et al. “Landscape genetics: combining landscape ecology and population genetics”. In: *Trends in Ecology & Evolution* 18.4 (2003), pp. 189–197. DOI: 10.1016/S0169-5347(03)00008-9.

[42] L. Mari et al. “Metapopulation persistence and species spread in river networks”. In: *Ecology Letters* 17.4 (2014). Ed. by F. Jordán, pp. 426–434. DOI: 10.1111/ele.12242.

[43] B. H. McRae. “ISOLATION BY RESISTANCE”. In: *Evolution* 60.8 (2006), p. 1551. DOI: 10.1554/05-321.1.

[44] B. H. McRae and P. Beier. “Circuit theory predicts gene flow in plant and animal populations”. In: *Proceedings of the National Academy of Sciences* 104.50 (2007), pp. 19885–19890. DOI: 10.1073/pnas.0706568104.

[45] G. Meszéna, I. Czibula, and S. Geritz. “Adaptive Dynamics in a 2-Patch Environment: A Toy Model for Allopatric and Parapatric Speciation”. In: *Journal of Biological Systems* 05.02 (1997), pp. 265–284. DOI: 10.1142/S0218339097000175.

[46] S. Mirrahimi and S. Gandon. “Evolution of specialization in heterogeneous environments: equilibrium between selection, mutation and migration”. In: *Genetics* 214.2 (2020), pp. 479–491. DOI: 10.1534/genetics.119.302868.

[47] R. Muneepeerakul et al. “Neutral metacommunity models predict fish diversity patterns in MississippiMissouri basin”. In: *Nature* 453.7192 (2008), pp. 220–222. DOI: 10.1038/nature06813.

[48] T. Nagylaki. “Geographical variation in a quantitative character.” In: *Genetics* 136.1 (1994), pp. 361–81.

[49] M. E. J. Newman. “Mixing patterns in networks”. In: *Physical Review E* 67.2 (2003), p. 026126. DOI: 10.1103/PhysRevE.67.026126. arXiv: 0209450 [cond-mat].

[50] L. Orsini et al. “Drivers of population genetic differentiation in the wild: Isolation by dispersal limitation, isolation by adaptation and isolation by colonization”. In: *Molecular Ecology* 22.24 (2013), pp. 5983–5999. DOI: 10.1111/mec.12561.

[51] F. Pelletier, D. Garant, and A. Hendry. “Eco-evolutionary dynamics”. eng. In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 364.1523 (2009), pp. 1483–1489. DOI: 10.1098/rstb.2009.0027.

[52] D. Petkova, J. Novembre, and M. Stephens. “Visualizing spatial population structure with estimated effective migration surfaces”. In: *Nature Genetics* 48.1 (2015), pp. 94–100. DOI: 10.1038/ng.3464.

[53] J. Polechová. “Is the sky the limit? On the expansion threshold of a species’ range”. In: *PLoS Biology* 16.6 (2018), pp. 1–18. DOI: 10.1371/journal.pbio.2005372.

[54] J. Polechová and N. H. Barton. “Limits to adaptation along environmental gradients”. In: *Proceedings of the National Academy of Sciences of the United States of America* 112.20 (2015), pp. 6401–6406. DOI: 10.1073/pnas.1421515112.

[55] C. Rackauckas and Q. Nie. “DifferentialEquations.jl a performant and feature-rich ecosystem for solving differential equations in Julia”. In: *Journal of Open Research Software* 5 (2017). DOI: 10.5334/jors.151.

[56] C. Rahbek and G. R. Graves. “Multiscale assessment of patterns of avian species richness”. In: *Proceedings of the National Academy of Sciences of the United States of America* 98.8 (2001), pp. 4534–4539. DOI: 10.1073/pnas.071034898.

[57] C. Rahbek et al. “Building mountain biodiversity: Geological and evolutionary processes”. In: *Science* 365.6458 (2019), pp. 1114–1119. DOI: 10.1126/science.aax0151.

[58] J. L. Richardson et al. “Microgeographic adaptation and the spatial scale of evolution”. In: *Trends in Ecology & Evolution* 29.3 (2014), pp. 165–176. DOI: 10.1016/j.tree.2014.01.002.

[59] M. Slatkin. “GENE FLOW AND SELECTION IN A CLINE”. In: *Genetics* 75.4 (1973), pp. 733–756. DOI: 10.1093/genetics/75.4.733.

[60] M. Slatkin. “Isolation by distance in equilibrium and non-equilibrium populations”. In: *Evolution* 47.1 (1993), pp. 264–279. DOI: 10.1111/j.1558-5646.1993.tb01215.x.

[61] M. Slatkin. “Spatial patterns in the distributions of polygenic characters”. In: *Journal of Theoretical Biology* 70.2 (1978), pp. 213–228. DOI: 10.1016/0022-5193(78)90348-X.

[62] A. Stein, K. Gerstner, and H. Kreft. “Environmental heterogeneity as a universal driver of species richness across taxa, biomes and spatial scales”. In: *Ecology Letters* 17.7 (2014), pp. 866–880. DOI: 10.1111/ele.12277.

[63] Y. Suzuki and E. P. Economo. “From species sorting to mass effects: spatial network structure mediates the shift between metacommunity archetypes”. In: *Ecography* (2021), p. 05453. DOI: 10.1111/ecog.05453.

[64] P. L. Thompson, B. Rayfield, and A. Gonzalez. “Loss of habitat and connectivity erodes species diversity, ecosystem functioning, and stability in metacommunity networks”. In: *Ecography* 40.1 (2017), pp. 98–108. DOI: 10.1111/ecog.02558.

[65] J. Tkadlec et al. “Population structure determines the tradeoff between fixation probability and fixation time”. In: *Communications Biology* 2.1 (2019), p. 138. DOI: 10.1038/s42003-019-0373-y.

[66] J. A. Veech and T. O. Crist. “Habitat and climate heterogeneity maintain beta-diversity of birds among landscapes within ecoregions”. In: *Global Ecology and Biogeography* 16.5 (2007), pp. 650–656. DOI: 10.1111/j.1466-8238.2007.00315.x.

[67] I. J. Wang and G. S. Bradburd. “Isolation by environment”. In: *Molecular Ecology* 23.23 (2014), pp. 5649–5662. DOI: 10.1111/mec.12938.

[68] M. C. Whitlock. “Evolutionary inference from Q ST”. In: *Molecular Ecology* 17.8 (2008), pp. 1885–1896. DOI: 10.1111/j.1365-294X.2008.03712.x.

[69] J. Wickman et al. “Determining selection across heterogeneous landscapes: A perturbation-based method and its application to modeling evolution in space”. In: *The American Naturalist* 189.4 (2017), pp. 381–395. DOI: 10.1086/690908.

[70] S. Yeaman and S. P. Otto. “Establishment and maintenance of adaptive genetic divergence under migration, selection, and drift”. In: *Evolution* 65.7 (2011), pp. 2123–2129. DOI: 10.1111/j.1558-5646.2011.01277.x.

2.A Supplementary Note

2.A.1 Mathematical construction of the model

The model is a measure-valued point process [1], so that individuals are represented as dirac functions $\delta_{x_k^{(i)}}$, where $x_k^{(i)} \in \mathcal{X}$ corresponds to the traits' value of individual k located on vertex v_i. Under this formalism, the population on v_i is represented as a sum of dirac functions $\nu^{(i)} = \sum_k^{N^{(i)}} \delta_{x_k^{(i)}}$, where $N^{(i)}$ is the local population size. It follows that the time variation of the process can be described by the so-called infinitesimal generator L, defined for all real valued functions ϕ as

$$L\phi(\nu_t^{(i)}) = \partial_t \mathbb{E}\left[\phi(\nu_t^{(i)})\right] \tag{S1}$$

(see [9] for an introduction to infinitesimal generators). Equation (S1) provides the expected time variation at time t of e.g. the population size by choosing $\phi(\nu_t^{(i)}) = \int_{\mathcal{X}} \nu_t^{(i)}(dx)$. Recall that we use $b^{(i)}$ to denote the birth rate on vertex v_i, d for the death rate, μ for the mutation probability, m for the migration probability, $\mathcal{M}(x,y) = \frac{1}{\sqrt{2\pi}\sigma_\mu} \exp\left(\frac{||x-y||^2}{2\sigma_\mu}\right)$ for the mutation kernel, K for the local carrying capacity, $A = (a_{i,j})_{1\leq i,j\leq M}$ for the adjacency matrix of the graph G, and $D = (d_1, d_2, \ldots, d_M)$ for the vector containing the degree of each vertex. In order to explicitly write the generator L, let us recall that five events of different natures can alter the number of individuals with trait x on vertex v_i:

- an individual on v_i with trait x can give birth to an offspring that does not experience mutations nor migration, at rate $(1-\mu)(1-m)b^{(i)}(x)$,
- an individual on v_i with trait y can give birth to an offspring with mutated trait x that does not experience migration, at rate $\mu(1-m)\mathcal{M}(x,y)b^{(i)}(y)$,
- an individual on v_i with trait x can die, at rate $d(N^{(i)}) = \frac{N^{(i)}}{K} = \frac{1}{K}\int_{\mathcal{X}} \nu_t^{(i)}(dx)$,
- an individual on v_j with trait x can give birth to an offspring that does not experience mutations and migrates to v_i, at rate $\frac{a_{i,j}}{d_j}(1-\mu)m\, b^{(j)}(x)$,
- an individual on v_j with trait y can give birth to an offspring with mutated trait x that migrates to v_i, at rate $\frac{a_{i,j}}{d_j}\mu m\mathcal{M}(x,y)b^{(j)}(x)$.

Summing over all all individuals and all vertices yields

$$
\begin{aligned}
L\phi(\nu_t^{(i)}) = &\int_{\mathcal{X}} \left\{ b^{(i)}(\mathbf{x})(1-\mu)(1-m)(\phi(\nu_t^{(i)} + \delta_{\mathbf{x}}) - \phi(\nu_t^{(i)})) \right\} \nu_t^{(i)}(d\mathbf{x}) && \text{births w/o mutations, w/} \\
&+ \int_{\mathcal{X}} \left\{ \mu(1-m) \int_{\mathcal{X}} b^{(i)}(y)(\phi(\nu_t^{(i)} + \delta_z) - \phi(\nu_t^{(i)}))\mathcal{M}(\mathbf{x}, y)dy \right\} \nu_t^{(i)}(d\mathbf{x}) && \text{births w/ mutations, w/} \\
&+ \iint_{\mathcal{X}} \left\{ \frac{1}{K}(\phi(\nu_t^{(i)} - \delta_{\mathbf{x}})) - \phi(\nu_t^{(i)}))\nu_t^{(i)}(dy)\, \nu_t^{(i)}(dx) \right\} \\
&+ \sum_{j \neq i} \frac{a_{i,j}}{d_j} \int_{\mathcal{X}} \mu m \left\{ \int_{\mathcal{X}} b^{(j)}(y)(\phi(\nu^{(j)} + \delta_{\mathbf{x}}) - \phi(\nu^{(j)}))\mathcal{M}(\mathbf{x}, y)dy \right\} \nu_t^{(j)}(d\mathbf{x}) && \text{migrations} \\
&+ \sum_{j \neq i} \frac{a_{i,j}}{d_j} \int_{\mathcal{X}} \left\{ b^{(j)}(\mathbf{x})(1-\mu)m(\phi(\nu^{(j)} + \delta_{\mathbf{x}}) - \phi(\nu^{(j)})) \right\} \nu_t^{(j)}(d\mathbf{x}). && \text{migrations w}
\end{aligned}
\tag{S2}
$$

Taking expectations in Eq. (S2), one can obtain an equation for the mean trajectory of the quantity of interest, $\mathbb{E}\left[\phi(\nu_t^{(i)})\right]$. Nonetheless, Eq. (S2) involves an integral with respect to $\nu_t^{(i)}(dx)\nu_t^{(i)}(dy)$, making it impossible to obtain an explicit solution. It is therefore unclear whether one can gain insight into the stochastic dynamics from Eq. (S2) without simplifying assumptions. We refer to [2] for a detailed discussion on the topic.

2.A.2 Deterministic approximation

One strategy to overcome the difficulties encountered above is to assimilate the process to its mean trajectory, assuming that $\mathbb{E}\left[\nu_t^{(i)}\right] \approx \nu_t^{(i)}$ and further approximating $\nu_t^{(i)}$ with a continuous deterministic function $n_t^{(i)}$. Such strategy inherently neglects the stochasticity of the process, which is reasonable provided that a force dampens the stochastic fluctuations of the quantity of interest.

Setting with no selection

Consider a setting with no selection and recall that in this setting where $x \equiv u \in \mathcal{X} = \mathcal{U}$ we define

$$b^{(i)}(x) \equiv b \tag{S3}$$

By applying the strategy mentioned above and choosing $\phi(n_t^{(i)}) = \int_{\mathcal{X}} n_t^{(i)}(x)dx$, Eq. (S2) transforms into the deterministic approximation of the population size dynamics given in the main-text by

$$\partial_t N_t^{(i)} = N_t^{(i)} \left[b(1-m) - \frac{N_t^{(i)}}{K} \right] + mb \sum_{j \neq i} \frac{a_{i,j}}{d_j} N_t^{(j)}. \tag{S4}$$

Competition stabilises the population size dynamics, which behaves deterministically. This is supported by Fig. S10a, which shows how Eq. (S4) accurately describes the population size for varying migration regimes. Nonetheless, stochastic fluctuations drive the dynamics of the neutral trait distribution. Attempting to characterise the neutral trait distribution with the same strategy, this time setting $\phi(n_t^{(i)}) = n_t^{(i)}(u)$, yields

$$\begin{aligned} \partial_t n_t^{(i)}(u) = n_t^{(i)}(u) \left[b(1-m)(1-\mu) - \frac{1}{K} \int_{\mathcal{U}} n_t^{(i)}(\mathbf{u})\, d\mathbf{u} \right) \right] \\ + (1-m)\mu b \int_{\mathcal{U}} n_t^{(i)}(\mathbf{u}) \mathcal{M}(u, \mathbf{u})\, d\mathbf{u} \\ + m\mu b \sum_{j \neq i} \frac{a_{i,j}}{d_j} \int_{\mathcal{U}} n_t^{(j)}(u) \mathcal{M}(u, \mathbf{u}) d\mathbf{u} \\ + m(1-\mu) b \sum_{j \neq i} \frac{a_{i,j}}{d_j} b\, n_t^{(j)}(u). \end{aligned} \tag{S5}$$

Solving for Eq. (S5), one can show that the variance of $n_t^{(i)}$ continuously grows in time (see Fig. S10) and tends to infinity as time goes to infinity, which is an unrealistic behaviour considering finite populations. Intuitively, this reflects the fact that no stabilising force acts on the neutral trait distribution, such that random fluctuations play a major role in driving the dynamics of the stochastic process. Figure S10 shows how IBM trajectories significantly differ from Eq. (S5), and Fig. S11 illustrates how diversity metrics obtained from Eq. (S5) do not match those obtained from simulations of the IBM.

Setting with heterogeneous selection

In contrast to the neutral trait dynamics, the adaptive distribution can successfully be approximated by a deterministic description because selection pressure acts as a stabilising force and stabilises the populations' adaptive trait, dampening the

stochastic fluctuations. Consider the setting with heterogeneous selection and recall that in this setting where $x \equiv (s,u) \in \mathcal{X} = \mathcal{S} \times \mathcal{U}$ we define

$$b^{(i)}(x) \equiv b(1 - p(s - \theta_i)^2). \tag{S6}$$

By applying the same strategy as above to characterise the adaptive trait distribution $n_t^{(i)}(s)$ by choosing $\phi(n_t^{(i)}) = n_t^{(i)}(s) \equiv \int_{\mathcal{U}} n_t^{(i)}(u,s)du$, Eq. (S2) transforms into

$$\begin{aligned}\partial_t n_t^{(i)}(s) = n_t^{(i)}(s) \left[b^{(i)}(s)(1-m)(1-\mu) - \frac{1}{K}\int_{\mathcal{S}} n_t^{(i)}(\mathbf{s})\, d\mathbf{s}) \right] \\ + (1-m)\mu \int_{\mathcal{S}} b^{(i)}(\mathbf{s})\, n_t^{(i)}(\mathbf{s}) \mathcal{M}(\mathbf{s},s)\, d\mathbf{s} \\ + m\mu \sum_{j\neq i} \frac{a_{i,j}}{d_j} \int_{\mathbb{R}} b^{(j)}(\mathbf{s})\, n_t^{(j)}(s) \mathcal{M}(\mathbf{s},s) d\mathbf{s} \\ + m(1-\mu) \sum_{j\neq i} \frac{a_{i,j}}{d_j} b^{(j)}(s)\, n_t^{(j)}(s).\end{aligned} \tag{S7}$$

Assuming that the variance of the mutation kernel is small, one can use a diffusion approximation for the mutation term [8, 3, 10]

$$\int_{\mathcal{S}} b^{(i)}(\mathbf{s})\, n_t^{(i)}(\mathbf{s}) \mathcal{M}(\mathbf{s},s)\, d\mathbf{s} = b^{(i)}(s,t)\, n_t^{(i)}(s) + \tfrac{1}{2}\sigma_\mu^2 \Delta_s (b^{(i)} n_t^{(i)})(s). \tag{S8}$$

Neglecting the terms in $m\mu$, we obtain

$$\begin{aligned}\partial_t n_t^{(i)}(s) = n_t^{(i)}(s) \left[b^{(i)}(s,t)(1-m-\mu) - \frac{1}{K}\int_{\mathcal{S}} n_t^{(i)}(\mathbf{s})\, d\mathbf{s}) \right] \\ + \mu \left[b^{(i)}(s,t)\, n_t^{(i)}(s) + \tfrac{1}{2}\sigma_\mu^2 \Delta_s (b^{(i)} n_t^{(i)})(s) \right] \\ + m \sum_{j\neq i} b^{(j)}(s,t) n_t^{(j)}(s) a_{i,j}\end{aligned} \tag{S9}$$

which, after rearranging terms, yields the elegant deterministic approximation of the adaptive trait dynamics

$$\partial_t n_t^{(i)}(s) = n_t^{(i)}(s) \left[b^{(i)}(s)(1-m) - \frac{1}{K}\int_{\mathcal{S}} n_t^{(i)}(\mathbf{s}) d\mathbf{s} \right] + m \sum_{j\neq i} b^{(j)}(s) \frac{a_{i,j}}{d_j} n_t^{(j)}(s) + \tfrac{1}{2}\mu\sigma_\mu^2 \Delta_s \left[b^{(i)}(s) n_t^{(i)}(s) \right]. \tag{S10}$$

Setting $m = 0$ [10] shows that Eq. (S10) admits a stationary solution that is Gaussian, with variance $\sqrt{\mu}\sigma_\mu^2/\sqrt{p}$. Therefore, the variance of the adaptive trait distribution stabilises to a finite value. Intuitively, this reflects the fact that the random fluctuations of the adaptive trait distribution are dampened by the stabilising force of selection. Provided that the selection strength p is large enough, Eq. (S10) is a good approximation of the adaptive trait distribution obtained from the stochastic

process. Figure S3 shows how IBM trajectories are similar to the ones obtained from Eq. (S5), and Fig. S4 illustrates how diversity metrics obtained from Eq. (S5) match those obtained from simulations of the IBM.

2.A.3 Trait-dependent competition

To test whether the effects of the metrics hold under more complex ecological processes, we designed an extra experiment considering heterogeneous selection and adaptive trait-dependent competition, where the death rate of individuals on v_i with traits $x_k^{(i)} = (u_k^{(i)}, s_k^{(i)}) \in \mathcal{U} \times \mathcal{S}$ is given by

$$d(x_k^{(i)}, \nu^{(i)}) = \frac{1}{K} \int_{\mathcal{S}} \exp\Big(-\frac{(s_k^{(i)} - \mathbf{s})^2}{2\sigma_\alpha^2}\Big) \nu^{(i)}(\mathbf{s}) \tag{S11}$$

where σ_α is the competition bandwidth. This competition kernel tends to increase the population size, as it decreases the overall competition. The adaptive dynamics theory predicts that when $m = 0$, competition promotes two distinct types of individuals at either side of the adaptive trait optimum for a competition bandwidth $\sigma_\alpha < 1/\sqrt{2p}$, while a single type is observed when $\sigma_\alpha > 1/\sqrt{2p}$ [5]. We performed simulations in both cases for graphs with $M = 7$ vertices and show results of the multivariate regression analyses in Table S5. The analyses demonstrate that the trends reported in the main manuscript remain unchanged in both cases.

2.A.4 Derivation of the habitat assortativity metric r_Θ in binary environments

We demonstrate here how the habitat assortativity r_Θ relates to the conditional probability of habitats being connected, and we show how r_Θ simplifies under mean field assumption.

Following the original definition of [11], habitat assortativity r_Θ is defined as the Pearson correlation of environmental conditions θ at either ends of the vertices V of graph G, that is

$$r_\Theta = \frac{\mathrm{Cov}(\Theta_\times, \Theta_\wedge)}{\sqrt{\mathrm{Var}(\Theta_\times)\mathrm{Var}(\Theta_\wedge)}} = \frac{\langle \Theta_\times \Theta_\wedge \rangle - \langle \Theta_\times \rangle \langle \Theta_\wedge \rangle}{\sqrt{(\langle \Theta_\times^2 \rangle - \langle \Theta_\times \rangle^2)(\langle \Theta_\wedge^2 \rangle - \langle \Theta_\wedge \rangle^2}} \tag{S12}$$

where $\Theta_\times$ and $\Theta_\wedge$ denote the sets of environmental conditions found at the toe and tip of each directed vertex of graph V, and $\langle\Theta_\times\rangle$ and $\langle\Theta_\wedge\rangle$ denote their respective mean values.

Let $P(\bullet,\bullet)$ be the proportion of edges that connect a vertex of habitat type $\bullet$ to a vertex of habitat type $\bullet$. One can also view $P(\bullet,\bullet)$ as the conditional probability that a vertex of type $\bullet$ is connected to a vertex of type $\bullet$. Let $P(\bullet)$ denote the proportion of vertices that are of type $\bullet$. First observe that for undirected graphs, one has $\langle\Theta_\times\rangle = \langle\Theta_\wedge\rangle$ and $\langle\Theta_\times^2\rangle = \langle\Theta_\wedge^2\rangle$. Assuming that habitats are symmetric and binary, it follows that $\theta_\bullet = -\theta_\bullet$. Then

$$\begin{aligned}\langle\Theta_\times\Theta_\wedge\rangle &= P(\bullet,\bullet)\theta_\bullet^2 + P(\bullet,\bullet)\theta_\bullet^2 + [P(\bullet,\bullet) + P(\bullet,\bullet)]\theta_\bullet\theta_\bullet \\ &= \theta_\bullet^2\left(P(\bullet,\bullet) + P(\bullet,\bullet) - [P(\bullet,\bullet) + P(\bullet,\bullet)]\right),\end{aligned} \tag{S13}$$

$$\begin{aligned}\langle\Theta_\times\rangle &= P(\bullet)\theta_\bullet + P(\bullet)\theta_\bullet \\ &= \theta_\bullet\left[P(\bullet) - P(\bullet)\right],\end{aligned} \tag{S14}$$

$$\begin{aligned}\langle\Theta_\times^2\rangle &= P(\bullet)\theta_\bullet^2 + P(\bullet)\theta_\bullet^2 \\ &= \theta_\bullet^2\left[P(\bullet) + P(\bullet)\right] \\ &= \theta_\bullet^2.\end{aligned} \tag{S15}$$

Combining Eq. (S13), Eq. (S14) and Eq. (S15) with Eq. (S12) one gets

$$\begin{aligned}r_\Theta &= \frac{\langle\Theta_\times\Theta_\wedge\rangle - \langle\Theta_\times\rangle\langle\Theta_\wedge\rangle}{\langle\Theta_\times^2\rangle - \langle\Theta_\times\rangle^2} \\ &= \frac{P(\bullet,\bullet) + P(\bullet,\bullet) - [P(\bullet,\bullet) + P(\bullet,\bullet)] - (P(\bullet) - P(\bullet))^2}{P(\bullet) + P(\bullet) - (P(\bullet) - P(\bullet))^2} \\ &= \frac{P(\bullet,\bullet) + P(\bullet,\bullet) - [P(\bullet,\bullet) + P(\bullet,\bullet)] - (P(\bullet) - P(\bullet))^2}{1 - (P(\bullet) - P(\bullet))^2}.\end{aligned} \tag{S16}$$

Assuming that habitats are homogeneously distributed, we have $P(\bullet) = P(\bullet) = \frac{1}{2}$ and thus we obtain

$$r_\Theta = P(\bullet,\bullet) + P(\bullet,\bullet) - [P(\bullet,\bullet) + P(\bullet,\bullet)]. \tag{S17}$$

The mean field approximation involves the assumption that all vertices with similar habitats are equivalent in terms of their connections with other habitats, so that $P(\bullet,\bullet) = P(\bullet,\bullet)$ and $P(\bullet,\bullet) = P(\bullet,\bullet)$, which yields $r_\Theta = 2\left(P(\bullet,\bullet) - P(\bullet,\bullet)\right)$.

References

[1] V. Bansaye and S. Méléard. “Some stochastic models for structured populations : scaling limits and long time behavior”. In: *Stochastic Models for Structured Populations: Scaling Limits and Long Time Behavior* (2015), pp. 1–107. DOI: 10.1007/978-3-319-21711-6. arXiv: 1506.04165.

[2] N. Champagnat, R. Ferrière, and S. Méléard. “Unifying evolutionary dynamics: From individual stochastic processes to macroscopic models”. In: *Theoretical Population Biology* 69.3 (2006), pp. 297–321. DOI: 10.1016/j.tpb.2005.10.004.

[3] F. Débarre, O. Ronce, and S. Gandon. “Quantifying the effects of migration and mutation on adaptation and demography in spatially heterogeneous environments”. In: *Journal of Evolutionary Biology* 26.6 (2013), pp. 1185–1202. DOI: 10.1111/jeb.12132.

[4] W.-N. Ding et al. “Ancient orogenic and monsoon-driven assembly of the world’s richest temperate alpine flora”. In: *Science* 369.6503 (2020), pp. 578–581. DOI: 10.1126/science.abb4484.

[5] M. Doebeli. *Adaptive diversification*. Monographs in population biology. Princeton, N.J: Princeton University Press, 2011.

[6] M. Jung et al. “A global map of terrestrial habitat types”. In: *Scientific Data* 7.1 (2020), p. 256. DOI: 10.1038/s41597-020-00599-8.

[7] D. N. Karger et al. “Climatologies at high resolution for the earth’s land surface areas”. In: *Scientific Data* 4.1 (2017), p. 170122. DOI: 10.1038/sdata.2017.122. arXiv: 1607.00217.

[8] M. Kimura. “A stochastic model concerning the maintenance of genetic variability in quantitative characters.” In: *Proceedings of the National Academy of Sciences of the United States of America* 54.3 (1965), pp. 731–736. DOI: 10.1073/pnas.54.3.731.

[9] H. Linke. “Applications of Brownian Motion”. In: 5114 (2015), pp. 199–213. DOI: 10.1142/9789814678940_0009.

[10] S. Mirrahimi and S. Gandon. “Evolution of specialization in heterogeneous environments: equilibrium between selection, mutation and migration”. In: *Genetics* 214.2 (2020), pp. 479–491. DOI: 10.1534/genetics.119.302868.

[11] M. E. J. Newman. “Mixing patterns in networks”. In: *Physical Review E* 67.2 (2003), p. 026126. DOI: 10.1103/PhysRevE.67.026126. arXiv: 0209450 [cond-mat].

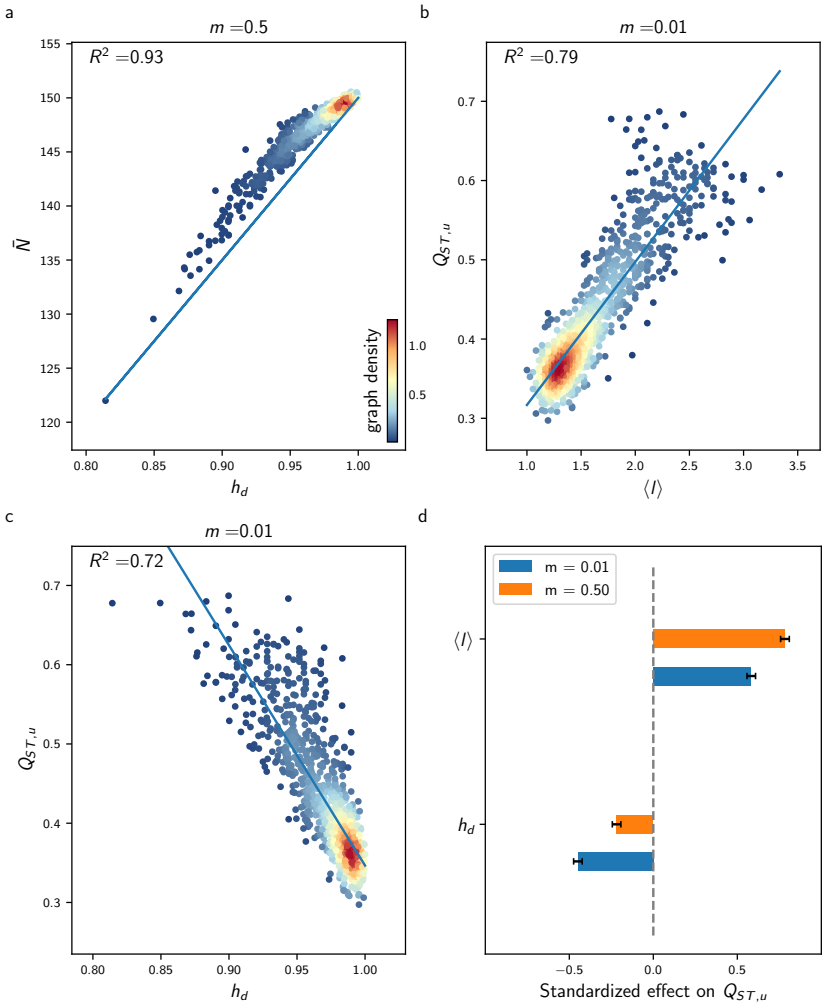

Fig. S1: Effect of $\langle l \rangle$ and h_d on average population size $\bar{N}$ and neutral differentiation $Q_{ST,u}$ under the setting with no selection, analogous to Fig. 2.2 but for 1126 of the 261,080 undirected connected graphs with $M = 9$ vertices.

2.B Supplementary Figures

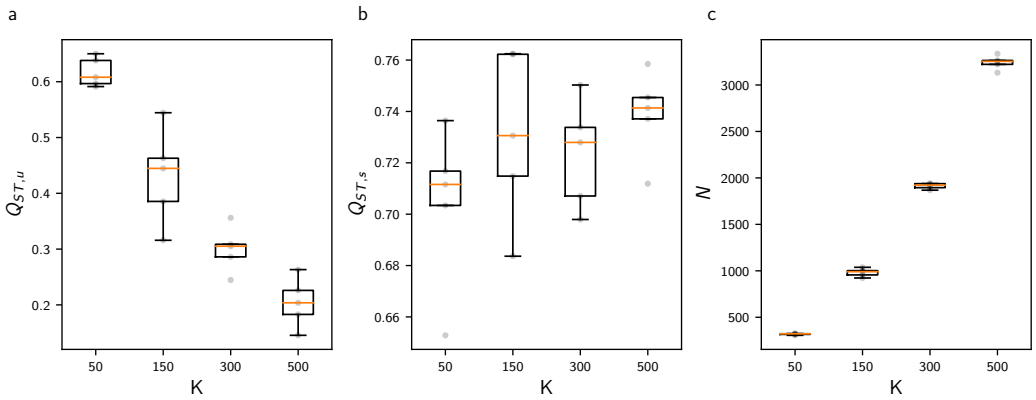

Fig. S2: Effect of the carrying capacity K on $Q_{ST,u}$, $Q_{ST,s}$ and metapopulation size N for the line graph with $M = 7$ vertices for $m = 0.1$. Decreasing K increases $Q_{ST,u}$ as it favours drift, but it does not influence $Q_{ST,s}$. Each boxplot is based on 5 replicate simulations of the IBM, and fade dots represent single values for each replicate.

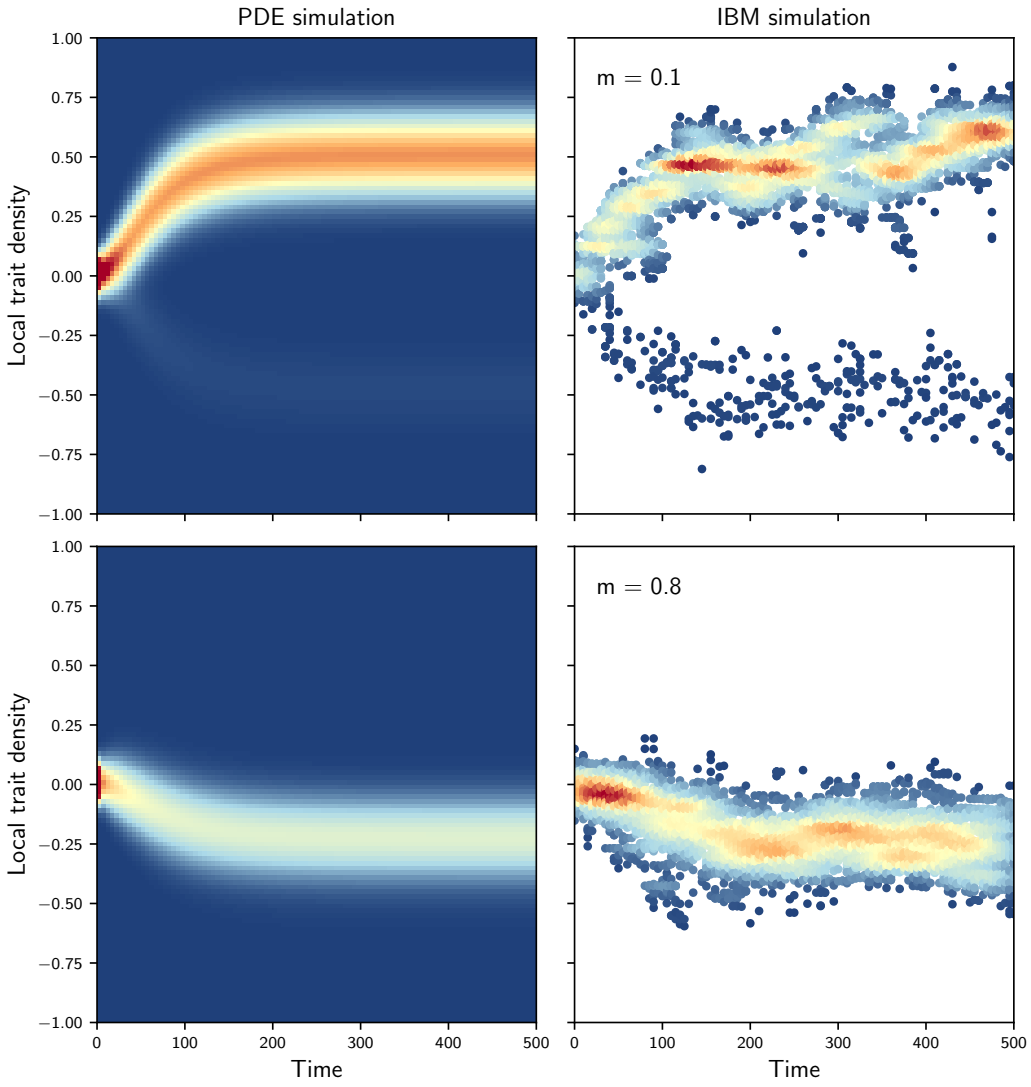

Fig. S3: Comparison of the adaptive trait density on one vertex obtained from Eq. (S10) (left) and from the IBM simulations (right) in the setting with heterogeneous selection, for the star graph with $M = 7$ vertices. The densities obtained from Eq. (S10) and from the IBM are qualitatively similar.

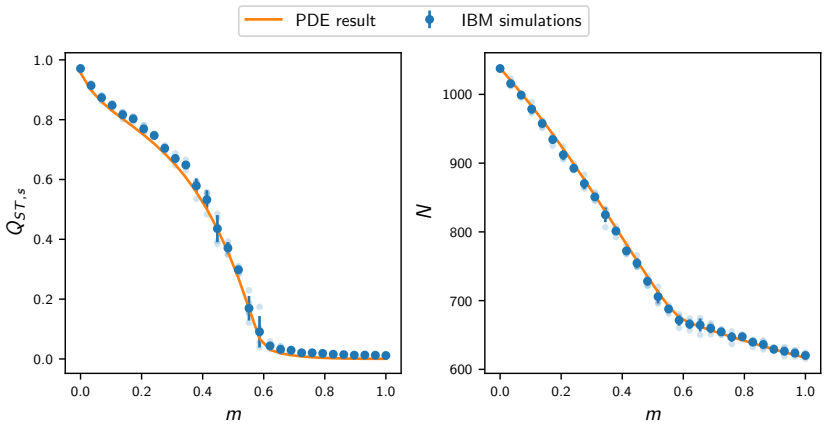

Fig. S4: Comparison of $Q_{ST,s}$ and N obtained from the deterministic approximation Eq. (S10) and from IBM simulations in the setting with heterogeneous selection, on the star graph with $M = 7$ vertices. $Q_{ST,s}$ and population size obtained from Eq. (S10) closely match the IBM simulations. Each plain dot represents average results from 5 replicate simulations of the IBM, bars represent one standard deviation, and each fade dot represents a single replicate value.

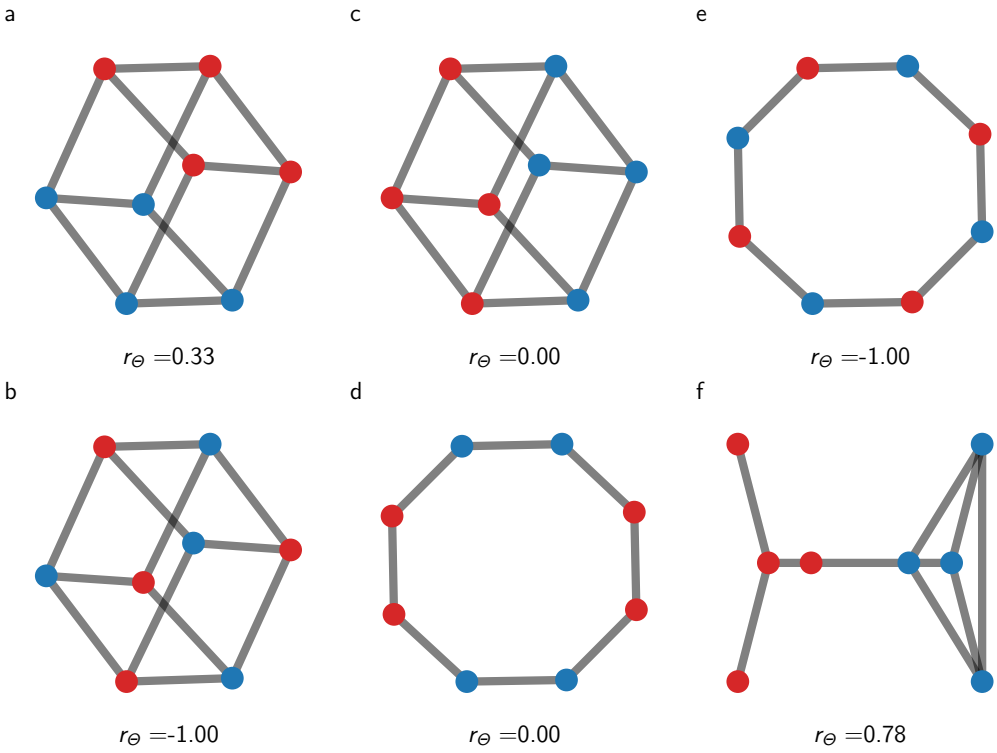

Fig. S5: Graphs with spatial distribution of habitat types corresponding to different habitat assortativity r_Θ. Graphs (ad) can be described exactly with a mean field approach, as blue and red vertices have an equivalent position on the graph.

