



Scalable agent-based models for optimized policy design: applications to the economics of biodiversity and carbon

Sharan S. Agrawal

August 2023

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500
<https://www.cl.cam.ac.uk/>

© 2023 Sharan S. Agrawal

The following report is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence:

<https://creativecommons.org/licenses/by/4.0/>

UNIVERSITY OF CAMBRIDGE and the Coat of Arms are registered trade marks of The Chancellor, Masters, and Scholars of the University of Cambridge (“University Marks”). For the avoidance of doubt, the University Marks are not included in this Creative Commons licence.

This technical report is based on a dissertation submitted June 2023 by the author for the degree of Master of Philosophy (Advanced Computer Science) to the University of Cambridge, Darwin College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

<https://www.cl.cam.ac.uk/techreports/>

ISSN 1476-2986

Abstract

As the world faces twinned crises of climate change and biodiversity loss, the need for integrated policy approaches addressing both is paramount. To help address this, a new agent-based model (ABM), the VDSK-B, was developed. Using Dasgupta’s review of the economics of biodiversity [1], it builds on the Dystopian Schumpeter meets Keynes (DSK) climate economics model [2] to link together the climate, economy and biosphere. To our knowledge, this is the first ABM proposed that integrates all 3 key elements.

Existing ABM frameworks struggled with global policy design needs due to their inability to scale to planetary-sized models, and optimize model parameters at the large scales needed for policy design. A new ABM framework called SalVO was built using a formalism for ABMs that expressed agent updates as recursive applications of pure agent functions. This formalism differs from existing computational ABM models but is shown to be expressive enough to emulate a Turing complete language. SalVO is built on a JAX backend and designed to be scalable, vectorized, and optimizable. Employing hardware acceleration, tests showed it was more performant and more able to scale on a single machine than any existing ABM framework, such as FLAME (GPU) [3].

Techniques for using backpropagation to create optimized policies differentiable, deterministic ABMs were further extended and implemented in SalVO. A novel protocol, GP-ABM, using William’s REINFORCE algorithm [4], was developed to optimize parameters in non-differentiable, stochastic ABMs. Both approaches are shown to be able to optimize ABMs for thousands of parameters, with backpropagation learning a highly non-trivial policy to move the centroid of a flock to a target location. This represents an innovation over current state-of-the-art techniques, such as Simulated Minimum Distance, which do not scale past fifty at most.

Finally, the VDSK-B model was implemented in SalVO, showing its capability of expressing highly complex ABMs. SalVO proved to be highly scalable, running a 5x bigger version of VDSK-B using just 4% of the time taken by the current open-source implementation, significantly strengthening its position as a preferred tool for large-scale ABM studies. While further work remains to be done on VDSK-B’s calibration and correctness, SalVO’s marriage of speed, scale and optimization has the potential to reshape how we approach, design, and apply agent-based models.

Acknowledgements

I would like to thank my supervisors Anil Madhavapeddy and Srinivasan Keshav for their guidance, patience, advice, ideas and time they kindly gave throughout the year. I'd like to thank them for introducing me to such a wonderful and impactful field of study and for their enthusiasm and vision for the work we conducted; I look forward to the work yet to be done.

Contents

1	Introduction	7
1.1	Thesis	7
1.2	Background	7
1.3	The importance of simulation in policy design	8
1.4	Agent-based models, and the economics of biodiversity	8
1.5	Summary of research	9
2	Gaps in existing research	11
2.1	Biodiversity modelling and ecological economics	11
2.1.1	ABMs in climate economics	11
2.2	Agent based model simulation frameworks	13
2.2.1	Comparable ABM computation frameworks	13
2.2.2	SalVO’s design strengths	14
2.2.3	Learning in ABMs	15
3	SalVO: Scalable, vectorized, and optimizable agent based model simulations	17
3.1	Introduction	17
3.2	ABMs formalized	18
3.2.1	SalVO ABM formalism	18
3.2.2	Shared state and dependency graphs	19
3.2.3	Emulating a Turing complete language	20
3.3	Achieving scale	20
3.3.1	Vectorization and hardware acceleration	21
3.4	Huge scale parameter optimisation and calibration	21
3.4.1	Parameter calibration with backpropagation	21
3.4.2	Policy gradient-based parameter calibration	22
3.4.3	GP-ABM	24
3.5	SalVO architecture and implementation	25
3.5.1	Design requirements	25
3.5.2	Platform	26
3.5.3	Core architecture and implementation	26
3.5.4	Example ABM implementations	29
4	Background in ecological economics	31
4.1	Review of the Dasgupta framework	31
4.1.1	Introduction	31
4.1.2	Summary of model	32
4.1.3	Criticisms	33
4.2	Review of the Dystopian Schumpeter meeting Keynes (DSK) model	33

5	The Vectorized Dystopian Schumpeter meeting Keynes, with Biodiversity (VDSK-B) Model	35
5.1	Filling in the blanks: vectorizing the DSK model	35
5.1.1	Capital good firms	35
5.1.2	Consumption good firms	36
5.1.3	Labour market	37
5.1.4	Energy industry	38
5.1.5	Climate box	38
5.1.6	Imitation process	39
5.2	Incorporating Dasgupta into VDSK	40
5.2.1	Modifications to incorporate Dasgupta	40
6	Evaluation	45
6.1	VDSK-B	45
6.1.1	Implementation	45
6.1.2	Speed and scale	46
6.2	SalVO evaluation	46
6.2.1	Speed and hardware acceleration	46
6.2.2	Learning	47
7	Discussion, conclusion and future work	50
7.1	Discussion and criticism	50
7.1.1	VDSK-B	50
7.1.2	SalVO	50
7.2	Conclusion	52
7.3	Future work	53
A	The Dystopian Schumpeter meeting Keynes (DSK) Model	63
A.1	Capital good firms	63
A.2	Consumption good firms	64
A.3	Labour	65
A.4	The banking sector	66
A.5	Energy industry	66
A.6	Climate box	67
B	Criticisms of and issues with the DSK model	69
C	Dasgupta on the economics of biodiversity	73
C.0.1	Introduction	73
C.0.2	The Impact Inequality	74
C.0.3	The model	75
C.0.4	Total Factor Productivity (TFP)	76
C.0.5	Human impact on the biosphere	78
C.0.6	The human population	78
C.0.7	Human capital, produced capital, innovation and investment	79
C.0.8	Summarizing assumptions	80
C.0.9	The Utility Maximization Problem	81
D	Additional Future Work	82
D.1	ABMs as timely data flow	82
D.2	Extensions to VDSK-B	82

Chapter 1

Introduction

1.1 Thesis

Integrated global policies addressing both biodiversity and climate change are desperately needed, and agent-based models (ABMs) are a powerful instrument to understand the effect of such policies on the world. Such ABMs will be highly complex and huge, and better simulation tools are needed for policymakers to express the model in a way that is easy to scale, fast, painless to calibrate, and optimizable.

1.2 Background

Our world is facing several simultaneous existential threats, including climate change and the decimation of global biodiversity [5]. So terrible has human impact been on our biosphere, that human activity is said to have triggered the sixth major extinction event in the history of our planet [6]. The implications of climate change and biodiversity loss are profound; we rely on the ecosystem for everything from fundamental biogeochemical phenomena, to raw material provision and agriculture, to regulating and minimizing extreme weather events [7].

While scientific advances are critical in addressing these challenges, transformations in the aggregate behaviour of society will be the defining factor in whether humanity can meet these challenges [8]. Having already exceeded the safe operating space for three out of nine identified planetary boundaries [5], that of climate change, biodiversity loss, and changes in the nitrogen cycle, the need for global policy changes capable of *simultaneously* addressing the multiple challenges we face has never been more pressing. Global calls for an **integrated** policy response towards both biodiversity and climate change [9] recognize the crucial interrelation of the two challenges and that they cannot be effectively addressed alone [10].

Ultimately humanity's advancement has come from treating biosphere services as “free“ [1]. The biosphere cannot be treated as external to humanity's progress; it must be treated as part of our collective wealth. Economic and social policies must incentivise changes in societal behaviours to reflect the value placed in our biosphere.

1.3 The importance of simulation in policy design

Global policymaking is extraordinarily hard: aside from geopolitical frictions, the systems we attempt to control are tremendously complex with many moving parts, and there is no guarantee the policies we design will work as intended. Policies are even harder to change once made; policies made now will have a lasting impact on our future. Consequently, while modelling these systems is daunting, and can be misleading, it is also essential in providing an environment where hypotheses can be tested. To paraphrase George E. P. Box no model is correct, but some are useful. A useful and understandable model and a powerful framework to express and optimize it, when carefully used, is an invaluable aid to policy making [11].

The field of ecological economics recognizes the trade-off inherent between economic growth, and ecological costs [12, 13]. Common modelling approaches use Integrated Assessment Models (IAM) [14], often with Dynamic Stochastic General Equilibrium (DSGE) ¹ models (e.g. [15]) for economic components. However, well-documented limitations in DSGE economic models exist, based on its strong assumptions of rationality, representative agents, and, most importantly, equilibrium [16] [17]; these challenges are by definition putting society into disequilibrium. A promising alternative approach are agent based models which don't need assumptions of rational individuals and can capture environments whose equilibrium is in flux and whose dynamics are immensely complex; ABMs are consequently widely recognized as powerful alternatives to DSGE models for climate economics [18] [17].

1.4 Agent-based models, and the economics of biodiversity

ABMs are digital laboratories in which you can test hypotheses, capture emergent and highly non-linear phenomena, and design interventions. By specifying the behaviour of individual agents and the rules of an environment in which they interact, ABMs allow the study of complex systems with emergent phenomena. However, ABMs are only as useful as their parameters; powerful calibration and optimisation of ABM parameters are needed to create better ABMs and derive more insights. As shown in §2, current ABM frameworks fall short of the scales and utility needed for policy design at the scales discussed here.

This research proposes extensions to ABM simulation theory that are better adapted to the massive-scale simulations needed to understand macroeconomic emergent phenomena. It develops calibration and optimisation tools rooted in machine learning techniques to develop possible economic policies to incentivize better aggregate behaviour to address these challenges. It uses techniques from neural network training and multi-agent reinforcement learning (MARL) to derive powerful methods to optimize and calibrate ABM parameters. The research implements a new Python-based library called SalVO (§3) in which massive ABMs can be built and easily deployed across large networks of heterogeneous devices, and use hardware acceleration effectively to achieve scale and efficiency. It can also leverage the learning frameworks derived to easily calibrate ABM parameters

¹DSGE models are a macroeconomic framework that is built around modelling demand and supply by assuming what "representative" agents of the economy (households, firms, etc.) and their utilities look like. It uses the assumption that the interaction between demand and supply will create an overall equilibrium.

and create optimized policies.

To start applying SalVO effectively to the simulation and optimisation of societal behaviours in the context of climate challenges, this work will also start operationalizing influential work by Dasgupta [1] on the economics of biodiversity. Dasgupta provides a view into the economics of biodiversity viewed from a traditional economic perspective, with equilibrium dynamics and rational agents forming the bedrock of the analysis but with an addition of natural capital into the mix. Traditionally economics treats biosphere services, such as climate regulation, as free services. The Dasgupta model treats it as another capital factor that forms the wealth of society. If society maximizes wealth over time in a traditional capitalist setting, it must also seek to maintain or improve natural capital, rather than degrade it. This influential perspective on the biosphere's role in the economy and society is currently based on a very theoretical view of the economy. This work will propose modifications of existing influential ABMs modelling climate economics to incorporate the key elements of the framework Dasgupta proposed. This has the key benefit of looking at the model from a disequilibrium, agent-based perspective, as well as producing simulation data that can be analysed for insight.

In summary, this work aims to achieve two key outcomes:

1. The development of a new ABM framework that has better performance and scalability, and allows for huge-scale parameter learning, optimized policy design, and ABM calibration.
2. The proposal and implementation in SalVO of a new ABM that captures the inter-related dynamics of society, the climate, and the biosphere in ways in which their study and policy design are useful. It will be based on operationalizing the Dasgupta report [1].

Ultimately, the quantity we are concerned with is the ability of our earth to sustain life and human enterprise. Biodiversity is often defined as the variety of different forms of life seen on our planet, and it isn't linked to the ability of our biosphere to sustain life. However, many studies have found that biodiversity is one of the most critical factors in our biosphere's ability to sustain itself and withstand stresses and shocks [19, 20, 21]. Therefore, when we refer to biodiversity, we use it interchangeably as both the variety in forms of life and the biosphere's ability to sustain life.

1.5 Summary of research

This work addresses three key areas which have either not been addressed, or in which better solutions can be created:

1. Biodiversity: no existing ABM marries biodiversity and the biosphere at a planetary scale with climate change and the economy.
 - Using a review of the principal components of the Dasgupta report [1] in §4, §5 integrates the key insights into a proposed ABM called the VDSK-B ² model, built to take advantage of the benefits offered by SalVO.
 - While VDSK-B is proposed and implemented, it is not fully operationalized due to the immense amount of research and calibration needed to actually match the biodiversity mechanisms with reality, beyond the scope of this report.

²Vectorized Dystopian Schumpeter meets Keynes, with Biodiversity

- By operationalizing the Dasgupta framework within an existing ABM of climate economics, we allow the design of integrated policies addressing both climate change and biodiversity loss.
2. ABM scalability: no existing ABM framework allows for the speed and easy scalability (defined more precisely in §3) achieved by SalVO.
 - An expressive formalism of ABMs allows us to build a computational model to easily scale ABMs across large networks of heterogeneous devices in a way that takes advantage of hardware acceleration using vectorization.
 - SalVO, presented in §3, implements this framework as a Python library, and §6 presents benchmarks of common ABMs vs. existing frameworks.
 3. ABM trainability: we develop a novel methodology to train ABMs using backpropagation and MARL (§3.4.3).
 - This ABM formalism also allows us to develop tools to develop optimized policies in the underlying ABMs.
 - The theory and implementation in SalVO are described in §3, and empirical tests are summarized in §6.

Chapter 2

Gaps in existing research

2.1 Biodiversity modelling and ecological economics

2.1.1 ABMs in climate economics

ABMs have already found a home in ecological economics simulations, from a farmer's decision-making affecting nitrogen run-off and water quality [22] and agroforestry [23], to emissions trading [24] and interactions of climate and monetary policy [25]. Since both ecology and economics involve dynamics that emerge from the interaction of millions or more heterogeneous participants whose combined (not necessarily equilibrium) interactions determine the overall state, ABMs have found widespread recognition as a useful modelling tool [26, 27].

However, while extensive studies of both the general and climate policy-specific studies of the interactions of the climate and economy exist, no ABM integrating the study of biodiversity as well into the climate/economic models at a global scale has been created. Existing approaches tend to focus on specific elements of the economy and its interaction with the biosphere, such as fisheries [28], green energy transitions [29], forestry [30], or the use of bioproducts [31]. A framework for describing broad-based interactions of the economy with the biosphere has been missing until the recent work by Dasgupta [1] on the economics of biodiversity. While key influential ABMs like ENGAGE [32], DSK [2] and *erin* [33] provide views into the effect of climate policies on the economy and climate through CO₂ emissions and global temperatures, and the extensive survey by Hardt [27] details several ABMs studying different interactions with economic agents and climate policies, none specifically capture the interaction between the economy and the biosphere.

Recent studies have shown that current policy is insufficient at preventing the loss of biodiversity, [34] and approaches such as market-based solutions are inadequate [35]. Integrating biodiversity into such ABMs is useful to help examine what policies can be introduced, either regulatory or market-based, to better protect biodiversity and the biosphere.

Rather than implementing an ABM capturing climate dynamics, the global economy, and the biosphere from scratch, we will build on an existing, influential ABM with peer-reviewed and accepted methodology. To determine which ABM is most appropriate, we carried out a gap analysis on some of the most influential and relevant ABMs in literature in table 2.1.

The categories being assessed include whether it captures the key state variables in the

Dasgupta framework, including the biosphere stock (some measure of the biosphere, e.g. the carbon stock it contains), human capital (changes in human populations and our earnings/ability to consume), technological innovation and productivity, produced capital (e.g. roads, machinery, infrastructure, etc.), and natural resource extraction (e.g. mining, forestry). We also look at whether market-based solutions, such as carbon markets, are included. Finally, we look for a model for biosphere impact, some measure of how the economy/climate affects biosphere dynamics, and the inclusion of emissions from economic activity into a dynamically updating climate model. It also includes a qualitative assessment of model features, such as how easy the model is to work with and extend, how configurable the simulation is, and whether the framework is modular.

		Model Biosphere Stock (S)	Model Human Capital (H)	Model Productivity (A)	Model Produced Goods (K)	Model Biosphere Services (R)	Model for Carbon Markets	Model for Biosphere Impact	Carbon Flux from Economic Activity	Simulation Configurable	Model Extensible	Framework Modular	Open source implementation
Macroeconomic / Climate Policy ABM	DSK [2]	✓	✓	✓	✓			✓	✓		✓	✓	✓
	ENGAGE [32]		✓	✓	✓							✓	✓
	ϵ -rin [33]		✓	✓	✓		✓		✓				
	MADIAMS [36]		✓	✓	✓								
	Lagom RegiO [37]		✓	✓	✓								
	Stock Flow [38]	✓	✓		✓	✓		✓	✓				
Carbon Credit Based ABM	CETS [39]				✓		✓		✓	✓			
	Auctions [40]				✓		✓		✓	✓			
	Andrade et al [41]			✓	✓		✓		✓	✓			
Biodiversity Focused ABM	Ng et al [22]	✓					✓						
	Noeldeke [23]	✓					✓						

Table 2.1: Gap analysis of existing ABMs vs. requirements for ABM being developed

Out of the surveyed models, the Dystopian Schumpeter meets Keynes (DSK) model [2] was the obvious candidate for extension. Not only did it capture a significant number of features needed for the Dasgupta model, it also was structured in a modular, extensible manner, and had an open-source implementation available. Other ABMs, such as the Stock Flow Fund Model [38] captured a number of important features but were not well structured and hard to work with, with no clear logical separation between the different components, and lacking an open source implementation.

2.2 Agent based model simulation frameworks

2.2.1 Comparable ABM computation frameworks

During the implementation of the proposed ABM, it became apparent that existing ABM frameworks were lacking in several important areas including:

- Scale: planetary-sized ABMs, as needed for global climate policy, will need an ABM framework that is easily distributed across a network of devices. An example of such a huge ABM is the EURACE ABM of the European economy [42].
- Hardware acceleration/speed: Frameworks running on the CPU on a single thread such as Mesa [43] will never be performant for large ABMs, and hardware acceleration is needed for fast simulations on large, complex models.
- Usability: Frameworks like FLAME [44] permit hardware acceleration but have a steep learning curve and complex DSL, and are difficult to debug, making developing truly huge models hard.
- Calibration: No framework found natively offers calibration as part of its toolkit. Calibration is widely recognized as one of the key challenges to ABM use [18], and solutions should be part of the framework.
- Creating optimized policies: No framework also allows for the automated design of optimal policies using global loss functions. We see this as a critical requirement for any ABM framework used for global policy design.