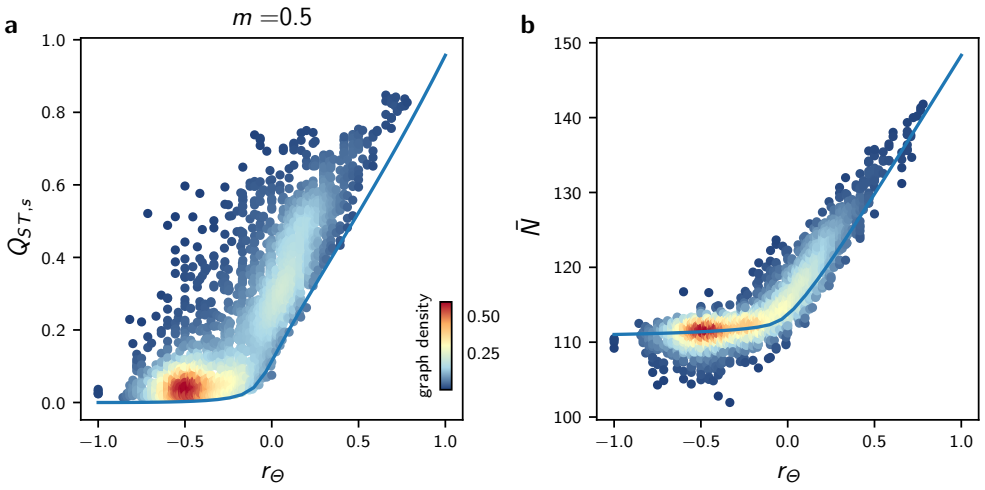

Fig. S6: Effects of habitat heterogeneity r_Θ on $Q_{ST,s}$ and average population size $\bar{N}$ for all undirected connected graphs with $M = 7$ vertices and varying r_Θ, obtained for similar simulations to those in Fig. 2.4 with $m = 0.5$. In (a) and (b), each dot represents average results from 5 replicate simulations of the IBM, the colour scale corresponds to the proportion of the graph with similar x and y axis values (graph density), and the blue lines correspond to results obtained from the mean field, deterministic approximation Eq. (2.5). Deviations from the mean field, deterministic approximation Eq. (2.5) can be explained by differences in $\langle l \rangle$ and h_d between the graphs.

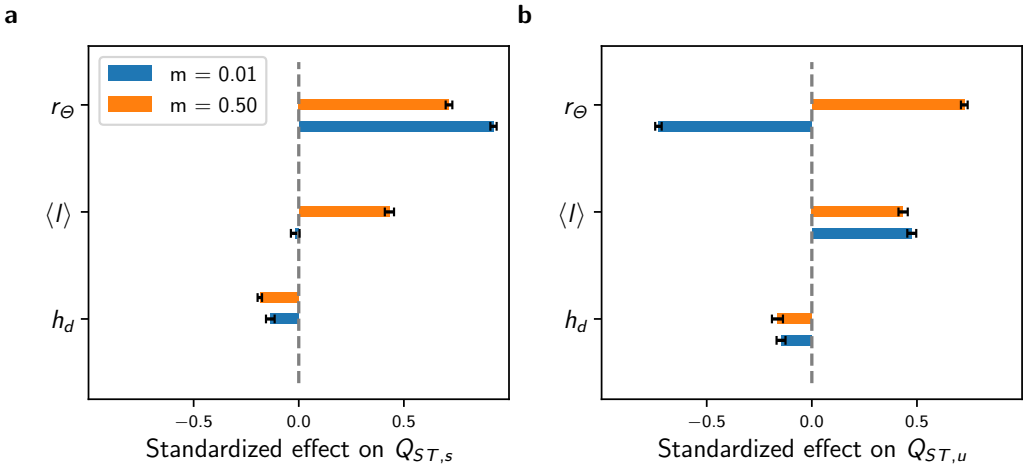

Fig. S7: Standardized effects of h_d, $\langle l \rangle$ and r_Θ on $Q_{ST,s}$ and $Q_{ST,u}$ obtained from multivariate regression models independently fitted for low and high migration regimes on average results from 5 replicate simulations of the IBM, analogous Fig. 2.5c–d but for 1126 of the 261,080 undirected connected graphs with $M = 9$ vertices and varying r_Θ (see Machine learning framework for ecosystem models for details). Error bars show 95% confidence intervals.

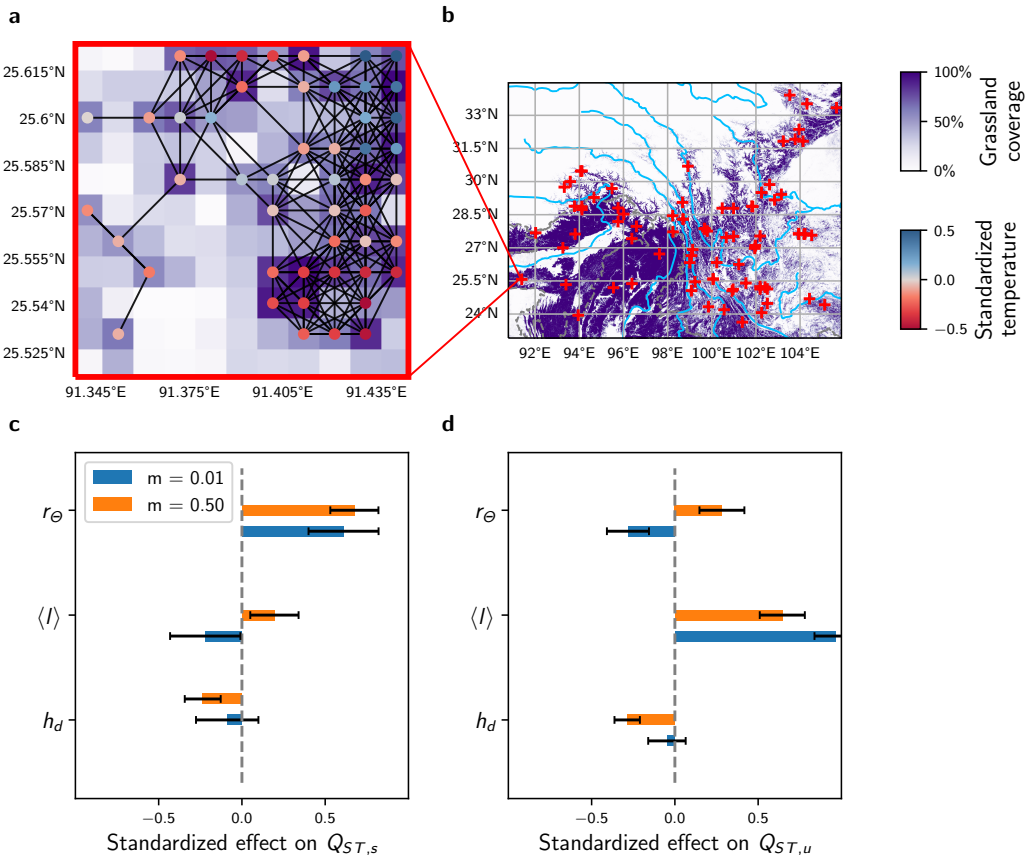

Fig. S8: Simulations on graphs with $M = 49$ vertices obtained from real spatial habitat datasets, in the setting with heterogeneous selection. The region from where graphs are obtained is centred on the Hengduan Mountains in Southwest China, one of the most species-rich temperate mountain biota globally [4]. (a) Graphical representation of a geographical area of size $0.11° \times 0.11°$. To create the graph, we considered biological populations living in grasslands, and used the dataset provided in [6] containing global grassland coverage at $0.01°$ resolution. We assigned a vertex to a geographical area of size $0.01° \times 0.01°$ if its grassland coverage was above a threshold arbitrarily set to 50%. We further assumed that two vertices were connected if their euclidean distance was below a certain dispersal range, which we let vary from 1 to 2.5 km. Local annual average temperature was considered as the value that captures environmental conditions at each vertex. Temperature data was obtained from the CHELSA dataset [7]. (b) Grassland coverage for the considered region. Blue lines correspond to rivers and dashed grey lines correspond to country borders. Red crosses indicate the locations of the 83 graphs sampled for the simulations used in (c–d). (c–d) Standardized effects of h_d, $\langle l \rangle$ and r_Θ on $Q_{ST,s}$ and $Q_{ST,u}$ obtained from multivariate regression models independently fitted for low and high migration regimes to average results from 5 replicate simulations of the IBM on the 83 graphs which location is illustrated in (c) (see Table S4 for simulation details). Error bars show 95% confidence intervals.

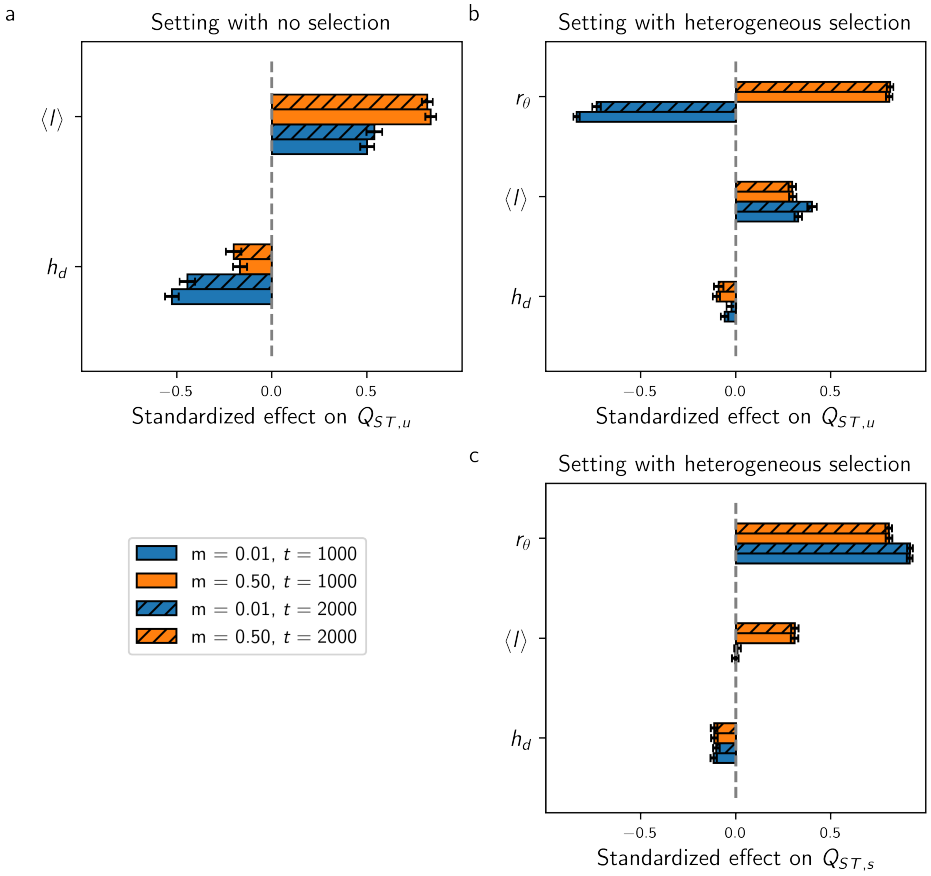

Fig. S9: Standardized effects of h_d, $\langle l \rangle$ and r_Θ on $Q_{ST,u}$ in the setting with no selection and in the setting with heterogeneous selection for the time horizons $t = 1000$ and $t = 2000$, obtained from multivariate regression models independently fitted for low and high migration regimes to average results from 5 replicate simulations of the IBM on all undirected connected graphs with $M = 7$ vertices and varying r_Θ (see Machine learning framework for ecosystem models for details). (a–c) illustrate that the effects of the topology metrics on $Q_{ST,u}$ and $Q_{ST,s}$ remain constant for $t > 1000$ in both the settings without selection and with heterogeneous selection. Error bars show 95% confidence intervals.

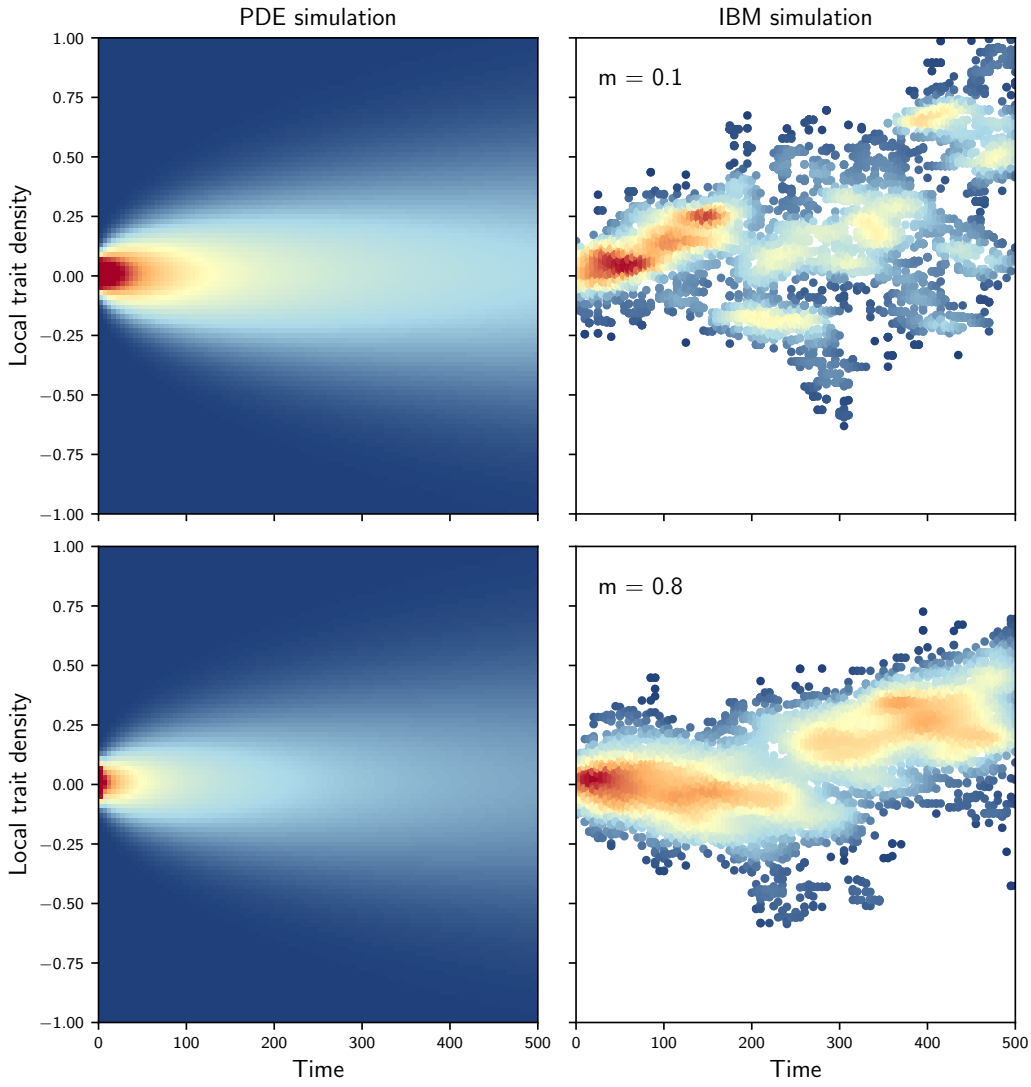

Fig. S10: Comparison of the neutral trait density on one vertex obtained from Eq. (S5) (left) and from the IBM simulations (right) in the setting with no selection, for the chain graph. The densities obtained from Eq. (S5) and from the IBM are dissimilar.

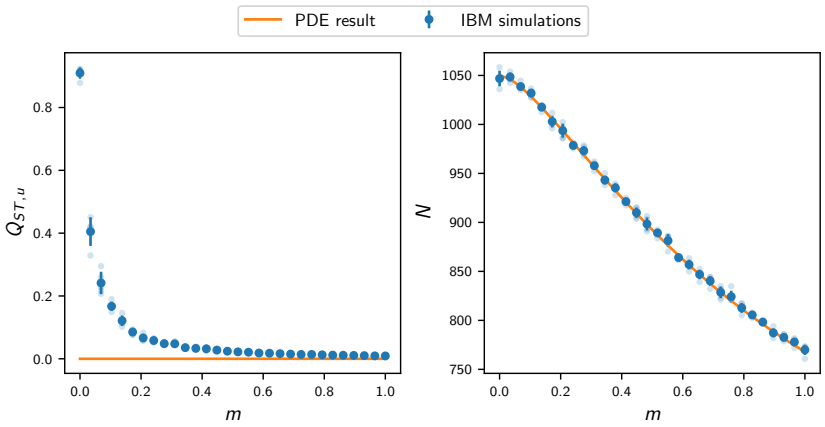

Fig. S11: Comparison of results obtained from the deterministic approximations Eqs. (S4) and (S5) and from IBM simulations in the setting with no selection, on the star graph with $M = 7$ vertices. While Eq. (S4) can capture population size, Eq. (S5) is not able to capture $Q_{ST,u}$. Each plain dot represents average results from 5 replicate simulations, bars represent one standard deviation, and each fade dot represents a single replicate value.

2.C Supplementary Tables

Tab. S1: Linear regression model coefficients for the effect of topology metrics on $Q_{ST,u}$ in the setting with no selection, based on all graphs with $M = 7$ vertices. *** $P < 0.001$

	$Q_{ST,u}$				$Q_{ST,u} - bN$	
m	0.01	0.50	0.01	0.50	0.01	0.50
(Intercept)	0.000	-0.000	-0.000	-0.000	-0.000	-0.000
	(0.023)	(0.017)	(0.023)	(0.025)	(0.023)	(0.028)
$\langle l \rangle$	0.739***	0.872***				
	(0.023)	(0.017)				
h_d			-0.753***	-0.674***	-0.753***	-0.143***
			(0.023)	(0.025)	(0.023)	(0.028)
Number of sim.	853	853	853	853	853	853
R^2	0.546	0.760	0.567	0.454	0.567	0.030

Tab. S2: Multivariate linear regression model coefficients for the effect of topology metrics on $Q_{ST,u}$ in the setting with no selection. *** $P < 0.001$

	$M = 7$		$M = 9$	
	$Q_{ST,u}$			
m	0.01	0.50	0.01	0.50
(Intercept)	-0.000	-0.000	0.000	-0.000
	(0.017)	(0.013)	(0.009)	(0.010)
h_d	-0.527***	-0.352***	-0.449***	-0.218***
	(0.019)	(0.014)	(0.013)	(0.013)
$\langle l \rangle$	0.500***	0.712***	0.583***	0.784***
	(0.019)	(0.014)	(0.013)	(0.013)
Number of sim.	853	853	1,126	1,126
R^2	0.766	0.858	0.899	0.896

Tab. S3: Multivariate linear regression model coefficients for the effect of the topology metrics on $Q_{ST,u}$ and $Q_{ST,s}$ in the setting with heterogeneous selection. *** $P < 0.001$

	$M = 7$				$M = 9$			
	$Q_{ST,s}$		$Q_{ST,u}$		$Q_{ST,s}$		$Q_{ST,u}$	
m	0.01	0.50	0.01	0.50	0.01	0.50	0.01	0.50
(Intercept)	-0.000	-0.000	-0.000	-0.000	0.000	0.000	0.000	0.000
	(0.008)	(0.009)	(0.009)	(0.009)	(0.008)	(0.008)	(0.008)	(0.008)
h_d	-0.117***	-0.114***	-0.060***	-0.102***	-0.135***	-0.185***	-0.146***	-0.164***
	(0.009)	(0.010)	(0.010)	(0.010)	(0.010)	(0.011)	(0.011)	(0.011)
$\langle l \rangle$	-0.004	0.308***	0.328***	0.300***	-0.017	0.431***	0.475***	0.434***
	(0.009)	(0.010)	(0.010)	(0.010)	(0.010)	(0.011)	(0.011)	(0.011)
r_Θ	0.914***	0.805***	-0.838***	0.807***	0.926***	0.715***	-0.730***	0.725***
	(0.008)	(0.009)	(0.009)	(0.009)	(0.008)	(0.008)	(0.008)	(0.008)
Number of sim.	2,548	2,548	2,548	2,548	2,250	2,250	2,250	2,250
R^2	0.845	0.808	0.808	0.799	0.870	0.853	0.862	0.851

Tab. S4: Multivariate linear regression model coefficients for the effect of topology metrics on $Q_{ST,u}$ and $Q_{ST,s}$ on real graphs with $M = 49$ vertices in the setting with heterogeneous selection. * $P < 0.05$, ** $P < 0.01$, *** $P < 0.001$

	$Q_{ST,s}$		$Q_{ST,u}$	
m	0.1	0.50	0.1	0.50
(Intercept)	-0.000	-0.000	0.000	-0.000
	(0.093)	(0.064)	(0.056)	(0.059)
h_d	-0.088	-0.235***	-0.048	-0.286***
	(0.094)	(0.065)	(0.057)	(0.060)
$\langle l \rangle$	-0.220*	0.195**	0.965***	0.645***
	(0.106)	(0.073)	(0.064)	(0.068)
r_Θ	0.610***	0.675***	-0.282***	0.282***
	(0.106)	(0.073)	(0.063)	(0.068)
Number of sim.	83	83	83	83
R^2	0.313	0.675	0.752	0.717

Tab. S5: Multivariate linear regression model coefficients for the effect of topology metrics on $Q_{ST,u}$ and $Q_{ST,s}$ in the setting of trait-dependent competition and heterogeneous selection (Section 2.A.3), based on all graphs with $M = 7$ vertices. *** $P < 0.001$

	$\sigma_a = 0.5 < 1/\sqrt{2p}$				$\sigma_a = 1 > 1/\sqrt{2p}$			
	$Q_{ST,s}$		$Q_{ST,u}$		$Q_{ST,s}$		$Q_{ST,u}$	
m	0.05	0.50	0.05	0.50	0.05	0.50	0.05	0.50
(Intercept)	0.000	-0.000	-0.000	-0.000	0.000	-0.000	0.000	-0.000
	(0.005)	(0.010)	(0.011)	(0.010)	(0.004)	(0.008)	(0.012)	(0.007)
h_d	-0.228***	-0.118***	-0.171***	-0.169***	-0.166***	-0.128***	-0.178***	-0.139***
	(0.006)	(0.011)	(0.012)	(0.012)	(0.004)	(0.009)	(0.013)	(0.008)
$\langle l \rangle$	0.084***	0.373***	0.461***	0.573***	0.002	0.296***	0.483***	0.286***
	(0.006)	(0.011)	(0.012)	(0.012)	(0.004)	(0.009)	(0.013)	(0.008)
r_Θ	0.922***	0.741***	-0.657***	0.508***	0.967***	0.816***	-0.585***	0.837***
	(0.005)	(0.010)	(0.011)	(0.010)	(0.004)	(0.008)	(0.012)	(0.007)
Number of sim.	2,548	2,548	2,548	2,548	2,548	2,548	2,548	2,548
R^2	0.934	0.768	0.716	0.732	0.962	0.828	0.659	0.861

3

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

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

[1] Unit of Land Change Science, Swiss Federal Research Institute for Forest, Snow and Landscape (WSL), Switzerland

[2] Landscape Ecology, Institute of Terrestrial Ecosystems, Department of Environmental Systems Science, ETH Zürich,

[3] Risklab, Department of Mathematics, ETH Zürich,

[4] School of Data Science and Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China,

[5] Applied Mathematics: Institute for Analysis and Numerics, Faculty of Mathematics and Computer Science, University of Münster,

arXiv:2205.03672

Under review at Journal of Partial Differential Equations and Applications

Nonlinear partial differential equations (PDEs) are used to model dynamical processes in a large number of scientific fields, ranging from finance to biology. In many applications standard local models are not sufficient to accurately account for certain non-local phenomena such as, e.g., interactions at a distance. In order to properly capture these phenomena non-local nonlinear PDE models are frequently employed in the literature. In this article we propose two numerical methods based on machine learning and on Picard iterations, respectively, to approximately solve non-local nonlinear PDEs. The proposed machine learning-based method is an extended variant of a deep learning-based splitting-up type approximation method previously introduced in the literature and utilizes neural networks to provide approximate solutions on a subset of the spatial domain of the solution. The Picard iterations-based method is an extended variant of the so-called full history recursive multilevel Picard *approximation scheme previously introduced in the literature and provides an approximate solution for a single point of the domain. Both methods are mesh-free and allow non-local nonlinear PDEs with Neumann boundary conditions to be solved in high dimensions. In the two methods, the numerical difficulties arising due to the dimensionality of the PDEs are avoided by (i) using the correspondence between the expected trajectory of reflected stochastic processes and the solution of PDEs (given by the Feynman–Kac formula) and by (ii) using a plain vanilla Monte Carlo integration to handle the non-local term. We evaluate the performance of the two methods on five different PDEs arising in physics and biology. In all cases, the methods yield good results in up to 10 dimensions with short run times. Our work extends recently developed methods to overcome the curse of dimensionality in solving PDEs.*

3.1 Introduction

In this article, we derive numerical schemes to approximately solve high-dimensional non-local nonlinear partial differential equations (PDEs) with Neumann boundary conditions. Such PDEs have been used to describe a variety of processes in physics, engineering, finance, and biology, but can generally not be solved analytically, requiring numerical methods to provide approximate solutions. However, traditional numerical methods are for the most part computationally infeasible for high-dimensional problems, calling for the development of novel approximation methods.

The need for solving non-local nonlinear PDEs has been expressed in various fields as they provide a more general description of the dynamical systems than their local counterparts [63, 29, 90]. In physics and engineering, non-local nonlinear PDEs are found, e.g., in models of Ohmic heating production [66], in the investigation of the fully turbulent behavior of real flows [20], in phase field models allowing non-local interactions [7, 41, 25, 49], or in phase transition models with conservation of mass [86, 88]; see [63] for further references. In finance, non-local PDEs are used, e.g., in jump-diffusion models for the pricing of derivatives where the dynamics of stock prices are described by stochastic processes experiencing large jumps [74, 22, 65, 1, 15, 90, 28, 26]. Penalty methods for pricing American put options such as in Kou's jump-diffusion model [58, 42], considering large investors where the agent policy affects the assets prices [5, 1], or considering default risks [83, 55] can further introduce nonlinear terms in non-local PDEs. In economics, non-local nonlinear PDEs appear, e.g., in evolutionary game theory with the so-called replicator-mutator equation capturing continuous strategy spaces [79, 62, 50, 3, 4] or in growth models where consumption is non-local [6]. In biology, non-local nonlinear PDEs are used, e.g., to model processes determining the interaction and evolution of organisms. Examples include models of morphogenesis and cancer evolution [71, 24, 91], models of gene regulatory networks [80], population genetics models with the non-local Fisher–Kolmogorov–Petrovsky–Piskunov (Fisher–KPP) equations [38, 51, 18, 82, 17, 57, 92], and quantitative genetics models where populations are structured on a phenotypic and/or a geographical space [19, 43, 16, 76, 77, 85, 30, 78]. In such models, Neumann boundary conditions are used, e.g., to model the effect of the borders of the geographical domain on the movement of the organisms.

Real world systems such as those just mentioned may be of considerable complexity and accurately capturing the dynamics of these systems may require models of high dimensionality [30], leading to complications in obtaining numerical approximations. For example, the number of dimensions of the PDEs may correspond in finance to the number of financial assets (such as stocks, commodities, exchange rates, and interest rates) in the involved portfolio; in evolutionary dynamics, to the dimension of the strategy space; and in biology, to the number of genes modelled [80] or to the dimension of the geographical or the phenotypic space over which the organisms are structured. Standard approximation methods for PDEs such as finite difference approximation methods, finite element methods, spectral Galerkin approximation methods, and sparse grid approximation methods all suffer from the so called

curse of dimensionality [14], meaning that their computational costs increase exponentially in the number of dimensions of the PDE under consideration.

Numerical methods exploiting stochastic representations of the solutions of PDEs can in some cases overcome the curse of dimensionality. Specifically, simple Monte Carlo averages of the associated stochastic processes have been proposed a long time ago to solve high-dimensional linear PDEs, such as, e.g., Black–Scholes and Kolmogorov PDEs [75, 8]. Recently, two novel classes of methods have proved successful in dealing with high-dimensional nonlinear PDEs, namely deep learning-based and full history recursive multilevel Picard approximation methods (in the following we will abbreviate *full history recursive multilevel Picard* by MLP). The explosive success of deep learning in recent years across a wide range of applications [69] has inspired a variety of neural network-based approximation methods for high-dimensional PDEs; see [10] for an overview. One class of such methods is based on reformulating the PDE as a stochastic learning problem through the Feynman–Kac formula [33, 52, 12]. In particular, the *deep splitting* scheme introduced in [11] relies on splitting the differential operator into a linear and a nonlinear part and in that sense belongs to the class of splitting-up methods [27, 48, 56]. The PDE approximation problem is then decomposed along the time axis into a sequence of separate learning problems. The deep splitting approximation scheme has proved capable of computing reasonable approximations to the solutions of nonlinear PDEs in up to 10000 dimensions. On the other hand, the MLP approximation method, introduced in [36, 60, 35], utilizes the Feynman–Kac formula to reformulate the PDE problem as a fixed point equation. It further reduces the complexity of the numerical approximation of the time integral through a multilevel Monte Carlo approach. However, neither the deep splitting nor the MLP method can, until now, account for non-localness and Neumann boundary conditions.

The goal of this article is to overcome these limitations and thus we generalize the deep splitting method and the MLP approximation method to approximately solve non-local nonlinear PDEs with Neumann boundary conditions. We handle the non-local term by a plain vanilla Monte Carlo integration and address Neumann boundary conditions by constructing reflected stochastic processes. While the MLP method can, in one run, only provide an approximate solution at a single point $x \in D$ of the spatial domain $D \subseteq \mathbb{R}^d$ where $d \in \mathbb{N} = \{1, 2, \dots\}$, the machine learning-based method can in principle provide an approximate solution on a full subset of the spatial domain D (however, cf., e.g., [53, 54, 46] for results on limitations on the performance of such approximation schemes). We use both methods to solve five non-local nonlinear PDEs arising in models from biology and physics and cross-validate the results of the simulations. We manage to solve the non-local nonlinear PDEs with reasonable accuracy in up to 10 dimensions.

For an account of classical numerical methods for solving non-local PDEs, such as finite differences, finite elements, and spectral methods, we refer the reader to the recent survey [29]. Several machine-learning based schemes for solving non-local PDEs can also be found in the literature. In particular, the *physics-informed neural network* and *deep Galerkin* approaches [84, 87], based on representing an approximation of the whole solution of the PDE as a neural network and using automatic differentiation to do a least-squares

minimization of the residual of the PDE, have been extended to fractional PDEs and other non-local PDEs [81, 72, 47, 2, 94]. While some of these approaches use classical methods susceptible to the curse of dimensionality for the non-local part [81, 72], mesh-free methods suitable for high-dimensional problems have also been investigated [47, 2, 94].

The literature also contains approaches that are more closely related to the machine learning-based algorithm presented here. Frey & Köck [39, 40] propose an approximation method for non-local semilinear parabolic PDEs with Dirichlet boundary conditions based on and extending the deep splitting method in [11] and carry out numerical simulations for example PDEs in up to 4 dimensions. Castro [21] proposes a numerical scheme for approximately solving non-local nonlinear PDEs based on [59] and proves convergence results for this scheme. Finally, Gonon & Schwab [45] provide theoretical results showing that neural networks with ReLU activation functions have sufficient expressive power to approximate solutions of certain high-dimensional non-local linear PDEs without the curse of dimensionality.

There is a more extensive literature on machine learning-based methods for approximately solving standard PDEs without non-local terms but with various boundary conditions, going back to early works by Lagaris et al. [68, 67] (see also [73]), which employed a grid-based method based on least-squares minimization of the residual and shallow neural networks to solve low-dimensional ODEs and PDEs with Dirichlet, Neumann, and mixed boundary conditions. More recently, approximation methods for PDEs with Neumann (and other) boundary conditions have been proposed using, e.g., physics-informed neural networks [72, 89, 93], the *deep Ritz* method (based on a variational formulation of certain elliptic PDEs) [34, 70, 23], or adversarial networks [95].

The remainder of this article is organized as follows. Section 3.2 discusses a special case of the proposed machine learning-based method, in order to provide a readily comprehensible exposition of the key ideas of the method. Section 3.3 discusses the general case, which is flexible enough to cover a larger class of PDEs and to allow more sophisticated optimization methods. Section 3.4 presents our extension of the MLP approximation method to non-local nonlinear PDEs, which we use to obtain reference solutions in Section 3.5. Section 3.5 provides numerical simulations for five concrete examples of (non-local) nonlinear PDEs.

3.2 Machine learning-based approximation method in a special case

In this section, we present in Lemma 3.2.2 in Section 3.2.3 below a simplified version of our general machine learning-based algorithm for approximating solutions of non-local nonlinear PDEs with Neumann boundary conditions proposed in Section 3.3 below. This simplified version applies to a smaller class of non-local heat PDEs, specified in Section 3.2.1 below. In Section 3.2.2 we introduce some notation related to the reflection of straight lines on the boundaries of a suitable subset $D \subseteq \mathbb{R}^d$ where $d \in \mathbb{N}$, which will be used to describe time-discrete reflected stochastic processes that are employed in our approximations

throughout the rest of the article. The simplified algorithm described in Section 3.2.3 below is limited to using neural networks of a particular architecture that are trained using plain vanilla stochastic gradient descent, whereas the full version proposed in Lemma 3.3.1 in Section 3.3.2 below is formulated in such a way that it encompasses a wide array of neural network architectures and more sophisticated training methods, in particular Adam optimization, minibatches, and batch normalization. Stripping away some of these more intricate aspects of the full algorithm is intended to exhibit more acutely the central ideas in the proposed approximation method.

The simplified algorithm described in this section as well as the more general version proposed in Lemma 3.3.1 in Section 3.3.2 below are based on the deep splitting method introduced in Beck et al. [11], which combines operator splitting with a previous deep learning-based approximation method for Kolmogorov PDEs [12]; see also Beck et al. [10, Sections 2 and 3] for an exposition of these methods.

3.2.1 Partial differential equations (PDEs) under consideration

Let $T \in (0, \infty)$, $d \in \mathbb{N}$, let $\mathbb{D} \subseteq \mathbb{R}^d$ be a closed set with sufficiently smooth boundary $\partial_{\mathbb{D}}$, let $\mathbf{n} \colon \partial_{\mathbb{D}} \to \mathbb{R}^d$ be an outer unit normal vector field associated to $\mathbb{D}$, let $g \in C(D, \mathbb{R})$, let $\nu_x \colon \mathcal{B}(\mathbb{D}) \to [0,1]$, $x \in \mathbb{D}$, be probability measures, let $f \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be measurable, let $u = (u(t,x))_{(t,x)\in[0,T]\times\mathbb{D}} \in C^{1,2}([0,T] \times \mathbb{D}, \mathbb{R})$ have at most polynomially growing partial derivatives, assume[1] for every $t \in (0,T]$, $x \in \partial_{\mathbb{D}}$ that $\langle \mathbf{n}(x), (\nabla_x u)(t,x) \rangle = 0$, and assume for every $t \in [0,T]$, $x \in \mathbb{D}$ that $u(0,x) = g(x)$, $\int_{\mathbb{D}} |f(u(t,x), u(t,\mathbf{x}))| \, \nu_x(\mathrm{d}\mathbf{x}) < \infty$, and

$$\big(\tfrac{\partial}{\partial t} u\big)(t,x) = (\Delta_x u)(t,x) + \int_{\mathbb{D}} f(u(t,x), u(t,\mathbf{x})) \, \nu_x(\mathrm{d}\mathbf{x}). \tag{3.1}$$

Our goal in this section is to approximately calculate under suitable hypotheses the solution $u \colon [0,T] \times \mathbb{D} \to \mathbb{R}$ of the PDE in (3.1).

3.2.2 Reflection principle for the simulation of time discrete reflected processes

Framework 3.2.1 (Reflection principle for the simulation of time discrete reflected processes)**.** *Let $d \in \mathbb{N}$, let $\mathbb{D} \subseteq \mathbb{R}^d$ be a closed set with sufficiently smooth boundary $\partial_{\mathbb{D}}$, let $\mathbf{n} \colon \partial_{\mathbb{D}} \to \mathbb{R}^d$ be a suitable outer unit normal vector field associated to $\mathbb{D}$, let $\mathfrak{c} \colon (\mathbb{R}^d)^2 \to \mathbb{R}^d$ satisfy for every $a, b \in \mathbb{R}^d$ that*

$$\mathfrak{c}(a,b) = a + \big[\inf(\{r \in [0,1] \colon a + r(b-a) \notin \mathbb{D}\} \cup \{1\})\big](b-a), \tag{3.2}$$

[1] Throughout this article we denote by $\langle \cdot, \cdot \rangle \colon \big(\bigcup_{n\in\mathbb{N}} (\mathbb{R}^n \times \mathbb{R}^n)\big) \to \mathbb{R}$ and $\|\cdot\| \colon \big(\bigcup_{n\in\mathbb{N}} \mathbb{R}^n\big) \to \mathbb{R}$ the functions which satisfy for all $n \in \mathbb{N}$, $v = (v_1, \ldots, v_n)$, $w = (w_1, \ldots, w_n) \in \mathbb{R}^n$ that $\langle v, w \rangle = \sum_{i=1}^n v_i w_i$ and $\|v\| = \sqrt{\langle v, v \rangle} = \big[\sum_{i=1}^n |v_i|^2\big]^{1/2}$.

let $\mathscr{R}\colon (\mathbb{R}^d)^2 \to (\mathbb{R}^d)^2$ *satisfy for every* $a, b \in \mathbb{R}^d$ *that*

$$\mathscr{R}(a,b) = \begin{cases} (a,b) & : \mathfrak{c}(a,b) = a \\ \big(\mathfrak{c}(a,b), b - 2\mathbf{n}(\mathfrak{c}(a,b))\langle b - \mathfrak{c}(a,b), \mathbf{n}(\mathfrak{c}(a,b))\rangle\big) & : \mathfrak{c}(a,b) \notin \{a,b\} \\ (b,b) & : \mathfrak{c}(a,b) = b, \end{cases} \tag{3.3}$$

let $P\colon (\mathbb{R}^d)^2 \to \mathbb{R}^d$ *satisfy for every* $a, b \in \mathbb{R}^d$ *that* $P(a,b) = b$, *let* $\mathcal{R}_n\colon (\mathbb{R}^d)^2 \to (\mathbb{R}^d)^2$, $n \in \mathbb{N}_0 = \{0\} \cup \mathbb{N}$, *satisfy for every* $n \in \mathbb{N}_0$, $x, y \in \mathbb{R}^d$ *that* $\mathcal{R}_0(x,y) = (x,y)$ *and* $\mathcal{R}_{n+1}(x,y) = \mathscr{R}(\mathcal{R}_n(x,y))$, *and let* $R\colon (\mathbb{R}^d)^2 \to \mathbb{R}^d$ *satisfy for every* $x, y \in \mathbb{R}^d$ *that*

$$R(x,y) = \lim_{n\to\infty} P(\mathcal{R}_n(x,y)). \tag{3.4}$$

3.2.3 Description of the proposed approximation method in a special case

Framework 3.2.2 (Special case of the machine learning-based approximation method)**.** *Assume Lemma 3.2.1, let* $T, \gamma \in (0,\infty)$, $N, M, K \in \mathbb{N}$, $g \in C^2(\mathbb{R}^d, \mathbb{R})$, $\mathfrak{d}, \mathfrak{h} \in \mathbb{N}\backslash\{1\}$, $t_0, t_1, \ldots, t_N \in [0,T]$ *satisfy* $\mathfrak{d} = \mathfrak{h}(N+1)d(d+1)$ *and*

$$0 = t_0 < t_1 < \ldots < t_N = T, \tag{3.5}$$

let $\tau_0, \tau_1, \ldots, \tau_N \in [0,T]$ *satisfy for every* $n \in \{0,1,\ldots,N\}$ *that* $\tau_n = T - t_{N-n}$, *let* $f\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ *be measurable, let* $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t\in[0,T]})$ *be a filtered probability space, let* $\xi^m\colon \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}$, *be i.i.d.* $\mathcal{F}_0/\mathcal{B}(\mathbb{R}^d)$*-measurable random variables, let* $W^m\colon [0,T]\times\Omega \to \mathbb{R}^d$, $m \in \mathbb{N}$, *be i.i.d. standard* $(\mathcal{F}_t)_{t\in[0,T]}$*-Brownian motions, for every* $m \in \mathbb{N}$ *let* $\mathcal{Y}^m\colon \{0,1,\ldots,N\}\times\Omega \to \mathbb{R}^d$ *be the stochastic process which satisfies for every* $n \in \{0,1,\ldots,N-1\}$ *that* $\mathcal{Y}^m_0 = \xi^m$ *and*

$$\mathcal{Y}^m_{n+1} = R\big(\mathcal{Y}^m_n, \mathcal{Y}^m_n + \sqrt{2}(W^m_{\tau_{n+1}} - W^m_{\tau_n})\big), \tag{3.6}$$

let $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}^d$ *satisfy for every* $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ *that*

$$\mathcal{L}(x) = \left(\frac{\exp(x_1)}{\exp(x_1)+1}, \ldots, \frac{\exp(x_d)}{\exp(x_d)+1}\right), \tag{3.7}$$

for every $\theta = (\theta_1, \ldots, \theta_{\mathfrak{d}}) \in \mathbb{R}^{\mathfrak{d}}$, $k, l, v \in \mathbb{N}$ *with* $v + l(k+1) \leq \mathfrak{d}$ *let* $A^{\theta,v}_{k,l}\colon \mathbb{R}^k \to \mathbb{R}^l$ *satisfy for every* $x = (x_1, \ldots, x_k) \in \mathbb{R}^k$ *that*

$$A^{\theta,v}_{k,l}(x) = \left(\theta_{v+kl+1} + \left[\sum_{i=1}^{k} x_i\,\theta_{v+i}\right], \ldots, \theta_{v+kl+l} + \left[\sum_{i=1}^{k} x_i\,\theta_{v+(l-1)k+i}\right]\right), \tag{3.8}$$

let $\mathbb{V}_n\colon \mathbb{R}^{\mathfrak{d}} \times \mathbb{R}^d \to \mathbb{R}$, $n \in \{0, 1, \ldots, N\}$, *satisfy for every* $n \in \{1, 2, \ldots, N\}$, $\theta \in \mathbb{R}^{\mathfrak{d}}$, $x \in \mathbb{R}^d$ *that* $\mathbb{V}_0(\theta, x) = g(x)$ *and*

$$\begin{aligned} &\mathbb{V}_n(\theta, x) = \\ &\big(A^{\theta,(\mathfrak{h}n+\mathfrak{h}-1)d(d+1)}_{d,1} \circ \mathcal{L} \circ A^{\theta,(\mathfrak{h}n+\mathfrak{h}-2)d(d+1)}_{d,d} \circ \ldots \circ \mathcal{L} \circ A^{\theta,(\mathfrak{h}n+1)d(d+1)}_{d,d} \circ \mathcal{L} \circ A^{\theta,\mathfrak{h}nd(d+1)}_{d,d}\big)(x), \end{aligned} \tag{3.9}$$

let $\nu_x\colon \mathcal{B}(\mathbb{D}) \to [0,1]$, $x \in \mathbb{D}$, *be probability measures, for every* $x \in \mathbb{D}$ *let* $Z^{n,m}_{x,k}\colon \Omega \to \mathbb{D}$, $k, n, m \in \mathbb{N}$, *be i.i.d. random variables which satisfy for every* $A \in \mathcal{B}(\mathbb{D})$ *that* $\mathbb{P}(Z^{1,1}_{x,1} \in A) = \nu_x(A)$, *let* $\Theta^n\colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$, $n \in \{0, 1, \ldots, N\}$, *be stochastic processes, for every* $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ *let* $\phi^{n,m}\colon \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}$ *satisfy for every* $\theta \in \mathbb{R}^{\mathfrak{d}}$, $\omega \in \Omega$ *that*

$$\begin{aligned} \phi^{n,m}(\theta,\omega) = \Bigg[&\mathbb{V}_n\big(\theta, \mathcal{Y}^m_{N-n}(\omega)\big) - \mathbb{V}_{n-1}\big(\Theta^{n-1}_M(\omega), \mathcal{Y}^m_{N-n+1}(\omega)\big) \\ &- \tfrac{(t_n - t_{n-1})}{K}\Bigg[\sum_{k=1}^{K} f\big(\mathbb{V}_{n-1}(\Theta^{n-1}_M(\omega), \mathcal{Y}^m_{N-n+1}(\omega)), \mathbb{V}_{n-1}(\Theta^{n-1}_M(\omega), Z^{n,m}_{\mathcal{Y}^m_{N-n+1}(\omega),k}(\omega))\big)\Bigg]\Bigg]^2, \end{aligned} \tag{3.10}$$

for every $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ *let* $\Phi^{n,m}\colon \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ *satisfy for every* $\theta \in \mathbb{R}^{\mathfrak{d}}$, $\omega \in \Omega$ *that* $\Phi^{n,m}(\theta,\omega) = (\nabla_\theta \phi^{n,m})(\theta,\omega)$, *and assume for every* $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ *that*

$$\Theta^n_m = \Theta^n_{m-1} - \gamma\, \Phi^{n,m}(\Theta^n_{m-1}). \tag{3.11}$$

As indicated in Section 3.2.1 above, the algorithm described in Lemma 3.2.2 computes an approximation for a solution of the PDE in Eq. (3.1), i.e., a function $u \in C^{1,2}([0,T] \times D, \mathbb{R})$ which has at most polynomially growing derivatives, which satisfies for every $t \in (0,T]$, $x \in \partial_{\mathbb{D}}$ that $\langle \mathbf{n}(x), (\nabla_x u)(t,x)\rangle = 0$ and which satisfies for every $t \in [0,T]$, $x \in \mathbb{D}$ that $u(0,x) = g(x)$, $\int_{\mathbb{D}} |f(u(t,x), u(t,\mathbf{x}))|\, \nu_x(\mathrm{d}\mathbf{x}) < \infty$, and

$$\big(\tfrac{\partial}{\partial t} u\big)(t,x) = (\Delta_x u)(t,x) + \int_{\mathbb{D}} f(u(t,x), u(t,\mathbf{x}))\, \nu_x(\mathrm{d}\mathbf{x}). \tag{3.12}$$

Let us now add some explanatory comments on the objects and notations employed in Lemma 3.2.2 above. The algorithm in Lemma 3.2.2 decomposes the time interval $[0,T]$ into N subintervals at the times $t_0, t_1, t_2, \ldots, t_N \in [0,T]$ (cf. Eq. (3.5)). For every $n \in \{1, 2, \ldots, N\}$ we aim to approximate the function $\mathbb{R}^d \ni x \mapsto u(t_n, x) \in \mathbb{R}$ by a suitable (realization function of a) fully-connected feedforward neural network. Each of these neural networks is an alternating composition of $\mathfrak{h} - 1$ affine linear functions from $\mathbb{R}^d$ to $\mathbb{R}^d$ (where we think of $\mathfrak{h} \in \mathbb{N}\backslash\{1\}$ as the *length* or *depth* of the neural network), $\mathfrak{h} - 1$ instances of a d-dimensional version of the standard logistic function and finally an affine linear function from $\mathbb{R}^d$ to $\mathbb{R}$. Every such neural network can be specified by means of $(\mathfrak{h} - 1)(d^2 + d) + d + 1 \leq \mathfrak{h}d(d+1)$ real parameters and so $N + 1$ of these neural networks can be specified by a parameter vector of length $\mathfrak{d} = \mathfrak{h}(N+1)d(d+1) \in \mathbb{N}$. Note that $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}^d$ in Lemma 3.2.2 above denotes the d-dimensional version of the standard logistic function (cf. Eq. (3.7)) and for every $k, l, v \in \mathbb{N}$, $\theta \in \mathbb{R}^{\mathfrak{d}}$ with $v + kl + l \leq \mathfrak{d}$ the function $A^{\theta,v}_{k,l}\colon \mathbb{R}^k \to \mathbb{R}^l$ in Lemma 3.2.2 denotes an affine linear function specified by means

of the parameters $v+1, v+2, \ldots, v+kl+l$ (cf. Eq. (3.8)). Furthermore, observe that for every $n \in \{1, 2, \ldots, N\}$, $\theta \in \mathbb{R}^{\mathfrak{d}}$ the function

$$\mathbb{R}^d \ni x \mapsto \mathbb{V}_n(\theta, x) \in \mathbb{R} \tag{3.13}$$

denotes a neural network specified by means of the parameters $\mathfrak{h}nd(d+1)+1, \mathfrak{h}nd(d+1)+2, \ldots, (\mathfrak{h}n+\mathfrak{h}-1)d(d+1)+d+1$.

The goal of the optimization algorithm in Lemma 3.2.2 above is to find a suitable parameter vector $\theta \in \mathbb{R}^{\mathfrak{d}}$ such that for every $n \in \{1, 2, \ldots, N\}$ the neural network $\mathbb{R}^d \ni x \mapsto \mathbb{V}_n(\theta, x) \in \mathbb{R}$ is a good approximation for the solution $\mathbb{R}^d \ni x \mapsto u(t_n, x) \in \mathbb{R}$ to the PDE in Eq. (3.12) at time t_n. This is done by performing successively for each $n \in \{1, 2, \ldots, N\}$ a plain vanilla stochastic gradient descent (SGD) optimization on a suitable loss function (cf. Eq. (3.11)).

Observe that for every $n \in \{1, 2, \ldots, N\}$ the stochastic process $\Theta^n \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ describes the successive estimates computed by the SGD algorithm for the parameter vector that represents (via $\mathbb{V}_n \colon \mathbb{R}^{\mathfrak{d}} \times \mathbb{R}^d \to \mathbb{R}$) a suitable approximation to the solution $\mathbb{R}^d \ni x \mapsto u(t_n, x) \in \mathbb{R}$ of the PDE in Eq. (3.12) at time t_n. Next note that $M \in \mathbb{N}$ in Lemma 3.2.2 above denotes the number of gradient descent steps taken for each $n \in \{1, 2, \ldots, N\}$ and that $\gamma \in (0, \infty)$ denotes the learning rate employed in the SGD algorithm. Moreover, observe that for every $n \in \{1, 2, \ldots, N\}$, $m \in \{1, 2, \ldots, M\}$ the function $\phi^{n,m} \colon \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}$ denotes the loss function employed in the mth gradient descent step during the approximation of the solution of the PDE in Eq. (3.12) at time t_n (cf. Eq. (3.10)). The loss functions employ a family of i.i.d. time-discrete stochastic processes $\mathcal{Y}^m \colon \{0, 1, \ldots N\} \times \Omega \to \mathbb{R}^d$, $m \in \mathbb{N}$, which we think of as discretizations of suitable reflected Brownian motions (cf. Eq. (3.6)). In addition, for every $n \in \{1, 2, \ldots, N\}$, $m \in \{1, 2, \ldots, M\}$, $x \in D$ the loss function $\phi^{n,m} \colon \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}$ employs a family of i.i.d. random variables $Z^{n,m}_{x,k} \colon \Omega \to D$, $k \in \mathbb{N}$, which are used for the Monte Carlo approximation of the non-local term in the PDE in Eq. (3.12) whose solution we are trying to approximate. The number of samples used in these Monte Carlo approximations is denoted by $K \in \mathbb{N}$ in Lemma 3.2.2 above.

Finally, for sufficiently large $N, M, K \in \mathbb{N}$ and sufficiently small $\gamma \in (0, \infty)$ the algorithm in Lemma 3.2.2 above yields for every $n \in \{1, 2, \ldots, N\}$ a (random) parameter vector $\Theta^n_M \colon \Omega \to \mathbb{R}^{\mathfrak{d}}$ which represents a function $\mathbb{R}^d \times \Omega \ni (x, \omega) \mapsto \mathbb{V}_n(\Theta^n_M(\omega), x) \in \mathbb{R}$ that we think of as providing for every $x \in \mathbb{D}$ a suitable approximation

$$\mathbb{V}_n(\Theta^n_M, x) \approx u(t_n, x). \tag{3.14}$$

3.3 Machine learning-based approximation method in the general case

In this section we describe in Lemma 3.3.1 in Section 3.3.2 below the full version of our deep learning-based method for approximating solutions of non-local nonlinear PDEs with

Neumann boundary conditions (see Section 3.3.1 for a description of the class of PDEs our approximation method applies to), which generalizes the algorithm introduced in Lemma 3.2.2 in Section 3.2.3 above and which we apply in Section 3.5 below to several examples of non-local nonlinear PDEs.

3.3.1 PDEs under consideration

Let $T \in (0, \infty)$, $d \in \mathbb{N}$, let $\mathbb{D} \subseteq \mathbb{R}^d$ be a closed set with sufficiently smooth boundary $\partial_{\mathbb{D}}$, let $\mathbf{n} \colon \partial_{\mathbb{D}} \to \mathbb{R}^d$ be an outer unit normal vector field associated to $\mathbb{D}$, let $g \colon \mathbb{D} \to \mathbb{R}$, $\mu \colon \mathbb{D} \to \mathbb{R}^d$, and $\sigma \colon \mathbb{D} \to \mathbb{R}^{d \times d}$ be continuous, let $\nu_x \colon \mathcal{B}(\mathbb{D}) \to [0,1]$, $x \in \mathbb{D}$, be probability measures, let $f \colon [0,T] \times \mathbb{D} \times \mathbb{D} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be measurable, let $u = (u(t,x))_{(t,x) \in [0,T] \times \mathbb{D}} \in C^{1,2}([0,T] \times \mathbb{D}, \mathbb{R})$ have at most polynomially growing partial derivatives, assume for every $t \in [0,T]$, $x \in \partial_{\mathbb{D}}$ that $\langle \mathbf{n}(x), (\nabla_x u)(t,x) \rangle = 0$, and assume for every $t \in [0,T]$, $x \in \mathbb{D}$ that $u(0,x) = g(x)$, $\int_{\mathbb{D}} \big| f\big(t, x, \mathbf{x}, u(t,x), u(t,\mathbf{x})\big) \big| \, \nu_x(\mathrm{d}\mathbf{x}) < \infty$, and

$$\begin{aligned} \big(\tfrac{\partial}{\partial t} u\big)(t,x) &= \int_{\mathbb{D}} f\big(t, x, \mathbf{x}, u(t,x), u(t,\mathbf{x})\big) \, \nu_x(\mathrm{d}\mathbf{x}) \\ &\quad + \big\langle \mu(x), (\nabla_x u)(t,x) \big\rangle + \tfrac{1}{2} \operatorname{Trace}\big(\sigma(x)[\sigma(x)]^* (\operatorname{Hess}_x u)(t,x)\big). \end{aligned} \tag{3.15}$$

Our goal is to approximately calculate under suitable hypotheses the solution $u \colon [0,T] \times \mathbb{D} \to \mathbb{R}$ of the PDE in (3.15).

3.3.2 Description of the proposed approximation method in the general case

Framework 3.3.1 (General case of the machine learning-based approximation method)**.** *Assume Lemma 3.2.1, let $T \in (0,\infty)$, $N, \varrho, \mathfrak{d}, \varsigma \in \mathbb{N}$, $(M_n)_{n \in \mathbb{N}_0} \subseteq \mathbb{N}$, $(K_n)_{n \in \mathbb{N}} \subseteq \mathbb{N}$, $(J_m)_{m \in \mathbb{N}} \subseteq \mathbb{N}$, $t_0, t_1, \ldots, t_N \in [0,T]$ satisfy*

$$0 = t_0 < t_1 < \ldots < t_N = T, \tag{3.16}$$

let $\tau_0, \tau_1, \ldots, \tau_N \in [0,T]$ satisfy for every $n \in \{0,1,\ldots,N\}$ that $\tau_n = T - t_{N-n}$, let $\nu_x \colon \mathcal{B}(\mathbb{D}) \to [0,1]$, $x \in \mathbb{D}$, be probability measures, for every $x \in \mathbb{D}$ let $Z^{n,m,j}_{x,k} \colon \Omega \to \mathbb{D}$, $k, n, m, j \in \mathbb{N}$, be i.i.d. random variables which satisfy for every $A \in \mathcal{B}(\mathbb{D})$ that $\mathbb{P}(Z^{1,1,1}_{x,1} \in A) = \nu_x(A)$, let $f \colon [0,T] \times \mathbb{D} \times \mathbb{D} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be measurable, let $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_t)_{t \in [0,T]})$ be a filtered probability space, for every $n \in \{1,2,\ldots,N\}$ let $W^{n,m,j} \colon [0,T] \times \Omega \to \mathbb{R}^d$, $m, j \in \mathbb{N}$, be i.i.d. standard $(\mathcal{F}_t)_{t \in [0,T]}$-Brownian motions, for every $n \in \{1,2,\ldots,N\}$ let $\xi^{n,m,j} \colon \Omega \to \mathbb{R}^d$, $m, j \in \mathbb{N}$, be i.i.d. $\mathcal{F}_0/\mathcal{B}(\mathbb{R}^d)$-measurable random variables, let $H \colon [0,T]^2 \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ be a function, for every $j \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^\varsigma$, $n \in \{0,1,\ldots,N\}$ let $\mathbb{V}^{j,\mathbf{s}}_n \colon \mathbb{R}^{\mathfrak{d}} \times \mathbb{R}^d \to \mathbb{R}$ be a function, for

every $n \in \{1, 2, \ldots, N\}$, $m, j \in \mathbb{N}$ *let* $\mathcal{Y}^{n,m,j} \colon \{0, 1, \ldots, N\} \times \Omega \to \mathbb{R}^d$ *be a stochastic process which satisfies for every* $k \in \{0, 1, \ldots, N-1\}$ *that* $\mathcal{Y}^{n,m,j}_0 = \xi^{n,m,j}$ *and*

$$\mathcal{Y}^{n,m,j}_{k+1} = H(\tau_{k+1}, \tau_k, \mathcal{Y}^{n,m,j}_k, W^{n,m,j}_{\tau_{k+1}} - W^{n,m,j}_{\tau_k}), \tag{3.17}$$

let $\Theta^n \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{d}}$, $n \in \{0, 1, \ldots, N\}$, *be stochastic processes, for every* $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^{\mathfrak{s}}$ *let* $\phi^{n,m,\mathbf{s}} \colon \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}$ *satisfy for every* $\theta \in \mathbb{R}^{\mathfrak{d}}$, $\omega \in \Omega$ *that*

$$\begin{aligned}
\phi^{n,m,\mathbf{s}}(\theta, \omega) &= \frac{1}{J_m} \sum_{j=1}^{J_m} \Bigg[\mathbb{V}^{j,\mathbf{s}}_n\big(\theta, \mathcal{Y}^{n,m,j}_{N-n}(\omega)\big) - \mathbb{V}^{j,\mathbf{s}}_{n-1}\big(\Theta^{n-1}_{M_{n-1}}(\omega), \mathcal{Y}^{n,m,j}_{N-n+1}(\omega)\big) \\
&\quad - \tfrac{(t_n - t_{n-1})}{K_n} \Bigg[\sum_{k=1}^{K_n} f\Big(t_{n-1}, \mathcal{Y}^{n,m,j}_{N-n+1}(\omega), Z^{n,m,j}_{\mathcal{Y}^{n,m,j}_{N-n+1}(\omega),k}(\omega), \\
&\quad \mathbb{V}^{j,\mathbf{s}}_{n-1}\big(\Theta^{n-1}_{M_{n-1}}(\omega), \mathcal{Y}^{n,m,j}_{N-n+1}(\omega)\big), \mathbb{V}^{j,\mathbf{s}}_{n-1}\big(\Theta^{n-1}_{M_{n-1}}(\omega), Z^{n,m,j}_{\mathcal{Y}^{n,m,j}_{N-n+1}(\omega),k}(\omega)\big)\Big) \Bigg] \Bigg]^2,
\end{aligned} \tag{3.18}$$

for every $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^{\mathfrak{s}}$ *let* $\Phi^{n,m,\mathbf{s}} \colon \mathbb{R}^{\mathfrak{d}} \times \Omega \to \mathbb{R}^{\mathfrak{d}}$ *satisfy for every* $\omega \in \Omega$, $\theta \in \{\vartheta \in \mathbb{R}^{\mathfrak{d}} \colon (\mathbb{R}^{\mathfrak{d}} \ni \eta \mapsto \phi^{n,m,\mathbf{s}}(\eta, \omega) \in \mathbb{R}) \text{ is differentiable at } \vartheta\}$ *that*

$$\Phi^{n,m,\mathbf{s}}(\theta, \omega) = (\nabla_\theta \phi^{n,m,\mathbf{s}})(\theta, \omega), \tag{3.19}$$

let $\mathcal{S}^n \colon \mathbb{R}^{\mathfrak{s}} \times \mathbb{R}^{\mathfrak{d}} \times (\mathbb{R}^d)^{\{0,1,\ldots,N\} \times \mathbb{N}} \to \mathbb{R}^{\mathfrak{s}}$, $n \in \{1, 2, \ldots, N\}$, *be functions, for every* $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ *let* $\psi^n_m \colon \mathbb{R}^{\varrho} \to \mathbb{R}^{\mathfrak{d}}$ *and* $\Psi^n_m \colon \mathbb{R}^{\varrho} \times \mathbb{R}^{\mathfrak{d}} \to \mathbb{R}^{\varrho}$ *be functions, and for every* $n \in \{1, 2, \ldots, N\}$ *let* $\mathbb{S}^n \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\mathfrak{s}}$ *and* $\Xi^n \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\varrho}$ *be stochastic processes which satisfy for every* $m \in \mathbb{N}$ *that*

$$\mathbb{S}^n_m = \mathcal{S}^n\big(\mathbb{S}^n_{m-1}, \Theta^n_{m-1}, (\mathcal{Y}^{n,m,i}_k)_{(k,i) \in \{0,1,\ldots,N\} \times \mathbb{N}}\big), \tag{3.20}$$

$$\Xi^n_m = \Psi^n_m(\Xi^n_{m-1}, \Phi^{n,m,\mathbb{S}^n_m}(\Theta^n_{m-1})), \qquad \text{and} \qquad \Theta^n_m = \Theta^n_{m-1} - \psi^n_m(\Xi^n_m). \tag{3.21}$$

In the setting of Lemma 3.3.1 above we think under suitable hypotheses for sufficiently large $N \in \mathbb{N}$, sufficiently large $(M_n)_{n \in \mathbb{N}_0} \subseteq \mathbb{N}$, sufficiently large $(K_n)_{n \in \mathbb{N}} \subseteq \mathbb{N}$, every $n \in \{0, 1, \ldots, N\}$, and every $x \in \mathbb{D}$ of $\mathbb{V}^{1,\mathbb{S}^n_{M_n}}_n(\Theta^n_{M_n}, x) \colon \Omega \to \mathbb{R}$ as a suitable approximation

$$\mathbb{V}^{1,\mathbb{S}^n_{M_n}}_n(\Theta^n_{M_n}, x) \approx u(t_n, x) \tag{3.22}$$

of $u(t_n, x)$ where $u = (u(t,x))_{(t,x) \in [0,T] \times \mathbb{R}^d} \in C^{1,2}([0,T] \times \mathbb{R}^d, \mathbb{R})$ is a function with at most polynomially growing derivatives which satisfies for every $t \in (0, T]$, $x \in \partial_{\mathbb{D}}$ that $\langle \mathbf{n}(x), (\nabla_x u)(t,x) \rangle = 0$ and which satisfies for every $t \in [0, T]$, $x \in \mathbb{D}$ that $u(0, x) = g(x)$, $\int_{\mathbb{D}} \big| f\big(t, x, \mathbf{x}, u(t,x), u(t, \mathbf{x})\big) \big| \, \nu_x(\mathrm{d}\mathbf{x}) < \infty$, and

$$\begin{aligned}
\big(\tfrac{\partial}{\partial t} u\big)(t,x) &= \int_{\mathbb{D}} f\big(t, x, \mathbf{x}, u(t,x), u(t, \mathbf{x})\big) \, \nu_x(\mathrm{d}\mathbf{x}) \\
&\quad + \big\langle \mu(x), (\nabla_x u)(t,x) \big\rangle + \tfrac{1}{2} \operatorname{Trace}\big(\sigma(x)[\sigma(x)]^* (\operatorname{Hess}_x u)(t,x)\big)
\end{aligned} \tag{3.23}$$

(cf. (3.15)). Compared to the simplified algorithm in Lemma 3.2.2 above, the major new elements introduced in Lemma 3.3.1 are the following:

(a) The numbers of gradient descent steps taken to compute approximations for the solution of the PDE at the times t_n, $n \in \{1, 2, \ldots, N\}$, are allowed to vary with n, and so are specified by a sequence $(M_n)_{n\in\mathbb{N}_0} \subseteq \mathbb{N}$ in Lemma 3.3.1 above.

(b) The numbers of samples used for the Monte Carlo approximation of the non-local term in the approximation for the solution of the PDE at the times t_n, $n \in \{1, 2, \ldots, N\}$, are allowed to vary with n, and so are specified by a sequence $(K_n)_{n\in\mathbb{N}_0} \subseteq \mathbb{N}$ in Lemma 3.3.1 above.

(c) The approximating functions $\mathbb{V}_n^{j,\mathbf{s}}$, $(j, \mathbf{s}, n) \in \mathbb{N} \times \mathbb{R}^{\varsigma} \times \{0, 1, \ldots, N\}$, in Lemma 3.3.1 above are not specified concretely in order to allow for a variety of neural network architectures. For the concrete choice of these functions employed in our numerical simulations, we refer the reader to Section 3.5.

(d) For every $m \in \{1, 2, \ldots, M\}$ the loss function used in the mth gradient descent step may be computed using a minibatch of samples instead of just one sample (cf. Eq. (3.18)). The sizes of these minibatches are specified by a sequence $(J_m)_{m\in\mathbb{N}} \subseteq \mathbb{N}$.

(e) Compared to Lemma 3.2.2 above, the more general form of the PDEs considered in this section (cf. Eq. (3.23)) requires more flexibility in the definition of the time-discrete stochastic processes $\mathcal{Y}^{n,m,j} \colon \{0, 1, \ldots, N\} \times \Omega \to \mathbb{R}^d$, $(n, m, j) \in \{1, 2, \ldots, N\} \times \mathbb{N} \times \mathbb{N}$, which are specified in Lemma 3.3.1 above in terms of the Brownian motions $W^{n,m,j} \colon [0, T] \times \Omega \to \mathbb{R}^d$, $(n, m, j) \in \{1, 2, \ldots, N\} \times \mathbb{N} \times \mathbb{N}$, via a function $H \colon [0, T]^2 \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ (cf. Eq. (3.17)). We refer the reader to Eq. (3.44) in Section 3.5.1 below, Eq. (3.46) in Section 3.5.2 below, Eq. (3.48) in Section 3.5.3 below, Eq. (3.50) in Section 3.5.4 below, and Eq. (3.72) in Section 3.5.5 below for concrete choices of H in the approximation of various example PDEs.

(f) For every $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ the optimization step in Eq. (3.21) in Lemma 3.3.1 above is specified generically in terms of the functions $\psi_m^n \colon \mathbb{R}^{\varrho} \to \mathbb{R}^{\mathfrak{d}}$ and $\Psi_m^n \colon \mathbb{R}^{\varrho} \times \mathbb{R}^{\mathfrak{d}} \to \mathbb{R}^{\varrho}$ and the random variable $\Xi_m^n \colon \Omega \to \mathbb{R}^{\varrho}$. This generic formulation covers a variety of SGD based optimization algorithms such as Adagrad [31], RMSprop, or Adam [64]. For example, in order to implement the Adam optimization algorithm, for every $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ the random variable Ξ_m^n can be used to hold suitable first and second moment estimates (see Eq. (3.42) and Eq. (3.43) in Section 3.5 below for the concrete specification of these functions implementing the Adam optimization algorithm).

(g) The processes $\mathbb{S}^n \colon \mathbb{N}_0 \times \Omega \to \mathbb{R}^{\varsigma}$, $n \in \{1, 2, \ldots, N\}$, and functions $\mathcal{S}^n \colon \mathbb{R}^{\varsigma} \times \mathbb{R}^{\mathfrak{d}} \times (\mathbb{R}^d)^{\{0,1,\ldots,N\}\times\mathbb{N}} \to \mathbb{R}^{\varsigma}$, $n \in \{1, 2, \ldots, N\}$, in Lemma 3.3.1 above can be used to implement batch normalization; see [61] for details. Loosely speaking, for every $n \in \{1, 2, \ldots, N\}$, $m \in \mathbb{N}$ the random variable $\mathbb{S}_m^n \colon \Omega \to \mathbb{R}^{\varsigma}$ then holds mean and variance estimates of the outputs of each layer of the approximating neural networks related to the minibatches that are used as inputs to the neural networks in computing the loss function at the corresponding gradient descent step.

3.4 Multilevel Picard approximation method for non-local PDEs

In this section we introduce in Lemma 3.4.1 in Section 3.4.1 below our extension of the full history recursive multilevel Picard approximation method for approximating solutions of non-local nonlinear PDEs with Neumann boundary conditions. The MLP method was first introduced in E et al. [35] and Hutzenthaler et al. [60] and later extended in a number of directions; see E et al. [32] and Beck et al. [10] for recent surveys. We also refer the reader to Becker et al. [13] and E et al. [37] for numerical simulations illustrating the performance of MLP methods across a range of example PDE problems.

In Section 3.4.2 below, we will specify five concrete examples of (non-local) nonlinear PDEs and describe how Lemma 3.4.1 can be specialized to compute approximate solutions to these example PDEs. These computations will be used in Section 3.5 to obtain reference values to compare the deep learning-based approximation method proposed in Section 3.3 above against.

3.4.1 Description of the proposed approximation method

Framework 3.4.1 (Multilevel Picard approximation method)**.** *Assume Lemma 3.2.1, let* $c, T \in (0,\infty)$*,* $\mathfrak{I} = \bigcup_{n\in\mathbb{N}} \mathbb{Z}^n$*,* $f \in C([0,T]\times D\times D\times\mathbb{R}\times\mathbb{R},\mathbb{R})$*,* $g \in C(D,\mathbb{R})$*,* $u \in C([0,T]\times\mathbb{D},\mathbb{R})$*, assume* $u|_{[0,T)\times\mathbb{D}} \in C^{1,2}([0,T)\times\mathbb{D},\mathbb{R})$*, let* $\nu_x\colon \mathcal{B}(\mathbb{D}) \to [0,1]$*,* $x \in \mathbb{D}$*, be probability measures, for every* $x \in \mathbb{D}$ *let* $Z_x^{\mathfrak{i}}\colon \Omega \to \mathbb{D}$*,* $\mathfrak{i} \in \mathfrak{I}$*, be i.i.d. random variables, assume for every* $A \in \mathcal{B}(\mathbb{D})$*,* $\mathfrak{i} \in \mathfrak{I}$ *that* $\mathbb{P}(Z_x^{\mathfrak{i}} \in A) = \nu_x(A)$*, let* $\phi_r\colon \mathbb{R}\to\mathbb{R}$*,* $r \in [0,\infty]$*, satisfy for every* $r \in [0,\infty]$*,* $y \in \mathbb{R}$ *that*

$$\phi_r(y) = \min\{r, \max\{-r, y\}\}, \tag{3.24}$$

let $(\Omega,\mathcal{F},\mathbb{P})$ *be a probability space, let* $\mathcal{V}^{\mathfrak{i}}\colon \Omega\to(0,1)$*,* $\mathfrak{i}\in\mathfrak{I}$*, be independent* $\mathcal{U}_{(0,1)}$*-distributed random variables, let* $V^{\mathfrak{i}}\colon [0,T]\times\Omega\to[0,T]$*,* $\mathfrak{i}\in\mathfrak{I}$*, satisfy for every* $t\in[0,T]$*,* $\mathfrak{i}\in\mathfrak{I}$ *that*

$$V_t^{\mathfrak{i}} = t + (T-t)\mathcal{V}^{\mathfrak{i}}, \tag{3.25}$$

let $W^{\mathfrak{i}}\colon [0,T]\times\Omega\to\mathbb{R}^d$*,* $\mathfrak{i}\in\mathfrak{I}$*, be independent standard Brownian motions, assume that* $(\mathcal{V}^{\mathfrak{i}})_{\mathfrak{i}\in\mathfrak{I}}$ *and* $(W^{\mathfrak{i}})_{\mathfrak{i}\in\mathfrak{I}}$ *are independent, let* $\mu\colon\mathbb{R}^d\to\mathbb{R}^d$ *and* $\sigma\colon\mathbb{R}^d\to\mathbb{R}^{d\times d}$ *be globally Lipschitz continuous, for every* $x\in\mathbb{R}^d$*,* $\mathfrak{i}\in\mathfrak{I}$*,* $t\in[0,T]$ *let* $X_t^{x,\mathfrak{i}} = (X_{t,s}^{x,\mathfrak{i}})_{s\in[t,T]}\colon [t,T]\times\Omega\to\mathbb{R}^d$ *be a stochastic process with continuous sample paths, let* $(K_{n,l,m})_{n,l,m\in\mathbb{N}_0}\subseteq\mathbb{N}$*, for every* $\mathfrak{i}\in\mathfrak{I}$*,*

$n, M \in \mathbb{N}_0$, $r \in [0,\infty]$ *let* $U^{\mathfrak{i}}_{n,M,r} \colon [0,T] \times \mathbb{R}^d \times \Omega \to \mathbb{R}^k$ *satisfy for every* $t \in [0,T]$, $x \in \mathbb{R}^d$ *that*

$$
\begin{aligned}
U^{\mathfrak{i}}_{n,M,r}(t,x) = \Bigg[\sum_{l=0}^{n-1} \frac{(T-t)}{M^{n-l}} \sum_{m=1}^{M^{n-l}} \frac{1}{K_{n,l,m}} \sum_{k=1}^{K_{n,l,m}} \Bigg[f\Big(V_t^{(\mathfrak{i},l,m)}, X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}}, Z^{(\mathfrak{i},l,m,k)}_{X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}}}, \\
\phi_r\Big(U^{(\mathfrak{i},l,m)}_{l,M,r}\big(V_t^{(\mathfrak{i},l,m)}, X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}} \big) \Big), \phi_r\Big(U^{(\mathfrak{i},l,m)}_{l,M,r}\big(V_t^{(\mathfrak{i},l,m)}, Z^{(\mathfrak{i},l,m,k)}_{X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}}} \big) \Big) \Big) \\
- \mathbb{1}_{\mathbb{N}}(l)\, f\Big(V_t^{(\mathfrak{i},l,m)}, X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}}, Z^{(\mathfrak{i},l,m,k)}_{X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}}}, \phi_r\Big(U^{(\mathfrak{i},l,-m)}_{\max\{l-1,0\},M,r}\big(V_t^{(\mathfrak{i},l,m)}, X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}} \big) \Big), \\
\phi_r\Big(U^{(\mathfrak{i},l,-m)}_{\max\{l-1,0\},M,r}\big(V_t^{(\mathfrak{i},l,m)}, Z^{(\mathfrak{i},l,m,k)}_{X^{x,(\mathfrak{i},l,m)}_{t,V_t^{(\mathfrak{i},l,m)}}} \big) \Big) \Big) \Bigg] \Bigg] + \frac{\mathbb{1}_{\mathbb{N}}(n)}{M^n} \Bigg[\sum_{m=1}^{M^n} g\big(X^{x,(\mathfrak{i},0,-m)}_{t,T} \big) \Bigg],
\end{aligned}
\tag{3.26}
$$

assume for every $t \in [0,T)$, $x \in \partial_D$ *that* $\langle \mathbf{n}(x), (\nabla_x u)(t,x) \rangle = 0$, *and assume for every* $t \in [0,T)$, $x \in \mathbb{D}$ *that* $\|u(t,x)\| \leq c(1 + \|x\|^c)$, $u(T,x) = g(x)$, *and*

$$
\begin{aligned}
\big(\tfrac{\partial}{\partial t} u\big)(t,x) + \tfrac{1}{2} \operatorname{Trace}\big(\sigma(x)[\sigma(x)]^* (\operatorname{Hess}_x u)(t,x) \big) + \langle \mu(x), (\nabla_x u)(t,x) \rangle \\
+ \int_D f(t,x,\mathbf{x},u(t,x),u(t,\mathbf{x}))\, \nu_x(\mathrm{d}\mathbf{x}) = 0.
\end{aligned}
\tag{3.27}
$$

3.4.2 Examples for the approximation method

Example 3.4.2 (Fisher–KPP PDEs with Neumann boundary conditions)**.** *In this example we specialize Lemma 3.4.1 to the case of certain Fisher–KPP PDEs with Neumann boundary conditions (cf., e.g., Bian et al. [18] and Wang et al. [92]).*

Assume Lemma 3.4.1, let $\epsilon \in (0,\infty)$ *satisfy* $\epsilon = \frac{1}{10}$, *assume that* $d \in \{1,2,5,10\}$, $D = [-1/2, 1/2]^d$, *and* $T \in \{1/5, 1/2, 1\}$, *assume for every* $n,l,m \in \mathbb{N}$ *that* $K_{n,l,m} = 1$, *assume for every* $t \in [0,T]$, $x, \mathbf{x} \in \mathbb{D}$, $y, \mathbf{y} \in \mathbb{R}$, $v \in \mathbb{R}^d$ *that* $g(x) = \exp(-\frac{1}{4}\|x\|^2)$, $\mu(x) = (0,\ldots,0)$, $\sigma(x)v = \epsilon v$, *and* $f(t,x,\mathbf{x},y,\mathbf{y}) = y(1-y)$, *and assume that for every* $x \in \mathbb{R}^d$, $\mathfrak{i} \in \mathfrak{I}$, $t \in [0,T]$, $s \in [t,T]$ *it holds* $\mathbb{P}$*-a.s. that*

$$
X^{x,\mathfrak{i}}_{t,s} = R\Big(x, x + \int_t^s \mu\big(X^{x,\mathfrak{i}}_{t,r}\big)\, \mathrm{d}r + \int_t^s \sigma\big(X^{x,\mathfrak{i}}_{t,r}\big)\, \mathrm{d}W^{\mathfrak{i}}_r \Big) = R(x, x + \epsilon(W^{\mathfrak{i}}_s - W^{\mathfrak{i}}_t)).
\tag{3.28}
$$