Existing ABM frameworks are diverse in their abilities. Several ABM frameworks aim at being usable and easy to develop in, but not performant. For instance, Mesa [43] is a very popular Python library for ABMs, built with the principle of being natively in Python. It is extremely usable, built by defining agent types as classes that contain within them a set of agents with state. However, computation happens via a single process on the CPU (bound by Python’s GIL), and it is consequently extremely slow and hard to scale [45]. MASON [46] and NetLogo [47] were among the earliest general-purpose ABM frameworks that were developed in Java. However, NetLogo, built with simplicity in mind, can’t handle complex ABMs, and MASON requires significant expertise and familiarity with a complex DSL to implement, and still suffers from poor performance due to the lack of hardware acceleration [45]. Repast [48] is a framework originally built in Java built on a tick-based event framework with complex scheduling that is very expressive, but again lacks scalability from distributional capabilities and is slow and hard to implement due to its dependence on Java. Several smaller frameworks such as ABCE [49] exist which are more specialized and field-specific, but still lack inbuilt scalability, and the use of hardware acceleration.

Several other frameworks attempt to be more performance and distributable. For instance, Agents.jl [50] is built in Julia and takes advantage of the massive speedup offered compared to languages like Python to achieve high speeds with little sacrifice on usability [45]. However, it still lacks the ability to take advantage of hardware acceleration and offers no easy way to distribute/scale. FLAME [44] is a powerful framework built in C++ using an X-Machine abstraction for its agents, turning them into state machines with state transitions, input/output messages etc. It uses MPI for message passing between agent groups and a common message boarding system and intelligent distribution strategies (for instance distributing agent sets who are neighbours in the simulation on adjacent nodes) to achieve scale. It also natively implements hardware acceleration using GPUs [3], where

a carefully written program can be compiled into CUDA kernel for GPU acceleration. However, there are several drawbacks to FLAME, found through experimentation. It is very hard to use, with debugging being difficult and scaling poorly with larger programs. It also uses a custom DSL with a steep learning curve and very careful specifications needed for hardware acceleration. Scaling and distribution are cumbersome and hard to set up. Hardware acceleration also relies on custom CUDA transpilers in FLAME which will never be as good as transpilers available through frameworks such as PyTorch [51] and TensorFlow [52], a fact clearly seen in the experiment described from previous work [53] below where GPU utilization using FLAME was between 5%-50% compared to consistently 100% for a TensorFlow based implementation.

Repast4Py [54] extends Repast to Python and achieves hardware acceleration and vectorization through the combined use of Numba for just-in-time compilation, and PyTorch for hardware acceleration. It also attempts to use similarly clever distribution strategies as did FLAME to locate similar neighborhoods on adjacent nodes. In doing so it gets closest to SalVO. However, its primary focus is on a design for agents communicating and interacting across multiple processes using libraries like MI4Py. While concepts such as shared projection are extremely useful, all of these IPCs are implemented within the framework, and PyTorch is used only for hardware acceleration when needed. Our thesis is that this approach will be less efficient than the approach SalVO takes, where the computation graph and device placement are optimized internally within frameworks like JAX, TensorFlow, and PyTorch (backed by huge companies with large engineering teams) and optimized using frameworks like TVM [55], assessed in §6. The LSD framework [56] achieves speed through the compilation of a very difficult-to-use and debug DSL into a C++ binary, but is hard to use and not scalable. Several other frameworks like D-MASON [57], Pandora [58] and Care HPS [59] use a combination of C++ and OpenMPI to achieve distribution but not with inbuilt hardware acceleration.

Finally, **all** of these frameworks lack the ability to calibrate and optimize ABM parameters as part of the framework at scale. Critically none of them create ABMs that are differentiable, necessary for our approach of learning ABM parameters through backpropagation.

These shortcomings resulted in the initial development of the "Dataflow ABM" by the author [53]. This was a framework built from first principles through a formalism of ABM where agent functions are pure functions of previous state, parameters, and exogenous messages. This permits a model of execution inspired by large-scale data processing frameworks, with vectorized execution using hardware acceleration, and using a scatter-gather programming paradigm, to scale across networks of heterogeneous devices. Benchmarks run in previous work [53] showed that implementation of the framework in TensorFlow using the Circles ABM benchmark [60] resulted in extraordinarily fast simulations, beating even FLAME [3], as seen in figure 2.1. This early framework will be extensively refined and extended in this work.

2.2.2 SalVO's design strengths

SalVO was developed with several critical design principles in mind, including ease of use, ease of scale, hardware acceleration through vectorization, and in-built learning capabilities. A comparison of these principles vs. current frameworks is summarized in table 2.2.2, showing the need for this framework.

Agents.jl supports BlackBoxOptim.jl that uses evolutionary algorithms for AB calibration,

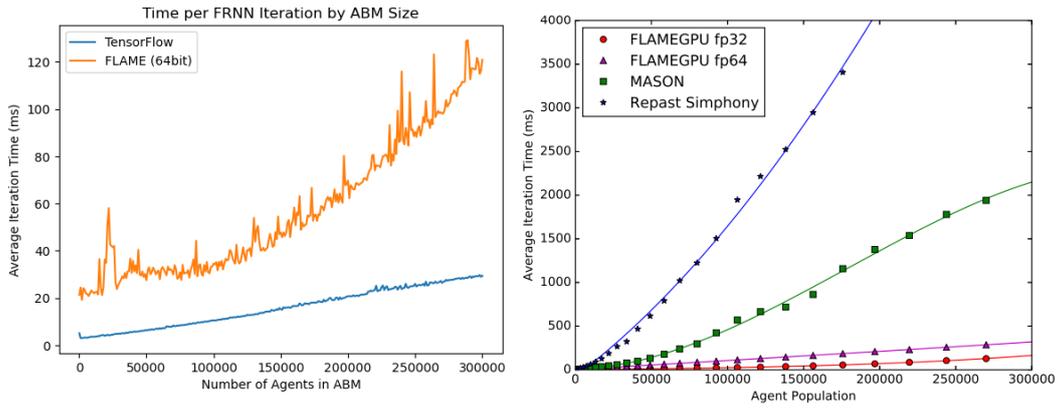


Figure 2.1: Comparison of execution speeds of the fixed radius nearest neighbours search benchmark by Chisholm [60] of FLAME and other ABM frameworks vs. the TensorFlow dataflow ABM implementation, from [53].

a slow and unscalable but built-in, approach. FLAME and Repast4Py both support distributed computing, but expert guidance is needed to deploy them in a multi-node setting. SalVO also doesn't yet have any in-built visualization tools; as its built-in Python, powerful visualization packages such as Seaborn can easily be used to visualize ABM execution traces.

2.2.3 Learning in ABMs

A key innovation of SalVO ABMs is that they are differentiable as long as the agent functions are differentiable. This is extraordinarily powerful since it allows for the training of ABM parameters at huge, neural network scales. Previous work by the author showed that training was trivially possible at the scale of 100,000+ parameters [61]. Comparative techniques like simulated minimum distance [62] or bayesian optimisation [63], or even mean-field approaches [64] do not scale to significant numbers of parameters, with experiments with bayesian optimisation failing to scale over 10 parameters, whereas backpropagation calibrated 100,000 parameters with much higher accuracy and less noise.

This backpropagation approach was developed independently, but very recent literature has also published some preliminary work on differentiable ABMs and training using backpropagation [65, 66]. It achieves powerful results, demonstrating large-scale parameter learning in the context of epidemiological ABMs. However, the published work has major drawbacks: it lacks the formalism and theoretical basis of the approach developed here and doesn't address the key limitations faced by differentiable ABMs, i.e. the need for deterministic, differentiable agents. This is an assumption that doesn't work for the majority of ABMs. In addition to a unified framework for extraordinarily fast, hardware-accelerated, and scalable ABM simulations, SalVO also provides an environment where backpropagation-based training can occur natively and also implements an approach that overcomes the differentiability and deterministic needs of backpropagation-based training.

Category	Strength	SalVO	FLAME [44]	Agents.jl [50]	Mesa [43]	Repast4Py [54]
Usability	Simulation and analysis both in Python	✓			✓	✓
	Inbuilt visualization tools		✓	✓	✓	✓
	Should have an easy-to-code API and low barriers to entry	✓		✓	✓	✓
Optimized policy design / calibration	Parameter calibration is an in-built feature of the package	✓		~		
	Parameter calibration using backpropagation	✓				
	Parameter calibration using policy gradient	✓				
	Supports auto-differentiation of ABMs	✓				
Parallelization and scalability	Supports in-built hardware acceleration	✓	✓			✓
	Supports (easy) distributed computing	✓	~			~
	Built on current high performance computing frameworks	✓				✓
	Should be highly scalable even on a single machine	✓	✓	✓		✓

Table 2.2: Comparison of SalVO’s key design principles and achievements vs. its closest competitors.

Chapter 3

SalVO: Scalable, vectorized, and optimizable agent based model simulations

SalVO stands for the 3 key properties of the novel Agent Based Modelling (ABM) simulation framework presented here: **ScAL**lable, **V**ectorized, and **O**ptimizable.

This section lays out the theoretical foundation of agent based models backing the approach adopted by SalVO, and discusses its implications on the computational models used to execute the simulations. It then builds SalVO, a novel Python-based ABM library implementing this approach, and details the design principles and decisions used in the library's implementation. It also reviews how the formalism developed enables ABM parameter learning and optimisation using backpropagation. It concludes by presenting an extension using a novel application of Multi-Agent Reinforcement Learning (MARL) techniques to overcome the limitations of using backpropagation but still learn parameters at similar scales.

3.1 Introduction

Intuitively an agent based model is quite simple: take many different agents, each with their own set of states and actions, place them in an environment in which they can interact, simulate them, and study the results. There are many different formalisms that could fit this pattern, for instance, the communicating X-Machines [67] based formalism used by FLAME [44]. We develop one from scratch by making a very basic observation of the system: the only input to the system is the initial state and configuration of the agents. Every subsequent state, and final output of the ABM, is simply a recursive implementation of the agent function. If S_0 where the initial state, A the agent function, and T^* the termination time, then the next state as a function of the previous one is $S_{t+1} = A(S_t)$, and the output of the ABM is simply: $S_{T^*} = A(A(\dots A(S_0)))$. In a formalism where the agent functions are pure, the simulation is a simple recursive application of arbitrarily complex agent functions on appropriate inputs. Finally, a user will want two things from these agents: a trace of their execution history, and measurements of their final state.

One useful result of a pure agent function was given by Grazzini [62, 63] who showed that with state calculated as recursive applications of an agent function, if an ABM is ergodic and m a measurement of the state and P the initial configuration, then:

$$\mathbb{E}[m(S_t)|t > T^*] = y_* = g^*(S_0, P)$$

I.e. an ABM is ergodic if it reaches the same statistical equilibrium for the same initial conditions. If this equilibrium is absorbing, then all initial conditions eventually lead to the same measurement: $g^*(S_0, P) = m^*(P)$. An extension to this model developed in previous work by the author [68] showed that if the initial conditions are taken to be random and the model ergodic, then the long run statistics are a stochastic function of the measurement: $y_* = m_*(P) + \epsilon_t$. This is an important result that will be used later.

3.2 ABMs formalized

The formalism presented below greatly extends prior work on "Dataflow ABMs" [61] [53] with crucially being able to handle multiple agent types parsimoniously. We also claim it is able to express most types of ABM simulations and show that the formalism can express any Turing complete program, thus demonstrating its expressivity.

3.2.1 SalVO ABM formalism

Definition 1. *SalVO ABM:*

1. A set of agents of J different types, $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_J$
2. A collected set of states of all agents $\mathcal{S}_t \in \mathbb{S}$
 - This can be subdivided into $\mathcal{S}_t^{(1)}, \mathcal{S}_t^{(2)}, \dots, \mathcal{S}_t^{(J)}, \mathcal{S}_t^{(shared)} \in \mathbb{S}$
 - $\mathcal{S}_t^{(j)}$ represents the collected state of each **agent type**, $j \in J$ that are belong solely to agents of that type and that only they can update
 - $\mathcal{S}_t^{(shared)}$ represents a shared state that is common across multiple agent types
3. A set of exogenous messages, $\mathcal{M}_t \in \mathbb{M}$ injected into the ABM
 - This allows agents to react to external stimuli rather than for all effects to be endogenous
 - This is a critical requirement to make agent functions pure but calibratable
4. A set of parameters $\mathcal{P} = \{\mathcal{P}_{fixed}, \mathcal{P}_{free}\} \in \mathbb{P}$ configuring the behaviour of the ABM
 - \mathcal{P}_{fixed} represent parameters that cannot be changed, such as physical constants
 - \mathcal{P}_{free} represent calibrated parameters for which the true value is only estimated
5. A set of dead or alive parameters $\gamma_t \in [0, 1]^{J \times \sum_{j \in J} N_j}$ representing whether agents are alive or dead, with one needed for each of the N_j agents of type j , for all J agent types
6. An edge set $\mathcal{E}_t \in [0, 1]^{\sum_{j \in J} N_j \times \sum_{j \in J} N_j}$ which is an adjacency matrix representing the connectivity of each agent to all others
7. An edge set generator $g : \mathbb{S} \times \mathbb{M} \times \mathbb{P} \times \gamma \times \mathbb{E} \rightarrow \mathbb{E}$ which generates the next edge set from the current collective state of the ABM
8. A set of J agent functions, $\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \dots, \mathcal{A}^{(J)}$, where each agent function is a map that generates the next state from the current state: $\mathcal{A}^{(j)} : \mathbb{S} \times \mathbb{M} \times \mathbb{P} \times \gamma \times \mathbb{E} \rightarrow \mathbb{S} \times \gamma$

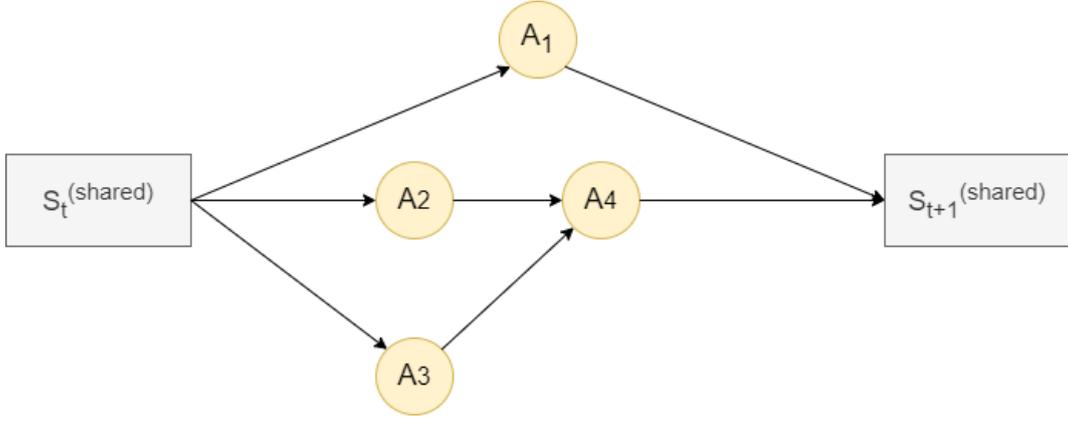


Figure 3.1: A DAG representing an ordering between agent types for a shared state update. Agent functions 1, 2, and 3 can be independently executed, but 4 can only be executed after 2 and 3.

(a) Each agent is responsible for updating both its section of the state, $\mathcal{S}_t^{(j)}$, and the shared component $\mathcal{S}_t^{(shared)}$

9. A set of measurement functions that summarize agent state and can be used for calibration, $m : \mathbb{S} \times \mathbb{M} \rightarrow \prod_{i \in H} \mathbb{R}^{M_i}$

This formalism is examined in more detail and operationalized in §3.5. Note this formalism assumes that time moves in discrete steps in the simulation, which is reasonable, and asynchronous activations can be modeled through external signals through \mathcal{M}_t or baked into the agent functions $\mathcal{A}^{(j)}$.

3.2.2 Shared state and dependency graphs

A key issue in definition 1 is that since $\mathcal{S}_{t+1}^{(shared)}$ comes from compositions of $\mathcal{A}^{(j)}$ applied to $\mathcal{S}_t^{(shared)}$, the ordering of the compositions is important and can determine the output. In most ABM simulations like Mesa [43], scheduling is customizable, but typical schedules include random agent activations, or simultaneous activations (all agents activated one after the other in deterministic order). In each of these options, there is not much information content in the choice of scheduler.

Since our formalism groups together agents of the same type into a single vectorized function, we can solve this problem by defining a directed acyclic graph (DAG) of dependencies between agent types. This is exemplified in figure 3.1, which represents computation in equation 3.1.

$$\mathcal{S}_t^{(shared)} \rightarrow \hat{\mathcal{S}}_t^{(shared)} = \text{ParMap} \left(\begin{array}{c} \mathcal{A}^{(1)}(\mathcal{S}_t^{(shared)}) \\ \mathcal{A}^{(2)}(\mathcal{S}_t^{(shared)}) \\ \mathcal{A}^{(3)}(\mathcal{S}_t^{(shared)}) \end{array} \right) \rightarrow \mathcal{S}_{t+1}^{(shared)} = \mathcal{A}^{(4)}(\hat{\mathcal{S}}_t^{(shared)}) \quad (3.1)$$

This is a powerful abstraction, since not only can it represent how updates to shared state are correctly made, it also gives an opportunity for parallel execution of vectorized agent types across available nodes.

3.2.3 Emulating a Turing complete language

There are clear restrictions in the formalism above compared to the free-flowing agent-level implementations achievable in frameworks like Mesa; for instance, all agents of the same time are expressed through one vectorized function. This is nonetheless a very expressive construct. As a sketch of the computational flexibility of this framework, we will show that, with a sufficiently expressive agent function, this formalism is capable of expressing programs in Brainfuck (BF) [69], a simple, 8 instruction, Turing complete language.