The solution $u \colon [0,T] \times \mathbb{D} \to \mathbb{R}$ *of the PDE in* (3.27) *then satisfies that for every* $t \in [0,T)$, $x \in \partial_{\mathbb{D}}$ *it holds that* $\langle \mathbf{n}(x), (\nabla_x u)(t,x) \rangle = 0$ *and that for every* $t \in [0,T)$, $x \in \mathbb{D}$ *it holds that* $u(T,x) = \exp(-\frac{1}{4}\|x\|^2)$ *and*

$$
\big(\tfrac{\partial}{\partial t} u\big)(t,x) + \tfrac{\epsilon^2}{2} (\Delta_x u)(t,x) + u(t,x)\big(1 - u(t,x)\big) = 0.
\tag{3.29}
$$

Example 3.4.3 (Non-local competition PDEs)**.** *In this example we specialize Lemma 3.4.1 to the case of certain non-local competition PDEs (cf., e.g., Doebeli & Ispolatov [30], Berestycki et al. [17], Perthame & Génieys [82], and Génieys et al. [43]).*

Assume Lemma 3.4.1, let $\mathfrak{s}, \epsilon \in (0,\infty)$ *satisfy* $\mathfrak{s} = \epsilon = \frac{1}{10}$*, assume that* $d \in \{1,2,5,10\}$*,* $D = \mathbb{R}^d$*, and* $T \in \{1/5, 1/2, 1\}$*, assume for every* $n,l,m \in \mathbb{N}$ *that* $K_{n,l,m} = 10$*, assume for every* $x \in \mathbb{R}^d$*,* $A \in \mathcal{B}(\mathbb{R}^d)$ *that* $\nu_x(A) = \pi^{-d/2}\mathfrak{s}^{-d} \int_A \exp\bigl(-\mathfrak{s}^{-2}\|x-\mathbf{x}\|^2\bigr)\,\mathrm{d}\mathbf{x}$*, assume for every* $t \in [0,T]$*,* $v,x,\mathbf{x} \in \mathbb{R}^d$*,* $y,\mathbf{y} \in \mathbb{R}$ *that* $g(x) = \exp(-\frac{1}{4}\|x\|^2)$*,* $\mu(x) = (0,\dots,0)$*,* $\sigma(x)v = \epsilon v$*, and* $f(t,x,\mathbf{x},y,\mathbf{y}) = y(1-\mathbf{y}\pi^{d/2}\mathfrak{s}^d)$*, and assume that for every* $x \in \mathbb{R}^d$*,* $\mathfrak{i} \in \mathfrak{I}$*,* $t \in [0,T]$*,* $s \in [t,T]$ *it holds* $\mathbb{P}$*-a.s. that*

$$X^{x,\mathfrak{i}}_{t,s} = x + \int_t^s \mu\bigl(X^{x,\mathfrak{i}}_{t,r}\bigr)\,\mathrm{d}r + \int_t^s \sigma\bigl(X^{x,\mathfrak{i}}_{t,r}\bigr)\,\mathrm{d}W^{\mathfrak{i}}_r = x + \epsilon(W^{\mathfrak{i}}_s - W^{\mathfrak{i}}_t). \tag{3.30}$$

The solution $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ *of the PDE in* (3.27) *then satisfies that for every* $t \in [0,T)$*,* $x \in \mathbb{R}^d$ *it holds that* $u(T,x) = \exp(-\frac{1}{4}\|x\|^2)$ *and*

$$\bigl(\tfrac{\partial}{\partial t}u\bigr)(t,x) + \tfrac{\epsilon^2}{2}(\Delta_x u)(t,x) + u(t,x)\biggl(1 - \int_{\mathbb{R}^d} u(t,\mathbf{x})\,\exp\bigl(-\tfrac{\|x-\mathbf{x}\|^2}{\mathfrak{s}^2}\bigr)\,\mathrm{d}\mathbf{x}\biggr) = 0. \tag{3.31}$$

Example 3.4.4 (Non-local sine-Gordon PDEs)**.** *In this example we specialize Lemma 3.4.1 to the case of certain non-local sine-Gordon type PDEs (cf., e.g., Hairer & Shen [49], Barone et al. [7], and Coleman [25]).*

Assume Lemma 3.4.1, let $\mathfrak{s}, \epsilon \in (0,\infty)$ *satisfy* $\mathfrak{s} = \epsilon = \frac{1}{10}$*, assume that* $d \in \{1,2,5,10\}$*,* $D = \mathbb{R}^d$*, and* $T \in \{1/5, 1/2, 1\}$*, assume for every* $n,l,m \in \mathbb{N}$ *that* $K_{n,l,m} = 10$*, assume for every* $x \in \mathbb{R}^d$*,* $A \in \mathcal{B}(\mathbb{R}^d)$ *that* $\nu_x(A) = \pi^{-d/2}\mathfrak{s}^{-d} \int_A \exp\bigl(-\mathfrak{s}^{-2}\|x-\mathbf{x}\|^2\bigr)\,\mathrm{d}\mathbf{x}$*, assume for every* $t \in [0,T]$*,* $v,x,\mathbf{x} \in \mathbb{R}^d$*,* $y,\mathbf{y} \in \mathbb{R}$ *that* $g(x) = \exp(-\frac{1}{4}\|x\|^2)$*,* $\mu(x) = (0,\dots,0)$*,* $\sigma(x)v = \epsilon v$*, and* $f(t,x,\mathbf{x},y,\mathbf{y}) = \sin(y) - \mathbf{y}\pi^{d/2}\mathfrak{s}^d$*, and assume that for every* $x \in \mathbb{R}^d$*,* $\mathfrak{i} \in \mathfrak{I}$*,* $t \in [0,T]$*,* $s \in [t,T]$ *it holds* $\mathbb{P}$*-a.s. that*

$$X^{x,\mathfrak{i}}_{t,s} = x + \int_t^s \mu\bigl(X^{x,\mathfrak{i}}_{t,r}\bigr)\,\mathrm{d}r + \int_t^s \sigma\bigl(X^{x,\mathfrak{i}}_{t,r}\bigr)\,\mathrm{d}W^{\mathfrak{i}}_r = x + \epsilon(W^{\mathfrak{i}}_s - W^{\mathfrak{i}}_t). \tag{3.32}$$

The solution $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ *of the PDE in* (3.27) *then satisfies that for every* $t \in [0,T)$*,* $x \in \mathbb{R}^d$ *it holds that* $u(T,x) = \exp(-\frac{1}{4}\|x\|^2)$ *and*

$$\bigl(\tfrac{\partial}{\partial t}u\bigr)(t,x) + \tfrac{\epsilon^2}{2}(\Delta_x u)(t,x) + \sin(u(t,x)) - \int_{\mathbb{R}^d} u(t,\mathbf{x})\,\exp\bigl(-\tfrac{\|x-\mathbf{x}\|^2}{\mathfrak{s}^2}\bigr)\,\mathrm{d}\mathbf{x} = 0. \tag{3.33}$$

Example 3.4.5 (Replicator-mutator PDEs)**.** *In this example we specialize Lemma 3.4.1 to the case of certain* d*-dimensional replicator-mutator PDEs (cf., e.g., Hamel et al. [50]).*

Assume Lemma 3.4.1, let $\mathfrak{m}_1,\mathfrak{m}_2,\dots,\mathfrak{m}_d,\mathfrak{s}_1,\mathfrak{s}_2,\dots,\mathfrak{s}_d,\mathfrak{u}_1,\mathfrak{u}_2,\dots,\mathfrak{u}_d \in \mathbb{R}$ *satisfy for every* $k \in \{1,2,\dots,d\}$ *that* $\mathfrak{m}_k = \frac{1}{10}$*,* $\mathfrak{s}_k = \frac{1}{20}$*, and* $\mathfrak{u}_k = 0$*, assume that* $d \in \{1,2,5,10\}$*,* $\mathbb{D} = \mathbb{R}^d$*, and* $T \in \{1/5, 1/2, 1\}$*, assume for every* $n,l,m \in \mathbb{N}$ *that* $K_{n,l,m} = 10$*, let* $a\colon \mathbb{R}^d \to \mathbb{R}$ *satisfy for every* $x \in \mathbb{R}^d$ *that* $a(x) = -\frac{1}{2}\|x\|^2$*, assume for every* $x \in \mathbb{R}^d$*,* $A \in \mathcal{B}(\mathbb{R}^d)$ *that* $\nu_x(A) = \int_{A\cap[-1/2,1/2]^d} \mathrm{d}\mathbf{x}$*, assume for every* $t \in [0,T]$*,* $v = (v_1,\dots,v_d)$*,* $x = (x_1,\dots,x_d) \in \mathbb{R}^d$*,* $\mathbf{x} \in$

$\mathbb{R}^d$, $y, \mathbf{y} \in \mathbb{R}$ *that* $g(x) = (2\pi)^{-d/2}\big[\prod_{i=1}^{d} |\mathfrak{s}_i|^{-1/2}\big] \exp\big(-\sum_{i=1}^{d} \frac{(x_i - \mathfrak{u}_i)^2}{2\mathfrak{s}_i}\big)$, $\mu(x) = (0, \ldots, 0)$, $\sigma(x)v = (\mathfrak{m}_1 v_1, \ldots, \mathfrak{m}_d v_d)$, *and*

$$f(t, x, \mathbf{x}, y, \mathbf{y}) = y(a(x) - \mathbf{y}a(\mathbf{x})), \tag{3.34}$$

and assume that for every $x \in \mathbb{R}^d$, $\mathfrak{i} \in \mathfrak{I}$, $t \in [0, T]$, $s \in [t, T]$ *it holds* $\mathbb{P}$*-a.s. that*

$$X_{t,s}^{x,\mathfrak{i}} = x + \int_t^s \mu\big(X_{t,r}^{x,\mathfrak{i}}\big)\, \mathrm{d}r + \int_t^s \sigma\big(X_{t,r}^{x,\mathfrak{i}}\big)\, \mathrm{d}W_r^{\mathfrak{i}} = x + \sigma(0)(W_s^{\mathfrak{i}} - W_t^{\mathfrak{i}}). \tag{3.35}$$

The solution $u\colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ *of the PDE in* (3.27) *then satisfies that for every* $t \in [0, T)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ *it holds that* $u(T, x) = (2\pi)^{-d/2}\big[\prod_{i=1}^{d} |\mathfrak{s}_i|^{-1/2}\big] \exp\big(-\sum_{i=1}^{d} \frac{(x_i - \mathfrak{u}_i)^2}{2\mathfrak{s}_i}\big)$ *and*

$$\big(\tfrac{\partial}{\partial t} u\big)(t, x) + u(t, x)\left(a(x) - \int_{[-1/2, 1/2]^d} u(t, \mathbf{x})\, a(\mathbf{x})\, \mathrm{d}\mathbf{x}\right) + \sum_{i=1}^{d} \tfrac{1}{2} |\mathfrak{m}_i|^2 \big(\tfrac{\partial^2}{\partial x_i^2} u\big)(t, x) = 0. \tag{3.36}$$

Example 3.4.6 (Allen–Cahn PDEs with conservation of mass)**.** *In this example we specialize Lemma 3.4.1 to the case of certain Allen–Cahn PDEs with cubic nonlinearity, conservation of mass, and no-flux boundary conditions (cf., e.g., Rubinstein & Sternberg [86]).*

Assume Lemma 3.4.1, let $\epsilon \in (0, \infty)$ *satisfy* $\epsilon = \frac{1}{10}$, *assume that* $d \in \{1, 2, 5, 10\}$, $D = [-1/2, 1/2]^d$, *and* $T \in \{1/5, 1/2, 1\}$, *assume for every* $n, l, m \in \mathbb{N}$ *that* $K_{n,l,m} = 10$, *assume for every* $x \in D$, $A \in \mathcal{B}(D)$ *that* $\nu_x(A) = \int_A \mathrm{d}\mathbf{x}$, *assume for every* $t \in [0, T]$, $x, \mathbf{x} \in \mathbb{D}$, $y, \mathbf{y} \in \mathbb{R}$, $v \in \mathbb{R}^d$ *that* $g(x) = \exp(-\frac{1}{4}\|x\|^2)$, $\mu(x) = (0, \ldots, 0)$, $\sigma(x)v = \epsilon v$, *and* $f(t, x, \mathbf{x}, y, \mathbf{y}) = y - y^3 - (\mathbf{y} - \mathbf{y}^3)$, *and assume that for every* $x \in \mathbb{R}^d$, $\mathfrak{i} \in \mathfrak{I}$, $t \in [0, T]$, $s \in [t, T]$ *it holds* $\mathbb{P}$*-a.s. that*

$$X_{t,s}^{x,\mathfrak{i}} = R\left(x, x + \int_t^s \mu\big(X_{t,r}^{x,\mathfrak{i}}\big)\, \mathrm{d}r + \int_t^s \sigma\big(X_{t,r}^{x,\mathfrak{i}}\big)\, \mathrm{d}W_r^{\mathfrak{i}}\right) = R(x, x + \epsilon(W_s^{\mathfrak{i}} - W_t^{\mathfrak{i}})). \tag{3.37}$$

The solution $u\colon [0, T] \times \mathbb{D} \to \mathbb{R}$ *of the PDE in* (3.27) *then satisfies that for every* $t \in [0, T)$, $x \in \partial_{\mathbb{D}}$ *it holds that* $\langle \mathbf{n}(x), (\nabla_x u)(t, x)\rangle = 0$ *and that for every* $t \in [0, T)$, $x \in \mathbb{D}$ *it holds that* $u(T, x) = \exp(-\frac{1}{4}\|x\|^2)$ *and*

$$\big(\tfrac{\partial}{\partial t} u\big)(t, x) + \tfrac{\epsilon^2}{2}(\Delta_x u)(t, x) + u(t, x) - [u(t, x)]^3 - \int_{[-1/2, 1/2]^d} u(t, \mathbf{x}) - [u(t, \mathbf{x})]^3\, \mathrm{d}\mathbf{x} = 0. \tag{3.38}$$

3.5 Numerical simulations

In this section we illustrate the performance of the machine learning-based approximation method proposed in Lemma 3.3.1 in Section 3.3.2 above by means of numerical simulations for five concrete (non-local) nonlinear PDEs; see Sections 3.5.1 to 3.5.5 below. In each of these numerical simulations we employ the general machine learning-based approximation method proposed in Lemma 3.3.1 with certain 4-layer neural networks and using the Adam optimizer (cf. (3.42) and (3.43) in Lemma 3.5.1 below and Kingma & Ba [64]).

More precisely, in each of the numerical simulations in Sections 3.5.1 to 3.5.5 the functions $\mathbb{V}_n^{j,\mathbf{s}} \colon \mathbb{R}^{\mathfrak{d}} \times \mathbb{R}^d \to \mathbb{R}$ with $n \in \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots, 8000\}$, $\mathbf{s} \in \mathbb{R}^{\varsigma}$ are implemented as N fully-connected feedforward neural networks. These neural networks consist of 4 layers (corresponding to 3 affine linear transformations in the neural networks) where the input layer is d-dimensional (with d neurons on the input layer), where the two hidden layers are both $(d + 50)$-dimensional (with $d + 50$ neurons on each of the two hidden layers), and where the output layer is 1-dimensional (with 1 neuron on the output layer). We refer to Fig. 3.1 for a graphical illustration of the neural network architecture used in the numerical simulations in Sections 3.5.1 to 3.5.5.

As activation functions just in front of the two hidden layers we employ, in Sections 3.5.1 to 3.5.4 below, multidimensional versions of the hyperbolic tangent function

$$\mathbb{R} \ni x \mapsto (e^x + e^{-x})^{-1}(e^x - e^{-x}) \in \mathbb{R}, \tag{3.39}$$

and we employ, in Section 3.5.5 below, multidimensional versions of the ReLU function

$$\mathbb{R} \ni x \mapsto \max\{x, 0\} \in \mathbb{R}. \tag{3.40}$$

In addition, in Sections 3.5.1, 3.5.2 and 3.5.4 we use the square function $\mathbb{R} \ni x \mapsto x^2 \in \mathbb{R}$ as activation function just in front of the output layer and in Sections 3.5.3 and 3.5.5 we use the identity function $\mathbb{R} \ni x \mapsto x \in \mathbb{R}$ as activation function just in front of the output layer. Furthermore, we employ Xavier initialization to initialize all neural network parameters; see Glorot & Bengio [44] for details. We did not employ batch normalization in our simulations.

Each of the numerical experiments presented below was performed with the **J**ulia library **H**ighDimPDE.jl on a NVIDIA TITAN RTX GPU with 1350 MHz core clock and 24 GB GDDR6 memory with 7000 MHz clock rate where the underlying system consisted of an AMD EPYC 7742 64-core CPU with 2TB memory running **J**ulia 1.7.2 on Ubuntu 20.04.3. We refer to **??** below for the employed **J**ulia source codes.

Framework 3.5.1. *Assume Lemma 3.3.1, assume* $\mathfrak{d} = (d+50)(d+1)+(d+50)(d+51)+(d+51)$, *let* $\varepsilon, \beta_1, \beta_2 \in \mathbb{R}$, $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0, \infty)$ *satisfy* $\varepsilon = 10^{-8}$, $\beta_1 = \frac{9}{10}$, *and* $\beta_2 = \frac{999}{1000}$, *let* $g \colon D \to \mathbb{R}$, $\mu \colon D \to \mathbb{R}^d$, *and* $\sigma \colon D \to \mathbb{R}^{d\times d}$ *be continuous, let* $u = (u(t,x))_{(t,x)\in[0,T]\times\mathbb{D}} \in C^{1,2}([0,T] \times \mathbb{D}, \mathbb{R})$ *have at most polynomially growing partial derivatives, assume for every* $t \in (0, T]$, $x \in \partial_{\mathbb{D}}$ *that* $\langle \mathbf{n}(x), (\nabla_x u)(t,x)\rangle = 0$, *assume for every* $t \in [0, T]$, $x \in \mathbb{D}$, $j \in \mathbb{N}$, $\mathbf{s} \in \mathbb{R}^{\varsigma}$ *that* $u(0,x) = g(x) = \mathbb{V}_0^{j,\mathbf{s}}(\theta, x)$, $\int_{\mathbb{D}} \big| f\big(t, x, \mathbf{x}, u(t,x), u(t,\mathbf{x})\big)\big| \, \nu_x(\mathrm{d}\mathbf{x}) < \infty$, *and*

$$\begin{aligned}\big(\tfrac{\partial}{\partial t}u\big)(t,x) &= \tfrac{1}{2}\,\mathrm{Trace}\big(\sigma(x)[\sigma(x)]^*(\mathrm{Hess}_x\, u)(t,x)\big) + \big\langle \mu(x), (\nabla_x u)(t,x)\big\rangle \\ &\quad + \int_{\mathbb{D}} f\big(t, x, \mathbf{x}, u(t,x), u(t,\mathbf{x})\big)\, \nu_x(\mathrm{d}\mathbf{x}),\end{aligned} \tag{3.41}$$

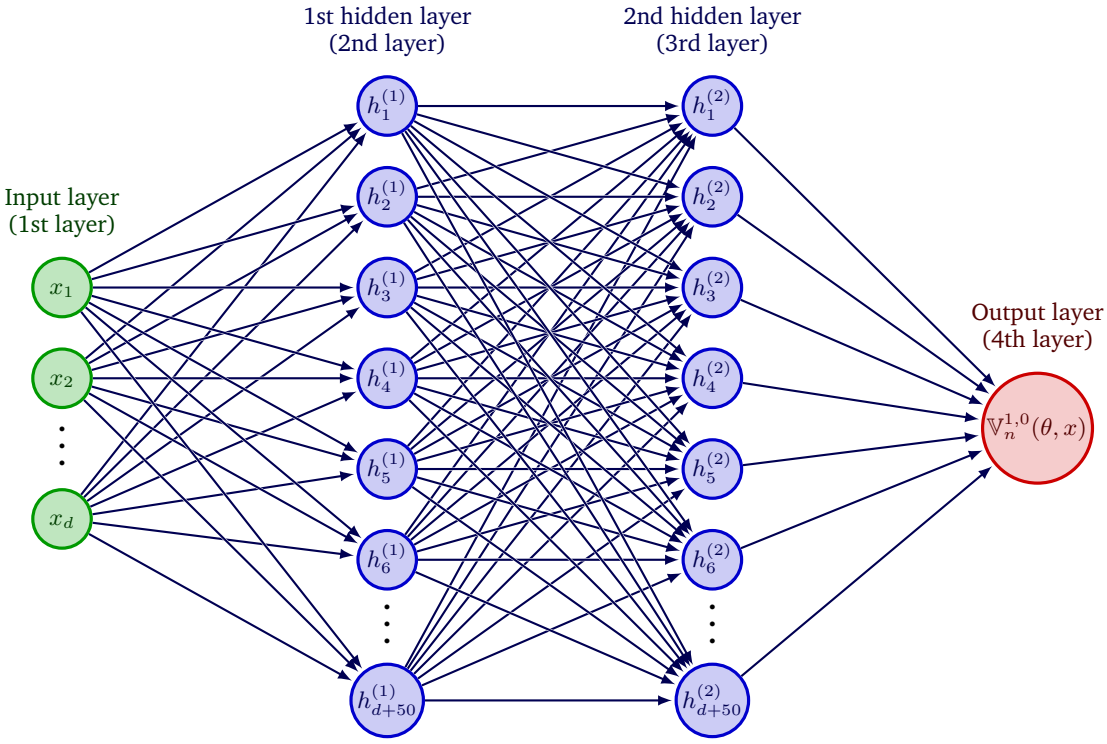

Fig. 3.1: Graphical illustration of the neural network architecture used in the numerical simulations. In Sections 3.5.1 to 3.5.5 we employ neural networks with 4 layers (corresponding to 3 affine linear transformations in the neural networks) with d neurons on the input layer (corresponding to a d-dimensional input layer), with $d + 50$ neurons on the 1st hidden layer (corresponding to a $(d + 50)$-dimensional 1st hidden layer), with $d + 50$ neurons on the 2nd hidden layer (corresponding to a $(d + 50)$-dimensional 2nd hidden layer), and with 1 neuron on the output layer (corresponding to a 1-dimensional output layer) in the numerical simulations.

assume for every $m \in \mathbb{N}$, $i \in \{0, 1, \dots, N\}$ *that* $J_m = 8000$, $t_i = \frac{iT}{N}$, *and* $\varrho = 2\mathfrak{d}$, *and assume for every* $n \in \{1, 2, \dots, N\}$, $m \in \mathbb{N}$, $x = (x_1, \dots, x_{\mathfrak{d}})$, $y = (y_1, \dots, y_{\mathfrak{d}})$, $\eta = (\eta_1, \dots, \eta_{\mathfrak{d}}) \in \mathbb{R}^{\mathfrak{d}}$ *that*

$$\Xi_0^n(x, y, \eta) = 0, \quad \Psi_m^n(x, y, \eta) = \big(\beta_1 x + (1 - \beta_1)\eta, \beta_2 y + (1 - \beta_2)((\eta_1)^2, \dots, (\eta_{\mathfrak{d}})^2)\big), \tag{3.42}$$

and

$$\psi_m^n(x, y) = \left(\left[\sqrt{\tfrac{|y_1|}{1-(\beta_2)^m}} + \varepsilon\right]^{-1} \frac{\gamma_m x_1}{1 - (\beta_1)^m}, \dots, \left[\sqrt{\tfrac{|y_{\mathfrak{d}}|}{1-(\beta_2)^m}} + \varepsilon\right]^{-1} \frac{\gamma_m x_{\mathfrak{d}}}{1 - (\beta_1)^m}\right). \tag{3.43}$$

3.5.1 Fisher–KPP PDEs with Neumann boundary conditions

In this subsection we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the solutions of certain Fisher–KPP PDEs with Neumann boundary conditions (cf., e.g., Bian et al. [18] and Wang et al. [92]).

Assume Lemma 3.5.1, let $\epsilon \in (0, \infty)$ satisfy $\epsilon = \frac{1}{10}$, assume that $d \in \{1, 2, 5, 10\}$, $D = [-1/2, 1/2]^d$, $T \in \{1/5, 1/2, 1\}$, $N = 10$, $K_1 = K_2 = \ldots = K_N = 1$, and $M_1 = M_2 = \ldots = M_N = 500$, assume for every $n, m, j \in \mathbb{N}$, $\omega \in \Omega$ that $\xi^{n,m,j}(\omega) = (0, \dots, 0)$, assume for every $m \in \mathbb{N}$ that $\gamma_m = 10^{-2}$, and assume for every $s, t \in [0, T]$, $x, \mathbf{x} \in \mathbb{D}$, $y, \mathbf{y} \in \mathbb{R}$, $v \in \mathbb{R}^d$ that $g(x) = \exp(-\frac{1}{4}\|x\|^2)$, $\mu(x) = (0, \dots, 0)$, $\sigma(x)v = \epsilon v$, $f(t, x, \mathbf{x}, y, \mathbf{y}) = y(1 - y)$, and

$$H(t, s, x, v) = R(x, x + \mu(x)(t - s) + \sigma(x)v) = R(x, x + \epsilon v) \tag{3.44}$$

(cf. (3.6) and (3.17)). The solution $u\colon [0, T] \times \mathbb{D} \to \mathbb{R}$ of the PDE in (3.41) then satisfies that for every $t \in (0, T]$, $x \in \partial_{\mathbb{D}}$ it holds that $\langle \mathbf{n}(x), (\nabla_x u)(t, x)\rangle = 0$ and that for every $t \in [0, T]$, $x \in \mathbb{D}$ it holds that $u(0, x) = \exp(-\frac{1}{4}\|x\|^2)$ and

$$\big(\tfrac{\partial}{\partial t} u\big)(t, x) = \tfrac{\epsilon^2}{2}(\Delta_x u)(t, x) + u(t, x)\big(1 - u(t, x)\big). \tag{3.45}$$

In (3.45) the function $u\colon [0, T] \times D \to \mathbb{R}$ models the proportion of a particular type of alleles in a biological population spatially structured over D. For every $t \in [0, T]$, $x \in \mathbb{R}^d$ the number $u(t, x) \in \mathbb{R}$ describes the proportion of individuals with a particular type of alleles located at position $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ at time $t \in [0, T]$. In Table 3.1 we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the mean of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \dots, 0))$, the standard deviation of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \dots, 0))$, the relative L^1-approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \dots, 0))$, the uncorrected sample standard deviation of the approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \dots, 0))$, and the average runtime in seconds needed for calculating one realization of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \dots, 0))$ based on 5 independent realizations (5 independent runs). The reference value, which is used as an approximation for the unknown value $u(T, (0, \dots, 0))$ of the exact solution of (3.45),

d	T	N	Mean of the approx. method	Standard deviation of the approx. method	Reference value	Relative L^1-approx. error	Standard deviation of the error	Average runtime in seconds
1	$1/5$	10	0.9995902	0.0000107	0.9996057	0.0000155	0.0000107	24.887
2	$1/5$	10	0.9991759	0.0000191	0.9991887	0.0000186	0.0000116	26.175
5	$1/5$	10	0.9979572	0.0000388	0.9979693	0.0000303	0.0000235	27.312
10	$1/5$	10	0.9959224	0.0000341	0.9959337	0.0000275	0.0000196	28.972
1	$1/2$	10	0.9992463	0.0000341	0.9992572	0.0000237	0.0000248	26.631
2	$1/2$	10	0.9984982	0.0000287	0.9985442	0.0000460	0.0000287	27.007
5	$1/2$	10	0.9962227	0.0000330	0.9962314	0.0000306	0.0000041	27.632
10	$1/2$	10	0.9925257	0.0001663	0.9921744	0.0003541	0.0001676	28.743
1	1	10	0.9991423	0.0000331	0.9989768	0.0001657	0.0000332	26.601
2	1	10	0.9982349	0.0000782	0.9982498	0.0000605	0.0000430	26.965
5	1	10	0.9956516	0.0000853	0.9957053	0.0000839	0.0000466	27.428
10	1	10	0.9912297	0.0001072	0.9904936	0.0007431	0.0001083	28.521

Tab. 3.1: Numerical simulations for the approximation method in Lemma 3.3.1 in the case of the Fisher–KPP PDEs with Neumann boundary conditions in (3.45) in Section 3.5.1.

has been calculated via the MLP approximation method for non-local nonlinear PDEs in Lemma 3.4.1 (cf. Lemma 3.4.2 and Beck et al. [9, Remark 3.3]).

3.5.2 Non-local competition PDEs

In this subsection we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the solutions of certain non-local competition PDEs (cf., e.g., Doebeli & Ispolatov [30], Berestycki et al. [17], Perthame & Génieys [82], and Génieys et al. [43]).

Assume Lemma 3.5.1, let $\mathfrak{s}, \epsilon \in (0,\infty)$ satisfy $\mathfrak{s} = \epsilon = \frac{1}{10}$, assume that $d \in \{1,2,5,10\}$, $\mathbb{D} = \mathbb{R}^d$, $T \in \{1/5, 1/2, 1\}$, $N = 10$, $K_1 = K_2 = \ldots = K_N = 5$, and $M_1 = M_2 = \ldots = M_N = 500$, assume for every $n,m,j \in \mathbb{N}$, $\omega \in \Omega$ that $\xi^{n,m,j}(\omega) = (0,\ldots,0)$, assume for every $m \in \mathbb{N}$ that $\gamma_m = 10^{-2}$, and assume for every $s,t \in [0,T]$, $v,x,\mathbf{x} \in \mathbb{R}^d$, $y,\mathbf{y} \in \mathbb{R}$, $A \in \mathcal{B}(\mathbb{R}^d)$ that $\nu_x(A) = \pi^{-d/2}\mathfrak{s}^{-d}\int_A \exp\!\big(-\mathfrak{s}^{-2}\|x-\mathbf{x}\|^2\big)\,\mathrm{d}\mathbf{x}$, $g(x) = \exp(-\frac{1}{4}\|x\|^2)$, $\mu(x) = (0,\ldots,0)$, $\sigma(x)v = \epsilon v$, $f(t,x,\mathbf{x},y,\mathbf{y}) = y(1-\mathbf{y}\mathfrak{s}^d\pi^{d/2})$, and

$$H(t,s,x,v) = x + \mu(x)(t-s) + \sigma(x)v = x + \epsilon v \tag{3.46}$$

(cf. (3.6) and (3.17)). The solution $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ of the PDE in (3.41) then satisfies that for every $t \in [0,T]$, $x \in \mathbb{R}^d$ it holds that $u(0,x) = \exp(-\frac{1}{4}\|x\|^2)$ and

$$\big(\tfrac{\partial}{\partial t}u\big)(t,x) = \tfrac{\epsilon^2}{2}(\Delta_x u)(t,x) + u(t,x)\bigg(1 - \int_{\mathbb{R}^d} u(t,\mathbf{x})\exp\!\big(-\tfrac{\|x-\mathbf{x}\|^2}{\mathfrak{s}^2}\big)\,\mathrm{d}\mathbf{x}\bigg). \tag{3.47}$$

d	T	N	Mean of the approx. method	Standard deviation of the approx. method	Reference value	Relative L^1-approx. error	Standard deviation of the error	Average runtime in seconds
1	$1/5$	5	1.1748404	0.0006512	1.1735975	0.0010591	0.0005549	20.571
2	$1/5$	5	1.2114236	0.0008700	1.2096305	0.0014823	0.0007193	25.042
5	$1/5$	5	1.2186650	0.0007070	1.2159038	0.0022709	0.0005814	54.644
10	$1/5$	5	1.2153864	0.0007789	1.2128666	0.0020776	0.0006422	74.331
1	$1/2$	5	1.4755801	0.0032738	1.4694976	0.0041392	0.0022278	20.182
2	$1/2$	5	1.6112576	0.0110426	1.5948898	0.0103067	0.0068414	25.178
5	$1/2$	5	1.6433913	0.0067468	1.6186897	0.0152602	0.0041681	53.618
10	$1/2$	5	1.6323552	0.0053956	1.6090688	0.0144720	0.0033532	73.648
1	1	5	2.0795628	0.0223341	2.0493301	0.0147525	0.0108982	19.836
2	1	5	2.5651031	0.0513671	2.4683060	0.0392160	0.0208107	24.700
5	1	5	2.6977694	0.0381160	2.5606137	0.0535636	0.0148855	52.343
10	1	5	2.6490054	0.0155291	2.5299994	0.0470380	0.0061380	73.186

Tab. 3.2: Numerical simulations for the approximation method in Lemma 3.3.1 in the case of the non-local competition PDEs in (3.47) in Section 3.5.2.

In (3.47) the function $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ models the evolution of a population characterized by a set of d biological traits under the combined effects of selection, competition and mutation. For every $t \in [0,T]$, $x \in \mathbb{R}^d$ the number $u(t,x) \in \mathbb{R}$ describes the number of individuals with traits $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ at time $t \in [0,T]$. In Table 3.2 we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the mean of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, the standard deviation of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, the relative L^1-approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, the uncorrected sample standard deviation of the approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, and the average runtime in seconds needed for calculating one realization of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$ based on 5 independent realizations (5 independent runs). The reference value, which is used as an approximation for the unknown value $u(T,(0,\ldots,0))$ of the exact solution of (3.47), has been calculated via the MLP approximation method for non-local nonlinear PDEs in Lemma 3.4.1 (cf. Lemma 3.4.3 and Beck et al. [9, Remark 3.3]).

3.5.3 Non-local sine-Gordon type PDEs

In this subsection we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the solutions of non-local sine-Gordon type PDEs (cf., e.g., Hairer & Shen [49], Barone et al. [7], and Coleman [25]).

Assume Lemma 3.5.1, let $\mathfrak{s},\epsilon \in (0,\infty)$ satisfy $\mathfrak{s} = \epsilon = \frac{1}{10}$, assume that $d \in \{1,2,5,10\}$, $\mathbb{D} = \mathbb{R}^d$, $T \in \{1/5, 1/2, 1\}$, $N = 10$, $K_1 = K_2 = \ldots = K_N = 5$, and $M_1 = M_2 = \ldots = M_N = 500$, assume for every $n,m,j \in \mathbb{N}$, $\omega \in \Omega$ that $\xi^{n,m,j}(\omega) = (0,\ldots,0)$, assume for every $m \in \mathbb{N}$ that $\gamma_m = 10^{-3}$, and assume for every $s,t \in [0,T]$, $v,x,\mathbf{x} \in \mathbb{R}^d$, $y,\mathbf{y} \in \mathbb{R}$, $A \in \mathcal{B}(\mathbb{R}^d)$

d	T	N	Mean of the approx. method	Standard deviation of the approx. method	Reference value	Relative L^1-approx. error	Standard deviation of the error	Average runtime in seconds
1	$1/5$	10	1.1363013	0.0000101	1.1366512	0.0003079	0.0000089	23.635
2	$1/5$	10	1.1678476	0.0000118	1.1685004	0.0005586	0.0000101	24.788
5	$1/5$	10	1.1731812	0.0000087	1.1740671	0.0007546	0.0000074	24.233
10	$1/5$	10	1.1704700	0.0000063	1.1715686	0.0009377	0.0000054	24.767
1	$1/2$	10	1.3514235	0.0000152	1.3529022	0.0010930	0.0000112	22.622
2	$1/2$	10	1.4393708	0.0000245	1.4423641	0.0020753	0.0000170	23.419
5	$1/2$	10	1.4546282	0.0000816	1.4598476	0.0035754	0.0000559	23.739
10	$1/2$	10	1.4473282	0.0000739	1.4503958	0.0021150	0.0000510	24.222
1	1	10	1.7114614	0.0000309	1.7136091	0.0012533	0.0000180	22.067
2	1	10	1.9019763	0.0000288	1.9062322	0.0022326	0.0000151	22.707
5	1	10	1.9364921	0.0000602	1.9411610	0.0024052	0.0000310	22.899
10	1	10	1.9223347	0.0001494	1.9272222	0.0025360	0.0000775	23.719

Tab. 3.3: Numerical simulations for the approximation method in Lemma 3.3.1 in the case of the non-local sine-Gordon PDEs in (3.49) in Section 3.5.3.

that $\nu_x(A) = \pi^{-d/2}\mathfrak{s}^{-d}\int_A \exp\big(-\mathfrak{s}^{-2}\|x-\mathbf{x}\|^2\big)\,\mathrm{d}\mathbf{x}$, $g(x) = \exp(-\frac{1}{4}\|x\|^2)$, $\mu(x) = (0,\ldots,0)$, $\sigma(x)v = \epsilon v$, $f(t,x,\mathbf{x},y,\mathbf{y}) = \sin(y) - \mathbf{y}\pi^{d/2}\mathfrak{s}^d$, and

$$H(t,s,x,v) = x + \mu(x)(t-s) + \sigma(x)v = x + \epsilon v \tag{3.48}$$

(cf. (3.6) and (3.17)). The solution $u\colon [0,T]\times\mathbb{R}^d \to \mathbb{R}$ of the PDE in (3.41) then satisfies that for every $t\in[0,T]$, $x\in\mathbb{R}^d$ it holds that $u(0,x) = \exp(-\frac{1}{4}\|x\|^2)$ and

$$\big(\tfrac{\partial}{\partial t}u\big)(t,x) = \tfrac{\epsilon^2}{2}(\Delta_x u)(t,x) + \sin(u(t,x)) - \int_{\mathbb{R}^d} u(t,\mathbf{x})\,\exp\big(-\tfrac{\|x-\mathbf{x}\|^2}{\mathfrak{s}^2}\big)\,\mathrm{d}\mathbf{x}. \tag{3.49}$$

In Table 3.3 we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the mean of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, the standard deviation of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, the relative L^1-approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, the uncorrected sample standard deviation of the approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$, and the average runtime in seconds needed for calculating one realization of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N,(0,\ldots,0))$ based on 5 independent realizations (5 independent runs). The reference value, which is used as an approximation for the unknown value $u(T,(0,\ldots,0))$ of the exact solution of (3.49), has been calculated via the MLP approximation method for non-local nonlinear PDEs in Lemma 3.4.1 (cf. Lemma 3.4.4 and Beck et al. [9, Remark 3.3]).

3.5.4 Replicator-mutator PDEs

In this subsection we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the solutions of certain replicator-mutator PDEs describing the

dynamics of a phenotype distribution under the combined effects of selection and mutation (cf., e.g., Hamel et al. [50]).

Assume Lemma 3.5.1, let $\mathcal{D} \subseteq \mathbb{R}^d$, $\mathfrak{m}_1, \mathfrak{m}_2, \ldots, \mathfrak{m}_d, \mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_d, \mathfrak{u}_1, \mathfrak{u}_2, \ldots, \mathfrak{u}_d, \mathfrak{t} \in \mathbb{R}$ satisfy for every $k \in \{1, 2, \ldots, d\}$ that $\mathfrak{m}_k = \frac{1}{10}$, $\mathfrak{s}_k = \frac{1}{20}$, $\mathfrak{u}_k = 0$, and $\mathfrak{t} = \frac{1}{50}$, assume that $d \in \{1, 2, 5, 10\}$, $\mathbb{D} = \mathbb{R}^d$, $T \in \{1/10, 1/5, 1/2\}$, $N = 10$, $K_1 = K_2 = \ldots = K_N = 5$, let $a \in C(\mathbb{R}^d, \mathbb{R})$, $\delta \in C(\mathbb{R}^d, (0, \infty))$ satisfy for every $x \in \mathbb{R}^d$ that $a(x) = -\frac{1}{2}\|x\|^2$, and assume for every $s, t \in [0, T]$, $v = (v_1, \ldots, v_d)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, $\mathbf{x} \in \mathbb{R}^d$, $y, \mathbf{y} \in \mathbb{R}$, $A \in \mathcal{B}(\mathbb{R}^d)$ that $\nu_x(A) = \int_{A \cap \mathcal{D}} \delta(\mathbf{x}) \, \mathrm{d}\mathbf{x}$, $g(x) = (2\pi)^{-d/2} \big[\prod_{i=1}^d |\mathfrak{s}_i|^{-1/2}\big] \exp\big(-\sum_{i=1}^d \frac{(x_i - \mathfrak{u}_i)^2}{2\mathfrak{s}_i}\big)$, $\mu(x) = (0, \ldots, 0)$, $\sigma(x)v = (\mathfrak{m}_1 v_1, \ldots, \mathfrak{m}_d v_d)$, $f(t, x, \mathbf{x}, y, \mathbf{y}) = y(a(x) - \mathbf{y}a(\mathbf{x})[\delta(\mathbf{x})]^{-1})$, and

$$H(t, s, x, v) = x + \mu(x)(t - s) + \sigma(x)v = x + (\mathfrak{m}_1 v_1, \ldots, \mathfrak{m}_d v_d) \tag{3.50}$$

(cf. (3.6) and (3.17)). The solution $u \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ of the PDE in (3.41) then satisfies that for every $t \in [0, T]$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$u(0, x) = (2\pi)^{-d/2} \left[\prod_{i=1}^d |\mathfrak{s}_i|^{-1/2}\right] \exp\Big(-\sum_{i=1}^d \tfrac{(x_i - \mathfrak{u}_i)^2}{2\mathfrak{s}_i}\Big) \tag{3.51}$$

and

$$\big(\tfrac{\partial}{\partial t} u\big)(t, x) = u(t, x)\left(a(x) - \int_{\mathcal{D}} u(t, \mathbf{x}) \, a(\mathbf{x}) \, \mathrm{d}\mathbf{x}\right) + \sum_{i=1}^d \tfrac{1}{2} |\mathfrak{m}_i|^2 \big(\tfrac{\partial^2}{\partial x_i^2} u\big)(t, x). \tag{3.52}$$

In (3.52) the function $u \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ models the evolution of the phenotype distribution of a population composed of a set of d biological traits under the combined effects of selection and mutation. For every $t \in [0, T]$, $x \in \mathbb{R}^d$ the number $u(t, x) \in \mathbb{R}$ describes the number of individuals with traits $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ at time $t \in [0, T]$. The function a models a quadratic Malthusian fitness function.

In Table 3.4 we use the machine learning-based method in Lemma 3.5.1 to approximately solve the PDE in Eq. (3.52) above in the case $\mathcal{D} = \mathbb{R}^d$. More precisely, we assume for every $n, m, j \in \mathbb{N}$ that $\xi^{n,m,j} = 0$, $\gamma_m = 1/100$, $M_n = 1000$ and we assume for every $\mathbf{x} \in \mathbb{R}^d$ that $\delta(\mathbf{x}) = (2\pi)^{-d/2} \mathfrak{t}^{-d} \exp\big(-\frac{\|\mathbf{x}\|^2}{2\mathfrak{t}^2}\big)$ to approximately calculate the mean of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \ldots, 0))$, the standard deviation of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \ldots, 0))$, the relative L^1-approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \ldots, 0))$, the uncorrected sample standard deviation of the approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \ldots, 0))$, and the average runtime in seconds needed for calculating one realization of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0, \ldots, 0))$ based on 5 independent realizations (5 independent runs). The value $u(T, (0, \ldots, 0))$ of the exact solution of (3.52) has been calculated by means of Lemma 3.5.2 below.

In Fig. 3.2 we use the machine learning-based method in Lemma 3.5.1 to approximate the solution $u \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ of the PDE in Eq. (3.52) above with $d = 5$, $T = 1/2$, and $\mathcal{D} = \mathbb{R}^d$. The right-hand side of Fig. 3.2 shows a plot of $[-1/4, 1/4] \ni x \mapsto u(t, (x, 0, \ldots, 0)) \in \mathbb{R}$ for $t \in \{0, 0.05, 0.1, 0.15\}$ where u is the exact solution of the PDE in Eq. (3.52) with $d = 5$, $T = 1/2$, and $\mathcal{D} = \mathbb{R}^d$ computed via Eq. (3.54) in Lemma 3.5.2 below. The left-hand side of Fig. 3.2 shows a plot of $[-1/4, 1/4] \ni x \mapsto \mathbb{V}_n^{1,0}(\Theta_{M_n}^n(\omega), (x, 0, \ldots, 0)) \in \mathbb{R}$ for $n \in \{0, 1, 2, 3\}$

d	T	N	Mean of the approx. method	Standard deviation of the approx. method	Reference value	Relative L^1-approx. error	Standard deviation of the error	Average runtime in seconds
1	$1/10$	10	1.7650547	0.0048907	1.7709574	0.0033330	0.0027616	43.949
2	$1/10$	10	3.1210874	0.0015513	3.1362901	0.0048474	0.0004946	45.002
5	$1/10$	10	17.1948978	0.0160821	17.4196954	0.0129048	0.0009232	45.934
10	$1/10$	10	295.8776489	0.0572639	303.4457874	0.0249407	0.0001887	47.750
1	$1/5$	10	1.7499938	0.0005580	1.7582066	0.0046711	0.0003174	43.129
2	$1/5$	10	3.0621917	0.0027811	3.0912904	0.0094131	0.0008996	44.443
5	$1/5$	10	16.3846066	0.0139748	16.8015567	0.0248162	0.0008318	45.019
10	$1/5$	10	268.2944397	0.0623432	282.2923073	0.0495864	0.0002208	45.612
1	$1/2$	10	1.7018557	0.0060157	1.7222757	0.0118564	0.0034929	42.092
2	$1/2$	10	2.8911286	0.0027431	2.9662336	0.0253200	0.0009248	42.657
5	$1/2$	10	14.2520916	0.1356645	15.1535149	0.0594861	0.0089527	43.338
10	$1/2$	10	201.6446228	0.3009756	229.6290127	0.1218678	0.0013107	44.190

Tab. 3.4: Numerical simulations for the approximation method in Lemma 3.3.1 in the case of the replicator-mutator PDEs in (3.52) in Section 3.5.4 where we assume for every $n, m, j \in \mathbb{N}$ that $\mathcal{D} = \mathbb{R}^d$, $\xi^{n,m,j} = 0$, $\gamma_m = 1/100$, and $M_n = 1000$ and where we assume for every $\mathbf{x} \in \mathbb{R}^d$ that $\delta(\mathbf{x}) = (2\pi)^{-d/2}\mathfrak{t}^{-d}\exp\big(-\frac{\|\mathbf{x}\|^2}{2\mathfrak{t}^2}\big)$.

and one realization $\omega \in \Omega$ where the functions $\mathbb{R}^d \ni x \mapsto \mathbb{V}_n^{1,0}(\Theta_{M_n}^n(\omega), x) \in \mathbb{R}$ for $n \in \{0, 1, 2, 3\}$, $\omega \in \Omega$ were computed via Lemma 3.5.1 as an approximation of the solution of the PDE in Eq. (3.52) with $d = 5$, $T = 1/2$, and $\mathcal{D} = [-1/2, 1/2]^d$. For the approximation, we take $M_1 = M_2 = \ldots = M_N = 2000$, $\gamma_1 = \gamma_2 = \cdots = \gamma_{2000} = 1/200$, and $\delta = \mathbb{1}_{\mathbb{R}^d}$ and we take $\xi^{n,m,j} \colon \Omega \to \mathbb{R}^d$, $n, m, j \in \mathbb{N}$, to be independent $\mathcal{U}_{[-1/2,1/2]^d}$-distributed random variables. Note that the solution of the PDE in Eq. (3.52) in the case $\mathcal{D} = [-R, R]^d$ with $R \in (0, \infty)$ sufficiently large is a good approximation of the solution $u \colon [0, T] \times \mathbb{R}^d \to \mathbb{R}$ of the PDE in Eq. (3.52) in the case $\mathcal{D} = \mathbb{R}^d$ since we have that for all $t \in [0, T]$ the value $u(t, x)$ of the solution u of the PDE in Eq. (3.52) in the case $\mathcal{D} = \mathbb{R}^d$ quickly tends to 0 as $\|x\|$ tends to ∞.

Lemma 3.5.2. *Let $d \in \mathbb{N}$, $\mathfrak{u}_1, \mathfrak{u}_2, \ldots, \mathfrak{u}_d \in \mathbb{R}$, $\mathfrak{m}_1, \mathfrak{m}_2, \ldots, \mathfrak{m}_d, \mathfrak{s}_1, \mathfrak{s}_2, \ldots, \mathfrak{s}_d \in (0, \infty)$, let $a \colon \mathbb{R}^d \to \mathbb{R}$ satisfy for every $x \in \mathbb{R}^d$ that $a(x) = -\frac{1}{2}\|x\|^2$, for every $i \in \{1, 2, \ldots, d\}$ let $\mathfrak{S}_i \colon [0, \infty) \to (0, \infty)$ and $\mathfrak{U}_i \colon [0, \infty) \to \mathbb{R}$ satisfy for every $t \in [0, \infty)$ that*

$$\mathfrak{S}_i(t) = \mathfrak{m}_i\left[\frac{\mathfrak{m}_i \sinh(\mathfrak{m}_i t) + \mathfrak{s}_i \cosh(\mathfrak{m}_i t)}{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)}\right] \quad \text{and} \quad \mathfrak{U}_i(t) = \frac{\mathfrak{m}_i \mathfrak{u}_i}{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)}, \tag{3.53}$$

and let $u \colon [0, \infty) \times \mathbb{R}^d \to \mathbb{R}$ satisfy for every $t \in [0, \infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ that

$$u(t, x) = (2\pi)^{-d/2}\left[\prod_{i=1}^{d} |\mathfrak{S}_i(t)|^{-1/2}\right] \exp\left(-\sum_{i=1}^{d} \frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right). \tag{3.54}$$

Then

(i) it holds that $u \in C^{1,2}([0, \infty) \times \mathbb{R}^d, \mathbb{R})$,

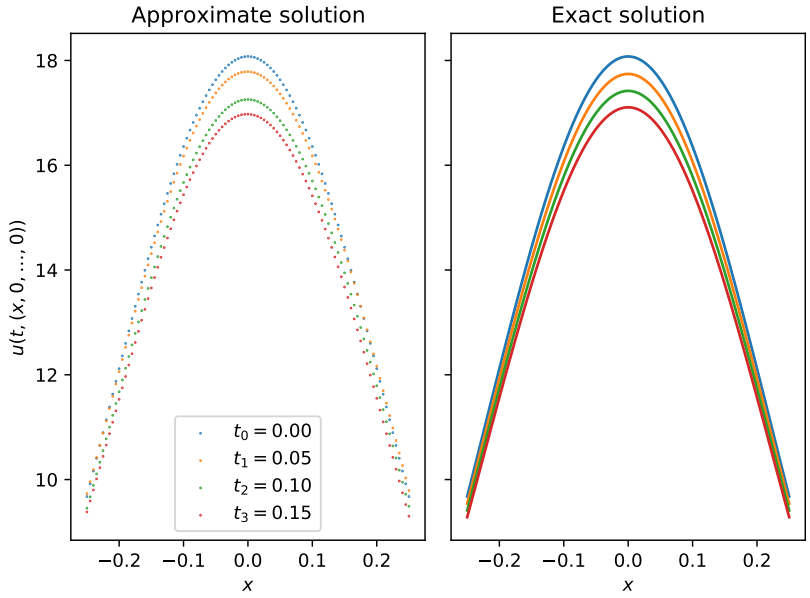

Fig. 3.2: Plot of a machine learning-based approximation of the solution of the replicator-mutator PDE in Eq. (3.52) in the case $d = 5$, $T = 1/2$, and $\mathcal{D} = \mathbb{R}^d$. The left-hand side shows a plot of $[-1/4, 1/4] \ni x \mapsto \mathbb{V}_n^{1,0}(\Theta_{M_n}^n(\omega), (x, 0, \ldots, 0)) \in \mathbb{R}$ for $n \in \{0, 1, 2, 3\}$ and one realization $\omega \in \Omega$ where the functions $\mathbb{R}^d \ni x \mapsto \mathbb{V}_n^{1,0}(\Theta_{M_n}^n(\omega), x) \in \mathbb{R}$ for $n \in \{0, 1, 2, 3\}$, $\omega \in \Omega$ were computed via Lemma 3.5.1 as an approximation of the solution of the PDE in Eq. (3.52) with $d = 5$, $T = 1/2$, and $\mathcal{D} = [-1/2, 1/2]^d$ where we take $M_1 = M_2 = \ldots = M_N = 2000$, $\gamma_1 = \gamma_2 = \cdots = \gamma_{2000} = 1/200$, and $\delta = \mathbb{1}_{\mathbb{R}^d}$ and where we take $\xi^{n,m,j} \colon \Omega \to \mathbb{R}^d$, $n, m, j \in \mathbb{N}$, to be independent $\mathcal{U}_{[-1/2,1/2]^d}$-distributed random variables. The right-hand side of Fig. 3.2 shows a plot of $[-1/4, 1/4] \ni x \mapsto u(t, (x, 0, \ldots, 0)) \in \mathbb{R}$ for $t \in \{0, 0.05, 0.1, 0.15\}$ where u is the exact solution of the PDE in Eq. (3.52) with $d = 5$, $T = 1/2$, and $\mathcal{D} = \mathbb{R}^d$.

(ii) it holds for every $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ *that*

$$u(0,x) = (2\pi)^{-d/2}\left[\prod_{i=1}^{d} |\mathfrak{s}_i|^{-1/2}\right] \exp\left(-\sum_{i=1}^{d} \frac{(x_i - \mathfrak{u}_i)^2}{2\mathfrak{s}_i}\right), \tag{3.55}$$

and

(iii) it holds for every $t \in [0,\infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ *that*

$$\left(\tfrac{\partial}{\partial t} u\right)(t,x) = u(t,x)\left(a(x) - \int_{\mathbb{R}^d} u(t,\mathbf{x})\, a(\mathbf{x})\, \mathrm{d}\mathbf{x}\right) + \sum_{i=1}^{d} \tfrac{1}{2}|\mathfrak{m}_i|^2 \left(\tfrac{\partial^2}{\partial x_i^2} u\right)(t,x). \tag{3.56}$$

Proof of Lemma 3.5.2. First, note that the fact that for every $i \in \{1,2,\ldots,d\}$ it holds that $\mathfrak{S}_i \in C^\infty([0,\infty),(0,\infty))$, the fact that for every $i \in \{1,2,\ldots,d\}$ it holds that $\mathfrak{U}_i \in C^\infty([0,\infty),\mathbb{R})$, and (3.54) establish Item (i). Moreover, observe that the fact that for every $i \in \{1,2,\ldots,d\}$ it holds that $\mathfrak{S}_i(0) = \mathfrak{s}_i$, the fact that for every $i \in \{1,2,\ldots,d\}$ it holds that $\mathfrak{U}_i(0) = \mathfrak{u}_i$, and (3.53) prove Item (ii). Next note that (3.54) ensures that for every $t \in [0,\infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$u(t,x) = \prod_{i=1}^{d}\left[(2\pi\mathfrak{S}_i(t))^{-1/2} \exp\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\right]. \tag{3.57}$$

The product rule hence implies that for every $t \in [0,\infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
&\left(\tfrac{\partial}{\partial t} u\right)(t,x) \\
&= \frac{\partial}{\partial t}\left(\prod_{i=1}^{d}\left[(2\pi\mathfrak{S}_i(t))^{-1/2} \exp\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\right]\right) \\
&= \sum_{i=1}^{d}\left[\left[\prod\nolimits_{j\in\{1,\ldots,d\}\setminus\{i\}}\left((2\pi\mathfrak{S}_j(t))^{-1/2} \exp\left(-\frac{(x_j - \mathfrak{U}_j(t))^2}{2\mathfrak{S}_j(t)}\right)\right)\right]\right. \\
&\quad \left.\cdot\left[\frac{\partial}{\partial t}\left((2\pi\mathfrak{S}_i(t))^{-1/2} \exp\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\right)\right]\right].
\end{aligned} \tag{3.58}$$

The chain rule, the product rule, and (3.57) therefore show that for every $t \in [0,\infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$
\begin{aligned}
&\left(\tfrac{\partial}{\partial t} u\right)(t,x) \\
&= \sum_{i=1}^{d} \left[\left[\prod\nolimits_{j \in \{1,\ldots,d\} \setminus \{i\}} \left((2\pi \mathfrak{S}_j(t))^{-1/2} \exp\!\left(-\frac{(x_j - \mathfrak{U}_j(t))^2}{2\mathfrak{S}_j(t)} \right) \right) \right] \right. \\
&\quad \cdot \left[\left(\frac{\partial}{\partial t} \left((2\pi \mathfrak{S}_i(t))^{-1/2} \right) \right) \exp\!\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)} \right) \right. \\
&\quad \left. \left. + (2\pi \mathfrak{S}_i(t))^{-1/2} \left(\frac{\partial}{\partial t} \left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)} \right) \right) \exp\!\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)} \right) \right] \right] \\
&= \sum_{i=1}^{d} \left[\left[\prod\nolimits_{j \in \{1,\ldots,d\} \setminus \{i\}} \left((2\pi \mathfrak{S}_j(t))^{-1/2} \exp\!\left(-\frac{(x_j - \mathfrak{U}_j(t))^2}{2\mathfrak{S}_j(t)} \right) \right) \right] \right. \\
&\quad \cdot \left[-(2\pi \mathfrak{S}_i(t))^{-1/2} \left[\frac{\left(\frac{\partial}{\partial t} \mathfrak{S}_i\right)(t)}{2\mathfrak{S}_i(t)} \right] \exp\!\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)} \right) \right. \\
&\quad + (2\pi \mathfrak{S}_i(t))^{-1/2} \left(\frac{2\left(\frac{\partial}{\partial t} \mathfrak{U}_i\right)(t)(x_i - \mathfrak{U}_i(t))}{2\mathfrak{S}_i(t)} \right. \\
&\quad \left. \left. \left. + \frac{(x_i - \mathfrak{U}_i(t))^2 \left(\frac{\partial}{\partial t} \mathfrak{S}_i\right)(t)}{2|\mathfrak{S}_i(t)|^2} \right) \exp\!\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)} \right) \right] \right] \\
&= u(t,x) \left[\sum_{i=1}^{d} \left(\frac{-\left(\frac{\partial}{\partial t} \mathfrak{S}_i\right)(t)}{2\mathfrak{S}_i(t)} + \frac{2\mathfrak{S}_i(t)\left(\frac{\partial}{\partial t} \mathfrak{U}_i\right)(t)(x_i - \mathfrak{U}_i(t)) + (x_i - \mathfrak{U}_i(t))^2 \left(\frac{\partial}{\partial t} \mathfrak{S}_i\right)(t)}{2|\mathfrak{S}_i(t)|^2} \right) \right].
\end{aligned}
\tag{3.59}
$$

Moreover, observe that (3.53), the chain rule, and the product rule ensure that for every $i \in \{1, \ldots, d\}$, $t \in [0,\infty)$ it holds that

$$
\begin{aligned}
\left(\tfrac{\partial}{\partial t} \mathfrak{U}_i\right)(t) &= \frac{\partial}{\partial t} \left(\frac{\mathfrak{m}_i \mathfrak{u}_i}{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)} \right) \\
&= -|\mathfrak{m}_i|^2 \mathfrak{u}_i \left[\frac{\mathfrak{m}_i \sinh(\mathfrak{m}_i t) + \mathfrak{s}_i \cosh(\mathfrak{m}_i t)}{\left[\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)\right]^2} \right] \\
&= -\mathfrak{S}_i(t) \mathfrak{U}_i(t)
\end{aligned}
\tag{3.60}
$$

and

$$
\begin{aligned}
&\left(\tfrac{\partial}{\partial t} \mathfrak{S}_i\right)(t) \\
&= \frac{\partial}{\partial t} \left(\mathfrak{m}_i \left[\frac{\mathfrak{m}_i \sinh(\mathfrak{m}_i t) + \mathfrak{s}_i \cosh(\mathfrak{m}_i t)}{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)} \right] \right) \\
&= |\mathfrak{m}_i|^2 \left[\frac{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)}{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)} \right] - |\mathfrak{m}_i|^2 \left[\frac{\mathfrak{m}_i \sinh(\mathfrak{m}_i t) + \mathfrak{s}_i \cosh(\mathfrak{m}_i t)}{\mathfrak{m}_i \cosh(\mathfrak{m}_i t) + \mathfrak{s}_i \sinh(\mathfrak{m}_i t)} \right]^2 \\
&= |\mathfrak{m}_i|^2 - |\mathfrak{S}_i(t)|^2.
\end{aligned}
\tag{3.61}
$$

Combining this with (3.59) implies that for every $i \in \{1, 2, \ldots, d\}$, $t \in [0, \infty)$ it holds that

$$\begin{aligned}
\big(\tfrac{\partial}{\partial t} u\big)(t,x) &= \frac{u(t,x)}{2} \sum_{i=1}^{d} \Bigg[\frac{-\big[|\mathfrak{m}_i|^2 - |\mathfrak{S}_i(t)|^2\big]}{\mathfrak{S}_i(t)} \\
&\quad + \frac{2|\mathfrak{S}_i(t)|^2 \mathfrak{U}_i(t)(\mathfrak{U}_i(t) - x_i) + (x_i - \mathfrak{U}_i(t))^2 (|\mathfrak{m}_i|^2 - |\mathfrak{S}_i(t)|^2)}{|\mathfrak{S}_i(t)|^2} \Bigg] \\
&= \frac{u(t,x)}{2} \sum_{i=1}^{d} \Bigg[|\mathfrak{m}_i|^2 \left(\left(\frac{x_i - \mathfrak{U}_i(t)}{\mathfrak{S}_i(t)} \right)^2 - \frac{1}{\mathfrak{S}_i(t)} \right) \\
&\quad + \mathfrak{S}_i(t) + 2\big(|\mathfrak{U}_i(t)|^2 - \mathfrak{U}_i(t)\, x_i\big) - \big(|x_i|^2 - 2\mathfrak{U}_i(t)\, x_i + |\mathfrak{U}_i(t)|^2\big) \Bigg] \\
&= \frac{u(t,x)}{2} \sum_{i=1}^{d} \left[|\mathfrak{m}_i|^2 \left(\left(\frac{x_i - \mathfrak{U}_i(t)}{\mathfrak{S}_i(t)} \right)^2 - \frac{1}{\mathfrak{S}_i(t)} \right) + \mathfrak{S}_i(t) + |\mathfrak{U}_i(t)|^2 - |x_i|^2 \right].
\end{aligned} \tag{3.62}$$

Furthermore, note that (3.57) and the product rule show that for every $i \in \{1, 2, \ldots, d\}$, $t \in [0, \infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
\big(\tfrac{\partial}{\partial x_i} u\big)(t,x) &= \frac{\partial}{\partial x_i} \left[\prod_{j=1}^{d} \left[(2\pi \mathfrak{S}_j(t))^{-1/2} \exp\!\left(-\frac{(x_j - \mathfrak{U}_j(t))^2}{2\mathfrak{S}_j(t)} \right) \right] \right] \\
&= \left[\frac{\partial}{\partial x_i} \left[(2\pi \mathfrak{S}_i(t))^{-1/2} \exp\!\left(-\frac{(x_i - \mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)} \right) \right] \right] \\
&\quad \cdot \prod_{j \in \{1,2,\ldots,d\} \setminus \{i\}} \left[(2\pi \mathfrak{S}_j(t))^{-1/2} \exp\!\left(-\frac{(x_j - \mathfrak{U}_j(t))^2}{2\mathfrak{S}_j(t)} \right) \right] \\
&= -u(t,x) \left(\frac{x_i - \mathfrak{U}_i(t)}{\mathfrak{S}_i(t)} \right) = u(t,x) \left(\frac{\mathfrak{U}_i(t) - x_i}{\mathfrak{S}_i(t)} \right).
\end{aligned} \tag{3.63}$$

The product rule therefore assures that for every $i \in \{1, 2, \ldots, d\}$, $t \in [0, \infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
\big(\tfrac{\partial^2}{\partial x_i^2} u\big)(t,x) &= \frac{\partial}{\partial x_i} \left(u(t,x) \left(\frac{\mathfrak{U}_i(t) - x_i}{\mathfrak{S}_i(t)} \right) \right) \\
&= \big(\tfrac{\partial}{\partial x_i} u\big)(t,x) \left(\frac{\mathfrak{U}_i(t) - x_i}{\mathfrak{S}_i(t)} \right) - \frac{u(t,x)}{\mathfrak{S}_i(t)} = u(t,x) \left[\left(\frac{x_i - \mathfrak{U}_i(t)}{\mathfrak{S}_i(t)} \right)^2 - \frac{1}{\mathfrak{S}_i(t)} \right].
\end{aligned} \tag{3.64}$$

Hence, we obtain that for every $t \in [0, \infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\sum_{i=1}^{d} \left[\tfrac{1}{2} |\mathfrak{m}_i|^2 \big(\tfrac{\partial^2}{\partial x_i^2} u\big)(t,x) \right] = \frac{u(t,x)}{2} \sum_{i=1}^{d} \left[|\mathfrak{m}_i|^2 \left(\left(\frac{x_i - \mathfrak{U}_i(t)}{\mathfrak{S}_i(t)} \right)^2 - \frac{1}{\mathfrak{S}_i(t)} \right) \right]. \tag{3.65}$$

Next observe that (3.57) and Fubini's theorem ensure that for every $t \in [0,\infty)$, $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
&u(t,x)\Big(a(x) - \int_{\mathbb{R}^d} u(t,\mathbf{x})\,a(\mathbf{x})\,\mathrm{d}\mathbf{x}\Big)\\
&= u(t,x)\left(-\frac{1}{2}\left[\sum_{i=1}^{d}|x_i|^2\right] - \int_{\mathbb{R}^d} -\frac{1}{2}\left[\sum_{i=1}^{d}|\mathbf{x}_i|^2\right] u(t,\mathbf{x})\,\mathrm{d}\mathbf{x}\right)\\
&= \frac{u(t,x)}{2}\left(-\left[\sum_{i=1}^{d}|x_i|^2\right]\right.\\
&\quad + \sum_{i=1}^{d}\left[\int_{\mathbb{R}}|\mathbf{x}_i|^2\,(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{(\mathbf{x}_i-\mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\mathrm{d}\mathbf{x}_i\right.\\
&\quad \left.\left.\cdot\left(\prod\nolimits_{j\in\{1,2,\ldots,d\}\setminus\{i\}}\int_{\mathbb{R}}(2\pi\mathfrak{S}_j(t))^{-1/2}\exp\left(-\frac{(\mathbf{x}_j-\mathfrak{U}_j(t))^2}{2\mathfrak{S}_j(t)}\right)\mathrm{d}\mathbf{x}_j\right)\right]\right).
\end{aligned} \tag{3.66}$$

This and the fact that for every $i \in \{1,2,\ldots,d\}$, $t \in [0,\infty)$ it holds that

$$\int_{\mathbb{R}}(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{(x-\mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\mathrm{d}x = 1 \tag{3.67}$$

imply that for every $t \in [0,\infty)$, $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned}
&u(t,x)\Big(a(x) - \int_{\mathbb{R}^d} u(t,\mathbf{x})\,a(\mathbf{x})\,\mathrm{d}\mathbf{x}\Big)\\
&= \frac{u(t,x)}{2}\sum_{i=1}^{d}\left[-|x_i|^2 + \int_{\mathbb{R}}|\mathbf{x}_i|^2\,(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{(\mathbf{x}_i-\mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\mathrm{d}\mathbf{x}_i\right].
\end{aligned} \tag{3.68}$$

Next observe that the integral transformation theorem demonstrates that for every $i \in \{1,2,\ldots,d\}$, $t \in [0,\infty)$ it holds that

$$\begin{aligned}
&\int_{\mathbb{R}} x^2\left[(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{(x-\mathfrak{U}_i(t))^2}{2\mathfrak{S}_i(t)}\right)\right]\mathrm{d}x\\
&= \int_{\mathbb{R}}(x+\mathfrak{U}_i(t))^2\left[(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{x^2}{2\mathfrak{S}_i(t)}\right)\right]\mathrm{d}x\\
&= \int_{\mathbb{R}} x^2\left[(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{x^2}{2\mathfrak{S}_i(t)}\right)\right]\mathrm{d}x\\
&\quad + \int_{\mathbb{R}}|\mathfrak{U}_i(t)|^2\left[(2\pi\mathfrak{S}_i(t))^{-1/2}\exp\left(-\frac{x^2}{2\mathfrak{S}_i(t)}\right)\right]\mathrm{d}x\\
&= \mathfrak{S}_i(t) + |\mathfrak{U}_i(t)|^2.
\end{aligned} \tag{3.69}$$

Combining this with (3.68) ensures that for every $t \in [0,\infty)$, $x = (x_1,\ldots,x_d) \in \mathbb{R}^d$ it holds that

$$u(t,x)\Big(a(x) - \int_{\mathbb{R}^d} u(t,\mathbf{x})\,a(\mathbf{x})\,\mathrm{d}\mathbf{x}\Big) = \frac{u(t,x)}{2}\sum_{i=1}^{d}\big(\mathfrak{S}_i(t) + |\mathfrak{U}_i(t)|^2 - |x_i|^2\big). \tag{3.70}$$

This and (3.65) demonstrate that for every $t \in [0,\infty)$, $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ it holds that

$$\begin{aligned} &u(t,x)\Big(a(x) - \int_D u(t,\mathbf{x})\, a(\mathbf{x})\, \mathrm{d}\mathbf{x}\Big) + \sum_{i=1}^{d} \tfrac{1}{2}|\mathfrak{m}_i|^2 \big(\tfrac{\partial^2}{\partial x_i^2} u\big)(t,x) \\ &= \frac{u(t,x)}{2} \sum_{i=1}^{d} \left[|\mathfrak{m}_i|^2 \left(\left(\frac{x_i - \mathfrak{U}_i(t)}{\mathfrak{S}_i(t)} \right)^2 - \frac{1}{\mathfrak{S}_i(t)} \right) + \mathfrak{S}_i(t) + |\mathfrak{U}_i(t)|^2 - |x_i|^2 \right]. \end{aligned} \tag{3.71}$$

Combining this with (3.62) proves Item (iii). The proof of Lemma 3.5.2 is thus complete. □

3.5.5 Allen–Cahn PDEs with conservation of mass

In this subsection we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the solutions of certain Allen–Cahn PDEs with cubic nonlinearity, conservation of mass and no-flux boundary conditions (cf., e.g., Rubinstein & Sternberg [86]).

Assume Lemma 3.5.1, let $\epsilon \in (0,\infty)$ satisfy $\epsilon = \frac{1}{10}$, assume that $d \in \{1,2,5,10\}$, $\mathbb{D} = [-1/2, 1/2]^d$, $T \in \{1/5, 1/2, 1\}$, $N = 10$, $K_1 = K_2 = \ldots = K_N = 5$, and $M_1 = M_2 = \ldots = M_N = 500$, assume that $\xi^{n,m,j}, n, m, j \in \mathbb{N}$, are independent $\mathcal{U}_D$-distributed random variables, assume for every $m \in \mathbb{N}$ that $\gamma_m = 10^{-2}$, and assume for every $s,t \in [0,T]$, $x, \mathbf{x} \in \mathbb{D}$, $y, \mathbf{y} \in \mathbb{R}$, $v \in \mathbb{R}^d$, $A \in \mathcal{B}(D)$ that $\nu_x(A) = \int_A \mathrm{d}\mathbf{x}$, $g(x) = \exp(-\frac{1}{4}\|x\|^2)$, $\mu(x) = (0,\ldots,0)$, $\sigma(x)v = \epsilon v$, $f(t,x,\mathbf{x},y,\mathbf{y}) = y - y^3 - (\mathbf{y} - \mathbf{y}^3)$, and

$$H(t,s,x,v) = R(x, x + \mu(x)(t-s) + \sigma(x)v) = R(x, x + \epsilon v) \tag{3.72}$$

(cf. (3.6) and (3.17)). The solution $u\colon [0,T] \times \mathbb{D} \to \mathbb{R}$ of the PDE in (3.41) then satisfies that for every $t \in (0,T]$, $x \in \partial_{\mathbb{D}}$ it holds that $\langle \mathbf{n}(x), (\nabla_x u)(t,x) \rangle = 0$ and that for every $t \in [0,T]$, $x \in \mathbb{D}$ it holds that $u(0,x) = \exp(-\frac{1}{4}\|x\|^2)$ and

$$\big(\tfrac{\partial}{\partial t} u\big)(t,x) = \tfrac{\epsilon^2}{2}(\Delta_x u)(t,x) + u(t,x) - [u(t,x)]^3 - \int_{[-1/2,1/2]^d} u(t,\mathbf{x}) - [u(t,\mathbf{x})]^3 \, \mathrm{d}\mathbf{x}. \tag{3.73}$$

In Table 3.5 we use the machine learning-based approximation method in Lemma 3.5.1 to approximately calculate the mean of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0,\ldots,0))$, the standard deviation of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0,\ldots,0))$, the relative L^1-approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0,\ldots,0))$, the uncorrected sample standard deviation of the approximation error associated to $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0,\ldots,0))$, and the average runtime in seconds needed for calculating one realization of $\mathbb{V}_N^{1,0}(\Theta_{M_N}^N, (0,\ldots,0))$ based on 5 independent realizations (5 independent runs). The reference value, which is used as an approximation for the unknown value $u(T,(0,\ldots,0))$ of the exact solution of (3.73), has been calculated via the MLP approximation method for non-local nonlinear PDEs in Lemma 3.4.1 (cf. Lemma 3.4.6 and Beck et al. [9, Remark 3.3]).

d	T	N	Mean of the approx. method	Standard deviation of the approx. method	Reference value	Relative L^1-approx. error	Standard deviation of the error	Average runtime in seconds
1	$1/5$	10	0.9947184	0.0021832	0.9932255	0.0015709	0.0021380	31.417
2	$1/5$	10	0.9908873	0.0027061	0.9868883	0.0040521	0.0027421	35.069
5	$1/5$	10	0.9942151	0.0052064	0.9710707	0.0238340	0.0053615	38.363
10	$1/5$	10	0.9792556	0.0203935	0.9514115	0.0292661	0.0214350	42.782
1	$1/2$	10	0.9870476	0.0014673	0.9880013	0.0014996	0.0007477	30.297
2	$1/2$	10	0.9763564	0.0030895	0.9750274	0.0024841	0.0021561	34.922
5	$1/2$	10	0.9518845	0.0051304	0.9431354	0.0092766	0.0054398	37.963
10	$1/2$	10	0.9249420	0.0052786	0.9063239	0.0205424	0.0058242	43.139
1	1	10	0.9823494	0.0003647	0.9780817	0.0043633	0.0003729	29.250
2	1	10	0.9659823	0.0004128	0.9658025	0.0003195	0.0003137	34.485
5	1	10	0.9209547	0.0019223	0.9158821	0.0055385	0.0020988	39.318
10	1	10	0.8693402	0.0029947	0.8683143	0.0030165	0.0015052	44.258

Tab. 3.5: Numerical simulations for the approximation method in Lemma 3.3.1 in the case of the Allen–Cahn PDEs with conservation of mass in (3.73) in Section 3.5.5.

References

[1] F. Abergel and R. Tachet. “A nonlinear partial integro-differential equation from mathematical finance”. In: *Discrete Contin. Dyn. Syst.* 27.3 (2010), pp. 907–917. DOI: 10.3934/dcds.2010.27.907.

[2] A. Al-Aradi et al. “Extensions of the Deep Galerkin Method”. In: *arXiv:1912.01455* (2019), 27 pp.

[3] M. Alfaro and R. Carles. “Replicator-mutator equations with quadratic fitness”. In: *Proc. Amer. Math. Soc.* 145.12 (2017), pp. 5315–5327. DOI: 10.1090/proc/13669.

[4] M. Alfaro and M. Veruete. “Evolutionary branching via replicator-mutator equations”. In: *J. Dynam. Differential Equations* 31.4 (2019), pp. 2029–2052. DOI: 10.1007/s10884-018-9692-9.

[5] A. L. Amadori. “Nonlinear integro-differential evolution problems arising in option pricing: a viscosity solutions approach”. In: *Differential Integral Equations* 16.7 (2003), pp. 787–811.

[6] M. Banerjee, S. V. Petrovskii, and V. Volpert. “Nonlocal reaction-diffusion models of heterogeneous wealth distribution”. In: *Mathematics* 9.4 (2021), Article No. 351, 18 pp. DOI: 10.3390/math9040351.

[7] A. Barone et al. “Theory and applications of the sine-Gordon equation”. In: *La Rivista del Nuovo Cimento* 1.2 (1971), pp. 227–267. DOI: 10.1007/BF02820622.

[8] W. F. Bauer. “The Monte Carlo Method”. In: *J. Soc. Ind. Appl. Math.* 6.4 (1958), pp. 438–451.

[9] C. Beck, W. E, and A. Jentzen. “Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations”. In: *J. Nonlinear Sci.* 29.4 (2019), pp. 1563–1619. DOI: 10.1007/s00332-018-9525-3.

[10] C. Beck et al. “An overview on deep learning-based approximation methods for partial differential equations”. In: *Revision requested from Discrete Contin. Dyn. Syst., arXiv:2012.12348* (2020), 22 pp.

[11] C. Beck et al. “Deep splitting method for parabolic PDEs”. In: *SIAM J. Sci. Comput.* 43.5 (2021), A3135–A3154. DOI: 10.1137/19M1297919.

[12] C. Beck et al. “Solving the Kolmogorov PDE by means of deep learning”. In: *J. Sci. Comput.* 88 (2021), Article No. 73, 28 pp.

[13] S. Becker et al. “Numerical Simulations for Full History Recursive Multilevel Picard Approximations for Systems of High-Dimensional Partial Differential Equations”. In: *Commun. Comput. Phys.* 28.5 (2020), pp. 2109–2138. DOI: https://doi.org/10.4208/cicp.OA-2020-0130.

[14] R. Bellman. *Dynamic Programming*. Princeton Landmarks in Mathematics. Reprint of the 1957 edition. Princeton University Press, Princeton, NJ, 2010, pp. xxx+340.

[15] F. E. Benth, K. H. Karlsen, and K. Reikvam. “Optimal portfolio selection with consumption and nonlinear integro-differential equations with gradient constraint: a viscosity solution approach”. In: *Finance Stoch.* 5.3 (2001), pp. 275–303. DOI: 10.1007/PL00013538.

[16] H. Berestycki, T. Jin, and L. Silvestre. “Propagation in a non local reaction diffusion equation with spatial and genetic trait structure”. In: *Nonlinearity* 29.4 (2016), pp. 1434–1466. DOI: 10.1088/0951-7715/29/4/1434.