Using the formalism from definition 1, let:

- \mathcal{P} contain the program
- \mathcal{M}_t be the user inputs at step t
- Let \mathcal{S}_0 contain:
 - A sufficiently large byte array (BF requires 30,000 bytes minimum)
 - An instruction pointer $l \in \mathbb{N}^+$ which indexes where in \mathcal{P} we are
 - A data pointer $d \in \mathbb{N}^+$ which indexes where in the byte array at step t we are
- \mathcal{A} implement the 8 BF instructions and update \mathcal{S}_t accordingly:
 - " $>$ / " $<$ " \rightarrow increment/decrement d , and increment l
 - " $+$ / " $-$ " \rightarrow increment/decrement $\mathcal{S}_t[d]$, and increment l
 - " $.$ " \rightarrow increment l
 - " $,$ " \rightarrow set $\mathcal{S}_t[d] = \mathcal{M}_t$, and increment l
 - " $[$ " \rightarrow if $\mathcal{S}_t[d]$ is zero, then set $l = next("]", \mathcal{P}) + 1$
 - " $]$ " \rightarrow if $\mathcal{S}_t[d]$ is non-zero, then set $l = last("[", \mathcal{P}) + 1$
- m be a measurement function that, if $\mathcal{P}[l] = "."$, records in the trace the byte at $\mathcal{S}_t[d]$

Then, any BF program can be expressed by applying \mathcal{A} on \mathcal{S}_t to generate \mathcal{S}_{t+1} until we reach $t = T$; $T = len(\mathcal{P})$. This choice of \mathcal{A} allows for the emulation of any program in BF, a Turing complete language. This shows that, while the computation model implied by the formalism differs from traditional ABM frameworks, it is capable of expressing a very wide range of programs and ABMs.

3.3 Achieving scale

We define "scale" in ABM simulation to be the ability to increase the number of agents, or environmental size, to very large levels successfully, without significantly needing to alter how the ABM is written, and without inducing very significant overhead. Scale is achieved through two means: increasing the throughput on a single device using hardware acceleration, and scaling computation to a network of devices when needed. The former is discussed below, and the latter in §3.5.1 and §7.3.

3.3.1 Vectorization and hardware acceleration

A consequence of the formalism we have chosen to adopt is that it naturally embeds vectorized execution of agent functions. An agent function of type j , $\mathcal{A}^{(j)}$ can express any number of agents, and their simultaneous calculation can take advantage of many modes of parallel execution.

Single instruction multiple data (SIMD) is a common mode of parallelism where a single instruction can be simultaneously applied across multiple data streams using array processors, implemented by most modern CPUs. Hardware acceleration using GPUs, for instance, using Nvidia’s CUDA [70], use a variant of SIMD called single instruction multiple threads (SIMT), which enables far greater parallelism by splitting single instructions into threads, each responsible for the execution of a block of instructions on a large array of processing units with their own registers and shared memory.

If agent functions can be made to operate on state represented in tensors [52], huge efficiencies through hardware acceleration are attainable, by compiling the application of an agent function on a state tensor into a kernel, which gets executed on the chunked tensorized data by threads operating on a block of processors on the GPUs. An example of such an ABM design is the VDSK presented in §5, and the results of hardware acceleration are clearly seen in §6.

A constraint introduced by vectorization is that if tensorized state and agent function dimensions were to change every step, the kernel would have to be recompiled, resulting in a significant slowdown introduced to the process. Implementations should ideally strive to have constant dimensionality throughout the simulation, which makes agent deaths hard to integrate. However, this can be dealt with by treating agents as toroidal (dead agents come back to life with a new state) or by keeping dead agents dead and summarizing their state using measurement functions.

3.4 Huge scale parameter optimisation and calibration

This section shows how backpropagation and techniques from MARL can be used to learn ABM parameters at huge scales. It builds on previous work by the author [61] implementing backpropagation-based learning using a simpler, less general formalism. It also presents a novel framework using policy gradient techniques to overcome the assumption of deterministic, differentiable agents needed for backpropagation.

3.4.1 Parameter calibration with backpropagation

The ultimate goal of ABMs is to calculate measurements of the final state: $m(\mathcal{S}_T)$. The final state is calculated from an application of the agent function (assuming one agent type): $\mathcal{S}_T = \mathcal{A}(\mathcal{S}_{T-1}, \mathcal{M}_{T-1}, \mathcal{P}, \gamma_{T-1}, \mathcal{E}_{T-1})$. This can further be recursively expanded: $\mathcal{S}_T = \mathcal{A}(\mathcal{A}(\mathcal{S}_{T-2}, \mathcal{M}_{T-2}, \mathcal{P}, \gamma_{T-2}, \mathcal{E}_{T-2}), \mathcal{M}_{T-1}, \mathcal{P}, \gamma_{T-1}, \mathcal{E}_{T-1})$. Since the edge sets also get generated as a function of previous states, this can be recursively expanded down to the initial state and parameter sets.

This formulation shows parallels with that of deep neural networks, where an agent function parallels activation functions, and state parallels weights. The other inputs are not relevant; either they are computable from state (γ, \mathcal{E}_t) , or an exogenous input. If \mathcal{A}_t denotes the application of the agent function of state \mathcal{S}_t (and other inputs), then parameter

sets can be learnt by using backpropagation through recursive applications of the chain rule until the derivative in equation 3.2 can be computed.

$$\frac{\partial m(\mathcal{S}_T)}{\partial \mathcal{P}} = \frac{\partial(m(\mathcal{S}_T))}{\partial \mathcal{A}_{T-1}} \frac{\partial \mathcal{A}_{T-1}}{\partial \mathcal{A}_{T-2}} \cdots \frac{\partial \mathcal{A}_0}{\partial \mathcal{P}} \quad (3.2)$$

This can be easily extended to multiple agent types using the dependency structure described earlier and insights from graph neural networks. Envisioning our vectorized ABM as a GNN with agent types as nodes, and the edges from the dependency graph from §3.2.2, then each node’s state gets updated at each timestep t , with connected edges as \mathcal{E}_t , using a nearly identical formulation to Hamilton’s for GNNs [71]:

$$\mathcal{S}_{t+1}^A = \text{UPDATE}(\mathcal{S}_t^A, \text{AGGREGATE}(\bigcup_{k \in \mathcal{E}_t} \mathcal{S}_t^k)) \quad (3.3)$$

While the necessary properties of permutation equivariance aren’t formally discussed in the context of definition 1, it is intuitive. In either case, once we can calculate the differential in equation 3.2, we can then proceed with parameter learning using gradient descent [61]:

$$\hat{\mathcal{P}} = \mathcal{P} - \eta \frac{\partial m(\mathcal{S}_T)}{\partial \mathcal{P}} \quad (3.4)$$

3.4.2 Policy gradient-based parameter calibration

Backpropagation enables huge-size parameter learning, but has two critical flaws: the agent functions have to all be differentiable, and the simulation (agent functions and external inputs) cannot be stochastic. These are key limitations that disallow vast classes of ABMs; e.g. most epidemiological simulations have probabilistic transmissions of diseases.

To circumvent these limitations, we look at reinforcement learning for inspiration. With a stochastic ABM, our performance measure will now be $J(\mathcal{P}) = \mathbb{E}[m(\mathcal{S}_T)|\mathcal{P}]$, called a reinforcement signal. A critical assumption is the signal needs to be stationary; this is a less restrictive assumption than no stochasticity, but still restrictive.

Now $\mathbb{E}[m(\mathcal{S}_T)|\mathcal{P}]$ is a fully defined but unknown function. We want to find a \mathcal{P}^* such that $\mathbb{E}[m(\mathcal{S}_T)|\mathcal{P}^*] \geq \mathbb{E}[m(\mathcal{S}_T)|\mathcal{P}] \quad \forall \mathcal{P} \in \mathbb{P}$. For this, we propose to use William’s REINFORCE algorithm [4], a type of policy gradient method, where, given a reward signal, parameter updates can be made as $\hat{\mathcal{P}} = \mathcal{P} - \eta \nabla_{\mathcal{P}} J$. The main problem is in determining the policy gradient, $\nabla_{\mathcal{P}} J$.

Say that parameters update according to equation 3.5:

$$\Delta \mathcal{P} = \underbrace{\alpha}_{\text{Learning rate}} (m(\mathcal{S}_T) - \underbrace{b}_{\text{Baseline}}) \frac{\partial \ln g}{\partial \mathcal{P}} \quad (3.5)$$

Here, William’s denoted the function g as the probability density function of a unit’s output, in a feedforward neural network [4]. In our case, g would represent the probability density function of an agent type of their output state, conditioned on parameters: $g(\xi, \mathcal{P}) = Pr\{\mathcal{S} = \xi|\mathcal{P}\}$.

REINFORCE stipulates that for an appropriate specification of reward, the average parameter update vector lies in the same direction as in which the expected measurement increases [4], i.e. $\langle \mathbb{E}[\Delta \mathcal{P} | \mathcal{P}], \nabla_{\mathcal{P}} \mathbb{E}[m(\mathcal{S}_T) | \mathcal{P}] \rangle \geq 0$. Williams shows that this is equivalent to saying that equation 3.5 is an unbiased estimate of $\partial \mathbb{E}[m(\mathcal{S}_T) | \mathcal{P}] / \partial \mathcal{P}$. This is an immense relaxation of the constraints of backpropagation. Now we need only that the output of agents is a stationary distribution of inputs and parameters.

Now our key goal is to fit $\Pr\{y_t = \xi | \mathcal{S}_t, \mathcal{P}\}$, which relies on stationarity. Fortunately, we can obtain this under certain conditions, as discussed in §3.1. This model is amenable to emulation via Gaussian Processes if the measurement function output is continuous and stationary. With these assumptions, we can use $y \sim GP(\mu(\mathcal{P}), \sigma(\mathcal{P}))$ as our Gaussian process, and the REINFORCE algorithm will be:

$$\Delta P = \alpha(m(S_T) - b) \frac{\partial GP}{\partial P}$$

There are several obvious drawbacks to this approach:

1. The formalism above works for a single functional mapping being recursively applied from the base state S_0 to the final state S_T , i.e. a single agent type or multiple agent types with no shared state. SalVO assumes that multiple agent types can be emulated, but it needs to be more formally described.
2. This only works for continuous measurement outputs, but any alternative discrete probabilistic model should also work.
3. The dimensionality over which we can fit a Gaussian Process without resorting to special tools for high dimensional GPs is low.
4. Fitting a Gaussian process is inherently tough:
 - (a) The choice of hyperparameters and kernel function is more art than science
 - (b) The number of data points we will need will scale non-linearly with the number of parameters.
5. If we fit a single Gaussian process for the overall model as done in [68], then the number of parameters that can be optimized is incredibly low and the quality of optimisation is poor.
6. If we do so at an agent level, then fitting the Gaussian process will be incredibly computationally expensive and likely also noisy; so we have to do so at an agent-type level. However, agents can be heterogenous even with the same type.
7. This is an ABM formulation with a global reward, but one that is achieved by maximizing each agent type's parameter sets independently. This would only be the case for agents which had independent transitions and block-reward independence, as per Amato [72], which aren't guaranteed to be the case in practice.
8. We don't explicitly include the exogenous messaging system in our ABM definition.
9. We will have to run many different iterations of the ABM simulation to gather enough data to fit a valid GP, particularly for many parameters.

All of these are significant issues that need deep theoretical investigations and resolution, which unfortunately there was not sufficient time for. Instead, we propose a practitioner's

empirical protocol which should be able to address a number of these concerns, including handling the effects of agent interaction and agent heterogeneity.

3.4.3 GP-ABM

This can be summarized into a proposed protocol for the GP-ABM:

1. We first fully specify a (possibly stochastic) ABM that runs from a specified initial state and parameters to its end.
2. We generate N different samples over a hypercube of P , with the sampling strategy to be specified later.
3. We run the simulations N times and record, for each agent of every different type, the output of the agents. Say for agent type $j \in J$, we have M different agents, this gives us a dataset of $J \times M \times N$ data points (assuming scalar outputs).
4. We use this dataset to calibrate J different Gaussian processes, one per agent type. We use the outputs of each agent as noisy observations to calibrate the kernel hyperparameters using MLE and fit the overall GP.
5. We now assume that each agent contains a copy of this Gaussian process. We can then optimize parameters at a single agent i level using the following REINFORCE update equation:

$$\Delta P_{i,j} = \alpha(m(S_T) - b) \frac{\partial GP_i^{(j)}}{\partial P} \Big|_{P=P_i}$$

We compute $\frac{\partial GP_i^{(j)}}{\partial P}$ by first calibrating the GP, and then carrying out inference at P_i to get a specification for its density, $g(\mu, \sigma)$ in equation 3.6. We can then differentiate the log density to calculate the gradient we use. Given a kernel function $\kappa(x, x)$ over observable data x , new data x^* and outcome y :

$$y \sim \mathcal{N}(\kappa(x^*, x)^T (\kappa(x, x) + \sigma^2 I)^{-1} y, \kappa(x^*, x^*) + \sigma^2 + \kappa(x^*, x) (\kappa(x, x) + \sigma^2 I)^{-1} \kappa(x^*, x)) \quad (3.6)$$

This is quite a powerful protocol and has numerous advantages, including:

1. By calibrating the GP for agents from observations from an agent type; we obtain M observations per run across N steps, which reduces the number of runs needed to calibrate the GP.
2. By using M observations, one from each agent, we handle agent heterogeneity as noise in the GP fitting process, whilst still retaining information about the overall influence of a parameter on the average behaviour of agents.
3. By allowing each agent to have its own copy of the Gaussian process we allow for the calibration of up to $J \times M \times 10$ parameters (assuming all J agent types have M agents).
4. By using the agent outputs at time T we are able to leverage the long-run stationarity of the simulation to obtain (theoretically) stationary observations to calibrate the GP.
5. This formulation only works for stationary rewards and outputs; this still needs to be carefully dealt with in practice, for instance by differencing outputs.

The implementation is discussed in the next section.

3.5 SalVO architecture and implementation

This section describes the implementation of the ABM framework we have laid out here into SalVO, a Python library implementing scalable, vectorized, and optimizable ABM simulations. We implement several common ABMs to demonstrate the power and speed of the framework, as well as its capacity for large-scale parameter learning.

3.5.1 Design requirements

It is worth being more prescriptive about some of SalVO’s design requirements. SalVO is intended to be easy to use, a contrast to FLAME [44] and LSD [56]. The purpose is to allow researchers and policymakers the ability to focus more on creating a useful model, than on worrying about its implementation. It aims to be accessible to a broader set of less technical users, users who likely have more value to add by creating a useful model than by coding in an incredibly complex DSL. This means it needs to be very easy to debug, ideally by being based in an interpreted language allowing for easier online debugging [73], based on a commonly used language accessible to non-computer scientists, and need little custom configuration. These are addressed in SalVO by basing it in Python, with the only setup needed being the installation of dependent packages and hardware acceleration libraries like CUDA.

Hardware acceleration is a very broad categorization. It can include SIMD vectorization using instruction sets like AVX, AVX-512, and SSE, or SIMT vectorization used by CUDA-based GPUs and other specialty hardware, discussed in §3.3.1. It can extend to other forms of parallelization, such as SPMD, where forks of the same program operate on multiple data streams. In SalVO, we aim to support the full gambit of vectorization across all forms of hardware by leveraging libraries like JAX and numpy, which implement Google’s XLA [74] and BLAS/MKL [75] respectively for hardware acceleration. XLA has the added advantage of compiling instructions for specialty hardware (GPUs and TPUs); using JAX as SalVO’s backend allows us to take advantage of this with little overhead.

Distributed computing here is defined as the ability to scale the ABM simulation to a network of nodes. Traditional approaches to the distributed data flow model adopted by SalVO include frameworks like timely dataflow [76], CIEL [77] and X-Stream [78]. However, the recent explosion in deep neural network architectures has created a litany of large-scale distributed deep learning frameworks optimizing execution across pools of specialized hardware like GPUs. These include parameter server frameworks like Geeps [79], Horovod [80] and DistBelief [81]. These have the added benefit of having implementations in libraries like TensorFlow and JAX, through which SalVO can take advantage of them to achieve effective distribution.

However, these approaches are generally optimized for neural network training and inference, and large-scale dataflow models are likely more scalable for SalVO. Timely dataflow as a computation model is explored for SalVO in §7.3, but not yet implemented in SalVO due to time constraints.

3.5.2 Platform

While previous work [53] explored implementations of ABMs in TensorFlow, SalVO was ultimately implemented with JAX [82] as its backend. While TensorFlow [52] and PyTorch [51] have many desirable features, such as inbuilt hardware acceleration, and handling of distribution and interprocess communication, they work on dynamic compute graphs that can be difficult to work with, and harder to debug. They also rely on precompiled kernels for various operations stitched together before the graph’s execution. This is powerful for standard expressions like neural network cells, but for arbitrarily complex agents. However, JAX’s just-in-time compilation uses XLA to generate code for the entire function. This allows it to take advantage of far more optimisations, like operator fusing, to generate more efficient compiled code.

JAX also operates via function transformations on pure functions, an approach pleasingly in-sync with our choice of pure agent functions. It is also incredibly easy to code in, with its API built on top of numpy while applying JIT compilation and hardware acceleration on top. Furthermore, it has incredibly powerful forward and backward mode auto-differentiation (AD), able to apply AD through branches in native Python code and on a wide variety of functions. It also has 3 crucial primitives for ABM simulations: `vmap`, which automatically vectorizes operations, `scan`, which allows for the main ABM loop to be JIT-compileable and differentiable, and `pmap`, which allows for trivial scaling of operations across multiple devices. Most critical are the in-built support for PyTrees in JAX, making highly complex agent state incredibly easy to express in SalVO while still making use of JIT compilation, hardware acceleration, and autodifferentiation. These make JAX the clear choice as the backend for SalVO.

3.5.3 Core architecture and implementation

SalVO is a transpiler of an ABM definition contained in the files in table 3.1. The core architecture is laid out in figure 3.2. The architecture operates on creating a state flow between pure, vectorized, agent functions in JAX. Agent functions are parsed by the SalVO transpiler, and their outputs are traced and collated with the dependency graph. Measurement functions are treated on parity with agent functions, where inputs are components of the state for any agent function, and outputs are measurements. These are inserted into the dependency graph as soon as the state it measures appears (for the last time). The agent deaths, γ , are treated as part of the state, and the edge set generator g is treated like an agent function (as shown in §3.5.4).

Once the state flow has been mapped, hardware information is also gathered. In the final product, this information will be used to commit state to the right device before agent function execution, and all independent agents will execute asynchronously across available hardware. However, the current implementation only supports a single device, since an optimal device placement strategy must be considered for good scaling performance.

This amalgamated information is then used to generate a function that runs a single step, `Orchestrate`. This outputs two PyTrees, the next state \mathcal{S}_t (and edge set \mathcal{E}_t , dead or alive parameters γ), and measurements, $m(\mathcal{S}_t)$. A function that runs the entire simulation using JAX’s `lax.scan` is also created, called `Run`, which is used for optimisation. External messages \mathcal{M}_t is passed by marking the parameter with `[iter]`, to pick up the message at the right simulation time.