[17] H. Berestycki et al. “The non-local Fisher-KPP equation: travelling waves and steady states”. In: *Nonlinearity* 22.12 (2009), pp. 2813–2844. DOI: 10.1088/0951-7715/22/12/002.

[18] S. Bian, L. Chen, and E. A. Latos. “Global existence and asymptotic behavior of solutions to a nonlocal Fisher-KPP type problem”. In: *Nonlinear Anal.* 149 (2017), pp. 165–176. DOI: 10.1016/j.na.2016.10.017.

[19] R. Burger and J. Hofbauer. “Mutation load and mutation-selection-balance in quantitative genetic traits”. In: *J. Math. Biol.* 32.3 (1994), pp. 193–218. DOI: 10.1007/BF00163878.

[20] E. Caglioti et al. “A special class of stationary flows for two-dimensional Euler equations: a statistical mechanics description. II”. In: *Comm. Math. Phys.* 174.2 (1995), pp. 229–260.

[21] J. Castro. “Deep Learning Schemes For Parabolic Nonlocal Integro-Differential Equations”. In: *arXiv:2103.15008* (2021), 28 pp.

[22] T. Chan. “Pricing contingent claims on stocks driven by Lévy processes”. In: *Ann. Appl. Probab.* 9.2 (1999), pp. 504–528. DOI: 10.1214/aoap/1029962753.

[23] J. Chen, R. Du, and K. Wu. “A Comparison Study of Deep Galerkin Method and Deep Ritz Method for Elliptic Problems with Different Boundary Conditions”. In: *Comm. Math. Res.* 36.3 (2020), pp. 354–376. DOI: https://doi.org/10.4208/cmr.2020-0051.

[24] L. Chen et al. “Mathematical models for cell migration: a non-local perspective”. In: *Philos. Trans. Roy. Soc. B* 375.1807 (2020), Article No. 20190379, 9 pp. DOI: 10.1098/rstb.2019.0379. arXiv: 1911.05200.

[25] S. Coleman. “Quantum sine-Gordon equation as the massive Thirring model”. In: *Bosonization*. Vol. 1. World Scientific, 1994, pp. 128–137. DOI: 10.1142/9789812812650_0013.

[26] R. Cont and P. Tankov. *Financial modelling with jump processes*. Chapman & Hall/CRC Financial Mathematics Series. Chapman & Hall/CRC, Boca Raton, FL, 2004, pp. xvi+535.

[27] S. Cox and J. van Neerven. “Pathwise Hölder convergence of the implicit-linear Euler scheme for semi-linear SPDEs with multiplicative noise”. In: *Numer. Math.* 125.2 (2013), pp. 259–345. DOI: 10.1007/s00211-013-0538-4.

[28] J. M. T. S. Cruz and D. Ševčovič. “On solutions of a partial integro-differential equation in Bessel potential spaces with applications in option pricing models”. In: *Jpn. J. Ind. Appl. Math.* 37.3 (2020), pp. 697–721. DOI: 10.1007/s13160-020-00414-2.

[29] M. D’Elia et al. “Numerical methods for nonlocal and fractional models”. In: *Acta Numerica* 29 (2020), pp. 1–124. DOI: 10.1017/S096249292000001X.

[30] M. Doebeli and I. Ispolatov. “Complexity and diversity”. In: *Science* 328.5977 (2010), pp. 494–497. DOI: 10.1126/science.1187468.

[31] J. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *J. Mach. Learn. Res.* 12.61 (2011), pp. 2121–2159.

[32] W. E, J. Han, and A. Jentzen. “Algorithms for Solving High Dimensional PDEs: From Nonlinear Monte Carlo to Machine Learning”. In: *Nonlinearity* 35.1 (2021), pp. 278–310. DOI: 10.1088/1361-6544/ac337f.

[33] W. E, J. Han, and A. Jentzen. “Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations”. In: *Commun. Math. Stat.* 5.4 (2017), pp. 349–380. DOI: 10.1007/s40304-017-0117-6.

[34] W. E and B. Yu. “The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems”. In: *Commun. Math. Stat.* 6.1 (2018), pp. 1–12.

[35] W. E et al. “Multilevel Picard iterations for solving smooth semilinear parabolic heat equations”. In: *Partial Differ. Equ. Appl.* 2.6 (2021), Article No. 80, 31 pp. DOI: 10.1007/s42985-021-00089-5.

[36] W. E et al. “On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations”. In: *J. Sci. Comput.* 79.3 (2019), pp. 1534–1571. DOI: 10.1007/s10915-018-00903-0.

[37] W. E et al. “On multilevel Picard numerical approximations for high-dimensional nonlinear parabolic partial differential equations and high-dimensional nonlinear backward stochastic differential equations”. In: *J. Sci. Comput.* 79.3 (2019), pp. 1534–1571.

[38] R. A. FISHER. “THE WAVE OF ADVANCE OF ADVANTAGEOUS GENES”. In: *Annals of Eugenics* 7.4 (1937), pp. 355–369. DOI: 10.1111/j.1469-1809.1937.tb02153.x.

[39] R. Frey and V. Köck. “Deep Neural Network Algorithms for Parabolic PIDEs and Applications in Insurance Mathematics”. In: *arXiv:2109.11403* (2021), 24 pp.

[40] R. Frey and V. Köck. “Deep Neural Network Algorithms for Parabolic PIDEs and Applications in Insurance Mathematics”. In: *Mathematical and Statistical Methods for Actuarial Sciences and Finance*. Ed. by M. Corazza et al. Cham: Springer International Publishing, 2022, pp. 272–277.

[41] H. Gajewski and K. Zacharias. “On a nonlocal phase separation model”. In: *J. Math. Anal. Appl.* 286.1 (2003), pp. 11–31. DOI: 10.1016/S0022-247X(02)00425-0.

[42] X. Gan, Y. Yang, and K. Zhang. “A robust numerical method for pricing American options under Kou’s jump-diffusion models based on penalty method”. In: *J. Appl. Math. Comput.* 62.1–2 (2020), pp. 1–21. DOI: 10.1007/s12190-019-01270-1.

[43] S. Génieys, V. Volpert, and P. Auger. “Pattern and waves for a model in population dynamics with nonlocal consumption of resources”. In: *Math. Model. Nat. Phenom.* 1.1 (2006), pp. 65–82. DOI: 10.1051/mmnp:2006004.

[44] X. Glorot and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y. W. Teh and M. Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010, pp. 249–256.

[45] L. Gonon and C. Schwab. “Deep ReLU neural networks overcome the curse of dimensionality for partial integrodifferential equations”. In: *arXiv:2102.11707* (2021), 35 pp. DOI: 10.48550/ARXIV.2102.11707. arXiv: 2102.11707.

[46] P. Grohs and F. Voigtlaender. “Proof of the Theory-to-Practice Gap in Deep Learning via Sampling Complexity bounds for Neural Network Approximation Spaces”. In: *arXiv:2104.02746* (2021), 42 pp. DOI: 10.48550/ARXIV.2104.02746. arXiv: 2104.02746.

[47] L. Guo et al. “Monte Carlo PINNs: deep learning approach for forward and inverse problems involving high dimensional fractional partial differential equations”. In: *arXiv:2203.08501* (2022), 18 pp.

[48] I. Gyöngy and N. Krylov. “On the splitting-up method and stochastic partial differential equations”. In: *Ann. Probab.* 31.2 (2003), pp. 564–591. DOI: 10.1214/aop/1048516528.

[49] M. Hairer and H. Shen. “The dynamical sine-Gordon model”. In: *Comm. Math. Phys.* 341.3 (2016), pp. 933–989. DOI: 10.1007/s00220-015-2525-3.

[50] F. Hamel et al. “Dynamics of adaptation in an anisotropic phenotype-fitness landscape”. In: *Nonlinear Anal. Real World Appl.* 54 (2020), Article No. 103107, 33 pp. DOI: 10.1016/j.nonrwa.2020.103107.

[51] F. Hamel and N. Nadirashvili. “Travelling fronts and entire solutions of the Fisher-KPP equation in $\mathbb{R}^N$”. In: *Arch. Ration. Mech. Anal.* 157.2 (2001), pp. 91–163. DOI: 10.1007/PL00004238.

[52] J. Han, A. Jentzen, and W. E. “Solving high-dimensional partial differential equations using deep learning”. In: *Proc. Natl. Acad. Sci. USA* 115.34 (2018), pp. 8505–8510. DOI: 10.1073/pnas.1718942115.

[53] S. Heinrich. “Monte Carlo complexity of global solution of integral equations”. In: *J. Complex.* 14.2 (1998), pp. 151–175.

[54] S. Heinrich and E. Sindambiwe. “Monte Carlo complexity of parametric integration”. In: *J. Complex.* 15.3 (1999), pp. 317–341.

[55] P. Henry-Labordère. “Counterparty Risk Valuation: A Marked Branching Diffusion Approach”. In: *arXiv:1203.2369* (2012), 17 pp. arXiv: 1203.2369.

[56] M. Hochbruck and A. Ostermann. “Explicit exponential Runge–Kutta methods for semilinear parabolic problems”. In: *SIAM J. Numer. Anal.* 43.3 (2005), pp. 1069–1090. DOI: 10.1137/040611434.

[57] B. Houchmandzadeh and M. Vallade. “Fisher waves: An individual-based stochastic model”. In: *Physical Review E* 96.1 (2017), pp. 1–13. DOI: 10.1103/PhysRevE.96.012414. arXiv: 1703.02835.

[58] J. Huang, Z. Cen, and A. Le. “A finite difference scheme for pricing American put options under Kou’s jump-diffusion model”. In: *J. Funct. Spaces Appl.* (2013), Article No. 651573, 11 pp. DOI: 10.1155/2013/651573.

[59] C. Huré, H. Pham, and X. Warin. “Deep backward schemes for high-dimensional nonlinear PDEs”. In: *Math. Comp.* 89 (2020), pp. 1547–1579.

[60] M. Hutzenthaler et al. “Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations”. In: *Proc. A.* 476.2244 (2020), Article No. 20190630, 25 pp. DOI: 10.1098/rspa.2019.0630.

[61] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, 2015, pp. 448–456.

[62] N. I. Kavallaris, J. Lankeit, and M. Winkler. “On a degenerate nonlocal parabolic problem describing infinite dimensional replicator dynamics”. In: *SIAM J. Math. Anal.* 49.2 (2017), pp. 954–983. DOI: 10.1137/15M1053840.

[63] N. I. Kavallaris and T. Suzuki. *Non-local partial differential equations for engineering and biology*. Vol. 31. Mathematics for Industry (Tokyo). Springer, Cham, 2018, pp. xix+300. DOI: 10.1007/978-3-319-67944-0.

[64] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv:1412.6980* (2014), 15 pp. arXiv: 1412.6980.

[65] S. G. Kou. “A Jump-Diffusion Model for Option Pricing”. In: *Manag. Sci.* 48.8 (2002), pp. 1086–1101. DOI: 10.1287/mnsc.48.8.1086.166. eprint: https://doi.org/10.1287/mnsc.48.8.1086.166.

[66] A. A. Lacey. “Thermal runaway in a non-local problem modelling Ohmic heating. I. Model derivation and some special cases”. In: *European J. Appl. Math.* 6.2 (1995), pp. 127–144. DOI: 10.1017/S095679250000173X.

[67] I. Lagaris, A. Likas, and D. Papageorgiou. “Neural-network methods for boundary value problems with irregular boundaries”. In: *IEEE Trans. Neural Netw.* 11.5 (2000), pp. 1041–1049. DOI: 10.1109/72.870037.

[68] I. E. Lagaris, A. Likas, and D. I. Fotiadis. “Artificial neural networks for solving ordinary and partial differential equations”. In: *IEEE Trans. Neural Netw.* 9.5 (1998), pp. 987–1000.

[69] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[70] Y. Liao and P. Ming. “Deep Nitsche Method: Deep Ritz Method with Essential Boundary Conditions”. In: *Comm. Comput. Phys.* 29.5 (2021), pp. 1365–1384. DOI: https://doi.org/10.4208/cicp.OA-2020-0219.

[71] A. Lorz et al. “Populational adaptive evolution, chemotherapeutic resistance and multiple anti-cancer therapies”. In: *ESAIM Math. Model. Numer. Anal.* 47.2 (2013), pp. 377–399. DOI: 10.1051/m2an/2012031.

[72] L. Lu et al. “DeepXDE: A Deep Learning Library for Solving Differential Equations”. In: *SIAM Review* 63.1 (2021), pp. 208–228. DOI: 10.1137/19M1274067. eprint: https://doi.org/10.1137/19M1274067.

[73] K. S. McFall and J. R. Mahan. “Artificial Neural Network Method for Solution of Boundary Value Problems With Exact Satisfaction of Arbitrary Boundary Conditions”. In: *IEEE Trans. Neural Netw.* 20.8 (2009), pp. 1221–1233. DOI: 10.1109/TNN.2009.2020735.

[74] R. C. Merton. “Option pricing when underlying stock returns are discontinuous”. In: *J. Financ. Econ.* 3.1–2 (1976), pp. 125–144. DOI: 10.1016/0304-405X(76)90022-2.

[75] N. Metropolis and S. Ulam. “The Monte Carlo Method”. In: *J. Amer. Statist. Assoc.* 44.247 (1949), pp. 335–341. DOI: 10.1080/01621459.1949.10483310. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1949.10483310.

[76] J. M. Nordbotten and N. C. Stenseth. “Asymmetric ecological conditions favor Red-Queen type of continued evolution over stasis”. In: *Proc. Natl. Acad. Sci. USA* 113.7 (2016), pp. 1847–1852. DOI: 10.1073/pnas.1525395113.

[77] J. M. Nordbotten et al. “Ecological and evolutionary dynamics of interconnectedness and modularity”. In: *Proc. Natl. Acad. Sci. USA* 115.4 (2018), pp. 750–755. DOI: 10.1073/pnas.1716078115.

[78] J. M. Nordbotten et al. “The dynamics of trait variance in multi-species communities”. In: *R. Soc. Open Sci.* 7.8 (2020), Article No. 200321, 20 pp. DOI: 10.1098/rsos.200321.

[79] J. Oechssler and F. Riedel. “Evolutionary dynamics on infinite strategy spaces”. In: *Econom. Theory* 17.1 (2001), pp. 141–162. DOI: 10.1007/PL00004092.

[80] M. Pájaro et al. “Stochastic modeling and numerical simulation of gene regulatory networks with protein bursting”. In: *J. Theoret. Biol.* 421 (2017), pp. 51–70. DOI: 10.1016/j.jtbi.2017.03.017.

[81] G. Pang, L. Lu, and G. E. Karniadakis. “fPINNs: Fractional Physics-Informed Neural Networks”. In: *SIAM Journal on Scientific Computing* 41.4 (2019), A2603–A2626. DOI: 10.1137/18M1229845. eprint: https://doi.org/10.1137/18M1229845.

[82] B. Perthame and S. Génieys. “Concentration in the nonlocal Fisher equation: the Hamilton-Jacobi limit”. In: *Math. Model. Nat. Phenom.* 2.4 (2007), pp. 135–151. DOI: 10.1051/mmnp:2008029.

[83] H. Pham. *Continuous-time stochastic control and optimization with financial applications*. Vol. 61. Stochastic Modelling and Applied Probability. Springer-Verlag, Berlin, 2009, pp. xviii+232. DOI: 10.1007/978-3-540-89500-8.

[84] M. Raissi, P. Perdikaris, and G. E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *J. Comput. Phys.* 378 (2019), pp. 686–707.

[85] L. Roques and O. Bonnefon. “Modelling Population Dynamics in Realistic Landscapes with Linear Elements: A Mechanistic-Statistical Reaction-Diffusion Approach”. In: *PLoS ONE* 11.3 (2016). Ed. by Z. Jin, Article No. e0151217, 20 pp. DOI: 10.1371/journal.pone.0151217.

[86] J. Rubinstein and P. Sternberg. “Nonlocal reaction-diffusion equations and nucleation”. In: *IMA J. Appl. Math.* 48.3 (1992), pp. 249–264. DOI: 10.1093/imamat/48.3.249.

[87] J. Sirignano and K. Spiliopoulos. “DGM: A deep learning algorithm for solving partial differential equations”. In: *J. Comput. Phys.* 375 (2018), pp. 1339–1364.

[88] I. Stoleriu. “Non-local models for solid-solid phase transitions”. In: *ROMAI J.* 7.1 (2011), pp. 157–170.

[89] N. Sukumar and A. Srivastava. “Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks”. In: *Comput. Methods Appl. Mech. Engrg.* 389 (2022), Article No. 114333. DOI: https://doi.org/10.1016/j.cma.2021.114333.

[90] S. Sunderasan. “Financial Modeling”. In: *Long-Term Investments*. Routledge India, 2020, pp. 33–51. DOI: 10.4324/9780367817909-3.

[91] C. Villa, M. A. J. Chaplain, and T. Lorenzi. “Evolutionary dynamics in vascularised tumours under chemotherapy: mathematical modelling, asymptotic analysis and numerical simulations”. In: *Vietnam J. Math.* 49.1 (2021), pp. 143–167. DOI: 10.1007/s10013-020-00445-9.

[92] F. Wang et al. “Global stabilization and boundary control of generalized Fisher/KPP equation and application to diffusive SIS model”. In: *J. Differential Equations* 275 (2021), pp. 391–417. DOI: 10.1016/j.jde.2020.11.031.

[93] S. Wang and P. Perdikaris. “Deep learning of free boundary and Stefan problems”. In: *J. Comput. Phys.* (2020), Article No. 109914, 27 pp. DOI: https://doi.org/10.1016/j.jcp.2020.109914.

[94] L. Yuan et al. “A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations”. In: *J. Comput. Phys.* (2022). Early access version available online, Article No. 111260. DOI: https://doi.org/10.1016/j.jcp.2022.111260.

[95] Y. Zang et al. “Weak adversarial networks for high-dimensional partial differential equations”. In: *Journal of Computational Physics* 411 (2020), Article No. 109409. DOI: https://doi.org/10.1016/j.jcp.2020.109409.

4

Mini-batching ecological data to improve ecosystem models with machine learning

by Victor Boussange[1,2], Pau Vilimelis Aceituno[3], and Loïc Pellissier[1,2]

[1] Swiss Federal Research Institute WSL, CH-8903 Birmensdorf, Switzerland

[2] Landscape Ecology, Institute of Terrestrial Ecosystems, Department of Environmental System Science, ETH Zürich, CH-8092 Zürich, Switzerland

[3] Institute of Neuroinformatics, ETH Zürich and University of Zürich, Zürich, Switzerland

bioRxiv:2022.07.25.501365
Under review at PLOS Computational Biology

4.1 Introduction

Ecosystems are complex systems involving many interacting functional entities which together play a major role in regulating global biogeochemical cycles [10] and delivering essential services to humans [49]. Ecosystems currently face intense disruption from anthropogenic pressure, through pollution and land use [21, 25], and from climate change [60]. In order to anticipate the responses of ecosystems to these disruptions, models that can extrapolate ecological dynamics beyond observations are required [12]. A major challenge is that the processes driving ecological dynamics are nonlinear, resulting in complex responses and feedbacks [80]. Nonlinearity greatly affects the capacity of modelling approaches that do not incorporate specific biological knowledge to reliably project current trends into the future [4]. For instance, while methods based on statistical descriptions [19] and nonparametric methods [93, 92, 20] have adequate interpolation capabilites, they are ill-suited for extrapolating beyond observed trends [4, 84]. In contrast, mechanistic ecosystem models integrate constraints on the expected dynamics by explicitly modelling interactions, feedback loops and dependencies between ecosystem components [31]. While this should ensure a more robust forecast under large disruptions [61], ecosystems models suffer in practice

from parametrization issues, i.e. inaccuracies in the mathematical formulation of the processes and issues in identifying the correct parameter values [17]. These drawbacks have limited their broad adoption [84]. Learning the parametrization of ecosystem models from observation data by blending specific biological knowledge and ML methods could improve our representation of ecosystem processes and help us to anticipate ecosystem responses to global changes.

The parametrization of ecosystem models can be indirectly learnt from observations by calibrating the parameters from the data collectively using inference methods. These methods proceed by maximizing the posterior probability of the parameters given the observations, but their success is subject to a number of issues, some of which specifically relating to ecosystem model properties. Among these issues, the exploration of the posterior landscape demands repeated model simulations, but ecosystem models are usually associated with a high computational cost that limits the number of possible runs [28]. Additionally, the complexity of processes requires a large number of parameters. Due to the curse of dimensionality [12], this complicates the exploration of the posterior distribution and can further leave many parameters poorly constrained [37]. The limited availability of observations, which are usually composed of multiple partial short-term time series [81, 79], accentuates the lack of parameter constraints. Moreover, ecosystem dynamics can be strongly dependent on the initial conditions (ICs) and show a chaotic behaviour [39, 42, 6], or can be associated with a large panel of dynamics depending on the parameter values. In such case, small perturbations of the ICs or parameter values lead to large divergences in the model outcomes, causing numerical problems in finding the most probable parameters [17]. Last but not least, in contrast to fields such as climate and weather modelling, the derivation of fundamental processes regulating ecosystems is far from being established [32, 79, 69], resulting in inaccurate mechanistic pathways and uncertain mathematical formulations [33], limiting extrapolation to unseen data. To summarize, the parametrization of ecosystem models requires inference methods that are robust despite the models' complexity, the limited observation data, and the inaccurate description of ecological processes.

A variety of data assimilation and ML methods are increasingly being used to parametrize ecosystem models. Bayesian inference with Markov Chain Monte Carlo methods, used in [52, 40, 90, 27, 76], offer the advantage of quantifying uncertainties by inferring the full posterior probability distribution of the unknown parameters. This is achieved by a global exploration of the parameter space, which makes Bayesian methods computationally expensive and particularly prone to the curse of dimensionality [36]. Simulated annealing [56], genetic algorithms [86], and sequential methods such as extended Kalman filtering and ensemble Kalman methods [7, 24, 34] have been used as alternatives, but are similarly subject to the curse of dimensionality and demand a large number of model evaluations. Variational methods rely on the model adjoint, i.e. the model sensitivity to the parameters, to explore more efficiently the parameter space, iteratively updating the parameter estimates using the gradient of the posterior landscape. Such methods therefore demand less evaluation [79], which explains their wide adoption in the field of artificial intelligence to train highly parametrized neural networks (up to the order of 10^8 parameters [85]) and their use in calibrating marine ecosystem models [26, 83, 89, 63] (see [79] for a review) and terrestrial

ecosystem models [94, 18, 16]. However, as the complex dynamics of ecosystem models tend to be associated with rugged posterior landscapes, variational methods are prone to converging to local minima, making variational methods very sensitive to the choice of initial model parameters [29, 79]. Ecosystem models are specified as differential equations that depend not only on parameters but also on ICs. The state-dependency of ecosystems means that neglecting the estimation of initial ICs might compromise the correct fitting of the parameters and the forecast skill [52]. However, few of the aforementioned studies have addressed the problem of IC estimation (but see [63]). Finally, the numerical implementation of variational methods is also challenging, as the model adjoint is difficult to obtain and maintain as the model is modified [51, 63, 34]. Novel methods for model parametrization are emerging, thanks to advances in the field of artificial intelligence [88, 45, 2, 64], providing new opportunities to better address these issues.

Here, we propose a ML framework relying on a mini-batch method inspired by multiple shooting methods [67] and on automatic differentiation and state-of-the-art optimizers to efficiently learn the parametrization of ecosystem models from observation data. The mini-batch method divides the training problem into mini-batches with a short time horizon. We show analytically how this learning strategy regularizes the ill-behaviour of the loss function arising from the strong nonlinearities of ecosystem models. We implement the mini-batch method in the software ecosystem SciML [71], which provides advanced optimizers and allows the automatic generation of efficient and accurate model adjoints, leading to excellent performance. The resulting ML framework makes it possible to efficiently combine the information contained in short, independent time series, and is a workhorse for performing model selection and improving model accuracy. We evaluate the performance of the ML framework in recovering the chaotic dynamics of simulated food webs. We assume a perfect-model setting and test the capacity of the ML framework to recover the true parameters and provide forecasts based on noisy and incomplete observations, and we explore its efficiency in combining the information from multiple time series. Additionally, we investigate whether the ML framework can recover the most appropriate model structure among candidate models. By blending biological knowledge and ML methods, the proposed ML framework is interpretable and data-efficient, and it facilitates mechanism discovery. The proposed approach thus shows promise in improving our ability to understand and forecast ecosystem dynamics.

4.2 Machine learning framework for ecosystem models

4.2.1 Ecosystem model parametrization as a learning problem

Ecosystem models

Ecosystem models generally consist of a system of ordinary differential equations (ODEs) of the form

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), p) \\ x(0) &= x_0 \\ y(t) &= h(x(t)) + \epsilon(t) \end{aligned} \tag{4.1}$$

where $x(t) \in \mathbb{R}^m$ is a vector of state variables that might represent species abundance, resources availability or functional group biomass, $y(t) \in \mathbb{R}^d$ is a vector of observables that contains a subset or aggregates of the state variables, and $p \in \mathbb{R}^q$ is the model parameter vector. h is a function that maps the state variables to the observables, and we assume that the observables are contaminated with a white noise ϵ of Gaussian type, with zero mean and variance–covariance matrix Σ_y. Denoting by $\theta = (x_0, p)$ the vector containing the ICs and the parameters, the model may be viewed as a map $\mathcal{M}$ parametrized by time t that takes the parameters θ to the state variables x

$$\begin{aligned} \mathcal{M}(t, \theta) &= x(t) \\ &= \int_0^t f(s, x(s), p) ds + x_0 \end{aligned} \tag{4.2}$$

Inverse modelling

Taking expectations over the noise realizations yields $\mathbb{E}[y(t)] = h(\mathcal{M}(t, \theta))$, and it follows that the conditional likelihood of each observation $y_k \equiv y(t_k)$, given the parameters θ and the model $\mathcal{M}$ denoted by $p(y_k|\theta, \mathcal{M})$, follows the distribution of the residuals $\epsilon_k \equiv \epsilon(t_k) = y(t_k) - h\left(\mathcal{M}(t_k, \theta)\right)$, which corresponds to the multivariate normal distribution $\mathcal{N}_{0,\Sigma_y}$. Following a Bayesian approach, the parametrization of the ecosystem model can be performed on the basis of the parameter and model posterior probability $p(\theta, \mathcal{M}|\mathbf{y}_{1:K})$, i.e. the conditional probability density of the parameter values θ and the model $\mathcal{M}$ given the data, given by

$$p(\theta, \mathcal{M}|\mathbf{y}_{1:K}) \propto p(\mathbf{y}_{1:K}|\theta, \mathcal{M}) p(\theta, \mathcal{M}) \tag{4.3}$$

where $\mathbf{y}_{1:K} = (y_1, \ldots, y_K)$, $p(\mathbf{y}_{1:K}|\theta, \mathcal{M})$ is the product of the conditional likelihood of each observation y_k

$$\begin{aligned} p(\mathbf{y}_{1:K}|\theta, \mathcal{M}) &= \prod_{i=1}^{K} p(y_i|\theta, \mathcal{M}) \\ &= \prod_{k=1}^{K} \frac{1}{\sqrt{(2\pi)^d |\Sigma_y|}} \exp\left(-\frac{1}{2}\epsilon_k^T \Sigma_y^{-1} \epsilon_k\right) \end{aligned} \tag{4.4}$$

and $p(\theta, \mathcal{M})$ is the prior distribution of the model and its associated parameter values. The model $\mathcal{M}$ is included in the probabilistic quantities in order to accommodate multiple candidate models (see Section 4.3.3).

A variational method to obtain a Bayesian estimate of θ involves maximizing $p(\theta, \mathcal{M}|\mathbf{y}_{1:k})$ to obtain the maximum a posteriori (MAP) estimator [9], which is equivalent to a maximum likelihood approach under a uniform prior distribution of the parameters, i.e. when no prior information on the parameter values is used [79]. Observing that maximizing $p(\theta, \mathcal{M}|\mathbf{y}_{1:K})$ is equivalent to minimizing $-\log p(\theta|\mathbf{y}_{1:K}, \mathcal{M})$ and assuming a normal prior distribution of the parameters $\mathcal{N}_{p_b, \Sigma_p}$, one can obtain the MAP $\hat{\theta}$

$$\hat{\theta} = \arg\min_{\theta} L_{\mathcal{M}}(\theta) \tag{4.5}$$

where

$$L_{\mathcal{M}}(\theta) = \frac{1}{2}\left[\sum_{k=1}^{K-1} \|y_k - h\left(\mathcal{M}(t_k, \theta)\right)\|_{\Sigma_y}^2 + \|p - p_b\|_{\Sigma_p}^2\right] \tag{4.6}$$

[82, 74] and where we use the notation $\|y\|_{\Sigma}^2 = y\Sigma^{-1}y^{\mathbf{T}}$. Eq. (4.6) is similar to a traditional least squares function commonly used in regression, where the second summand is the analogue of a regularization term for the weights and biases of e.g. a neural network.

Gradient-based optimizers can then be used to efficiently obtain $\hat{\theta}$ in Eq. (4.5), iteratively updating the parameter vector θ_m given the gradient of the loss function, denoted by $\nabla_\theta L_{\mathcal{M}}$, to navigate the surface defined by $L_{\mathcal{M}}$ with the aim to find the global minimum where $\nabla_\theta L_{\mathcal{M}}(\hat{\theta}) = 0$. As an example, the plain vanilla gradient descent algorithm is given by

$$\theta_{m+1} = \theta_m - \gamma \nabla_\theta L_{\mathcal{M}}(\theta_m) \tag{4.7}$$

where γ is the learning rate. Other gradient-based algorithms, such as the ADAM optimizer used in the section below, employ more advanced updating strategies to avoid convergence to local minima but stay in the spirit of Eq. (4.7).

Information indigestion

A naive minimization of $L_{\mathcal{M}}(\theta)$ with gradient-based methods is likely to fail, as its associated surface cannot be navigated properly. As illustrated in Fig. 4.1**A**, the loss surface associated

with models characterized by complex dynamics consists of multiple local minima, which cause problems of convergence in efforts to reach the global minimum (depicted by the orange curve in Fig. 4.1**B**). Furthermore, in a neighbourhood of the global minimum, the gradient of the loss function is very large (a "ravine" with almost vertical walls), leading the optimizer to overshoot the true parameter values (depicted by the green curve in Fig. 4.1**B**). In Section 4.A we show in a general setting that these problematic features arise from the dynamical properties of ecosystem models: when the dynamics are chaotic or exhibit a limit cycle (as is often the case for ecological dynamics [8, 39, 42, 6]), the dynamical trajectories exhibit high sensitivity to the model parameters and ICs. This means that a small modification of the parameters or ICs leads to large divergences over time. The prevalence of large deviations causes discontinuities, appearing as many sub-optimal local minima on the loss surface. Moreover, the true minimum can only be found in a narrow ravine that becomes narrower as the number of data points increases. Such surfaces are hardly navigable with gradient descent methods, but since the behaviour of $L_{\mathcal{M}}(\theta)$ critically depends on the time horizon, we reformulate Eq. (4.6) in the following section by splitting the time series into mini-batches with a short time horizon.

4.2.2 ML framework for ecosystem models

Description of the mini-batch method

We propose a mini-batch method that splits the data into mini-batches with a short time horizon. Under perturbed parameters and ICs, chaotic or limit cycle dynamics only diverge after some characteristic simulation time; by splitting the time series into small mini-batches, discontinuities that cause the poor navigability of the landscape are therefore avoided. By averaging the associated losses during the training, the mini-batch method regularizes the loss function and makes it possible to combine the information contained in independent time series.

Consider the availability of S independent time series, where each time series $s \in \{1, \dots, S\}$ contains $K^{(s)}$ observations $\mathbf{y}^{(s)}_{1:K^{(s)}}$. To improve the ill-behaviour of Eq. (4.6), we split each time series into $M^{(s)}$ mini-batches, each of which defines a loss denoted by $L^{(s,m)}_{\mathcal{M}}$. Averaging the losses $L^{(s,m)}_{\mathcal{M}}$ leads to a reformulation of the loss function in Eq. (4.6), yielding

$$
\begin{aligned}
L^{\star}_{\mathcal{M}}(\theta) &= \tfrac{1}{S}\sum_{s=1}^{S} \tfrac{1}{M^{(s)}} \sum_{m=0}^{M^{(s)}-1} L^{(s,m)}_{\mathcal{M}}(\theta) \\
&= \tfrac{1}{S}\sum_{s=1}^{S} \tfrac{1}{M^{(s)}} \sum_{m=0}^{M^{(s)}-1} \Bigg[\tfrac{1}{K^{(s,m)}} \sum_{k=1}^{K^{(s,m)}} \| y^{(s)}_{k+mK^{(s)}/M^{(s)}} - h\left(\mathcal{M}(t_{k+mK^{(s)}}, p, x_0^{(s,m)})\right) \|_{\Sigma_y} \\
&\qquad + \| y^{(s)}_{mK^{(s)}/M^{(s)}} - h\left(x_0^{(s,m)}\right) \|_{\Sigma_{x_0}} + \| p_b - p \|_{\Sigma_p} \Bigg]
\end{aligned}
\tag{4.8}
$$

where $\theta = (p, x_0^{(1,1)}, \dots x_0^{(1,M_1)}, x_0^{(2,1)} \dots, x_0^{(S,M_S)})$ consists of the augmented parameter vector containing $\sum_{s=1}^{S}\left[M^{(s)} - 1\right]$ additional ICs to be estimated and denoted by $x_0^{(s,m)}$,

$K^{(s,m)} = \begin{cases} K^{(s)}/M^{(s)} + 1 & : m < M \\ K^{(s)}/M^{(s)} & : m = M \end{cases}$ indicates the number of points to include in the mth batch, and p_b designates the prior knowledge of the parameter values. We refer to Fig. 4.1**C** for a graphical representation of Eq. (4.8). In contrast to Eq. (4.6), the ICs for each batch are constrained by an extra term with weight $\Sigma_{x_0}^{-1}$, which is needed in practice for better convergence. The ICs $x_0^{(s,m)}$ are nuisance parameters that augment the dimensionality of the parameter space as they are inferred for each batch. Nonetheless, the efficiency of gradient-based methods, together with the suggested numerical implementation detailed in the section below, largely circumvent the additional cost associated with the augmented dimensionality. $K^{(s,m)}$ expresses the overlap between each batch of data for $m < M^{(s)}$ and ensures that all the available information is utilized to constrain the parameter vector p. By smoothing the ruggedness of the loss surface (orange curve in Fig. 4.1**A**), the proposed mini-batch method yields an improved navigability (orange curve in Fig. 4.1**B**). We show analytically in Section 4.A that using $L^{\star}_{\mathcal{M}}$ yields a more navigable loss surface than if $L_{\mathcal{M}}(\theta)$ is used in Eq. (4.6).

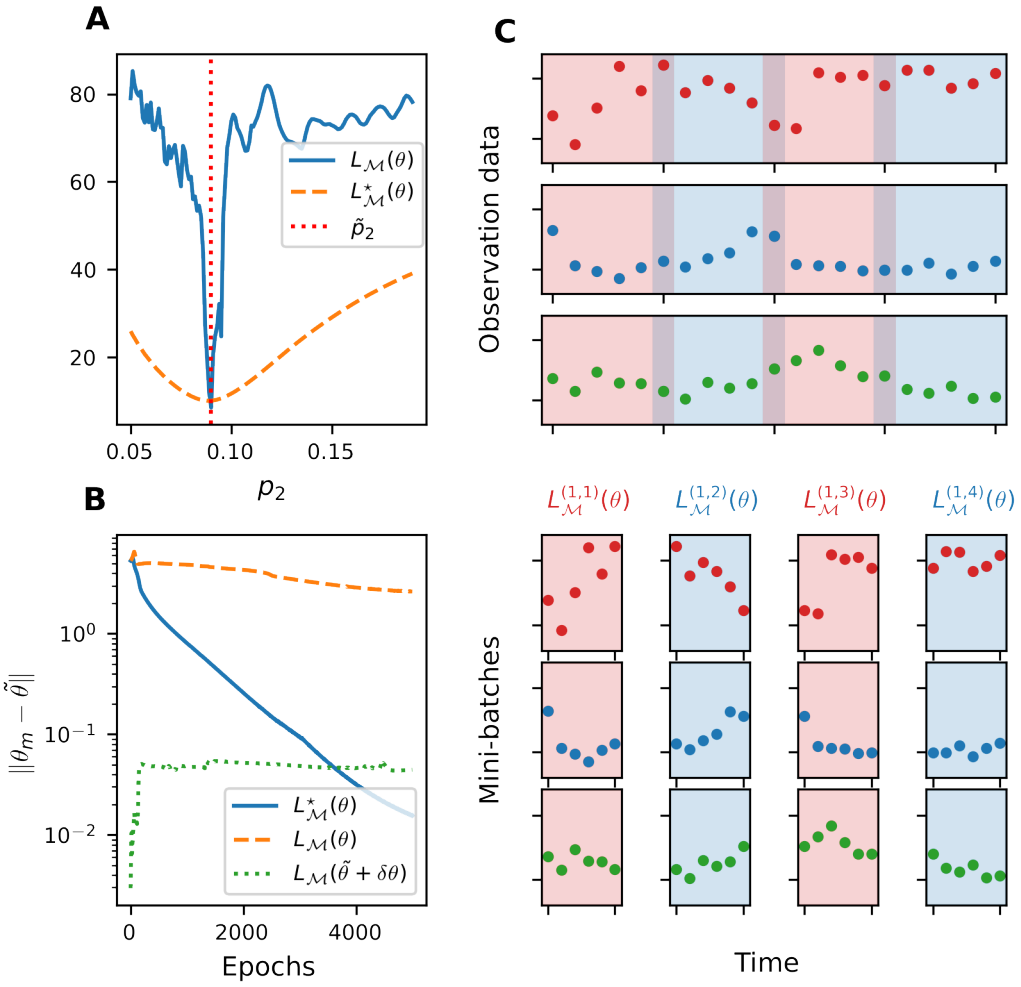

Fig. 4.1: Illustration of the mini-batch method. **A** Characteristic features of the naive loss function $L_{\mathcal{M}}(\theta)$ and the mini-batch loss function $L^{\star}_{\mathcal{M}}(\theta)$. The blue and orange lines correspond to cross sections of $L_{\mathcal{M}}(\theta)$ and $L^{\star}_{\mathcal{M}}(\theta)$, respectively, obtained from the ecosystem model presented in the section Simulated food-web model as a case study. While the cross section of $L_{\mathcal{M}}(\theta)$ presents many local minima and a very large gradient in the neighbourhood of the true parameter, which renders the navigability of the loss surface difficult, the cross section of $L^{\star}_{\mathcal{M}}(\theta)$ is smooth and shows a single minimum, illustrating the regularization induced by the mini-batch method. **B** Convergence of gradient descent algorithms applied to $L_{\mathcal{M}}$ and $L^{\star}_{\mathcal{M}}$. The blue, orange and green lines correspond to the loss function evaluated against the epochs (number of parameter updates) using $L^{\star}_{\mathcal{M}}$, $L_{\mathcal{M}}$ starting from a parameter value close to the true parameters, and $L^{\star}_{\mathcal{M}}$, respectively. The ill-behaviour of $L_{\mathcal{M}}$ leads to the convergence to a local minimum, while $L^{\star}_{\mathcal{M}}$ is associated with a smooth convergence to the true parameters. **C** Graphical representation of the proposed mini-batch method. To improve the navigability of the posterior landscape, the algorithm splits the time series into mini-batches with short time horizons (blue and red portions of the time series). Since mini-batches are treated independently, the method naturally extends to independent time series.