File	Description
parameters.json	All static ABM parameters and initial values.
config.py	The dependency graph between agent types, and the state flow definitions are defined here.
simulation.py	This pulls together the parameters and collates all inputs needed for simulation, including \mathcal{M}_t .
agent_functions.py	Contains all (vectorized) agent functions.
measurements.py	Contains all (vectorized) measurement functions.
run.py	The Run Environment - collates simulation and runs it.
backpropagation.py	Contains the target parameters being optimized, the gradient descent optimizer and associated parameters.
policy_gradient.py	Contains the target parameters for optimisation and their space definitions, the GP kernel definition, and acquisition function.

Table 3.1: Description of files in SalVO ABM definition.

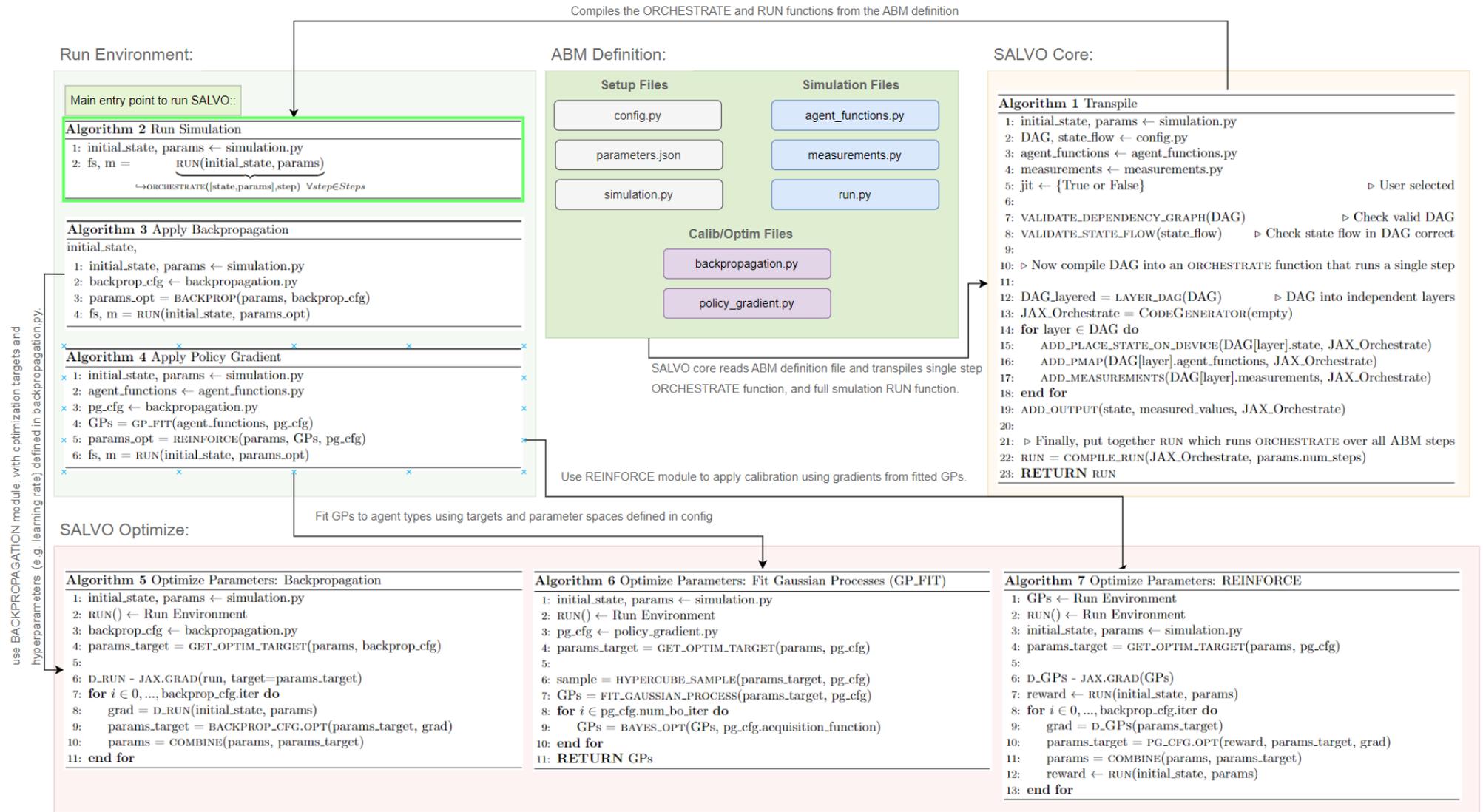


Figure 3.2: Outline of SalVO architecture and implementation of key subcomponents. Everything stems from the ABM definition, where each ABM implements the 6 required files for simulation, and the optional 2 files for the 2 methods of calibration implemented. This definition gets transpiled into the Run Environment, where the simulation can be run and optimized policies can be derived.

```

agent_functions.py
@jax.jit
def agent_function(agent_positions, i, radius, k_rep, k_att):
    """
    Vectorize and carry out computation
    """
    cf = partial(
        hlp.compute_force_vector,
        agent_positions=agent_positions,
        radius=radius,
        k_rep=k_rep,
        k_att=k_att,
    )
    fv = jax.vmap(cf)
    force_vector = fv(agent_positions)
    agent_positions += force_vector
    return agent_positions, agent_positions

simulation.py
param_fpath = pathlib.Path(__file__).parent.resolve()
parameters = core.load_parameters(str(param_fpath) + "/parameters.json")

key = random.PRNGKey(0)
w = parameters.w
e_dim = parameters.e_dim
radius = parameters.radius
a_pop = parameters.a_pop
k_att = parameters.k_att
k_rep = parameters.k_rep
rounds = parameters.rounds

agent_positions = random.uniform(key, shape=(int(a_pop), e_dim)) * w

config.py
dependency = {
    "agent_function_orchestrate": ["identity"],
    "identity": [],
}

state_flow = {
    "agent_function_orchestrate": {
        "inputs": ["agent_positions", "radius", "k_att", "k_rep"],
        "outputs": ["agent_positions"],
    },
    "identity": {
        "inputs": ["agent_positions"],
        "outputs": ["agent_positions_measured"],
    },
}

parameters.json
{
    "w": 100,
    "e_dim": 2,
    "radius": 20,
    "a_pop": 100,
    "k_att": 1e-2,
    "k_rep": 1e-2,
    "rounds": 10
}

run.py
import salvo_core as core
from circles import (simulation, agent_functions,
                    measurements, config)

sim = core.build_simulation(
    simulation, config.dependency,
    config.state_flow, agent_functions,
    measurements, True
)

exec(sim.code)

```

Figure 3.3: Implementation of the Circles ABM [60] in SalVO

3.5.4 Example ABM implementations

Circles

The Circles ABM is based on an ABM benchmark by Chisholm [60] for fixed radius nearest neighbour (FRNN) finding. It consists of circles that are attracted/repelled from each other depending on the distance to all nearby balls. It was used by previous work [53] in a TensorFlow-based implementation benchmarked against FLAME, and is implemented in SalVO to demonstrate its speed and usability.

The implementation can be seen in figure 3.3. A single agent type is present *agent_functions.py*, implemented in one agent function that takes advantage of JAX’s auto-vectorization through *vmap*. The dependency graph and state flow in *config.py* contain the measurement function as well as the agent function, and the SalVO transpiler automatically identifies measurement functions and places them appropriately, and figures out end-to-end state flow and measurements correctly. The simulation is set through static parameters in *parameters.json*, from which a complex initial state is calculated in *simulation.py*.

Forest Fires

Forest Fires is a more complex ABMs, taken from Agents.jl [50], which showcases a more complex agent-type dependency structure, as well as the flexibility of the state flow paradigm. Agents are a set of trees spread over a $N \times N$ grid, which can either be dead, alive, or on fire. Trees are dead if they were on fire in the last round, and on fire, if they were alive in the last round and an adjacent tree was on fire. Dead trees also come back to life with probability p , and alive trees catch fire with probability f .

This was implemented by setting tree state to be $\mathbb{S} = \{0, 1, 2\}$ signifying the dead, alive or on-fire states. Note how the dead or alive parameter, γ , was taken into account in the state. A convolution filter was then used to identify if adjacent trees were on fire in a vectorized way, a standard approach in SalVO.

For simplicity, only *config.py* is shown in figure 3.4. There are 2 measurement functions, *number_of_trees_on_fire* and *number_of_trees_live*, that depend on *regrowth*

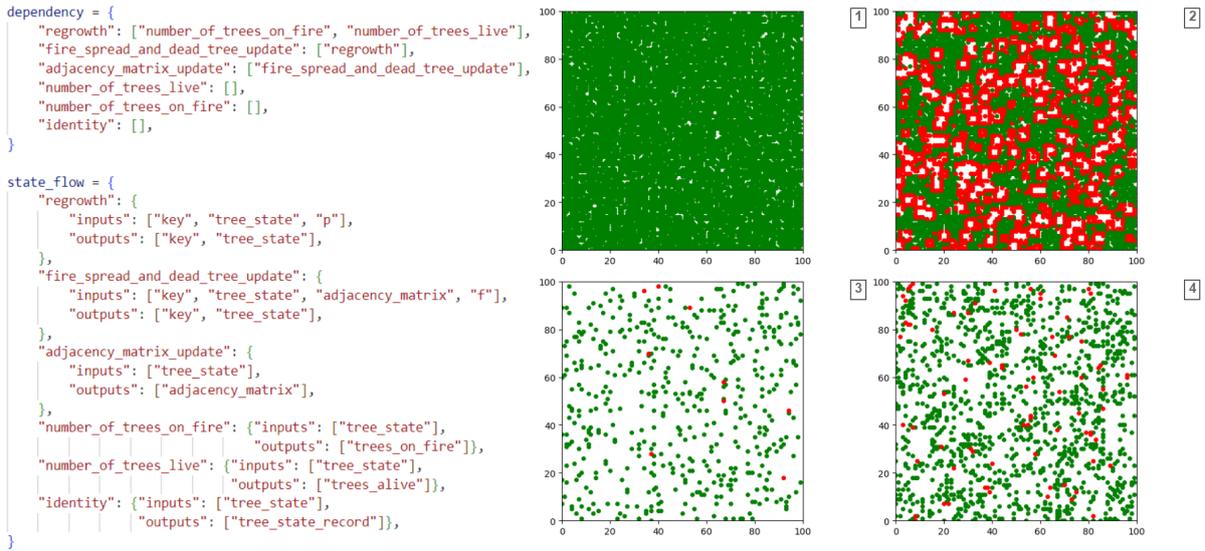


Figure 3.4: [Left] Dependency graph and state flow for the Forest Fires ABM. [Right] Screenshots from simulation, with the initial forest being torn down by fires, followed by regrowth and reaching a steady state.

finishing for the step, which in turn depends on `fire_spread_and_dead_tree_update`. This shows the power of SalVO’s abstraction, where agent functions aren’t necessarily just agent types but general computation steps that need to happen in the dependency graph in the ABM’s execution. The careful design here also increases the potential for parallelization. The results of this ABM are shown in figure 3.4.

Bird Flocking

The bird flocking variation of the classic boid flocking algorithm [83] developed in previous work [53] was also implemented in SalVO. Here, a number of birds move towards the centroid of the flock with a velocity of $v = \max(v_{max}, \alpha D)$ where D is the (vector) distance from the centroid of the flock, α is the control parameter, and v_{max} is the max velocity. Birds essentially all flock to the centroid with diminishing velocity as they reach the centroid of the flock. The original version [53] included stochastic wind, which was removed from this implementation.

Chapter 4

Background in ecological economics

Now that we have a powerful ABM framework in which to express our ABMs, we turn our attention to the creation of an ABM integrating the economy, climate, and biosphere, based on the DSK model. This section reviews the background necessary for the operationalization of this ABM. A more full accounting is given in appendix C.

4.1 Review of the Dasgupta framework

4.1.1 Introduction

One of the key challenges in addressing recent dramatic declines in biodiversity and our biosphere has been a mismatch of incentives from those who wish to conserve it, and those that get economic benefits from exploiting it [84]. Unfortunately, those who benefit from exploitation are also those who degrade it the most through the extraction of resources and the use of our biosphere as a sink for waste.

To summarize this model succinctly: traditional economic approaches assume that human activities cannot affect the biosphere enough to destabilise it, and in that sense biosphere services are effectively "free". Unbounded GDP growth then follows assuming a positive rate of technological innovation - i.e. we can do more and more with what we have, over time. If, however, you accept that human activities can infact devastate the biosphere and that in turns destroys human activities, GDP can only grow at at most a rate at which by our impact on the biosphere is offset by its natural regeneration, or at least that we don't push it over some critical boundary into instability.

Dasgupta argues our prices and wealth should reflect our value of the biosphere, both from the perspective of future economic value we can derive from a biosphere better sustained, and the intangible value we reap from it. He also argues that we cannot rely on indefinite technological innovation, and that technological innovation is ultimately bounded by the rules of entropy in a "bounded" earth, i.e. one where planetary boundaries exist whose traversal will tip the earth's ecosystem into instability. This assumption is key for any attempt to operationalize this model, and we will revisit it later.

4.1.2 Summary of model

A foundational element of the Dasgupta framework is the impact inequality, relating the rate of regrowth of the biosphere ¹, $G(S(t))$, to humanity's impact on it. Denote global output (GDP) as $Y(t) = Ny(t)$, with a population of N , then Dasgupta formulates the impact of resource extraction (e.g. through mining, forestry, etc.) from the biosphere on its growth rate as $Ny(t)/\alpha_x$, and the impact of waste as $Ny(t)/\alpha_z$. This can then be used to express the dynamics of the biosphere in equation 4.1.

$$\frac{dS(t)}{dt} = G(S(t)) - \frac{Y(t)}{\alpha_x} - \frac{Y(t)}{\alpha_z}, \quad \alpha_z^* \geq \alpha_z > 0 \quad (4.1)$$

Dasgupta formulates a Cobb-Douglas-like formula for GDP as a function of capital inputs in equation 4.2. Crucially, GDP depends on the traditional capital inputs of K , produced capital (infrastructure, machinery, etc.), H , human capital (health, education, earnings, etc.) and A (productivity and innovation). Additionally Dasgupta includes R , resources extracted from the biosphere (wood, metals, minerals, etc.), and S , biosphere regulating services (climate regulation, nitrogen cycle, etc.).

$$Y(t) = A(t)S(t)^\beta K(t)^a H(t)^b R(t)^{1-a-b}, \quad a, b, (1-a-b) > 0, \beta > 0 \quad (4.2)$$

Dasgupta also models the dynamics of the human population assuming that humans “target“ a population size depending on the level of human capital available, $J(h)$, in equation 4.3. This is backed by empirical studies [85] that found a higher standard of living generally reduced desired family size, and vice versa.

$$\frac{dN(t)}{dt} = N(t)(J(h) - N(t)), \quad J > 0, dJ(h)/dh < 0 \quad (4.3)$$

Similarly, produced capital depreciates with a certain rate λ , and is offset by investment, $I_K(t)$, leading to equation 4.4. Innovation and productivity is likewise boosted by investment in research and development, in equation 4.5, and investment in human capital $I_H(t)$ (building schools, hospitals, parks, and other infrastructure to improve quality of life) leads to increases in human capital, in equation 4.6.

$$\frac{dK(t)}{dt} = I_K(t) - \lambda K(t) = Y(t) - C(t) - I_H(t) = I_A(t) - \lambda K(t) \quad (4.4)$$

$$\frac{dA(t)}{dt} = I_A(t) \quad (4.5)$$

$$\frac{dH(t)}{dt} = N(t)\frac{dh(t)}{dt} + h(t)\frac{dN(t)}{dt} = I_H(t) \quad (4.6)$$

Given the dynamics of the system defined above, Dasgupta attempts to maximize a joint utility function, $n(c(t), S(t))$. Dasgupta notes that the key control variables available to the designer are $c(t)$, aggregate consumption, $I_H(t)$, investment in human capital, $I_A(t)$, investment in innovation, and $R(t)$, resource extraction. More details are in appendix C.

¹Dasgupta “heroically“ assumes the biosphere is representable in a single scalar value S

4.1.3 Criticisms

Aside from the extremely broad and often vaguely justified assumptions driving many of these dynamics, there is a crucial missing element in the equations presented above. This is the impact that a declining biosphere has on several key state variables, such as:

- $N(t)$: a worsening biosphere resulting in natural disasters, desertification, the loss of livable land, all will likely also affect birth rates and our population size. In Dasgupta, $N(t)$ is affected by $h(t)$, which only relies on $I_H(t)$, a control variable, but not $S(t)$.
- $K(t)$: increased natural disasters from a worsening biosphere will affect our capital stock; changing climates, extreme weather patterns, etc. will all destroy or impact our machinery (e.g. extreme storms destroying off-shore wind power plants)
- $A(t)$: the pace of innovation will be affected by biosphere degradation. As the issue becomes more important, more research will go into mitigation. However, with increased global catastrophes, resources will be diverted towards dealing with contemporaneous issues than research.

These will be addressed as part of the VDSK-B ABM proposal in §5.

4.2 Review of the Dystopian Schumpeter meeting Keynes (DSK) model

As discussed in §2, the DSK model from Lamperti et al [2] was chosen as the base model on which to develop to operationalize Dasgupta’s investigations into the economics of biodiversity. Using the DSK model presents 2 key technical challenges:

1. Scale: the model is highly complex, with over 16,000 parameters in the open source implementation by Pereira [86], even with only 250 agents.
2. Integration of Dasgupta: the model will have to be modified to endogenously contain the critical features of Dasgupta’s framework presented in §4.

The model was developed over the last 13 years, and we work from 5 key papers by Dosi, Lamperti et al [2, 87, 88, 89, 90]. It is highly complex, and its implementation in the Sant’Anna LSD simulator [86] even more so. An outline of the key components and interactions is given in figure 4.1. At a high-level, the model is an Agent Based Integrated Assessment Model consisting of the following agents/mechanisms:

1. A labour force
2. An energy industry choosing between green and dirty power plants
3. A capital goods industry making machinery for the consumption goods industry
4. A consumption goods industry making products for the labour force to consume
5. A financial industry supplying loans to the capital and consumption goods industries
6. A climate box recording the effects of emissions on the climate and generating ”disasters”

Appendix A contains a detailed account of the published version of the DSK model collated across the 5 papers cited. However, a critical review of the published papers in

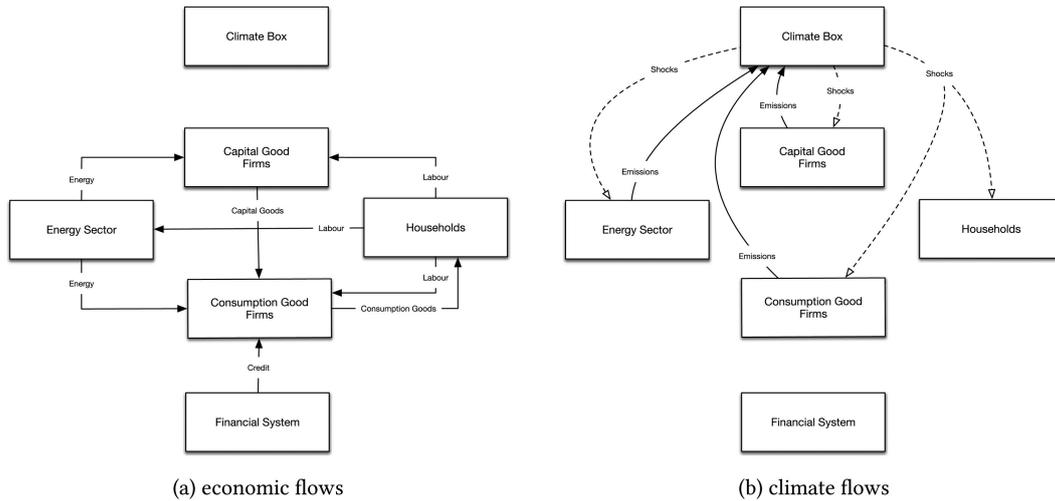


Figure 4.1: An outline of the DSK model from [2]. Economic flows occur between economic agents, whose emissions alter the climate, leading to shocks applied to the economic agents in terms of their inventory and productivity.

appendix B reveals that published works omit numerous key details of the DSK models and it is impossible to implement the model just the published works. Examples of details omitted include how output is constrained by the available labour, how to determine required labour from the current production, or how to calculate inflation or average productivity (some of these elements were obtained from the code). Additionally, the papers contain numerous errors which would affect the output of the model, and it is not clear how or whether these were resolved by the authors in their publications. For instance, a mistake in the coefficients of the DICE-2013R temperature model [91] cause temperatures to collapse to 2 degC after the model starts if implemented as the paper describes (see appendix B for detail). The only available open source implementation by Pereira [86] using the LSD simulation framework [56] differs greatly from published works, including using a different climate model and several different types of agents not mentioned in the DSK or other related papers, and so cannot easily be used to fill in the gaps. The criticisms and issues are summarized in appendix B.

All these issues will be addressed in the Vectorized-DSK re-implementation in §5, with details filled in from the open source implementation [86] where possible and assumed where not.

Chapter 5

The Vectorized Dystopian Schumpeter meeting Keynes, with Biodiversity (VDSK-B) Model

5.1 Filling in the blanks: vectorizing the DSK model

Now that we have a powerful framework, SalVO, in which to express our ABM, we will modify Lamperti's DSK model [2] into a vectorized form that we can express in SalVO. In doing so, we will also demonstrate the numerous benefits of SalVO, including its succinctness, speed through hardware acceleration, and scale. It will also allow us to leverage auto-differentiation and MARL techniques to allow for parameter learning.

A more thorough introduction to the DSK model is given in appendix A, with its transformation into VDSK contained here. For this work, assume we have N capital good firms and M consumption good firms. We further assume consumption firms replace their machinery after J years, and energy firms replace their dirty power plants every K years.

Let \odot will represent the Hadamard product, and $e_I \in \mathbf{1}^I$ be a vector of ones of length I .

5.1.1 Capital good firms

The state for capital good firms is deeply complex. To begin with, we have each of the 6 key variables characterising machinery produced by the firms, $\{A|B\}_{L|EF|EE}(t) \in \mathbb{R}^{N \times 1}$ representing the labour productivity (L), environmental friendliness (EF), and energy efficiency (EE) respectively affecting the output of capital good firms (B) and consumption good firms (A). Since each capital good firm is only producing one machine at a time, we only need to keep track of the coefficients for the latest vintage, aiding in vectorization. We can then compute the state updates in a vectorized way. Capital firms incur cost $c(t)$ per unit for producing machinery at with market wage $w(t)$ and cost of energy $c_{en}(t)$, and they sell the machinery to consumption good firms with a fixed markup μ at price $p(t)$, in equations 5.1 and 5.2.

$$c(t) = w(t)B_L^{-1} + c_{en}(t)B_{EE}^{-1}(t) \quad (5.1)$$

$$p(t) = c(t)(1 + \mu) \quad (5.2)$$

As per DSK [2], a random adjacency matrix representing connections between capital good and consumption good firms is instantiated, $\mathcal{E}_0 \in \text{Ber}(0.5)^{N \times M}$. Consumption firms can only place orders with those consumption firms they are connected to. For now, this is held static, but further improvements will add random changes. Since no demand clearing mechanism is specified by Lamperti [2], we allocate demand from consumption good firms, $Q^{cons}(t) \in \mathbb{R}^M$, equally across capital good firms $Q^{cap}(t) = \mathcal{E}_t Q^{cons}(t) (\mathcal{E}_t^T e_N)^{-1}$.

A research and development process works to improve technology stochastically through an innovation process, improving machinery energy efficiency, environmental friendliness, and productivity. Firms invest a fraction of past sales, $RD(t) = \nu S(t-1)$ into R&D. Innovation success is then a draw from a Bernoulli random vector $I^S \sim \text{Ber}(\theta)$ where $I_i^S \perp I_j^S \forall i \neq j$, and $\theta = 1 - e^{-\xi RD(t)}$. If innovation is successful, then coefficients are updated from a random draw from a beta distribution, as per standard DSK.

DSK also includes an imitation process, where firms have a probability of setting their coefficients to that of a close neighbour based on a random probability. This process was harder to vectorize and is not fundamentally different from innovation, so in VDSK we incorporate imitation through a correlation in the innovation vector. Intuitively, innovative firms are likely to be copied, resulting in a higher probability of innovations elsewhere. We add this by adding a correlation, ρ , to the Bernoulli innovation success draw. This is detailed in §5.1.6.

Innovations don't necessarily have to be positive in each dimension. They can include productivity increases that reduce environmental friendliness (e.g. our recent uptick in AI use). Firms choose which machinery to produce based on a payback period, detailed in appendix A.

Finally, the profit of capital good firms is: $\Pi(t) = (p(t) - c(t)) \odot Q^{cap}(t) - RD(t)$.

5.1.2 Consumption good firms

Consumption good firms use machinery produced by capital good firms to produce a good consumed by the population. They own a capital base of machines, $K^{cons}(t)$, consisting of machines of various vintages. Each machine vintage encodes the technology capital good firms were making at the time (through their coefficients $\{A|B\}^{L|EE|EF}(t)$). This could be the agricultural industry, for example, producing food from agricultural machinery produced by capital good firms.

Consumption good firms calculate the orders they expect according to $Q^d(t) = D^e(t) + N^d(t) - N(t-1)$, where $D^e(t)$ is the expected demand (set to the last period's demand), $N^d(t)$ is the desired inventory, and $N(t-1)$ is leftover inventory from the last period. This leads to the desired level of capital of $K_d^{cons}(t) = Q^d(t)$. The actual amount that firms can produce is limited based on their available capital (the stock of machinery they have purchased so far), their labour force allocation, and energy allocation (see §5.1.3 for labour and §5.1.4 for energy) as per equation 5.3. Firms then calculate an expected investment to reach their desired level of capital as $EI(t) = \max(K_d^{cons}(t) - K^{cons}(t), 0)$. The authors [2] also did not specify a mechanism for consumers to choose the firm they consume from, so in VDSK demand is allocated proportionally to $1/\sqrt{p(t)}$.

$$\hat{Q}^d(t) = \min(K^{cons}(t), L_{cons}^S/C_L, E_{cons}^{available}(t)) \quad (5.3)$$

Firms keep track of the set of coefficients across each one of their vintages and replace

vintages older than J , so they only need to track $A_{L,EE,EF}(t) \in \mathbb{R}^{M \times J}$. We compute replacement in a vectorized way by knowing that the vintage in the current step is $t \bmod J$ in this matrix, and updating that column per step.

We compute consumption good firms cost through a weighted average of the machines they own. Say the fraction each firm owns in each vintage is $M(t) \in \mathbb{R}^{N \times \eta}$, such that if $e_I \in 1^I$, then $M(t)e_\eta = e_M$ and $K(t) \odot M(t)e_\eta = K(t)$. Then we can compute the total unit cost of production as:

$$c(t) = w(t)[(A_L(t) \odot M(t))e_\eta]^{-1} + c_{en}(t)[(A_E E(t) \odot M(t))e_\eta]^{-1} \quad (5.4)$$

Firms also replace machines when better technology becomes available, when the price earned by new machines $p^*(t)$, their cost $c^*(t)$, and weigh it against their current weighted cost $c(t)$. They buy new machines when $p^*(t) - b(c(t) - c^*(t)) \leq 0$. We compute a mask: $M_R(t) = [p^*(t) - b(c(t) - c^*(t)) \leq 0] \in \mathbb{R}^{M \times J}$, and calculate new orders as: $\text{NewOrders}(t) = ((M(t) \odot M_R(t))e_\eta) \odot K(t)$. Total investment is then: $I(t) = EI(t) + \text{Replacement}(t) + \text{NewOrders}(t)$.

Pricing is then computed through a straightforward vectorization of the mechanism described in appendix A. The price consumption good firms charge is calculated from this markup and average cost as $p(t) = (1 + \mu(t))c(t)$.

Consumption good firms can also take on debt to meet their required investments, $I(t)$, only partially funded by net wealth. Say $NW(t)$ is the net wealth of all consumption good firms. The total debt ceiling the bank sector is allowed to give out is $MTC(t) = e^T NW(t)$, and each firm is also similarly constrained on the amount of debt they can take on by the sales to debt ratio, $\alpha(\text{Deb}(t) \odot S(t)^{-1}) = \Lambda(t)$. The original specification did not detail how to handle the debt ceiling being reached, so in VDSK we specify $\hat{c}(t)$ as an adjusted fraction of investments coming from debt. This can be computed from the desired loan amount vs. the actual loan amount. Say that $lev(t) = e^T \text{Deb}(t-1)$, $MTC(t) = e^T NW(t-1)$ and $req(t) = e^T I(t)(1-c)$, then:

1. If $req(t) + lev(t) \leq MTC(t)$ then all loan requests can be fulfilled
2. Else we fulfill $c_1(t) = \frac{MTC(t) - lev(t)}{req(t)}$ of requests
3. Now at the firm level, if $\text{Deb}(t-1) + (1-c)I(t) \geq \Lambda(t)$ then we fulfill $\frac{\Lambda(t) - \text{Deb}(t-1)}{(1-c)I(t)} = c_2(t)$ of requests
4. So finally we fulfill $\tilde{c}(t) = \min(c_1(t), c_2(t))$ of $(1-c)I(t)$, so $\hat{c}(t) = 1 - \tilde{c}(t)(1-c)$

Lamperti et al [2] also did not specify a mechanism to pay off debt, so we pay off a fraction, b , from profit. We can then have the debt update as $\text{Deb}(t) = \text{Deb}(t-1) + (1 - \hat{c}(t))I(t)$, and, given a profit of $\Pi(t) = (p(t) - c(t)) \odot Q_L(t) - r\text{Deb}(t)$, we compute final net wealth as $NW(t) = NW(t-1) + (1-b)\Pi(t) - \hat{c}(t)I(t)$.

5.1.3 Labour market

To compute the wage, we need 3 scalar inputs: $\hat{A}B(t)$, the average worker productivity, $cpi(t)$, the average price inflation, and $U(t)$, the unemployment rate. The calculation of $cpi(t)$ and $\hat{A}B(t)$ was omitted from the original specification, so we infer it in VDSK as:

$$\hat{A}B(t) = \frac{p^{cap}(t)^T Q^{cap}(t) + p^{cons}(t)^T Q^{cons}(t)}{L^D}$$

$$cpi(t) = \frac{e_N^T p^{cap}(t)}{N} + \frac{e_M^T p^{cons}(t)}{M}$$

Published papers specified that labour was demanded by firms, but neglected to mention how such demand is calculated. Pereira's implementation is highly complex and well beyond any published details [86], so in VDSK, we proxy it using a coefficient of labour, C_L , representing how much labour is needed per unit of output, and the amount of output produced in the last period by consumption good, capital good, and energy firms. If the labour demand exceeds the supply, then it is prorated according the last output produced by industry, and output is constrained by the amount of labour available as: $Q_{industry}^{max}(t) = \max(K_{industry}(t), L_{industry}^S(t))$

$$L^D = \max\left(e_N^T \frac{Q^{cap}(t-1)}{C_L} + e_M^T \frac{Q^{cons}(t-1)}{C_L} + \frac{Q^{energy}(t-1)}{C_L}, L^S\right)$$

Finally the unemployment rate is calculated as: $U(t) = L^S - L^D$. Here L^S is the labour market supply which is calculated with a static growth rate over time from an initial amount: $L^S(t) = (1 + g)^Y L^S(0)$.

5.1.4 Energy industry

Almost everything in the energy industry is a single scalar agent, so vectorization is only used to deal with dirty power plants of varying vintages. We have at most F dirty power plants, with $A_{de}(t) \in \mathbb{R}^{1 \times F}$ being the coefficients of production efficiency, and $em_{de}(t) \in \mathbb{R}^{1 \times F}$ the emissions the dirty plant produces. We again use $t \bmod F$ as the current index of machines we operate on. Say we have $M_{de}(t) \in \mathbb{R}^{1 \times F}$ as the normalized capacity across the power plants such that $(M_{de}(t) \odot A_{de}(t))e_F = \hat{A}_{de}(t)$. Then, the realized cost of the dirty plants is then: $\hat{c}(t) = p^*(t)/\hat{A}_{de}(t)$, where $p^*(t)$ is the cost of the fossil fuel used to generate electricity, and total capacity is $K_{de}(t) \in \mathbb{R}^{1 \times 1}$.

Now capacity is replaced every F periods, and we expand capacity if there is excess demand over current capacity, in which case we create new capacity using green plants if the cost of green plants is lower than the cost of dirty plants $IC_{ge}(t) \leq b\hat{c}_{de}(t)$, or else dirty power plants (for free). The innovation process proceeds exactly as described in the DSK model in appendix A. We can compute the effective emissions coefficient across the vector of dirty plants as above: $(M_{de}(t) \odot em_{de}(t))e_F = \hat{em}_{de}(t)$.

Another thing missing from published DSK papers is how lack of energy capacity affects output, or how firms utilize it. Similar to the labour market, VDSK prorates energy capacity across industries based on the last period's output, and max output is limited by $Q_{industry}^{max}(t) = \max(K_{industry}(t), E_{industry}^{avail}(t))$.

5.1.5 Climate box

The climate box proceeds as per our interpretation the DSK specification, in appendix A, with the exception of how emissions are calculated, which is vectorized. We have the following environmental friendliness coefficients for each agent type:

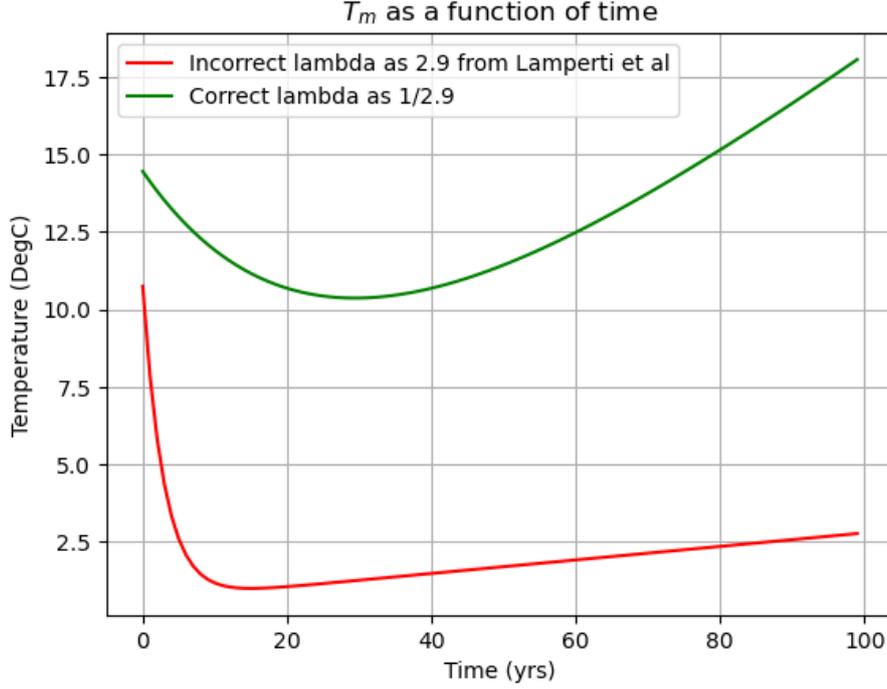


Figure 5.1: Temperature as a function of time with emissions growth of 1.2% per year over 100 years. Incorrect λ used in Lamperti et al causes temperatures to plunge and stay low. Even with corrected coefficient, the model shows temperature plunging until emissions increase it again after 20 years.

1. In a consumption good firm, we have $A_t^{EF} \in \mathbb{R}^{M \times J}$
2. In a capital good firm, coefficients are $B_t^{EF} \in \mathbb{R}^{N \times 1}$
3. In the energy sector, we have $e\hat{m}_{de}(t) \in \mathbb{R}^{1 \times F}$

Then, given the output vectors across the different industries, $Q^{cap}(t)$, $Q^{cons}(t)$, $Q^{de}(t)$, total emissions are:

$$em(t) = Q^{cons}(t)^T (M(t) \odot A_t^{EF}) e_J + e_N^T (Q^{cap}(t) \odot B_t^{EF}) + Q^{de}(t) e\hat{m}_{de}(t)$$

One significant error in how the DICE-2013R [91] temperature model is implemented uses an incorrect coefficient of 2.9 for λ in equation A.1, which should instead be $1/2.9$. This causes temperatures to plunge and stay low, although even when corrected temperatures initially plunge before recovering, seen in figure 5.1.

5.1.6 Imitation process

The innovation process samples from an uncorrelated Bernoulli vector with coefficients $\theta = 1 - e^{\xi RD(t)}$. In the traditional DSK model, the exponent is the amount invested in innovation and imitation respectively. In VDSK, we replace it with the total R&D expenditure, and instead add a correlation to the multivariate Bernoulli distribution $Ber(\theta)$.