Numerical implementation of the ML framework

The choice of the optimizer and the correct calculation of the model sensitivity to the parameters and ICs, upon which the gradient of the loss function $\nabla L^{\star}_{\mathcal{M}}(\theta)$ depends, play an essential role in the success of the minimization of the loss function and the subsequent correct estimation of the MAP. To accelerate the learning process and make it more robust, we propose to combine the mini-batch method with modern optimizers and automatic differentiation. Building upon the software ecosystem SciML [71], we use **DifferentialEquations.jl** for the forward integration of the ecosystem model, as it provides highly efficient ODE solvers [70]. **DifferentialEquations.jl** is additionally compatible with automatic differentiation and includes an extensive set of sensitivity analysis methods [54], enabling the automatic generation of the model sensitivity to the parameters and ICs and guaranteeing their accuracy. This automatic generation greatly reduces the effort and potential errors associated with the adjoint code construction, enabling continuous development of the models. The accuracy is also an essential feature, as the model sensitivities are critically involved in the minimization of Eq. (4.6) and their inaccuracies can compromise the convergence of the gradient-based optimizers [35]. The interoperability of the SciML ecosystem further makes it possible to benefit from the tooling of the deep learning library **Flux.jl** and the nonlinear optimization library **Optim.jl** [44], providing state-of-the-art optimizers that are computationally efficient and well suited for highly parameterized models [77]. We use the adaptive, momentum-based Adam optimizer [47] to converge in the basin of attraction of the true parameters, which we substitute with the limited memory BroydenFletcherGoldfarbShanno optimizer (L-BFGS) [53] for the final training epochs to ensure faster and more accurate convergence.

The reformulation of the learning problem in Eq. (4.8), together with the numerical implementation suggested above, define the proposed ML framework, which we benchmark with a concrete case scenario in the next section.

4.3 Simulated food-web model as a case study

We evaluate the performance of the ML framework by considering a food-web ecosystem composed of three functional compartments including a resource, consumers and predators. We use a reference model to generate the observation data and first assume a perfect-model setting, evaluating the performance of the ML framework in parametrizing the reference model for different noise levels, with incomplete observations, and with an increasing number of independent time series. Second, we relax the perfect-model assumption by considering two plausible candidate models capturing contrasting hypotheses regarding the ecological processes, and test whether the ML framework can provide support for the true generating model by combining it with information-based model selection.

4.3.1 Three-compartment food-web ecosystem

We use a reference food-web model investigated in [38, 59, 58, 48] where a resource R is eaten by consumers C, which in turn are fed upon by predators P (model $\mathcal{M}_1$ in Fig. 4.2). We further consider an "omnivory variant" of the reference model introduced in [57], where predators are omnivorous and can feed upon the resource with a determined strength ω (model $\mathcal{M}_2$ in Fig. 4.2). These models generate fluctuations that resemble the behaviour of observed ecological time series [8], they produce chaotic dynamics that are notoriously challenging to forecast for a wide range of realistic parameters [68], and they have been used as benchmarks for proposed ecosystem forecasting methods (see [65, 20, 92]).

After nondimensionalization, the three-compartment model and the omnivory variant comprise a total of six and nine parameters, respectively: the mass-specific metabolic rate of consumers and predators x_C and x_P, the ingestion rate per unit metabolic rate of consumers and predators y_C and y_P (decomposed into y_{PC}, y_{PR} for the omnivory variant), the half saturation densities for the type II functional responses of the consumers and predators R_0 (decomposed into R_0, R_{02} for the omnivory variant) and C_0, and the omnivory strength ω for the omnivory variant (see Section 4.B for the ODE details). Time is nondimensionalized by the resource growth rate and set to the biologically realistic value of 100% biomass increase per day, so that one unit of time corresponds to one day. The parameters in the simulations are set to the biologically realistic values proposed by [58, 57], which additionally ensure that the dynamics of the system are chaotic or show oscillations (see Section 4.B for details).

We generate the observation data by sampling the simulated ecosystem dynamics and by contaminating the samples with noise. The noise variance–covariance matrix Σ_y is set to be diagonal, with entries that are proportional to the sample variances of the observables

$$\operatorname{diag} \Sigma_y = r^2[\operatorname{Var}(\tilde{y}_1), \ldots, \operatorname{Var}(\tilde{y}_d)] \tag{4.9}$$

where r indicates the noise level and $\tilde{y}$ corresponds to the noiseless data generated with the true parameter values $\tilde{\theta}$. We sample the simulated dynamics after a long burn-in time ($t > 500$) to ensure that transient dynamics are not observed. A visual representation of the generated data is displayed in Fig. 4.2**B**. We assume a uniform distribution of the parameter priors, and randomly draw initial parameter estimates from a uniform distribution so that the initial parameter estimates follow $\mathcal{U}(0, 2\tilde{p})$.

We consider two different settings: one where all compartment abundances are observable, i.e. the observing system map h is the identity, and one where only predator and consumer abundances are available, i.e. discarding the resource abundance data. For both settings, structural identifiability is tested with the Julia library **StructuralIdentifiability.jl** [22] and verified globally. This means that in theory, the unique observation of predator and consumer abundances carries the information required for a complete characterization of all the model parameters.

For the meta-parameters of the ML framework, we set $\Sigma_{x_0} = \frac{M^{(1)}}{K^{(1)}}\Sigma_y$, we use Adam with $\gamma_m = \mathbb{1}_{[0,2000]}(m)10^{-1} + \mathbb{1}_{[0,2000]}(m)10^{-2} + \mathbb{1}_{[0,2000]}(m)10^{-3}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for the first 6000 epochs, where γ_m corresponds to the learning rate of the mth epoch, and we use L-BFGS for the last 200 epochs.

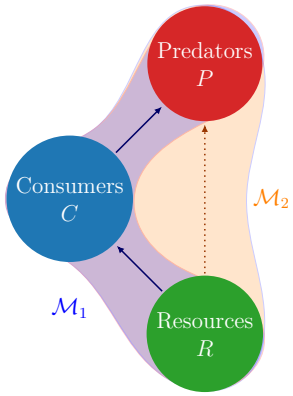

Fig. 4.2: Reference food web systems considered. In Section 4.3.2, the blue model $\mathcal{M}_1$ from [38] is considered, where a resource is eaten by consumers, which are themselves eaten by predators. In Section 4.3.3, the orange model $\mathcal{M}_2$, corresponding to the omnivory variant introduced in [57], is also considered.

4.3.2 Parameter learning in a perfect-model setting

We generate time series from the reference food-web model (blue model in Fig. 4.2) under varying noise levels and for both settings with total and partial observations, sampling from the model simulations every four days (see Fig. 4.1**C** for an illustration of the generated data). We then apply the ML method to the generated data, with a focus on evaluating its performance in recovering the true parameters $\tilde{p}$ and on its forecast skill.

We evaluate the performance in recovering the true parameters with two metrics, namely the coefficient of determination between the true parameters $\tilde{x}_P$ and the estimated parameters $\hat{x}_P$, denoted by $R^2_{x_P}$, and the relative parameter error for the ensemble of training simulations, denoted by $|(\hat{p}-\tilde{p})/\tilde{p}|$ and calculated as the median relative parameter error across the six estimated parameters. To evaluate the out-of-sample forecast skill, we simulate the model beyond the training time span by using the estimated ICs of the last batch of each independent time series. Further, we quantify the forecast skill, denoted by ρ^2, by computing the mean squared correlation across all the independent time series between the prey abundance generated with the true parameter $\tilde{\theta}$ and the predicted prey abundance. We obtain summary statistics of the metrics by varying the critical parameter value x_P, generating a total of 50 simulations for each noise level and setting considered. While only x_P is varied, all the parameters together with the ICs are collectively fitted.

Robustness of the ML framework against noise and incomplete observations

We set the number of time series to $S = 1$, the time series length to $K = 80$, and the number of mini-batches to $M = 8$. We investigate the ML framework performance against observational noise and under the setting with complete or partial observations.

In the complete observation setting, the ML framework can very accurately recover the true parameter values under a moderate observational noise, with mean $|(\hat{p}-\tilde{p})/\tilde{p}| = 6\%$ and $R^2_{x_P} = 0.99$ for 20% observational noise ($r = 0.2$; see Fig. 4.3**A**, red dots). In the partial observation setting, fair results are also obtained with $|(\hat{p}-\tilde{p})/\tilde{p}| = 16\%$ and $R^2_{x_P} = 0.89$ (Fig. 4.3**A**, blue triangles). On top of being accurate, the ML framework further shows a very short inference time, i.e. 36 seconds in the complete observation setting and 34 seconds in the partial observation setting (see Table S1 for details). To investigate systematically how the ML framework accommodates different levels of observational noise, we vary the noise level from $r = 0.$ to $r = 1$ and calculate $|(\hat{p}-\tilde{p})/\tilde{p}|$. Results reported in Fig. 4.3**B** show that the ML framework handles observational noise well and the response of the logarithm of $|(\hat{p}-\tilde{p})/\tilde{p}|$ to r only differs by a constant factor between the complete and partial observation settings.

Given that the food-web dynamics are chaotic (Section 4.B), excellent performance in parameter estimation might not be sufficient to provide accurate forecasts. We therefore also test the ML algorithm by evaluating how the forecast skill ρ^2 is affected by the noise level under complete or partial observations. Results for the complete and partial observation settings reported in Fig. 4.3**C** show good forecast skill under moderate observational noise, where ρ^2 linearly decreases with the time horizon considered.

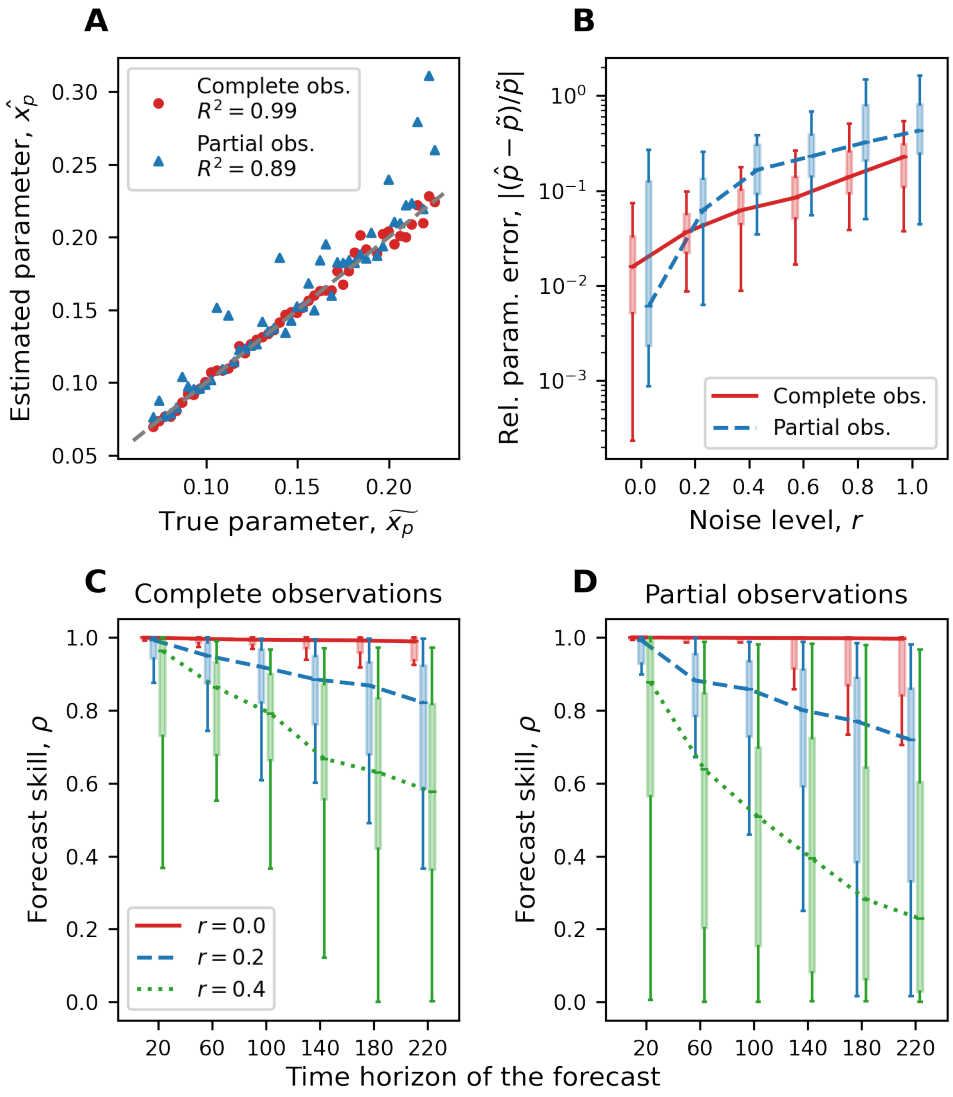

Fig. 4.3: Performance of the proposed ML framework for varying noise levels, under the complete and the partial observation setting. A True parameter $\tilde{x_P}$ against estimated parameter $\hat{x_P}$ for $r = 0.2$ under the complete and the partial observation setting. Although the parameter estimation is more accurate when complete observations of abundance are available, parameters show a fair fit when estimated with partial observations. **B** Relative parameter error $|(\hat{p}-\tilde{p})/\tilde{p}|$ for varying noise levels under the setting of complete or partial observations. **B** supports the above observation for varying noise levels. **C** Forecast skill ρ^2 of the trained model under the complete observation setting. **D** Analogous data under the partial observation setting. In **A–D**, the batch size is set to $m = 6$.

We further investigate the ability of the ML framework to process and combine information from independent datasets. We reduce the time horizon of the observation data by setting the time series length to $K = 12$, generate two datasets comprising $S = 1$ and $S = 6$ independent time series, respectively, – obtained from independent ICs – and set the number of batches for each time series s to $M^{(s)} = 2$. In both the complete and partial observation settings, we find that the relative parameter error $|(\hat{p}-\tilde{p})/\tilde{p}|$ is consistently lower in the simulations with a larger number of time series (Fig. 4.4**A**). The forecast skill is also consistently improved as more independent time series are processed (Fig. 4.4**B**), and the forecast skill for long-term predictions considerably increases. These results confirm the robustness of the ML framework against noise and partial observations, and show that the ML framework can efficiently harness the information from disparate observation datasets.

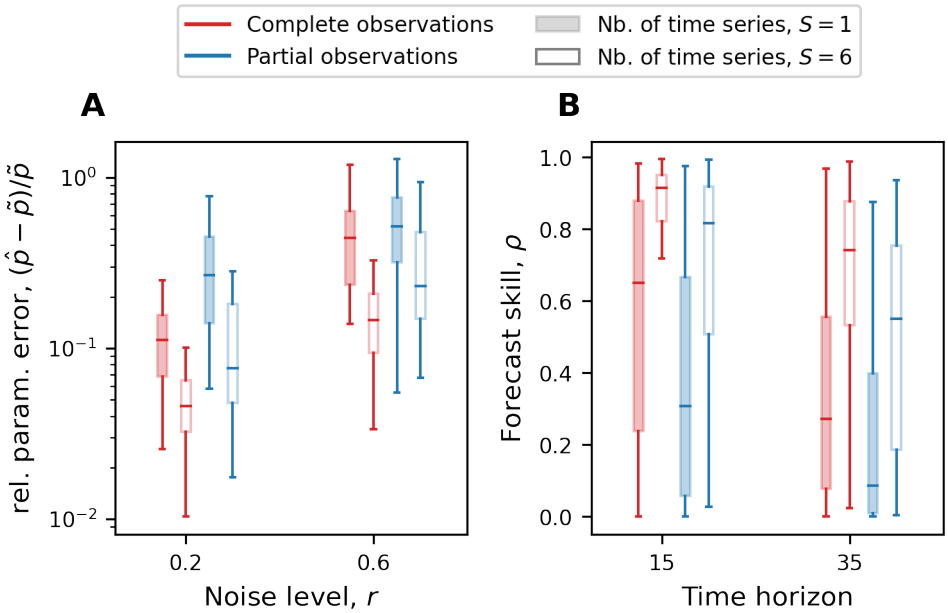

Fig. 4.4: Performance of the ML framework in processing and combining the information of multiple independent data sets. A Relative parameter error $|(\hat{p}-\tilde{p})/\tilde{p}|$ for different numbers of time series and levels of noise, under the complete and the partial observation setting. **B** Forecast skill for different numbers of time series and time horizons of the forecasts, under the complete and the partial observation setting. $r = 0.1$. In **A–B**, $|(\hat{p}-\tilde{p})/\tilde{p}|$ decreases while ρ^2 increases as the number of time series processed increases, demonstrating the capacity of the ML framework to process and combine the information from independent time series. In **A–B**, each box plot corresponds to 100 independent simulations where $\tilde{x_P}$ varies $\tilde{x_P} \in [0.071, 0.225]$, and the batch size is set to $m = 6$.

4.3.3 Elucidating mechanistic pathways

Finally, we relax the perfect-model assumption and investigate whether the ML framework can provide statistical support for the true generating model among several candidates with information-based model selection. Specifically, we investigate whether the ML framework can detect omnivory from single observations of time series. We generate multiple observation datasets from the omnivory variant model $\mathcal{M}_2$ for different omnivory strengths ω and noise levels r. We consider both the standard model $\mathcal{M}_1$ and the omnivory variant model $\mathcal{M}_2$ as two plausible candidate models (see Fig. 4.2 for a graphical illustration of the models). We use the Akaike information criterion (AIC) to select the model with the strongest support in relation to the data [55]. In the specific case of our framework, we calculate the AIC as

$$\text{AIC}_{\mathcal{M}_i} = -2\ln(p(\hat{\theta}, M_i|\mathbf{y}_{1:K})) + 2k_{\mathcal{M}_i} \tag{4.10}$$

where $p(\hat{\theta}, \mathcal{M}_i|y_{1:K})$ corresponds to the maximum value of the likelihood of the model $\mathcal{M}_i$ given the data, and $k_{\mathcal{M}_i}$ is the number of parameters in the model $\mathcal{M}_i$. The AIC ranks the most probable models by penalizing complexity to balance information loss and parsimony, where candidate models with the lowest scores are ranked as the most likely. We consider the Akaike weights $w_{\mathcal{M}_i} = \frac{\exp(-\Delta\text{AIC}_{\mathcal{M}_i}/2)}{\sum_j \exp(-\Delta\text{AIC}_{\mathcal{M}_j}/2)}$, where $\Delta\text{AIC}_{\mathcal{M}_i} = \text{AIC}_{\mathcal{M}_i} - \min_j \text{AIC}_{\mathcal{M}_j}$, which can be directly interpreted as the probability that $\mathcal{M}_i$ is the most appropriate model given the data (see [46]). We expect that the Akaike weights provide support for the generating model $\mathcal{M}_2$ only across values where $\omega > 0$, as $\mathcal{M}_1$ is equivalent to $\mathcal{M}_2$ when $\omega = 0$ and $\mathcal{M}_2$ is penalized by its three additional parameters.

In the complete observation setting, for moderate observational noise ($r = 0.1$) we find that $\mathcal{M}_1$ is given strong support for $\omega < 0.07$ ($w_{\mathcal{M}_1} > 98\%$) and that $\mathcal{M}_2$ if favored for $\omega > 0.08$ ($w_{\mathcal{M}_2} > 99\%$), providing overall strong support for the true model over a large range of ω values (Fig. 4.5**A**). As the observational noise increases the support strength naturally decreases, leading to an increased range of ω values where the simplest model $\mathcal{M}_1$ is favored or where no model is given strong support (Fig. 4.5**B**-**C**). On the other hand, in the partial observation setting, the lack of data prevents the correct estimation of the omnivory variant model parameters, leading model $\mathcal{M}_1$ to be supported for an even larger range of ω values (Fig. S1).

Overall, the ML framework provides statistical support for the model embedding the most appropriate hypotheses given the available data. With appropriate data, the proposed ML framework can therefore elucidate mechanistic pathways and infer ecological processes by utilizing information-based model selection.

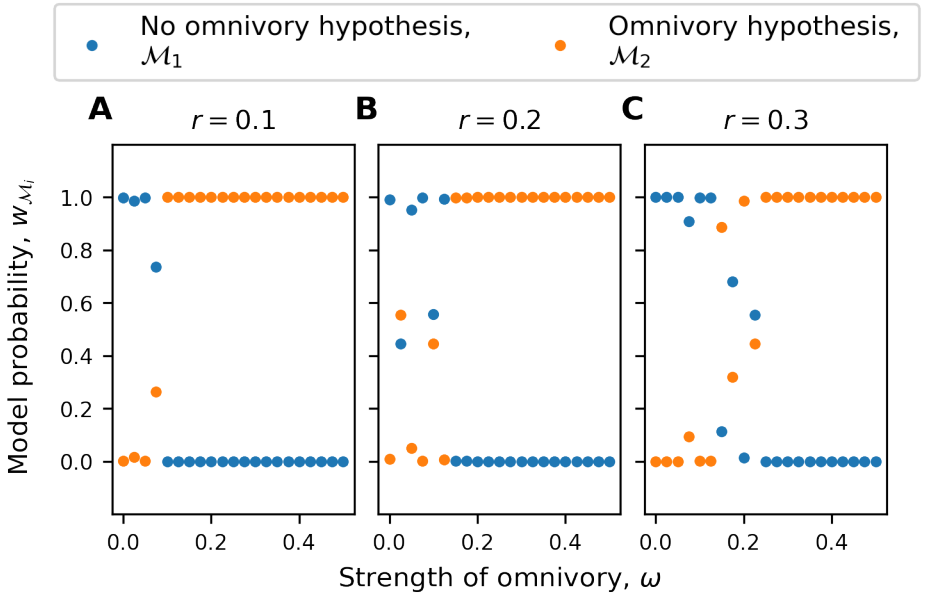

Fig. 4.5: Performance of the ML framework in supporting the predator omnivory hypothesis in a food web. **A–C** Hypothesis testing for levels of noise $r = 0.1, 0.2, 0.3$. Blue dots correspond to $w_{\mathcal{M}_1}$, the Akaike weights of the simple food-web model, and orange dots correspond to $w_{\mathcal{M}_2} = 1 - w_{\mathcal{M}_1}$, the Aikaike weights of the omnivory model, which can be interpreted as model probabilities. **A–C** indicate that the ML framework can detect omnivory, as the omnivory model $\mathcal{M}_2$ is given strong support ($w_{\mathcal{M}_2} > 99\%$) for most of the ω range investigated.

4.4 Discussion

We propose a ML framework combining a mini-batch method inspired by multiple shooting methods [67] with automatic differentiation [71] and state-of-the-art variational optimizers [47] to efficiently and accurately parametrize complex dynamical models. We show formally that splitting the data into mini-batches with a short time horizon regularizes the loss function associated with dynamical models characterized by complex dynamics, such as chaotic dynamics and limit cycles (Section 4.A). We demonstrate numerically that this reformulation ensures the success of gradient-based optimizers to parametrize ecosystem models (Figs. 4.1, 4.3 and 4.4). This mini-batch method is also relevant beyond variational methods and applies to any inferential method navigating the posterior landscape, such as evolutionary algorithms [87, 75] or Markov Chain Monte Carlo methods [52, 40, 90, 27, 76]. The proposed approach is particularly relevant for the parametrization of ecosystem models incorporating realistic ecological and adaptive mechanisms [84], which are generally associated with strong nonlinearities due to the complexity of processes linking interacting ecological compartments [8, 39, 42, 6]. It further integrates the practical constraints of available ecological datasets [23], accommodating incomplete, noisy, shallow and independent observation data. Overall, the ML framework successfully blends ML methods with mechanistic ecosystem models to learn from ecological time series, and it could therefore

improve our quantitative understanding of ecosystem dynamics and help to anticipate their responses to global changes [84].

Our work contributes to the ongoing effort to better assimilate observational data into mechanistic models [79, 72, 45], with a specific focus on the parametrization of ecosystem models with strong nonlinearities. Recently, [91] proposed an alternative framework dubbed "systems biology informed deep learning", where a neural network is fitted to the data and the additional mechanistic model constraints are integrated. This alternative framework extends previous colocation methods [73, 14] and has the advantage of being able to parametrize stochastic models. As it requires the selection of a neural network architecture and a "goodness of fit" parameter, it nevertheless imposes an additional layer of complexity, which might negatively affect the model parametrization [91]. In contrast, the ML framework proposed here trains the model directly against data, using automatic differentiation and sensitivity analysis in order to apply variational optimizers directly to the model simulations. This makes it possible to bypass the use of neural networks, rendering the parametrization process simpler and more amenable to model selection [73].

By integrating the practical constraints imposed by ecological datasets, the ML framework can learn from short time series with partial and noisy observations (Figs. 4.3 and 4.4). Local ecosystem surveys, such as marine trawling surveys or local terrestrial surveys ([66, 23, 13] and references therein), provide time series that are generally shallow in time but composed of many replicates [41, 15], in part due to the practical difficulties of long-term monitoring [92]. Our results show that the inclusion of multiple independent time series in the training dataset reduces the error in the parameter estimates and increases the forecast skill (Fig. 4.4). This indicates that the proposed ML framework could, in practice, efficiently harness the information available in current ecological datasets. Instead of directly comparing simulated and observed data, matching time-averaged statistics between observations and simulations (e.g. means and covariances) could further yield an improved assimilation of observations from diverse data sources, such as global observations of productivity from satellites and local surveys, as proposed for climate models [82]. Overall, the proposed ML framework accommodates the specificities of current ecological time series and can improve the assimilation of ecological data into mechanistic ecosystem models.

Our work can help elucidate mechanistic pathways by contrasting hypotheses embedded in model variants. Using information-criterion-based model selection, we demonstrate with a case study that the ML framework is able to provide statistical support for the true generating model among two different candidates (Fig. 4.5). Importantly, the ML framework can perform model selection on complex models, incorporating key mechanisms such as trait–species interactions, evolutionary potential and responses to environmental conditions, which have been shown to be important in mediating ecosystem dynamics and must be refined in models to improve predictive accuracy [84]. The ML framework can therefore lead to the improvement of current ecosystem models and knowledge, which is crucially needed given that key ecological processes are only partially described in most ecosystem models [79]. AIC can also be used to ensure the interpretability of the model parameters, and should be preferred to estimating the parameter uncertainty through e.g.

the Cramer Rao inqequality [46]: by favouring models with less complexity, model selection techniques disqualify uninformative parameters to ensure interpretability [46]. This has the extra benefit of reducing the dimensionality of the parameter space, hence improving the estimation of other parameters. Following recent novel approaches to investigate ecological hypotheses [16], our method contributes to the development of a process understanding of ecosystem functions and provides a path forward to better link ecological theory and data.

The proposed approach still presents a set of limitations, which might hamper its success under specific situations. First, while the use of mini-batches smooths the loss surface and ensures better convergence, it also flattens the loss surface around the true parameter value, which consequently deteriorates the precision of the inferred parameters because the loss function takes similar values in an extended neighbourhood of the true parameters. To circumvent this issue, iterative training can be performed, where the learning is initiated by a short batch length $K^{(s)}$ to identify the region with the most probable parameters, and in subsequent iterations the batch length is increased to improve the precision of the inference. Iterative training could also improve the lack of statistical support obtained in hypothesis testing experiments (see simulations in Figs. S1 and 4.5 where none of the models is given statistical support), as it would increase differences in likelihood for parameter values around the neighbourhood of the true parameter values. Second, our results highlight that the data might not provide enough constraints for a correct parametrization (Fig. 4.4, partial observation setting and $S = 1$). Pre-experimental analyses with simulated synthetic data might therefore be required to design the sampling protocol and campaign to ensure an adequate sampling effort [3, 50]. Third, while in Eq. (4.8) it is assumed that the parameter values are the same across the time series, strong regional variability might also be observed among the spatially replicated data, causing parameter values to vary across the replicates. The knowledge of this variability could motivate partial pooling [5] or parametrization of the parameters in terms of environmental conditions [62], to account for the independence of the parameter values across the replicates. Finally, while the proposed ML framework greatly improves convergence, it could still be that – even with a large amount of data – poor initial parameter estimates, a large number of free parameters, or high noise levels prevent convergence to the true minimum. Performing multiple runs with varying initial parameter estimates can ensure that the maximum a priori estimate is reliable. If this is not the case, stochasticity could further be introduced within the ML framework to prevent the convergence to local minima, where only a subset of mini-batches are fitted at each epoch [11].

4.5 Conclusion

We proposed a ML framework based on a mini-batch method combined with automatic differentiation and state-of-the-art optimizers to estimate the parameters and improve the forecast skill of complex ecosystem models from observation data. The ML framework was benchmarked with a realistic ecosystem model characterized by strong nonlinearities and delivered excellent performance, accommodating the practical constraints imposed by the

quality and availability of ecological datasets. Our experiments have further illustrated the ability of the ML framework to discriminate between several candidate models, enabling the testing of ecological theories against data and the improvement of current mechanistic models. Given the increasing number of ecological datasets following the development of monitoring technologies such as environmental DNA [78], remote sensing [43], bioaccoustics [1], and citizen observations [30], the proposed ML framework opens up new opportunities for the quantitative investigation of current ecosystem functions [16] and the prediction of ecosystem responses to increasing disruptions [84].

4.6 Acknowledgements

L.P. and V.B. were supported by the SNF grant 310030E_205556. P.V.A. was supported by an ETH postdoctoral fellowship.

4.7 Code availability

The ML framework is implemented in the multi-purpose Julia package **MiniBatchInference.jl** available at https://github.com/vboussange/MiniBatchInference.jl, and the simulation code is available at https://github.com/vboussange/mini-batching-ecological-data.

References

[1] T. M. Aide et al. “Real-time bioacoustics monitoring and automated species identification”. In: *PeerJ* 1.1 (2013), e103. DOI: 10.7717/peerj.103.

[2] M. Alber et al. “Integrating machine learning and multiscale modelingperspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences”. In: *npj Digital Medicine* 2.1 (2019), p. 115. DOI: 10.1038/s41746-019-0193-y. arXiv: 1910.01258.

[3] H. Banks et al. “PARAMETER ESTIMATION FOR AN ALLOMETRIC FOOD WEB MODEL”. In: *International Journal of Pure and Apllied Mathematics* 114.1 (2017), pp. 143–160. DOI: 10.12732/ijpam.v114i1.12.

[4] A. D. Barnosky et al. “Approaching a state shift in Earth’s biosphere”. In: *Nature* 486.7401 (2012), pp. 52–58. DOI: 10.1038/nature11018. arXiv: 9605103 [cs].

[5] M. A. Beaumont. “Approximate Bayesian Computation in Evolution and Ecology”. In: *Annual Review of Ecology, Evolution, and Systematics* 41.1 (2010), pp. 379–406. DOI: 10.1146/annurev-ecolsys-102209-144621. arXiv: 1212.1417.

[6] E. Benincà et al. “Chaos in a long-term experiment with a plankton community”. In: *Nature* 451.7180 (2008), pp. 822–825. DOI: 10.1038/nature06512.

[7] L. Bertino, G. Evensen, and H. Wackernagel. “Sequential Data Assimilation Techniques in Oceanography”. In: *International Statistical Review* 71.2 (2003), pp. 223–241. DOI: 10.1111/j.1751-5823.2003.tb00194.x.

[8] O. N. Bjørnstad and B. T. Grenfell. “Noisy Clockwork: Time Series Analysis of Population Fluctuations in Animals”. In: *Science* 293.5530 (2001), pp. 638–643. DOI: 10.1126/science.1062226.

[9] M. Bocquet et al. “Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models”. In: *Nonlinear Processes in Geophysics* 26.3 (2019), pp. 143–162. DOI: 10.5194/npg-26-143-2019.

[10] G. B. Bonan. “Forests and Climate Change: Forcings, Feedbacks, and the Climate Benefits of Forests”. In: *Science* 320.5882 (2008), pp. 1444–1449. DOI: 10.1126/science.1155121.

[11] L. Bottou. “Stochastic gradient descent tricks”. In: *Neural networks: Tricks of the trade*. Springer, 2012, pp. 421–436.

[12] I. L. Boyd. “The Art of Ecological Modeling”. In: *Science* 337.6092 (2012), pp. 306–307. DOI: 10.1126/science.1225049.

[13] M. T. Burrows et al. “Ocean community warming responses explained by thermal affinities and temperature gradients”. In: *Nature Climate Change* 9.12 (2019), pp. 959–963. DOI: 10.1038/s41558-019-0631-5.

[14] J. Cao, G. F. Fussmann, and J. O. Ramsay. “Estimating a PredatorPrey Dynamical Model with the Parameter Cascades Method”. In: *Biometrics* 64.3 (2008), pp. 959–967. DOI: 10.1111/j.1541-0420.2007.00942.x.

[15] A. T. Clark et al. “Spatial convergent cross mapping to detect causal relationships from short time series”. In: *Ecology* 96.5 (2015), pp. 1174–1181. DOI: 10.1890/14-1479.1.

[16] A. Curtsdotter et al. “Ecosystem function in predatorprey food websconfronting dynamic models with empirical data”. In: *Journal of Animal Ecology* 88.2 (2019). Ed. by D. Stouffer, pp. 196–210. DOI: 10.1111/1365-2656.12892.

[17] D. L. DeAngelis and S. Yurek. “Equation-free modeling unravels the behavior of complex ecological systems”. In: *Proceedings of the National Academy of Sciences* 112.13 (2015), pp. 3856–3857. DOI: 10.1073/pnas.1503154112.

[18] J. P. DeLong, T. C. Hanley, and D. A. Vasseur. “Predator-prey dynamics and the plasticity of predator body size”. In: *Functional Ecology* 28.2 (2014). Ed. by M. Pfrender, pp. 487–493. DOI: 10.1111/1365-2435.12199.

[19] B. Deneu et al. “Convolutional neural networks improve species distribution modelling by capturing the spatial structure of the environment”. In: *PLOS Computational Biology* 17.4 (2021). Ed. by A. C. Martiny, e1008856. DOI: 10.1371/journal.pcbi.1008856.

[20] E. R. Deyle et al. “Tracking and forecasting ecosystem interactions in real time”. In: *Proceedings of the Royal Society B: Biological Sciences* 283.1822 (2016), p. 20152258. DOI: 10.1098/rspb.2015.2258.

[21] S. C. Doney. “The Growing Human Footprint on Coastal and Open-Ocean Biogeochemistry”. In: *Science* 328.5985 (2010), pp. 1512–1516. DOI: 10.1126/science.1185198.

[22] R. Dong et al. “Differential elimination for dynamical models via projections with applications to structural identifiability”. In: (2021), pp. 1–37. arXiv: 2111.00991.

[23] M. Dornelas et al. “BioTIME: A database of biodiversity time series for the Anthropocene”. In: *Global Ecology and Biogeography* 27.7 (2018), pp. 760–786. DOI: 10.1111/geb.12729.

[24] M. Doron et al. “Stochastic estimation of biogeochemical parameters from Globcolour ocean colour satellite data in a North Atlantic 3D ocean coupled physical-biogeochemical model”. In: *Journal of Marine Systems* 117-118 (2013), pp. 81–95. DOI: 10.1016/j.jmarsys.2013.02.007.

[25] E. C. Ellis. “Anthropogenic transformation of the terrestrial biosphere”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 369.1938 (2011), pp. 1010–1035. DOI: 10.1098/rsta.2010.0331.

[26] K. Fennel et al. “Testing a marine ecosystem model: sensitivity analysis and parameter optimization”. In: *Journal of Marine Systems* 28.1-2 (2001), pp. 45–63. DOI: 10.1016/S0924-7963(00)00083-X.

[27] J. Fiechter et al. “A Bayesian parameter estimation method applied to a marine ecosystem model for the coastal Gulf of Alaska”. In: *Ecological Modelling* 258 (2013), pp. 122–133. DOI: 10.1016/j.ecolmodel.2013.03.003.

[28] R. A. Fisher et al. “Vegetation demographics in Earth System Models: A review of progress and priorities”. In: *Global Change Biology* 24.1 (2018), pp. 35–54. DOI: 10.1111/gcb.13910.

[29] A. Gábor and J. R. Banga. “Robust and efficient parameter estimation in dynamic models of biological systems”. In: *BMC Systems Biology* 9.1 (2015), p. 74. DOI: 10.1186/s12918-015-0219-2.

[30] GBIF: The Global Biodiversity Information Facility. “What is GBIF?” In: (2022).

[31] W. L. Geary et al. “A guide to ecosystem models and their environmental applications”. In: *Nature Ecology & Evolution* 4.11 (2020), pp. 1459–1471. DOI: 10.1038/s41559-020-01298-8.

[32] M. Gehlen et al. “Building the capacity for forecasting marine biogeochemistry and ecosystems: recent advances and future developments”. In: *Journal of Operational Oceanography* 8.sup1 (2015), s168–s187. DOI: 10.1080/1755876X.2015.1022350.

[33] W. Gentleman et al. “Functional responses for zooplankton feeding on multiple resources: a review of assumptions and biological dynamics”. In: *Deep Sea Research Part II: Topical Studies in Oceanography* 50.22-26 (2003), pp. 2847–2875. DOI: 10.1016/j.dsr2.2003.07.001.

[34] M. Gharamti et al. “Ensemble data assimilation for ocean biogeochemical state and parameter estimation at different sites”. In: *Ocean Modelling* 112 (2017), pp. 65–89. DOI: 10.1016/j.ocemod.2017.02.006.

[35] A. Gholaminejad, K. Keutzer, and G. Biros. “ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. Vol. 2019-Augus. California: International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 730–736. DOI: 10.24963/ijcai.2019/103. arXiv: 1902.10298.

[36] S. Gosh, P. Birrell, and D. De Angelis. “Variational inference for nonlinear ordinary differential equations”. In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* 130.29 (2021), pp. 2719–2727.

[37] R. N. Gutenkunst et al. “Universally Sloppy Parameter Sensitivities in Systems Biology Models”. In: *PLoS Computational Biology* 3.10 (2007). Ed. by A. P. Arkin, e189. DOI: 10.1371/journal.pcbi.0030189.

[38] A. Hastings and T. Powell. “Chaos in a Three-Species Food Chain”. In: *Ecology* 72.3 (1991), pp. 896–903. DOI: 10.2307/1940591.

[39] A. Hastings et al. “Chaos in Ecology: Is Mother Nature a Strange Attractor?” In: *Annual Review of Ecology and Systematics* 24.1 (1993), pp. 1–33. DOI: 10.1146/annurev.es.24.110193.000245.

[40] S. I. Higgins, S. Scheiter, and M. Sankaran. “The stability of African savannas: insights from the indirect estimation of the parameters of a dynamic model”. In: *Ecology* 91.6 (2010), pp. 1682–1692. DOI: 10.1890/08-1368.1.

[41] C. Hsieh, C. Anderson, and G. Sugihara. “Extending Nonlinear Analysis to Short Ecological Time Series”. In: *The American Naturalist* 171.1 (2008), pp. 71–80. DOI: 10.1086/524202.

[42] J. Huisman and F. J. Weissing. “Biodiversity of plankton by species oscillations and chaos”. In: *Nature* 402.6760 (1999), pp. 407–410. DOI: 10.1038/46540.