This is achieved by defining a Gaussian standard normal vector $Z \sim N(0, 1)$ and a static correlation $\rho \in \mathbb{R}^{1 \times 1}$. We construct the covariance matrix using the static correlation parameter ρ assuming unit variances, and then sample from $Z' \sim N(0, \Sigma_\rho)$. A Bernoulli

draw can then be simulated from this by first determining a threshold $\tau = \Phi^{-1}(\theta)$, and then defining the Bernoulli draw as $1_{\{z \geq \tau\}}$.

5.2 Incorporating Dasgupta into VDSK

There are some crucial similarities between the Dasgupta framework presented in §4, and the DSK model. In particular, the output $Y(t)$ which measures the amount of economic output produced by the economy in the Dasgupta framework is directly observable as the sum of the machinery, consumption good, and energy firms output ($Q_{cons} + Q_{cap} + Q_{energy}$). The produced capital, $K(t)$, will effectively be the total value of machinery produced by the capital goods industry, used by the consumption goods industry. We don't have other forms of capital in this model, such as buildings, roads, or otherwise. Human capital, $H(t)$ can be measured using a variety of approaches, but a common way is the income approach [92], proxied by simply sum the wages across the labour force.

Now, two key components of the Dasgupta framework are not present in the DSK model, which we will propose modifications for in this chapter and detail the additional research needed to operationalize fully. These are:

1. The biosphere is represented as a single scalar value $S(t)$. The biosphere's dynamics are driven by three things:
 - (a) The natural dynamics of $S(t)$, driven by the growth rate $G(S(t))$
 - (b) Resources extracted from the biosphere to manufacture goods and machinery, $R(t)$ in Dasgupta's notation
 - (c) Waste dumped back into the biosphere from our consumption, $Y(t)/\alpha_z$ in Dasgupta's notation
2. Changes in population sizes as a function of human capital per person. The cross dynamics between the economy, the population size, and the biosphere is an important second-order effect: a degrading biosphere causing extreme weather events, desertification and natural disasters would reduce economic output which would reduce the target population size, reducing the labour available for the economy.

These two components form the basis of the modifications we make to the VDSK model to construct the Vectorized Dystopian Schumpeter meets Keynes, with Biodiversity (VDSK-B) model. This will be a powerful ABM capable of endogenously modelling both biodiversity and climate change and the economy's co-evolution with them.

5.2.1 Modifications to incorporate Dasgupta

Introducing biodiversity into a model of climate economics is very non-trivial and requires a lot of design and parameter choices to match reality. In this section, we lay out a framework in which such choices can be made. However, actually making the correct choices is beyond the scope of this report as it will require careful investigations into the biosphere, how it evolves over time, to what extent and how waste and natural resource extraction affects it, etc.

The biosphere: $S(t)$

The critical addition to the DSK model is the addition of Dasgupta's single scalar parameter, $S(t)$, "heroically" representing the entire biosphere. The requirements for this

parameter are that:

1. It is proxies well the ability of our biosphere to sustain life
2. Biosphere dynamics (growth and decline rates) are well captured and explained
3. Resource extraction linearly negatively affects the global growth rate of the parameter
4. Waste and pollution also cause linear declines in the global growth rate of this parameter

There is significant supporting literature that both 1. and 2. are very well proxied by the amount of carbon in our biosphere [93, 94, 95, 96].

The effects of resource extraction are more complex; certain activities like forestry both directly deplete carbon in the biosphere, and have knock-on impacts on biodiversity from habitat loss, whereas activities like mining deplete ecosystem services indirectly. In both cases, literature is supportive of carbon loss acting as an (imperfect) measure of biosphere damage from resource extraction since the impacts tend to be damage to the local ecosystem that degrades its ability to sustain life and store biomass [97, 98].

Biosphere carbon is also an appropriate measure for the degradation of the biosphere from waste and pollution. Waste comes in many different forms; agricultural waste for instance includes fertilizer run-off, soil degradation, biocides and pesticides entering the water table, all of which degrade the ability of the local ecosystem to sustain life [99]. The deposits of persistent chemicals and plastics into the environment actively damage flora and fauna and their ability to reproduce [100]. All of these reduce the amount of biomass in the biosphere.

As a result, we augment VDSK-B with an additional state variable, $S(t)$, in the climate box, representing the GtC present in the biosphere, initially set to 3,700 GtC as per Managi [101].

Resources extracted from biosphere, and pollution

Looking at equation 4.1, there are two detractors from the natural growth of the biosphere. Resources extracted from the biosphere, $R(t)$, are measured by Dasgupta as $Y(t)/\alpha_x$ through the coefficient α_x . Pollution and waste flow back to the biosphere, degrading its growth rate through $Y(t)/\alpha_z$.

We also note the parsimony between now Dasgupta models the effect of waste and resource extraction, $Y(t)/\alpha$, and how DSK models emissions, Q_{cons}/A_{EF} . Therefore, propose to model this within the VDSK-B model by introducing a set of new coefficients:

- A_R, B_R, A_R^{de} : These model the efficiency with which resources are extracted from the biosphere to machine manufacture (A_R) and consumption good manufacture (B_R , a function of machinery's ability). and dirty power plants from the extraction of fossil fuels.
 - The removal of carbon from the biosphere is then Q_{cons}/B_R for consumption good firms, Q_{cap}/A_R for capital good firms, and Q_{energy}^{de}/A_R^{de} for dirty power plants.
- A_W, B_W, A_W^{de}, C_W : These model the biosphere exhaustion coming from waste and pollution from capital good firms, consumption good firms, dirty power plants, and

household consumption.

- The reduction in biosphere growth rate is then $Q_{cons} \frac{C_W + B_W}{C_W B_W}$ for consumption good firms and households (who consume what consumption good firms make and throw away plastic and other waste), Q_{cap}/A_W for capital good firms, and Q_{energy}^{de}/A_W^{de} for dirty power plants.

These coefficients are subject to same innovation process as in VDSK that accumulates innovation capital, $A(t)$, over time. The calibration of the initial value of coefficients, as well as the impact of innovation processes, are key to using this mechanism. However, this is beyond the scope of this work and is left for future work.

Integration into pricing

Biosphere degradation (and also emissions) can be incorporated in the pricing of consumption and capital goods products by modifying the cost equations 5.1 and 5.4 to:

$$c_{cons}(t) = \frac{w(t)}{B_L} + \frac{c_{en}(t)}{B_L} + \phi_1^{cons} \frac{1}{B_{EF}} + \phi_2^{cons} \frac{1}{B_W} + \phi_3^{cons} \frac{1}{B_R} \quad (5.5)$$

$$c_{cap}(t) = \frac{w(t)}{A_L} + \frac{c_{en}(t)}{A_L} + \phi_1^{cap} \frac{1}{A_{EF}} + \phi_2^{cap} \frac{1}{A_W} + \phi_3^{cap} \frac{1}{A_R} \quad (5.6)$$

This is extraordinarily powerful since a key control variable in Dasgupta is the aggregate consumption, $C(t)$, and no mechanism to control it exists in the DSK model. By introducing coefficients $\phi_{1,2,3}^{cons|cap}$ not only do we introduce a mechanism to control consumption through pricing, and a target for optimization via SalVO.

Biosphere dynamics

We now incorporate $S(t)$ into the Climate Box. At each step, the growth rate of the biosphere stock, $G(S(t))$ is calculable using Dasgupta's proposed formulation. The drain to this growth rate comes from resources extracted and waste produced from the previous step. The final state equation for the biosphere update is:

$$S(t+1) = \underbrace{G(S(t))}_{\text{Growth rate of biosphere}} \quad (5.7)$$

$$- \underbrace{Q_{cons} \frac{B_W + C_W}{C_W B_W} - \frac{Q_{cons}}{B_W} - \frac{Q_{cap}}{A_W} - \frac{Q_{energy}^{de}}{A_W^{de}}}_{\text{Degradation from waste}} \quad (5.8)$$

$$- \underbrace{\frac{Q_{cons}}{B_R} - \frac{Q_{cons}}{B_R} - \frac{Q_{cap}}{A_R} - \frac{Q_{energy}^{de}}{A_R^{de}}}_{\text{Degradation from resource extraction}} \quad (5.9)$$

DSK also contains a term for Net Primary Production coming from atmospheric CO2 absorption into the biosphere. The goal of this term is to update the atmospheric CO2 concentration, not to actually model changes to biomass over time. As a result, while it captures atmospheric CO2 updates, it doesn't necessarily truly model biomass changes accurately; the model is based on models of the carbon cycle from Goudriaan [94] and

Oeschger [102] and doesn't include biomass from fauna reproduction, etc. Therefore, we suggest modelling $G(S(t))$ as:

$$G(S(t)) = \hat{r}S(t)\left(1 - \frac{S(t)}{\underline{S}}\right)\left(\frac{S(t) - L}{\underline{S}}\right) + NPP(t) \quad (5.10)$$

Here the coefficient \hat{r} needs to be calibrated to included components not coming from NPP, the selection is beyond the scope of this report. Adding $NPP(t)$ into this has the added benefit of integrating global temperature into biosphere dynamics.

Climate shocks to the economy

Degradation of the biosphere should lead to similar shocks as seen from rising temperatures in the DSK model, as a result of effects such as desertification leading to droughts and food shortages. We can therefore make the climate disaster generating function a function of both temperature and $S(t)$, so disasters get worse if biosphere stock gets depleted and if temperature worsens.

Realistic damage functions are one the key items recognized as not adequately addressed by existing literature [18]. In VDSK-B we incorporate climate damage by extending the DSK temperature-based damage model by sampling damages from a bivariate beta distribution that is a function of both $S(t)$ and $T_m(t)$, sampled over the support $D \in [0, 0.5]^2$:

$$D \sim \beta \left(\left(\begin{array}{c} a_0^{T_m}(1 + \log T_m(t)) \\ a_0^S(1 + \log S(t)) \end{array} \right), \left(\begin{array}{c} b_0^{T_m}\sigma_{10y}^{T_m}(0)/\sigma_{10y}^{T_m}(t) \\ b_0^S\sigma_{10y}^S(0)/\sigma_{10y}^S(t) \end{array} \right) \right) \quad (5.11)$$

The key parameters \hat{r}, a_0^S, b_0^S need to be calibrated (beyond report scope). The shocks are then applied as $e_2^T D$.

This misses several important considerations in the Dasgupta report, such as the ineffable value of the beauty of nature, and that loss of biodiversity equates to loss of life, a concept entirely separate from economics. However, given the bounds of the DSK model, we can keep the scope limited for now.

Population

Population size in DSK is calculated using a constant growth rate, g , as $N(t) = (1 + g)^t N(0)$. In VDSK-B, to incorporate human population dynamics from Dasgupta, we will make g a function of the unfilled consumer demand, i.e. the demand made for goods from the population L^S on consumption good firms, Q_{demand} , and the consumption good firms ability to meet them based on their machine's productivity and available labour and energy, Q_{cons} . There is evidence that while the human population grows in the presence of plentiful resources, there is a subsistence level below the ideal needs of the society beyond which it will start to shrink [85], f_s . We will use a logistic function to model this as:

$$g(t) = \frac{G}{1 - e^{k(1 - f_s Q_{demand}/Q_{cons})}} - \frac{G}{2} \quad (5.12)$$

Whenever supplied consumption goods exceed the minimum subsistence needed by consumers, the population will grow up to G percent per year, and vice versa. With a variable population size, the simulation should result in interesting dynamics where shocks from

climate disasters lead to a worse economy, lower wages and a smaller population target size and less demand in the future for companies. This will have to be very carefully designed and tested in future work.

Chapter 6

Evaluation

6.1 VDSK-B

6.1.1 Implementation

The VDSK-B model from §5 was implemented in SalVO by first deriving a vectorized form of the DSK model, and implementing it using the same initial parameter configurations as those given in Lamperti et al [2]. The only exception is the innovation correlation parameter, ρ , which was arbitrarily set to 0.1. This implementation was exceedingly complex, ending in the creation of an ABM with 132 different state components. However, table 6.1 shows SalVO’s succinctness and ease of use; the VDSK model was implemented in only 2104 lines of code (LOC), compared to the 9058 needed by the open source implementation, which is albeit more complex with a more nuanced climate model, banking sector and labour allocation mechanism [86].

The complexity of the model is significant and the description of the implementation in Lamperti et al [2] is insufficient. As a result, the outputs of the model, while reasonable, do not match the expectations set by the paper or Pereira’s implementation of the DSK model. The differences need further investigation to determine if they are coming from flaws in the implementation or from the added complexity in Pereira’s implementation.

The biosphere modifications inspired by the Dasgupta report in §5.2 were also introduced. These were successfully included in a vectorized way to make the VDSK-B model, and the additional LOC and time benchmarks are included in table 6.1. However, the results of the model are useless without the correct set of parameters needed for the additional components in §5.2, which is beyond the scope of this work. Therefore, while the model was implemented, showing SalVO’s speed and ease of use, its calibration and simulation are left to future work.

Model	LOC	Time taken (s)	5× bigger time taken (s)
SalVO - VDSK-B	2104	7.24 ± 0.29	7.49 ± 0.64
LSD - Pereira DSK [86]	9058	14.13 ± 0.64	151.34 ± 2.82

Table 6.1: Lines of code (LOC) and the average time taken across 5 runs of VDSK and VDSK-B in SalVO, and Pereria’s DSK implementation in LSD. A model with the number of agents increased by 5x was also tested and timed.

Benchmark	Rubric	SalVO	Agents.jl	Mesa*	FLAME (GPU)
Circles [60]	Time/iteration (ms)	0.3 ± 0.1	-	-	22.2 ± 1.8
Flocking [45]	Total sim. time (ms)	24.0 ± 3.8	184.7 ± 21.2	23,974.1	-
Forest Fires [50]	Total sim. time (ms)	146.2 ± 21.7	445.7 ± 32.8	136,842.2	-

Table 6.2: Running times across example ABMs across frameworks. SalVO executed on the GPU with JAX. ABM configurations are identical to cited sources. Note that JIT compilation times were excluded from the SalVO benchmarks. *Mesa times are inferred from benchmarks in [45].

6.1.2 Speed and scale

As seen in table 6.1, the SalVO implementation of VDSK-B running on the GPU is exceedingly fast, running 2x as fast as Pereira’s implementation in LSD, which compiles a C++ binary from a DSL (using all 20 logical processors for execution). The benchmark was generated for identical initial configurations, with 50 capital good agents, 200 consumption good agents, and a labour force of 3000 at the start.

The remarkable observation in table 6.1 is that scaling the model by $5\times$ to 250 capital good firms, 1000 capital good firms and 15,000 consumers, causes a $10\times$ slowdown in LSD, but since it can still easily fit into GPU VRAM, hardly changes the SalVO runtime. The additional overhead in memory transfer barely registers. Once the GPU runs out of memory, SalVO may not be able to complete the computation without additional batching, but this demonstrates the strong scalability achieved by SalVO for single-machine simulations. While not explicitly tested, this also shows how efficient multi-device scaling will be in SalVO.

6.2 SalVO evaluation

6.2.1 Speed and hardware acceleration

A critical design requirement for SalVO is that it can utilize vectorization and subsequent hardware acceleration to achieve phenomenal speed and scale on a single machine. Table 6.2 shows the results of benchmarks for the example ABMs defined in §3.5.4. In all cases, SalVO is significantly more performant, with the difference vs. FLAME (GPU) for circles particularly remarkable. A key reason for this are optimisations such as operator fusing applied by the JAX JIT compiler, as well as much more full GPU utilization achieved by JAX over FLAME.

For other ABMs, Agents.jl achieves remarkable performance, but since it lacks hardware acceleration, it is unable to reach the speeds SalVO can do for both the Flocking and Forest Fires ABM. This combined with the VDSK runtimes shows SalVO’s ability to take advantage of vectorization and hardware acceleration to reach significant speeds.

Tests on multi-device scalability, unfortunately, could not easily be performed due to the lack of time and a multi-GPU setup. In theory, SalVO should be able to scale better than most frameworks from its use of JAX’s data sharding and computation distribution framework, but these have not been evaluated in this work.

6.2.2 Learning

This section evaluates the application of the backpropagation and policy gradient modules in SalVO to training example ABMs. Unfortunately, due to time constraints, training could not be applied directly to the VDSK-B model since it first needs research on parameter choices, but they are instead applied on ABMs in §3.5.4.

Backpropagation

The bird flocking ABM from §3.5.4 was used to test the application of backpropagation to ABM training. Note this ABM is differentiable in SalVO since JAX [82] supports auto-differentiation of the max function except at the discontinuity. Since previous work [61] showed the application of backpropagation (outside of SalVO) in an optimisation context by maximizing GDP without any specific target, this work shows backpropagation within SalVO applied to a calibration task. Our objective is seemingly simple but actually very hard: to move the centroid of the flock to a target specified by the user.

In particular, if $\mathcal{S}_t \in \mathbb{R}^{N \times 2}$ is the vector of bird locations, $C \in \mathbb{R}^{1 \times 2}$ the target, and e the identity vector, then we set our objective function as:

$$\mathcal{L} = \sum_{t \in T} \left\| \frac{e^T \mathcal{S}_t}{N} - C \right\|_2 \quad (6.1)$$

We target $\alpha \in \mathbb{R}^{T \times N}$ (see §3.5.4) as the optimisation target, with $T = 100$ and $N = 500$, resulting in 50,000 parameters. Changing the centroid of the flock is **extremely** non-trivial: as birds collapse to the centroid, their velocities reach 0, so no choice of α moves them. The only choice the ABM has to optimize equation 6.1 is to move the centroid itself, rather than individual birds.

Backpropagation is applied through SalVO by specifying a `backpropagation.py` file. Since ABMs are very "deep", we apply clipped gradient descent to control explosive gradients. Parameters such as `learning_rate`, `clips`, `training_iterations` are specified here, as is a function `update`, which applies the gradient descent step (allowing for custom optimizers). It also contains the loss function and a `targetspec` which contains the state-flow for the loss function and the target of the optimisation.

The results can be seen in figure 6.2, with the training curve and learnt α policies seen in figure 6.1. The achievements are remarkable: SalVO is able to learn all 50,000 parameters with a very non-trivial optimized policy, where certain birds are flung out away from the centroid in figure 6.2 so that the centroid itself moves towards the target, before grouping them back. The only bird with the 0 alpha is the one at the centroid already. The remaining birds have their alpha collapse to the initial value of 0.1. This shows both the optimized policy creation and calibration capability of SalVO.

GP-ABM

The Forest Fires ABM was used to demonstrate learning using the GP-ABM framework from §3.4.3. Since fires spread from adjacent trees, a dense forest (i.e. a strongly connected set of trees) will always burn down. As seen in figure 6.3, as the forest regrows, it either reaches an equilibrium with the rate of Forest Fires, or it regrows so quickly that it starts to form dense patches which again cause the forest to burn down. We want to find a regrowth probability, p , that maximizes the average amount of trees alive through the

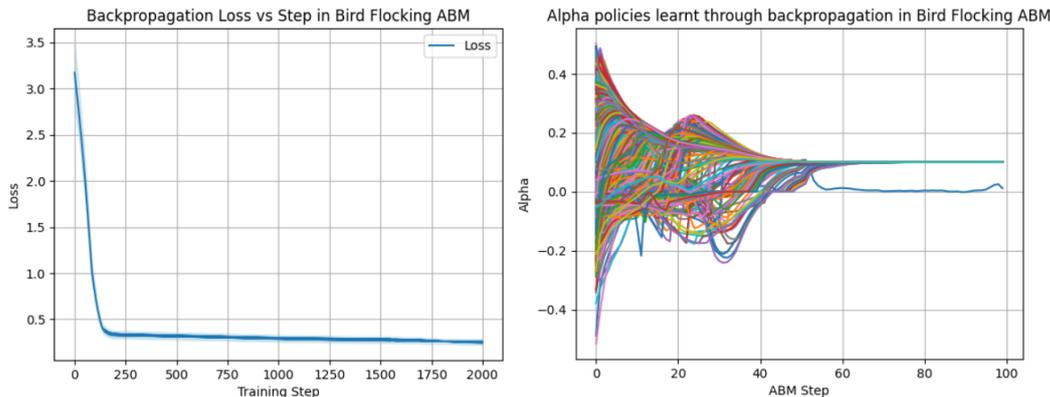


Figure 6.1: [Left] The training curve over 10 iterations. [Right] Learnt α policies for all individual birds.

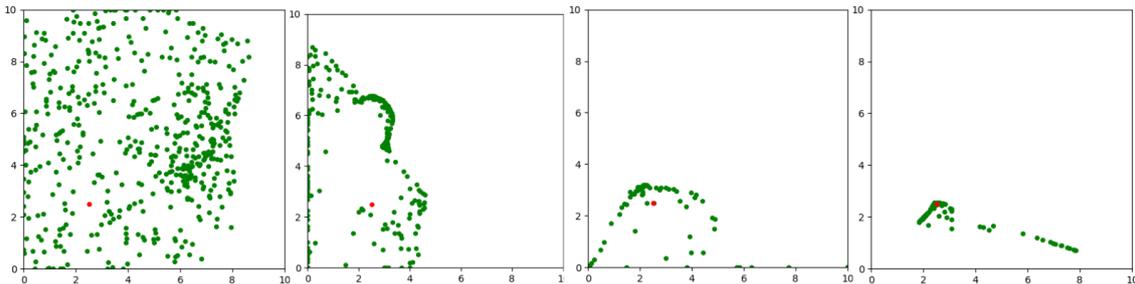


Figure 6.2: Bird flocking optimized to move the centroid from the center to the red dot. SalVO learns to cast birds far to move the centroid, before collapsing at the target.

forest’s history. This will be such that it doesn’t cause dense forests that burn down, and not so little that we haven’t yet reached equilibrium.

We imbue each tree with a static probability of regrowth $p \in [0, 1]^N$ for 10,000 trees. The GP-ABM is then learning 10,000 parameters. Now, this ABM is both stochastic and non-differentiable, leading to backpropagation not being a viable option. However, it is, as per Grazzini [62], ergodic and has an absorbing equilibrium for small values of p . I.e. the number of trees alive is dependent only on its parameters and has a stationary distribution, as shown by figure 6.3. This satisfies William’s conditions [4] for reward stationarity, enabling us to apply the GP-ABM.

We follow the protocol laid out in §3.4.3. We fit a Gaussian Process using agent-level data on the tree state, $S \in 0, 1$: $S \sim GP(\mu(p), \sigma(p))$ using the RBF kernel. We then use this calibrated GP to generate a density given the value of p for an agent: $g(S)|_{p=p_i}$. We then use JAX’s auto-differentiation to calculate $\partial \ln g / \partial p$. We further use a dynamic baseline in REINFORCE from Weaver and Tao [103]. Note that we use a GP that predicts a distribution over a single tree’s state given p to optimize the overall average number of trees alive.

The results of this calibration can be seen in figure 6.4, where the training curve clearly shows an increase in the average number of trees alive up to an asymptote. The learnt trajectories show a clear, smooth convergence over time to 0.09 for all 10,000 parameters. The outcome can be seen in figure 6.3, where a $p = 0.15$ leads to unstable behaviour. Now, it’s easy to see that a stable packing of trees with none adjacent would fit 2,500 trees in a 100x100 grid. Our GP-ABM reaches 2150 trees. The stable packing is only reachable with $p = 0$ for trees that should stay dead. This is discussed more in §7.1.2.

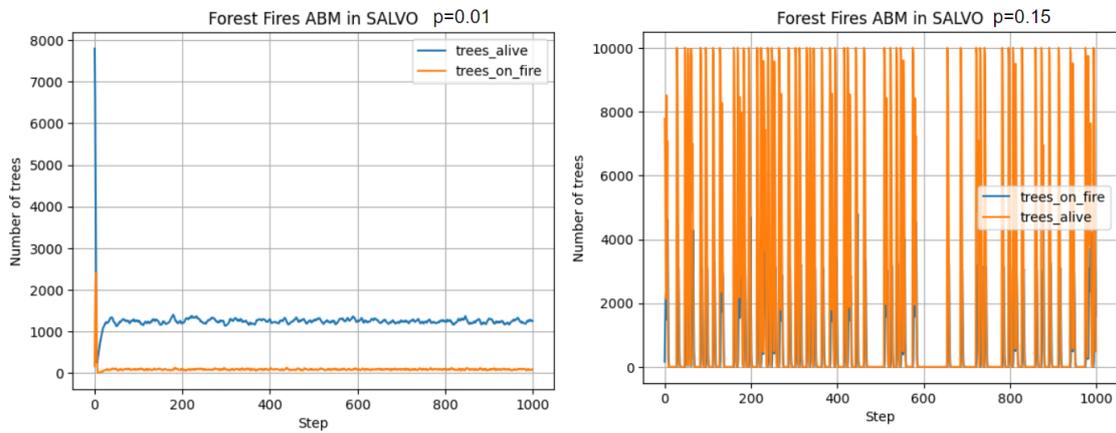


Figure 6.3: [Left] Trees alive and on fire under baseline settings. [Right] Trees alive/on fire under an unstable setting, where dense forest regrowth triggers full burns.

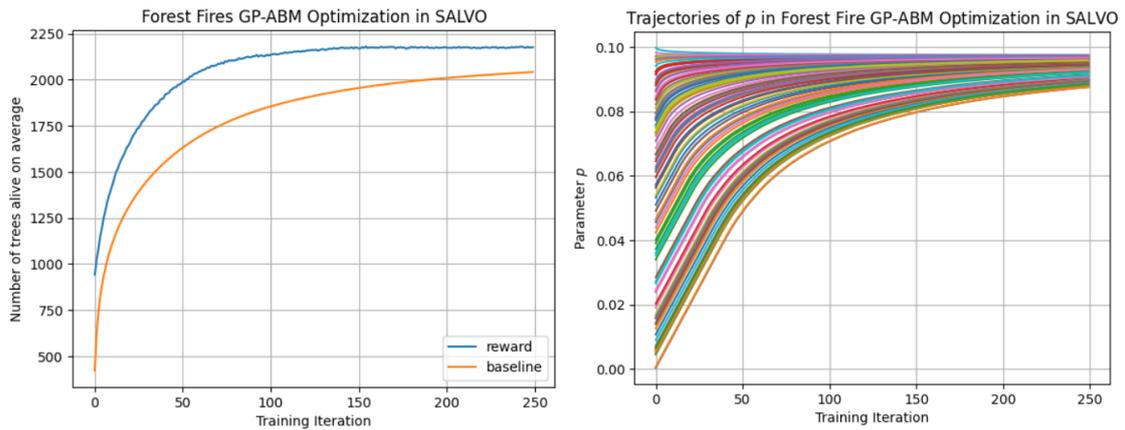


Figure 6.4: [Left] Training curve of the GP-ABM. [Right] How p changes for each agent with training, from its initial distribution from $Uniform(0, 0.1)$.

Chapter 7

Discussion, conclusion and future work

7.1 Discussion and criticism

7.1.1 VDSK-B

The VDSK-B model creates crucial linkages through the tracking of the additional climate state variable, the biosphere, $S(t)$, from Dasgupta [1] and links it to the economy with the climate through the use of a joint damage function. It is able to represent nearly every crucial element of the framework laid out by Dasgupta, including population dynamics, the biosphere stock levels and dynamics, waste, and resource mining. The integration of it into VDSK, along with the success of SAVLO's optimizable policy design suite, unlocks powerful potential studies using Dasgupta's key control variables of consumption patterns (through price controls), investment in innovation and human capital (through unemployment wages), and limitations on resource mining.

However, the model is extraordinarily complex and difficult to implement. This work tried to implement the model using descriptions from Lamperti et al [2], and while a model was successfully designed and implemented, it doesn't display behaviour in line with published results or Pereira's implementation [86]. For instance, in VDSK prices generally fall as innovation happens, but they tend to trend sideways in Pereira's implementation. This could be due to implementation issues in VDSK but is more likely from incorrect extrapolations, necessary to fill in missing details in the DSK-published papers. Much more work is needed here to understand the gaps and close them. Similarly, VDSK-B requires research on parameters matching reality for stable simulations.

The lack of useful implementations is a stumbling block for future research, but the successful implementation of VDSK and VDSK-B into SalVO, and the simulation performance achieved, clearly demonstrate SalVO's ability to handle highly complex ABMs. While the speed isn't a totally fair comparison (since Pereira's implementation contains more complexity than described in Lamperti's original work) it does not contain twice as much complexity, and the results at a $5\times$ scale are incontrovertible.

7.1.2 SalVO

SalVO's implementation based on the ABM formalism in definition 1 shows clear benefits. The expressivity of the formalism is borne out by the numerous examples of ABMs, and

the highly complex VDSK, implemented successfully in SalVO. §6 contains clearly positive results when evaluating SalVO against its design goals of being scalable, vectorized, and optimizable. In every test that was run, SalVO, using GPU acceleration, outperformed its closest competitors in terms of speed and scale on a single machine. The VDSK implementation ran in half the time of the LSD [56] library that compiles its DSL into a C++ binary and in 4% of the time taken by LSD with a $5\times$ scale up. It outperformed Agents.jl on the forest fire and flocking benchmarks by between 2-3x. It even outperformed FLAME’s GPU implementation on the Circles benchmark [60].

However, speed isn’t everything, and there are clear downsides to this framework. While SalVO is much faster by using hardware acceleration, these aren’t really fair benchmarks since neither LSD nor Agents.jl use GPUs. SalVO’s performance using JAX’s JIT on CPUs would be a more fair test, but this wasn’t formally assessed here due to time constraints. Furthermore, Agents.jl runs on the CPU not necessarily using its entire capacity across cores, which means that some simultaneous runs of an ABM model should be possible on a single machine. However, for parallel model runs in SalVO, we’d need likely multiple GPUs. This isn’t the case for LSD, however, which used all available CPU logical processors for its DSK simulation.

Furthermore, while SalVO claims to be scalable to multiple devices by leveraging JAX’s data sharding and pmap capabilities, there wasn’t sufficient time to assess this in this work. Additionally, SalVO’s performance is heavily dependent on the choice of GPU. For instance in §6.2.1 in the Circles benchmark, SalVO’s blazing speed is attributable to an operator fusion optimisation carried out by JAX’s JIT compiler. However, this also actually *prevented* single-machine scale in this case, since it caused the GPU to go out of memory at a mere 35,000 agents, whereas both FLAME and a previous TensorFlow implementation by the author [53] scaled to 100,000+ agents. Smaller GPUs with insufficient VRAM could end up being a bottleneck in SalVO’s performance, a fact not sufficiently explored due to time constraints.

One benefit of SalVO is that it is intuitive and easy to use; anecdotally a colleague was able to implement an ABM in SalVO unassisted in under an hour. Comparable frameworks like LSD/FLAME took the author several hours to just get familiar with the DSL. However, one key drawback is that ABMs in SalVO have to be expressed in a way that permits vectorization. This does require more thought than generic for-loops, but the results are well worth it.

SalVO did demonstrate a strong ability to learn parameters both for differentiable ABMs through backpropagation, and non-differentiable/stochastic ABMs using policy gradients. However, it is important to note that ABMs can be very “deep“ (i.e. have many steps). This can cause issues with vanishing or explosive gradients. The latter is handled using clipped gradient descent, but the former remains important to remediate. The backpropagation approach, while clearly powerful, can only be applied to differentiable, deterministic ABMs, a crucial drawback if not applied carefully.

The policy gradient-based approach overcomes both these limitations and was also seen to learn a large number of parameters successfully. It is also surprisingly sample efficient, requiring data from only a single run of the Forest Fires ABM to calibrate a useful Gaussian process successfully. This is because it was learning a GP for tree state as a function of the tree’s p , and so it obtained 10,000 observations from a single ABM run. However, this approach has clear drawbacks. It is only applicable in the case where the reward is stationary, otherwise, the assumptions behind William’s REINFORCE [4] would break down. It also needs all useful information to be encoded into the learnt probabilistic

model, which is impossible for ABMs. For example, by learning a GP for a single tree, we missed the joint effect of all the tree’s regrowth probabilities on ultimate tree states. So we were not able to learn a policy for p where there are no adjacent trees, leading to 2,500 trees alive, since it had no idea that setting some p ’s to 0 would enable others to survive. However, even by missing this crucial interaction from other agent parameters, it still was able to create an optimized policy.

In both learning approaches, this work is limited in that it only evaluated two simple ABM cases where they worked well. There was not sufficient time to properly assess a full gambit of ABMs, particularly in cases where they would fail or would be hard to implement. SalVO also doesn’t implement traditional calibration techniques such as simulated minimum distance or Bayesian optimisation natively yet.

7.2 Conclusion

To build an ABM that captures the integrated dynamics of the economy, climate and biosphere, this work recognized gaps in both existing ABM computation frameworks and current ecological economic ABM models. No existing ABM captures the interaction of these 3 elements. Building on Lamperti et al’s DSK model [2], a vectorized form, the VDSK, was derived. Using Dasgupta’s work [1], the biosphere was integrated into the ABM, creating the VDSK-B ABM.

Recognizing the need for large-scale calibration and optimized policy design to be an essential component of any ABM framework, this work built SalVO (§3). Based on an ABM formalism proven to be expressive enough to emulate programs in a Turing complete language, SalVO is a library in which backpropagation and policy gradient techniques from machine learning literature can be used for truly large-scale parameter calibration. Both proposed learning approaches need many evaluations of the ABM in order to generate optimized parameters, motivating the need for fast ABM simulations. SalVO achieved extraordinarily fast simulations by leveraging vectorization and using JAX’s [82] XLA compiler to generate kernels capable of hardware acceleration across CPUs, GPUs and TPUs.

Results showed that simulations in SalVO ran significantly faster than comparable frameworks, such as FLAME (GPU) [3]. Scaling ABMs on a single machine also showed significant promise, with a $5\times$ scale-up in VDSK resulting in a negligibly slower simulation. ABM training through backpropagation was also successfully demonstrated, with a very non-trivial policy learnt across 50,000 parameters to guide a flock centroid to a target location. Policy gradient-based training also successfully enabled parameter learning for stochastic, non-differentiable ABMs.

The VDSK-B ABM was successfully implemented in SalVO, demonstrating its ability to handle complex ABMs. However, simulation results don’t match those in published papers or open-source implementations, showing the need for further research for proper operationalization.

While SalVO shows much promise, there are many drawbacks to address, including implementing scalability across a network of devices. Policy gradient-based training (the GP-ABM protocol) contains many theoretical issues discussed in §3.4.3 which merit further investigation. A more thorough evaluation of SalVO across more complex ABMs is also warranted.

However, it is also the first ABM framework to introduce large-scale parameter learning

as a key feature for both differentiable, deterministic, and non-differentiable, stochastic ABMs. It unlocks a world of possibilities for optimized policy design and calibration, as well as easy scalability to large ABM sizes without the need for complex DSLs and specialist knowledge, democratising performant, scalable and optimizable ABMs for non-computer scientist researchers.

7.3 Future work

The clear avenue for future work is the proper operationalization of the VDSK-B model. The current implementation is based on descriptions in published works since Pereira’s implementation differs significantly. However, inconsistent results show the need for a deeper investigation. Furthermore, the VDSK-B model introduces a number of new parameters for the biosphere element which need calibration to real-world data before the model can be useful.

A fuller evaluation of SalVO’s training capabilities across different ABMs would be valuable to better explore the usefulness of the GP-ABM and backpropagation protocols. Scale to multiple devices using JAX’s data-sharding and pmap capabilities also needs full implementation and testing. Finally, SalVO’s GP-ABM protocol only works with stationary rewards; more powerful training tools are needed to relax these constraints further.

Additional further work is captured in appendix D

Bibliography

- [1] P. Dasgupta, *The economics of biodiversity: the Dasgupta review: full report*. London: HM Treasury, updated: 18 february 2021 ed., 2021.
- [2] F. Lamperti, G. Dosi, M. Napoletano, A. Roventini, and A. Sapio, “Faraway, So Close: Coupled Climate and Economic Dynamics in an Agent-based Integrated Assessment Model,” *Ecological Economics*, vol. 150, pp. 315–339, 2018.
- [3] P. Richmond and M. K. Chimeh, “FLAME GPU: Complex System Simulation Framework,” in *2017 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 11–17, 2017.
- [4] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Reinforcement learning*, pp. 5–32, 1992. Publisher: Springer.
- [5] J. Rockström, W. Steffen, K. Noone, Persson, F. S. Chapin III, E. Lambin, T. M. Lenton, M. Scheffer, C. Folke, H. J. Schellnhuber, and others, “Planetary boundaries: exploring the safe operating space for humanity,” *Ecology and society*, vol. 14, no. 2, 2009. Publisher: JSTOR.
- [6] F. S. Chapin III, E. S. Zavaleta, V. T. Eviner, R. L. Naylor, P. M. Vitousek, H. L. Reynolds, D. U. Hooper, S. Lavorel, O. E. Sala, S. E. Hobbie, M. C. Mack, and S. Díaz, “Consequences of changing biodiversity,” *Nature*, vol. 405, pp. 234–242, May 2000.
- [7] C. Parmesan and G. Yohe, “A globally coherent fingerprint of climate change impacts across natural systems,” *Nature*, vol. 421, pp. 37–42, Jan. 2003.
- [8] S. Díaz, J. Settele, E. S. Brondízio, H. T. Ngo, J. Agard, A. Arneth, P. Balvanera, K. A. Brauman, S. H. M. Butchart, K. M. A. Chan, L. A. Garibaldi, K. Ichii, J. Liu, S. M. Subramanian, G. F. Midgley, P. Miloslavich, Z. Molnár, D. Obura, A. Pfaff, S. Polasky, A. Purvis, J. Razzaque, B. Reyers, R. R. Chowdhury, Y.-J. Shin, I. Visseren-Hamakers, K. J. Willis, and C. N. Zayas, “Pervasive human-driven decline of life on Earth points to the need for transformative change,” *Science*, vol. 366, no. 6471, p. eaax3100, 2019. eprint: <https://www.science.org/doi/pdf/10.1126/science.aax3100>.
- [9] C. Turney, A.-G. Ausseil, and L. Broadhurst, “Urgent need for an integrated policy framework for biodiversity loss and climate change,” *Nature Ecology & Evolution*, vol. 4, pp. 996–996, Aug. 2020.
- [10] M. Bálint, S. Domisch, C. H. M. Engelhardt, P. Haase, S. Lehrian, J. Sauer, K. Theissinger, S. U. Pauls, and C. Nowak, “Cryptic biodiversity loss linked to global climate change,” *Nature Climate Change*, vol. 1, pp. 313–318, Sept. 2011.