[43] W. Jetz et al. “Essential biodiversity variables for mapping and monitoring species populations”. In: *Nature Ecology and Evolution* 3.4 (2019), pp. 539–551. DOI: 10.1038/s41559-019-0826-1.

[44] P. K Mogensen and A. N Riseth. “Optim: A mathematical optimization package for Julia”. In: *Journal of Open Source Software* 3.24 (2018), p. 615. DOI: 10.21105/joss.00615. arXiv: arXiv:1710.07708.

[45] K. Kashinath et al. “Physics-informed machine learning: Case studies for weather and climate modelling”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (2021). DOI: 10.1098/rsta.2020.0093.

[46] D. R. A. Kenneth P. Burnham and Model. *Model Selection and Multimodel Inference*. Ed. by K. P. Burnham and D. R. Anderson. New York, NY: Springer New York, 2002. DOI: 10.1007/b97636.

[47] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv:1412.6980* (2014), 15 pp. arXiv: 1412.6980.

[48] A. Klebanoff and A. Hastings. “Chaos in three species food chains”. In: *Journal of Mathematical Biology* 32.5 (1994), pp. 427–451. DOI: 10.1007/BF00160167.

[49] C. Kremen. “Managing ecosystem services: what do we need to know about their ecology?” In: *Ecology Letters* 8.5 (2005), pp. 468–479. DOI: 10.1111/j.1461-0248.2005.00751.x.

[50] A. N. Laubmeier et al. “From theory to experimental designQuantifying a trait-based theory of predator-prey dynamics”. In: *PLOS ONE* 13.4 (2018). Ed. by M. S. Crowther, e0195919. DOI: 10.1371/journal.pone.0195919.

[51] L. M. Lawson et al. “A data assimilation technique applied to a predator-prey model”. In: *Bulletin of Mathematical Biology* 57.4 (1995), pp. 593–617. DOI: 10.1007/BF02460785.

[52] R. Lignell et al. “Getting the right parameter values for models of the pelagic microbial food web”. In: *Limnology and Oceanography* 58.1 (2013), pp. 301–313. DOI: 10.4319/lo.2013.58.1.0301.

[53] D. C. Liu and J. Nocedal. “On the limited memory BFGS method for large scale optimization”. In: *Mathematical Programming* 45.1-3 (1989), pp. 503–528. DOI: 10.1007/BF01589116.

[54] Y. Ma et al. “A Comparison of Automatic Differentiation and Continuous Sensitivity Analysis for Derivatives of Differential Equation Solutions”. In: *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. 2. IEEE, 2021, pp. 1–9. DOI: 10.1109/HPEC49654.2021.9622796. arXiv: 1812.01892.

[55] N. M. Mangan et al. “Model selection for dynamical systems via sparse regression and information criteria”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2204 (2017), p. 20170009. DOI: 10.1098/rspa.2017.0009. arXiv: 1701.01773.

[56] R. J. Matear. “Parameter optimization and analysis of ecosystem models using simulated annealing: A case study at Station P”. In: *Journal of Marine Research* 53.4 (1995), pp. 571–607. DOI: 10.1357/0022240953213098.

[57] K. McCann and A. Hastings. “Reevaluating the omnivorystability relationship in food webs”. In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 264.1385 (1997), pp. 1249–1254. DOI: 10.1098/rspb.1997.0172.

[58] K. McCann and P. Yodzis. “Biological Conditions for Chaos in a Three-Species Food Chain”. In: *Ecology* 75.2 (1994), pp. 561–564. DOI: 10.2307/1939558.

[59] K. McCann and P. Yodzis. “Nonlinear Dynamics and Population Disappearances”. In: *The American Naturalist* 144.5 (1994), pp. 873–879. DOI: 10.1086/285714.

[60] G. Midgley and L. Hannah. “Extinction Risk from Climate Change”. In: *Biodiversity and Climate Change*. Yale University Press, 2019, pp. 294–296. DOI: 10.2307/j.ctv8jnzw1.37.

[61] J. Norberg et al. “Eco-evolutionary responses of biodiversity to climate change”. In: *Nature Climate Change* 2.10 (2012), pp. 747–751. DOI: 10.1038/nclimate1588.

[62] M. Pahlow et al. “Adaptive model of plankton dynamics for the North Atlantic”. In: *Progress in Oceanography* 76.2 (2008), pp. 151–191. DOI: 10.1016/j.pocean.2007.11.001.

[63] J. S. Pelc et al. “Application of model reduced 4D-Var to a 1D ecosystem model”. In: *Ocean Modelling* 57-58 (2012), pp. 43–58. DOI: 10.1016/j.ocemod.2012.09.003.

[64] G. C. Y. Peng et al. “Multiscale Modeling Meets Machine Learning: What Can We Learn?” In: *Archives of Computational Methods in Engineering* 28.3 (2021), pp. 1017–1037. DOI: 10.1007/s11831-020-09405-5. arXiv: 1911.11958.

[65] C. T. Perretti, G. Sugihara, and S. B. Munch. “Nonparametric forecasting outperforms parametric methods for a simulated multispecies system”. In: *Ecology* 94.4 (2013), pp. 794–800. DOI: 10.1890/12-0904.1.

[66] M. L. Pinsky et al. “Marine Taxa Track Local Climate Velocities”. In: *Science* 341.6151 (2013), pp. 1239–1242. DOI: 10.1126/science.1239352.

[67] V. F. Pisarenko and D. Sornette. “Statistical methods of parameter estimation for deterministically chaotic time series”. In: *Physical Review E* 69.3 (2004), p. 036122. DOI: 10.1103/PhysRevE.69.036122. arXiv: 0308059 [physics].

[68] D. M. Post, M. E. Conners, and D. S. Goldberg. “Prey preference by a top predator and the stability of linked food chains”. In: *Ecology* 81.1 (2000), pp. 8–14. DOI: 10.1890/0012-9658(2000)081[0008:PPBATP]2.0.CO;2.

[69] D. Purves et al. “Time to model all life on Earth”. In: *Nature* 493.7432 (2013), pp. 295–297. DOI: 10.1038/493295a.

[70] C. Rackauckas and Q. Nie. “DifferentialEquations.jl a performant and feature-rich ecosystem for solving differential equations in Julia”. In: *Journal of Open Research Software* 5 (2017). DOI: 10.5334/jors.151.

[71] C. Rackauckas et al. “Universal Differential Equations for Scientific Machine Learning”. In: (2020). arXiv: 2001.04385.

[72] M. Raissi, P. Perdikaris, and G. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (2019), pp. 686–707. DOI: 10.1016/j.jcp.2018.10.045.

[73] J. O. Ramsay et al. “Parameter estimation for differential equations: a generalized smoothing approach”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.5 (2007), pp. 741–796. DOI: 10.1111/j.1467-9868.2007.00610.x.

[74] A. Raue et al. “Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood”. In: *Bioinformatics* 25.15 (2009), pp. 1923–1929. DOI: 10.1093/bioinformatics/btp358.

[75] M. Rodriguez-Fernandez, P. Mendes, and J. R. Banga. “A hybrid approach for efficient and robust parameter estimation in biochemical pathways”. In: *Biosystems* 83.2-3 (2006), pp. 248–265. DOI: 10.1016/j.biosystems.2005.06.016.

[76] B. Rosenbaum et al. “Estimating Parameters From Multiple Time Series of Population Dynamics Using Bayesian Inference”. In: *Frontiers in Ecology and Evolution* 6.JAN (2019). DOI: 10.3389/fevo.2018.00234.

[77] S. Ruder. “An overview of gradient descent optimization algorithms”. In: (2016), pp. 1–14. arXiv: 1609.04747.

[78] K. M. Ruppert, R. J. Kline, and M. S. Rahman. “Past, present, and future perspectives of environmental DNA (eDNA) metabarcoding: A systematic review in methods, monitoring, and applications of global eDNA”. In: *Global Ecology and Conservation* 17 (2019), e00547. DOI: 10.1016/j.gecco.2019.e00547.

[79] M. Schartau et al. “Reviews and syntheses: parameter identification in marine planktonic ecosystem modelling”. In: *Biogeosciences* 14.6 (2017), pp. 1647–1701. DOI: 10.5194/bg-14-1647-2017.

[80] M. Scheffer et al. “Catastrophic shifts in ecosystems”. In: *Nature* 413.6856 (2001), pp. 591–596. DOI: 10.1038/35098000.

[81] S. Scheiter, L. Langan, and S. I. Higgins. “Nextgeneration dynamic global vegetation models: learning from community ecology”. In: *New Phytologist* 198.3 (2013), pp. 957–969. DOI: 10.1111/nph.12210.

[82] T. Schneider et al. “Earth System Modeling 2.0: A Blueprint for Models That Learn From Observations and Targeted HighResolution Simulations”. In: *Geophysical Research Letters* 44.24 (2017), pp. 12,396–12,417. DOI: 10.1002/2017GL076101. arXiv: 1709.00037.

[83] Y. Spitz et al. “Data assimilation and a pelagic ecosystem model: parameterization using time series observations”. In: *Journal of Marine Systems* 16.1-2 (1998), pp. 51–68. DOI: 10.1016/S0924-7963(97)00099-7.

[84] M. C. Urban et al. “Improving the forecast for biodiversity under climate change”. In: *Science* 353.6304 (2016). DOI: 10.1126/science.aad8466.

[85] A. Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[86] B. A. Ward et al. “Parameter optimisation techniques and the problem of underdetermination in marine biogeochemical models”. In: *Journal of Marine Systems* 81.1-2 (2010), pp. 34–43. DOI: 10.1016/j.jmarsys.2009.12.005.

[87] C. O. Wilke et al. “Evolution of digital organisms at high mutation rates leads to survival of the flattest”. In: *Nature* 412.6844 (2001), pp. 331–333. DOI: 10.1038/35085569.

[88] J. Willard et al. “Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems”. In: 1.1 (2020), pp. 1–35. arXiv: 2003.04919.

[89] Y. Xiao and M. A. M. Friedrichs. “The assimilation of satellitederived data into a onedimensional lower trophic level marine ecosystem model”. In: *Journal of Geophysical Research: Oceans* 119.4 (2014), pp. 2691–2712. DOI: 10.1002/2013JC009433.

[90] T. Xu et al. “Probabilistic inversion of a terrestrial ecosystem model: Analysis of uncertainty in parameter estimation and model prediction”. In: *Global Biogeochemical Cycles* 20.2 (2006), n/a–n/a. DOI: 10.1029/2005GB002468.

[91] A. Yazdani et al. “Systems biology informed deep learning for inferring parameters and hidden dynamics”. In: *PLOS Computational Biology* 16.11 (2020). Ed. by V. Hatzimanikatis, e1007575. DOI: 10.1371/journal.pcbi.1007575.

[92] H. Ye and G. Sugihara. “Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality”. In: *Science* 353.6302 (2016), pp. 922–925. DOI: 10.1126/science.aag0863.

[93] H. Ye et al. “Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling”. In: *Proceedings of the National Academy of Sciences* 112.13 (2015), E1569–E1576. DOI: 10.1073/pnas.1417063112.

[94] Q. Zhu and Q. Zhuang. “Ecosystem biogeochemistry model parameterization: Do more flux data result in a better model in predicting carbon flux?” In: *Ecosphere* 6.12 (2015), art283. DOI: 10.1890/ES15-00259.1.

4.A Supplementary Information

We show that the loss function $L_{\mathcal{M}}(\theta)$ in the main manuscript in Eq. (4.6) is ill-behaved for models with complex dynamics when the time horizon is large. We proceed by first analysing the dynamics of models with complex dynamics showing chaotic behaviour or limit cycles, and approximate the divergence of perturbed dynamical trajectories. We then show that the divergence in dynamics translates into a loss function whose surface is rugged in most of the parameter space, and that the gradient of the loss function around the true parameters becomes exponentially steeper with time. We conclude by formally discussing how the proposed mini-batch method regularizes the ill-behaviour of the loss function.

4.A.1 Dynamics under perturbations

Perturbed initial conditions

Consider the trajectory of the state variables

$$\begin{aligned} x(t) &= \mathcal{M}(t, p, x_0) \\ &= \int_0^t f(s, x(s), p) ds + x_0 \end{aligned} \tag{S1}$$

and consider the perturbed trajectory

$$x_{\delta x_0}(t) = \mathcal{M}(t, p, x_0 + \delta x_0) \tag{S2}$$

whose initial conditions (ICs) x_0 are perturbed by δx_0. Assuming that the system is chaotic and that δx_0 is small, the distance between the perturbed trajectory and the original one grows as

$$\|x(t) - x_{\delta x_0}(t)\| \sim e^{\lambda t} \delta x_0 \tag{S3}$$

where λ is the largest Lyapunov exponent of the system [5]. After enough time, the trajectories diverge so much that they effectively become independent samples of the phasespace: the trajectories forget their ICs, and ergodic theory ensures that the positions of $x(t)$ and $x_{\delta x_0}(t)$ are better described by a random variable $\mathcal{R}$ with probability density given by the density of orbits in the chaotic attractor, the so-called invariant measure of the chaotic attractor [2]. The distance between $x(t)$ and $x_{\delta x_0}(t)$ can therefore be described as

$$\|x(t) - x_{\delta x_0}(t)\| \sim \|\mathcal{R}_1 - \mathcal{R}_2\| \tag{S4}$$

Considering the observation function h, it follows that

$$\begin{aligned} \|h(x(t)) - h(x_{\delta x_0}(t))\| &\sim \left.\frac{\partial h}{\partial x}\right|_{x(t)} e^{\lambda t}\delta x_0 \text{ for } t \ll \frac{1}{\lambda} \\ \|h(x(t)) - h(x_{\delta x_0}(t))\| &\sim \|h(\mathcal{R}_1) - h(\mathcal{R}_2)\| \text{ for } t \gg \frac{1}{\lambda} \end{aligned} \tag{S5}$$

(see Fig. S2 for an illustration of the divergence behaviour over time).

Perturbed model parameters

Consider now a trajectory $x_{\delta p}$ with a small perturbation of the parameters δp . From Eqs. (S1) and (S2), it follows that

$$\begin{aligned} \dot{x}_{\delta p}(t) &= f(t, x_{\delta p}(t), p + \delta p) \\ &\sim f(t, x_{\delta p}(t), p) + \frac{\partial f(t, x_{\delta p}(t), p)}{\partial p}\delta p \\ &\sim f(t, x(t), p) + \frac{\partial f(t, x_{\delta p}(t), p)}{\partial p}\delta p + \frac{\partial f(t, x(t), p)}{\partial x(t)}\frac{\partial x(t)}{\partial p}\delta p \end{aligned} \tag{S6}$$

which is dominated by the first term under small values of δp, and is thus subject to chaotic dynamics. Similar to a perturbation of the ICs, the small perturbation δp generates a divergence in the dynamical trajectories that grows exponentially until they become uncorrelated. For small δp, the distance between the true and deviated trajectory can therefore be approximated as

$$\|x(t) - x_{\delta p}(t)\| \sim \int_0^t e^{\lambda(t-s)}\|x(s) - x_{\delta p}(s)\|ds \sim e^{\lambda t} g(\delta p) \tag{S7}$$

where $g(\delta p) = \|\frac{\partial f(t, x_{\delta p}(t), p)}{\partial p} + \frac{\partial f(t, x(t), p)}{\partial x(t)}\frac{\partial x(t)}{\partial p}\|$ gives the scale of the divergence between the two trajectories as a function of δp. Similar to a perturbation of the ICs, the difference in trajectories grows to the point where the trajectories become effectively independent after a long time. Hence, it follows that

$$\begin{aligned} \|h(x(t)) - h(x_{\delta p}(t))\| &\sim \left.\frac{\partial h}{\partial x}\right|_{x(t)} e^{\lambda t} g(\delta p) \text{ for } t \ll \frac{1}{\lambda} \\ \|h(x(t)) - h(x_{\delta p}(t))\| &\sim \|h(\mathcal{R}_1) - h(\mathcal{R}_2)\| \text{ for } t \gg \frac{1}{\lambda} \end{aligned} \tag{S8}$$

(see Fig. S3 for an illustration of the divergence behaviour over time).

In the following section, we call the first divergence regime the informative divergence regime, where the loss grows with the distance to the true parameters, and we call the second divergence regime the mixed divergence regime, where the loss

is dominated by the random-like behaviour. Given that δp and δx_0 behave similarly, we employ θ and $\delta\theta$ to encompass both perturbations of parameters and ICs, and denote by $g(\delta\theta)$ the function that gives the scale of the divergence in trajectories for both $g(\delta p)$ and $\|\delta x_0\|$.

Transition in the parameter space between the informative and the mixed regime

For a fixed time horizon t, and depending on the shape of the chaotic attractor, the magnitude of the perturbation determines the divergence regime. If the perturbation is small, the trajectories will be aligned, but for large perturbations they will effectively become two independent trajectories.

The transition between the two regimes can be studied by noting that the informative divergence should remain in the same order of magnitude as the mixed divergence. The reason is that the expected value of the squared divergence between two trajectories $x(t)$ and $x_{\delta\theta}(t)$ for large t is

$$\mathbb{E}\left[\|h\left(\mathcal{R}_1\right)-h\left(\mathcal{R}_2\right)\|^2\right]=2\mathrm{Var}\left[h\left(\mathcal{R}\right)\right]\lesssim\max\left|h(\mathcal{R})-\mathbb{E}\left[h(\mathcal{R})\right]\right|^2 \tag{S9}$$

meaning that the expected value of the squared divergence in the mixed regime is in the same order of magnitude as the maximum distance within the phasespace. On the other hand, the divergence of any two trajectories in the chaotic attractor cannot be larger than the maximum distance between two points in the attractor, which is itself bounded through the triangle inequality as $2\max\left|h(\mathcal{R})-\mathbb{E}\left[h(\mathcal{R})\right]\right|$.

Since the growth of the informative regime has to remain in the same order of magnitude as the mixed regime, at the regime transition we must have

$$e^{\lambda t}g(\delta\theta)\sim\mathbb{E}\left[\|h\left(\mathcal{R}\right)\|\right] \tag{S10}$$

Equation (S10) implies that, for a given time horizon t, the magnitude of the critical perturbation $\delta\theta^*$ associated with the regime transition satisfies

$$\|\delta\theta^*\|\sim e^{-\lambda t} \tag{S11}$$

Limit cycles

While in the section above a chaotic system was assumed to provide an approximation for the divergence of the trajectories, a similar approximation applies for

systems characterized by limit cycles. Considering a system $x(t)$ with a limit cycle characterized by the phase ωt with frequency ω, i.e.

$$x(t) = \mathfrak{f}(\phi(t)) = \mathfrak{f}(\omega t) \mod 2\pi \tag{S12}$$

a perturbation of the parameters δp might lead to a perturbed frequency $\delta\omega$, further leading to a difference in phases

$$\delta\phi(t) = \phi(t) - \phi_{\delta\omega}(t) = \delta\omega t \mod 2\pi \tag{S13}$$

For $t \lesssim \frac{1}{\delta\omega}$, $\delta\phi(t)$ grows linearly with $\delta\omega$, but once $t \gg \frac{1}{\delta\omega}$, the change of phase $\delta\phi(t)$ is affected by the modulo operation. As this operation is nonlinear, a small random perturbation $\delta\omega$ results in a random uniform phase over the interval $[0, 2\phi]$. For a large time horizon, $x_{\delta\omega}$ is thus uniformly spread over the circular line given by the phasespace of the dynamical system. Hence, the approximation in Eq. (S8) applies for cyclic dynamics, except that the initial divergence is linear rather than exponential, and that in contrast to Eq. (S5), a change affecting the initial position will not grow over time.

4.A.2 Consequences for the shape of the loss surface

The approximation of the divergence of trajectories in Eqs. (S5) and (S8), together with the transition boundary determined by Eq. (S11), can be used to characterize the surface associated with the loss function.

Omitting the term corresponding to the priors and the variance–covariance matrix Σ_y for simplicity, the loss function presented in the main text is expressed as

$$\begin{aligned} L_{\mathcal{M}}(\theta) &= \frac{1}{K}\sum_{k=1}^{K} ||y_k - h(\mathcal{M}(t_k, \theta))||^2 \\ &= \frac{1}{K}\sum_{k=1}^{K} ||h(\tilde{x}(t_k)) + \epsilon(t_k) - h(\mathcal{M}(t_k, \theta))||^2 \end{aligned} \tag{S14}$$

where the parameter vector θ is decomposed into the model parameter vector p and the ICs x_0, y_k correspond to the observations, $\tilde{x}$ corresponds to the true trajectory and $\epsilon(t)$ is the observational noise. As the noise is independent of the dynamics,

it is uncorrelated with $h(\tilde{x}(t_k)) - \mathcal{M}(t_k, \theta)$, meaning that the loss can be split in expectation

$$\mathbb{E}\left[L_{\mathcal{M}}(\theta)\right] = \frac{1}{K}\sum_{k=1}^{K} ||h(\tilde{x}(t_k)) - h(\mathcal{M}(t_k, \theta))||^2 + \mathrm{Var}\left[\epsilon\right] \tag{S15}$$

where the noise term $(\mathrm{Var}\left[\epsilon\right])$ is independent from the parameters θ. Assuming that $\theta = \tilde{\theta} + \delta\theta$, where $\tilde{\theta}$ correspond to the true parameters and ICs, every term in the sum corresponds to a squared distance between the true trajectory $\tilde{x}(t)$ and a perturbed trajectory $\tilde{x}_{\delta\theta}(t)$. Using Eqs. (S5) and (S8) to (S10) we obtain the loss function approximation

$$L_{\mathcal{M}}(\theta) \sim \sum_{k=1}^{\min\{K,K^*\}} e^{2\lambda t_k} g^2(\delta\theta) + \sum_{k=\min\{K,K^*\}+1}^{K} \mathrm{Var}\left[h(\mathcal{R})\right] + \mathrm{Var}\left[\epsilon\right] \tag{S16}$$

where K^* is the observation index corresponding to the time horizon t_{K^*}, where the transition between the informative and the mixed regime happens for the perturbation $\delta\theta$, obtained from Eq. (S11).

The distribution of the observation times and the magnitude of the perturbation $\delta\theta$ determine whether the loss is dominated by the informative or by the mixed divergence regime. For a fixed perturbation $\delta\theta$, assuming that the observations y_k are uniformly distributed over the time interval $[0, t]$, Eq. (S11) yields for $t \ll \frac{\log(|\delta\theta|)}{\lambda}$ that the loss $L_{\mathcal{M}}$ is dominated by the informative divergence regime, whereas if $t \gg \frac{\log(|\delta\theta|)}{\lambda}$ the loss is dominated by the mixed divergences. In the region where the loss is dominated by mixed divergences, the loss $L_{\mathcal{M}}$ has an expected value of order $\mathcal{O}\left(\mathrm{Var}\left[h(\mathcal{R})\right] + \mathrm{Var}\left[\epsilon\right]\right)$ and does not grow monotonically with $\delta\theta$. It corresponds to a "random-like" surface populated with local minima, and is consequently characterized by an uninformative gradient preventing local optimizers from converging to the true parameters $\tilde{\theta}$ (see Fig. 4.1B, orange dashed curve). On the other hand, in the region where the divergence in trajectories belongs to the informative regime, the loss is convex and grows with $\delta\theta$. Its associated gradient $\nabla_\theta L_{\mathcal{M}}(\theta) \sim \sum_{k=1}^{K} e^{2\lambda t_k} \nabla_\theta g^2(\delta\theta)$ consequently contains relevant information for the use of variational optimizers, but the loss surface becomes exponentially steeper as the time horizon increases. As shown by the green dotted curve in Fig. 4.1B, this large gradient in the vicinity of the optimal parameters is likely to lead gradient-based optimizers to overshoot and not converge to the true parameters $\tilde{\theta}$. Eq. (S11) further indicates that the volume of the region in the parameter space where the loss is informative shrinks exponentially as the time horizon increases, implying that for large time horizons, the uninformative region is predominant.

4.A.3 Regularizing the loss surface with mini-batches

To prevent the situation with a mixed divergence regime and to decrease the gradient in a vicinity of the true parameters, we reformulate the loss function as the average of loss functions defined over mini-batches of short time horizons. In the following section, we compare the properties of the naive loss function $L_{\mathcal{M}}$ in Eq. (S15) with the mini-batch loss function $L^{\star}_{\mathcal{M}}$ proposed in Eq. (4.8) in the main manuscript, and further discuss the limitations of the method in the presence of noise.

Omitting the term corresponding to the priors and ICs and the noise shape Σ_y, and assuming a single time series for simplicity, the mini-batch loss function presented in the main text can be expressed as

$$L^{\star}_{\mathcal{M}}(\theta) = \frac{1}{M}\sum_{m=0}^{M-1}\frac{1}{K}\sum_{k=1}^{K^{(m)}} \| y_{k+mK/M} - h\left(\mathcal{M}(t_{k+mK/M}, p, x_0^{(m)})\right) \|^2 \tag{S17}$$

where M is the number of mini-batches, $x_0^{(m)}$ corresponds to the ICs for mini-batch m inferred at time $t_{mK/M}$, and $K^{(m)} = \begin{cases} K/M+1 & : m < M \\ K/M & : m = M \end{cases}$ is the number of data points in the mth batch. The loss function can be split in expectation as

$$\mathbb{E}\left[L^{\star}_{\mathcal{M}}(\theta)\right] = \frac{1}{M}\sum_{m=0}^{M-1}\frac{1}{K}\sum_{k=1}^{K^{(m)}} \|h(\tilde{x}(t_{k+mK/M})) - \mathcal{M}(t_{k+mK/M}, p, x_0^{(m)})\|^2 + \mathrm{Var}\left[\epsilon\right] \tag{S18}$$

Assuming that $x_0^{(m)} = \tilde{x}(t_{mK/M}) + \delta x_0^{(m)}$ when $\delta x_0^{(m)}$ is small, and assuming that the observation times are regularly spaced so that $t_{k+1} - t_k = \Delta t$, the time length of simulated trajectories (i.e. the time elapsed between the time when the perturbation is applied and the end time of the simulated trajectory) is divided by the number of mini-batches, in comparison to the time length of the simulated trajectory in Eq. (S15) ($t = t_{K/M+1} - t_0 = K/M\Delta t$ in Eq. (S18), in comparison to $t = K\Delta t$ in Eq. (S15)). Further assuming that $K\Delta t >> \frac{\log(|\delta\theta|)}{\lambda}$ and choosing the number of mini-batches M so that $K/M\Delta t << \frac{\log(|\delta\theta|)}{\lambda}$, we apply the approximation Eq. (S16), which leads to

$$\begin{aligned} L_{\mathcal{M}}(\theta) &\sim \mathrm{Var}\left[\epsilon\right] + \mathrm{Var}\left[h\left(\mathcal{R}\right)\right] \\ L^{\star}_{\mathcal{M}}(\theta) &\sim \mathrm{Var}\left[\epsilon\right] + g^2(\delta\theta)\sum_{m=0}^{M-1} e^{2\lambda(K/M\Delta t)} \end{aligned} \tag{S19}$$

While $L_{\mathcal{M}}$ is dominated by the mixed regime, $L^{\star}_{\mathcal{M}}$ is dominated by the informative regime because the simulation time remains small, permitting the successful use of variational optimizers.

The number of mini-batches M should be determined by considering the dynamical behaviour of the system and the level of noise in the observation, because a large number of mini-batches M smooths out the loss surface but also entails more sensitivity to the level of noise. Indeed, in $L_{\mathcal{M}}(\theta)^{\star}$ the relative effect of the second term corresponding to the observational noise ϵ increases when the number of mini-batches M increases. The value of M should therefore be chosen wisely to balance the benefits of mini-batches, i.e. widening the region of the parameter space where $L_{\mathcal{M}}$ is well behaved and reducing the overshooting problem, and their cons, i.e. their tendency to increase the importance of noise.

4.B Three-compartment food-web models

4.B.1 Reference food-web model

We used the three-species chaotic food-web model from [1], formulated as

$$\begin{aligned}
\tfrac{d}{dt}R &= R(1-R) - x_C y_C \frac{CR}{R+R_0} \\
\tfrac{d}{dt}C &= x_C C\left[-1 + x_C \frac{R}{R+R_0}\right] - x_P y_P \frac{PC}{C+C_0} \\
\tfrac{d}{dt}P &= x_P P\left[-1 + y_P \frac{C}{C+C_0}\right],
\end{aligned} \tag{S20}$$

with the biologically realistic parameter values $x_C = 0.4$, $0.071 \leq x_P \leq 0.225$, $y_C = 2.01$, $y_P = 5$, $R_0 = 0.16129$, and $C_0 = 0.5$ [4]. The dynamics of the system are chaotic for this set of parameter values.

4.B.2 Omnivory variant food-web model

We used the three-species food-web model from [3], formulated as

$$\begin{aligned}
\tfrac{d}{dt}R &= R(1-R) - x_C x_C \frac{CR}{R+R_0} - \omega x_P y_{PR} \frac{PR}{R_{02} + (1-\omega)C + \omega R} \\
\tfrac{d}{dt}C &= x_C C\left[-1 + x_C \frac{R}{R+R_0}\right] - (1-\omega) x_P y_{PC} \frac{PC}{\omega R + (1-\omega)C + C_0} \\
\tfrac{d}{dt}P &= x_P P\left[-1 + (1-\omega) x_P y_{PC} \frac{C}{\omega R + (1-\omega)C + C_0} + \omega x_P y_{PR} \frac{R}{\omega R + (1-\omega)C + R_{02}}\right]
\end{aligned} \tag{S21}$$

with the biologically realistic parameter values $x_C = 0.4$, $x_P = 0.08$, $y_C = 2.009$, $y_{PR} = 2$, $y_{PC} = 5$, $R_0 = 0.16129$, $C_0 = 0.5$, and $0 \leq \omega \leq 0.5$. For this set of parameter values, the dynamics of the system are chaotic for $\omega \lesssim 0.20$, consist of a limit cycle for $0.20 \lesssim \omega \lesssim 0.35$, and consist of dampened oscillations for $0.35 \lesssim \omega$.

References

[1] A. Hastings and T. Powell. "Chaos in a Three-Species Food Chain". In: *Ecology* 72.3 (1991), pp. 896–903. DOI: 10.2307/1940591.

[2] J. Jost. *Dynamical systems: examples of complex behaviour*. Springer Science & Business Media, 2005.

[3] K. McCann and A. Hastings. "Reevaluating the omnivorystability relationship in food webs". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 264.1385 (1997), pp. 1249–1254. DOI: 10.1098/rspb.1997.0172.

[4] K. McCann and P. Yodzis. "Biological Conditions for Chaos in a Three-Species Food Chain". In: *Ecology* 75.2 (1994), pp. 561–564. DOI: 10.2307/1939558.

[5] S. H. Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

4.C Supplementary Figures

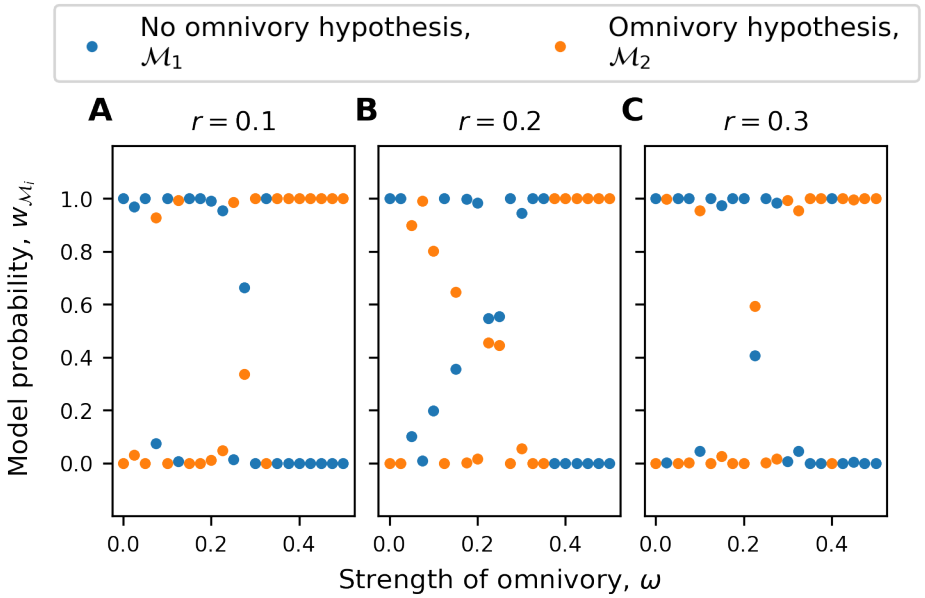

Fig. S1: Performance of the ML framework in supporting the predator omnivory hypothesis in a food web for the partial observation setting. In **A**, **B** and **C** where $r = 0.1, 0.2, 0.3$, the lack of data prevents the correct estimation of the omnivory variant model parameters, leading model $\mathcal{M}_1$ to be supported for a wider range of ω in contrast to the complete observation setting (Fig. 4.5A).

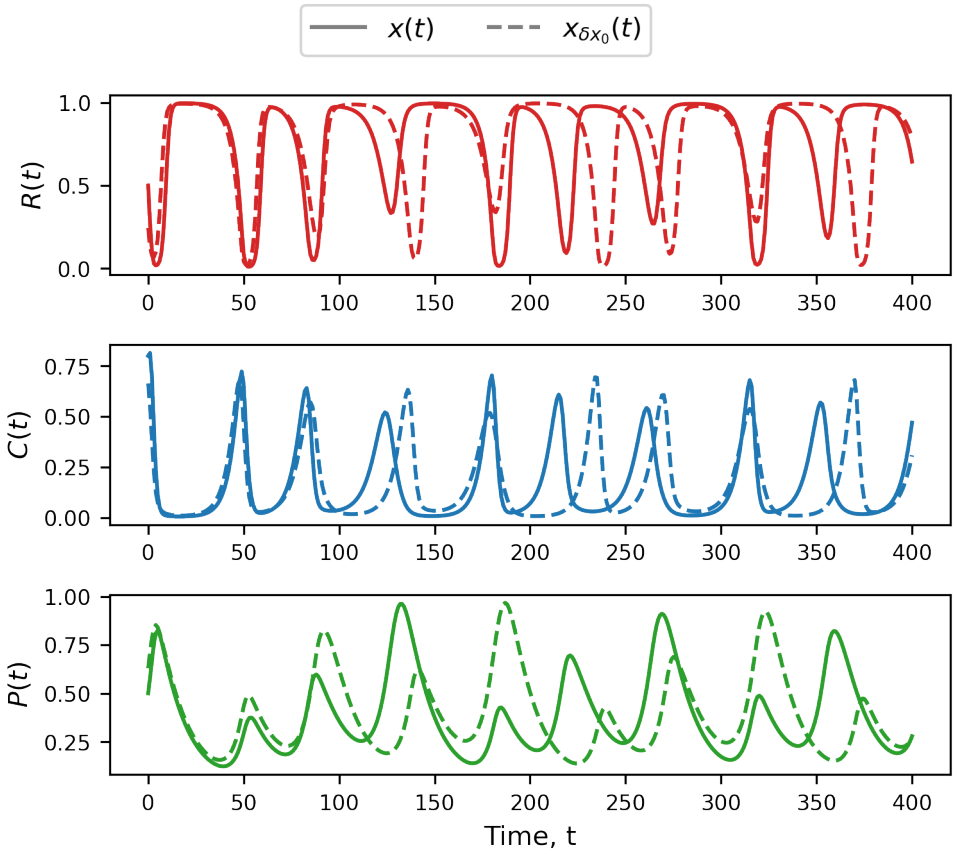

Fig. S2: Divergence between the trajectory $x(t)$ and a perturbed trajectory $x_{\delta x_0}(t)$, obtained from the reference food-web model from [1] and detailed in Section 4.B. For $t \lesssim 100$, $x(t)$ and $x_{\delta x_0}(t)$ are correlated and the divergence regime is informative, but for $t \gtrsim 100$ the trajectories become essentially uncorrelated, corresponding to the mixed divergence regime.

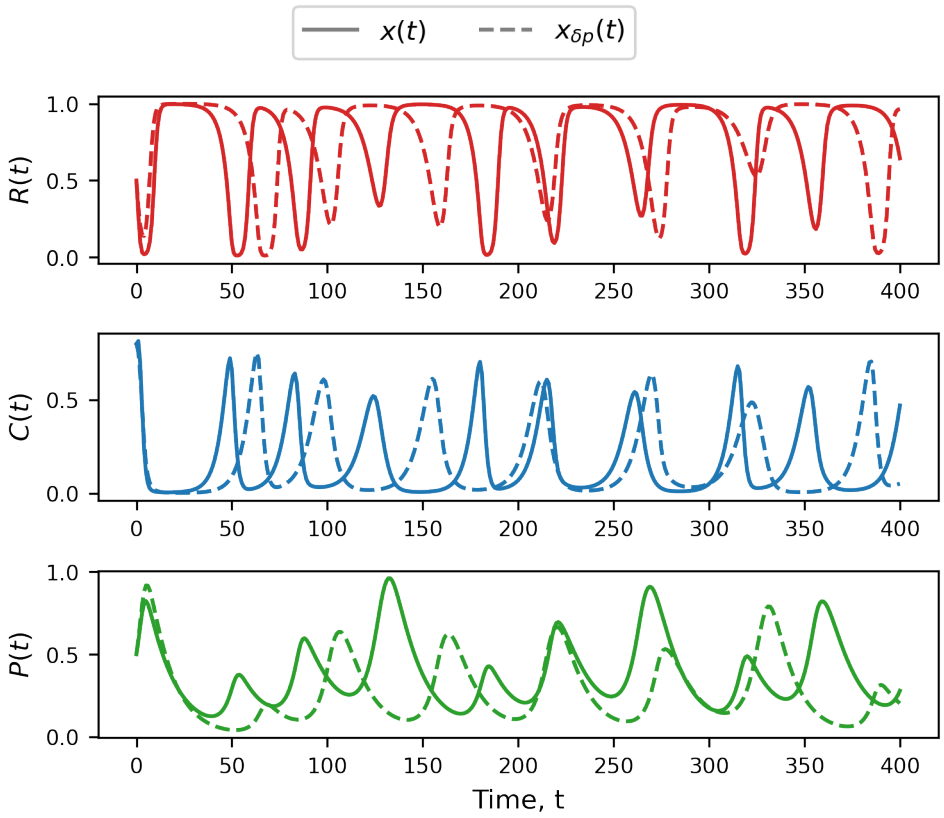

Fig. S3: **Divergence between the trajectory $x(t)$ and a perturbed trajectory $x_{\delta p}(t)$, obtained from the reference food-web model from [1] and detailed in Section 4.B.** For $t \lesssim 40$, $x(t)$ and $x_{\delta p}(t)$ are correlated and the divergence regime is informative, but for $t \gtrsim 40$ the trajectories become essentially uncorrelated, corresponding to the mixed divergence regime.

4.D Supplementary Tables

Setting	Median simulation time	Mean simulation time	Std. simulation time
Complete observations	35.9977436	39.5980174	20.5789433
Partial observations	34.1293998	39.1896534	21.8191709

Tab. S1: Simulation time for the complete and partial observation settings.

5

Econobiology: quantifying interactions and evolution in economic systems

6 Discussion

6.0.1 Limitation of PDE methods

[**Akesson2021**] : PDE methods are probably not as adapted as trait based ODEs. Those simpler models can already address important questions regarding climate change.

6.1 Conclusion

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Zürich, July 26, 2022

Victor Boussange