- [11] P.-M. Boulanger and T. Bréchet, “Models for policy-making in sustainable development: The state of the art and perspectives for research,” *Ecological Economics*, vol. 55, no. 3, pp. 337–350, 2005.
- [12] H. Daly and J. Farley, *Ecological Economics: Principles and Applications*. 2nd edition ed., Jan. 2011.
- [13] J. Gowdy and J. D. Erickson, “The approach of ecological economics,” *Cambridge Journal of Economics*, vol. 29, pp. 207–222, Mar. 2005. _eprint: <https://academic.oup.com/cje/article-pdf/29/2/207/4768301/bei033.pdf>.
- [14] D. Pearce, “The Social Cost of Carbon and its Policy Implications,” *Oxford Review of Economic Policy*, vol. 19, pp. 362–384, Sept. 2003. _eprint: <https://academic.oup.com/oxrep/article-pdf/19/3/362/1208529/grg002.pdf>.
- [15] M. Golosov, J. Hassler, P. Krusell, and A. Tsyvinski, “Optimal Taxes on Fossil Fuel in General Equilibrium,” *Econometrica*, vol. 82, no. 1, pp. 41–88, 2014. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.3982/ECTA10217>.
- [16] J. E. Stiglitz, “Where modern macroeconomics went wrong,” *Oxford Review of Economic Policy*, vol. 34, pp. 70–106, Jan. 2018.
- [17] G. Fagiolo and A. Roventini, “Macroeconomic Policy in DSGE and Agent-Based Models Redux: New Developments and Challenges Ahead,” *Journal of Artificial Societies and Social Simulation*, p. 48, Jan. 2017.
- [18] J. D. Farmer, C. Hepburn, P. Mealy, and A. Teytelboym, “A Third Wave in the Economics of Climate Change,” *Environmental and Resource Economics*, vol. 62, pp. 329–357, Oct. 2015.
- [19] S. Barrett, A. Dasgupta, P. Dasgupta, W. N. Adger, J. Anderies, J. van den Bergh, C. Bledsoe, J. Bongaarts, S. Carpenter, F. S. r. Chapin, A.-S. Crépin, G. Daily, P. Ehrlich, C. Folke, N. Kautsky, E. F. Lambin, S. A. Levin, K.-G. Mäler, R. Naylor, K. Nyborg, S. Polasky, M. Scheffer, J. Shogren, P. S. Jørgensen, B. Walker, and J. Wilen, “Social dimensions of fertility behavior and consumption patterns in the Anthropocene,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 117, pp. 6300–6307, Mar. 2020. Place: United States.
- [20] D. U. Hooper, F. S. Chapin III, J. J. Ewel, A. Hector, P. Inchausti, S. Lavorel, J. H. Lawton, D. M. Lodge, M. Loreau, S. Naeem, B. Schmid, H. Setälä, A. J. Symstad, J. Vandermeer, and D. A. Wardle, “EFFECTS OF BIODIVERSITY ON ECOSYSTEM FUNCTIONING: A CONSENSUS OF CURRENT KNOWLEDGE,” *Ecological Monographs*, vol. 75, no. 1, pp. 3–35, 2005. _eprint: <https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.1890/04-0922>.
- [21] M. Wackernagel, B. Beyers, and K. Rout, *Ecological footprint : managing our bio-capacity budget*. Gabriola Island, BC, Canada: New Society Publishers, 2019.
- [22] T. L. Ng, J. W. Eheart, X. Cai, and J. B. Braden, “An agent-based model of farmer decision-making and water quality impacts at the watershed scale under markets for carbon allowances and a second-generation biofuel crop,” *Water Resources Research*, vol. 47, no. 9, 2011. _eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2011WR010399>.

- [23] B. Noeldeke, E. Winter, and E. B. Ntawuhiganayo, “Representing human decision-making in agent-based simulation models: Agroforestry adoption in rural Rwanda,” *Ecological Economics*, vol. 200, p. 107529, 2022.
- [24] I. Bakam and R. B. Matthews, “Emission trading in agriculture: a study of design options using an agent-based approach,” *Mitigation and Adaptation Strategies for Global Change*, vol. 14, p. 755, Oct. 2009.
- [25] I. Monasterolo and M. Raberto, “The EIRIN Flow-of-funds Behavioural Model of Green Fiscal Policies and Green Sovereign Bonds,” *Ecological Economics*, vol. 144, pp. 228–243, Feb. 2018.
- [26] S. Heckbert, T. Baynes, and A. Reeson, “Agent-based modeling in ecological economics,” *Annals of the New York Academy of Sciences*, vol. 1185, no. 1, pp. 39–53, 2010. Publisher: Wiley Online Library.
- [27] L. Hardt and D. W. O’Neill, “Ecological Macroeconomic Models: Assessing Current Developments,” *Ecological Economics*, vol. 134, pp. 198–211, 2017.
- [28] J. R. Nielsen, E. Thunberg, D. S. Holland, J. O. Schmidt, E. A. Fulton, F. Bastardie, A. E. Punt, I. Allen, H. Bartelings, M. Bertignac, E. Bethke, S. Bossier, R. Buckworth, G. Carpenter, A. Christensen, V. Christensen, J. M. Da-Rocha, R. Deng, C. Dichmont, R. Doering, A. Esteban, J. A. Fernandes, H. Frost, D. Garcia, L. Gasche, D. Gascuel, S. Gourguet, R. A. Groeneveld, J. Guillén, O. Guyader, K. G. Hamon, A. Hoff, J. Horbowy, T. Hutton, S. Lehuta, L. R. Little, J. Leonart, C. Macher, S. Mackinson, S. Mahevas, P. Marchal, R. Mato-Amboage, B. Mapstone, F. Maynou, M. Merzéréaud, A. Palacz, S. Pascoe, A. Paulrud, E. Plaganyi, R. Prellezo, E. I. van Putten, M. Quaas, L. Ravn-Jonsen, S. Sanchez, S. Simons, O. Thébaud, M. T. Tomczak, C. Ulrich, D. van Dijk, Y. Vermard, R. Voss, and S. Waldo, “Integrated ecological–economic fisheries models—Evaluation, review and challenges for implementation,” *Fish and Fisheries*, vol. 19, no. 1, pp. 1–29, 2018. .eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/faf.12232>.
- [29] A. Welfle, P. Thornley, and M. Röder, “A review of the role of bioenergy modelling in renewable energy research & policy development,” *Biomass and Bioenergy*, vol. 136, p. 105542, May 2020.
- [30] P. J. Verkerk, P. Anttila, J. Eggers, M. Lindner, and A. Asikainen, “The realisable potential supply of woody biomass from forests in the European Union,” *Forest Ecology and Management*, vol. 261, pp. 2007–2015, June 2011.
- [31] A. Pyka, G. Cardellini, H. v. Meijl, and P. J. Verkerk, “Modelling the bioeconomy: Emerging approaches to address policy needs,” *Journal of Cleaner Production*, vol. 330, p. 129801, 2022.
- [32] M. D. Gerst, P. Wang, A. Roventini, G. Fagiolo, G. Dosi, R. B. Howarth, and M. E. Borsuk, “Agent-based modeling of climate policy: An introduction to the ENGAGE multi-level model framework,” *Environmental Modelling & Software*, vol. 44, pp. 62–75, 2013.
- [33] I. Monasterolo and R. Marco, “A Hybrid System Dynamics – Agent Based Model to Assess the Role of Green Fiscal and Monetary Policies,” *SSRN*, Mar. 2016.
- [34] D. Tilman, M. Clark, D. R. Williams, K. Kimmel, S. Polasky, and C. Packer, “Future threats to biodiversity and pathways to their prevention,” *Nature*, vol. 546, pp. 73–81, June 2017.

- [35] R. Pirard, “Market-based instruments for biodiversity and ecosystem services: A lexicon,” *Environmental Science & Policy*, vol. 19-20, pp. 59–68, 2012.
- [36] K. Hasselmann and D. V. Kovalevsky, “Simulating animal spirits in actor-based environmental models,” *Environmental Modelling & Software*, vol. 44, pp. 10–24, 2013.
- [37] S. Wolf, S. Fürst, A. Mandel, W. Lass, D. Lincke, F. Pablo-Martí, and C. Jaeger, “A multi-agent model of several economic regions,” *Environmental Modelling & Software*, vol. 44, pp. 25–43, 2013.
- [38] Y. Dafermos, M. Nikolaidi, and G. Galanis, “A stock-flow-fund ecological macroeconomic model,” *Ecological Economics*, vol. 131, pp. 191–207, 2017.
- [39] L. Tang, J. Wu, L. Yu, and Q. Bao, “Carbon emissions trading scheme exploration in China: A multi-agent-based model,” *Energy Policy*, vol. 81, pp. 152–169, 2015.
- [40] L. Tang, J. Wu, L. Yu, and Q. Bao, “Carbon allowance auction design of China’s emissions trading scheme: A multi-agent-based approach,” *Energy Policy*, vol. 102, pp. 30–40, 2017.
- [41] P. R. d. Andrade, A. M. V. Monteiro, and G. Camara, “From Input-Output Matrixes to Agent-Based Models: A Case Study on Carbon Credits in a Local Economy,” in *2010 Second Brazilian Workshop on Social Simulation*, pp. 58–65, 2010.
- [42] C. Deissenberg, S. Van Der Hoog, and H. Dawid, “EURACE: A massively parallel agent-based model of the European economy,” *Applied mathematics and computation*, vol. 204, no. 2, pp. 541–552, 2008. Publisher: Elsevier.
- [43] D. Masad and J. Kazil, “MESA: an agent-based modeling framework,” in *14th python in Science Conference*, vol. 2015, pp. 53–60, Citeseer, 2015.
- [44] M. Kiran, P. Richmond, M. Holcombe, L. S. Chin, D. Worth, and C. Greenough, “FLAME: Simulating Large Populations of Agents on Parallel Hardware Architectures,” in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS ’10, (Richland, SC), pp. 1633–1636, International Foundation for Autonomous Agents and Multiagent Systems, 2010. event-place: Toronto, Canada.
- [45] G. Datsaris, A. R. Vahdati, and T. C. DuBois, “Agents.jl: a performant and feature-full agent-based modeling software of minimal code complexity,” *SIMULATION*, vol. 0, p. 003754972110688, Jan. 2022. Publisher: SAGE Publications.
- [46] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, “MASON: A Multiagent Simulation Environment,” *SIMULATION*, vol. 81, no. 7, pp. 517–527, 2005. eprint: <https://doi.org/10.1177/0037549705058073>.
- [47] S. Tisue and U. Wilensky, “Netlogo: A simple environment for modeling complexity,” in *International conference on complex systems*, vol. 21, pp. 16–21, Citeseer, 2004.
- [48] N. Collier, “Repast: An extensible framework for agent simulation,” *The University of Chicago’s Social Science Research*, vol. 36, p. 2003, 2003.
- [49] D. Taghawi-Nejad, R. H. Tanin, R. M. Del Rio Chanona, A. Carro, J. D. Farmer, T. Heinrich, J. Sabuco, and M. J. Straka, “ABCE: A Python Library for Economic

- Agent-Based Modeling,” in *Social Informatics* (G. L. Ciampaglia, A. Mashhadi, and T. Yasserli, eds.), (Cham), pp. 17–30, Springer International Publishing, 2017.
- [50] A. R. Vahdati, “Agents. jl: Agent-based modeling framework in Julia,” *Journal of Open Source Software*, vol. 4, no. 42, p. 1611, 2019.
- [51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, and others, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.
- [52] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and others, “TensorFlow: a system for large-scale machine learning,” in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.
- [53] S. Agrawal, “Dataflow Agent Based Models: Using Large Scale Data Processing to Scale ABMs,” Jan. 2023.
- [54] N. T. Collier, J. Ozik, and E. R. Tatara, “Experiences in Developing a Distributed Agent-based Modeling Toolkit with Python,” in *2020 IEEE/ACM 9th Workshop on Python for High-Performance and Scientific Computing (PyHPC)*, pp. 1–12, 2020.
- [55] T. Chen, T. Moreau, Z. Jiang, L. Zheng, E. Yan, H. Shen, M. Cowan, L. Wang, Y. Hu, L. Ceze, and others, “TVM: An automated end-to-end optimizing compiler for deep learning,” in *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pp. 578–594, 2018.
- [56] M. Valente and M. Pereira, “Library for Simulation Development (LSD 8.0),” May 2022.
- [57] G. Cordasco, R. D. Chiara, A. Mancuso, D. Mazzeo, V. Scarano, and C. Spagnuolo, “Bringing together efficiency and effectiveness in distributed simulations: The experience with D-Mason,” *SIMULATION*, vol. 89, no. 10, pp. 1236–1253, 2013. [.eprint: https://doi.org/10.1177/0037549713489594](https://doi.org/10.1177/0037549713489594).
- [58] X. Rubio-Campillo, “Pandora: A Versatile Agent-Based Modelling Platform for Social Simulation,” Jan. 2014.
- [59] F. Borges, A. Gutierrez-Milla, E. Luque, and R. Suppi, “Care HPS: A high performance simulation tool for parallel and distributed agent-based modeling,” *Future Generation Computer Systems*, vol. 68, pp. 59–73, Mar. 2017.
- [60] R. Chisholm, P. Richmond, and S. Maddock, “A Standardised Benchmark for Assessing the Performance of Fixed Radius Near Neighbours,” *Euro-Par 2016 Workshops*, pp. 311–321, 2017.
- [61] S. Agrawal, “Agent Based Networks: Multi-Agent Learning for Creating Optimal Policies in Agent Based Models,” Mar. 2023.
- [62] J. Grazzini and M. Richiardi, “Estimation of ergodic agent-based models by simulated minimum distance,” *Journal of Economic Dynamics and Control*, vol. 51, pp. 148–165, 2015. Publisher: Elsevier.
- [63] J. Grazzini, M. G. Richiardi, and M. Tsionas, “Bayesian estimation of agent-based models,” *Journal of Economic Dynamics and Control*, vol. 77, pp. 26–47, 2017. Publisher: Elsevier.

- [64] W. Lu, G. Xiong, X. Liu, S. Liu, P. Ye, and L. Zhang, “Fast Calibration of Agent-Based Model using Mean-Field Approach,” in *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pp. 266–269, 2021.
- [65] A. Chopra, A. Rodríguez, J. Subramanian, B. Krishnamurthy, B. A. Prakash, and R. Raskar, “Differentiable agent-based epidemiological modeling for end-to-end learning,” in *ICML 2022 Workshop AI for Agent-Based Modelling*, 2022.
- [66] A. Quera-Bofarull, A. Chopra, A. Calinescu, M. Wooldridge, and J. Dyer, “Bayesian calibration of differentiable agent-based models,” *arXiv preprint arXiv:2305.15340*, 2023.
- [67] J. Barnard, J. Whitworth, and M. Woodward, “Communicating X-machines,” *Information and Software Technology*, vol. 38, pp. 401–407, June 1996.
- [68] S. Agrawal, S. Olowu, B. Mutua, and I. Iraoui, “Bayesian Emulation and Optimization of Agent Based Models Applied to COVID-19 Interventions,” Jan. 2023.
- [69] U. Muller, “Brainfuck: An Eight Instruction, Turing-Complete Programming Language,” 1993.
- [70] M. Garland, S. Le Grand, J. Nickolls, J. Anderson, J. Hardwick, S. Morton, E. Phillips, Y. Zhang, and V. Volkov, “Parallel Computing Experiences with CUDA,” *IEEE Micro*, vol. 28, no. 4, pp. 13–27, 2008.
- [71] Hamilton, *Graph Representation Learning*, vol. 14 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. 2020.
- [72] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer, “Decentralized control of partially observable Markov decision processes,” in *52nd IEEE Conference on Decision and Control*, pp. 2398–2405, IEEE, 2013.
- [73] T. G. Evans and D. L. Darley, “On-Line Debugging Techniques: A Survey,” in *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, pp. 37–50, New York, NY, USA: Association for Computing Machinery, 1966.
- [74] A. Sabne, “XLA : Compiling Machine Learning for Peak Performance,” 2020.
- [75] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, and others, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020. Publisher: Nature Publishing Group UK London.
- [76] D. G. Murray, F. McSherry, R. Isaacs, M. Isard, P. Barham, and M. Abadi, “Naiad: A Timely Dataflow System,” in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, SOSP ’13*, (New York, NY, USA), pp. 439–455, Association for Computing Machinery, 2013. event-place: Farmington, Pennsylvania.
- [77] D. G. Murray, M. Schwarzkopf, C. Snowton, S. Smith, A. Madhavapeddy, and S. Hand, “\${Sciel}\$: A Universal Execution Engine for Distributed \${Ddataflow}\$ Computing,” in *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*, 2011.
- [78] A. Roy, I. Mihailovic, and W. Zwaenepoel, “X-Stream: Edge-Centric Graph Processing Using Streaming Partitions,” in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, SOSP ’13*, (New York, NY, USA),

- pp. 472–488, Association for Computing Machinery, 2013. event-place: Farminton, Pennsylvania.
- [79] H. Cui, H. Zhang, G. R. Ganger, P. B. Gibbons, and E. P. Xing, “Geeps: Scalable deep learning on distributed gpus with a gpu-specialized parameter server,” in *Proceedings of the eleventh european conference on computer systems*, pp. 1–16, 2016.
- [80] A. Sergeev and M. Del Balso, “Horovod: fast and easy distributed deep learning in TensorFlow,” *arXiv preprint arXiv:1802.05799*, 2018.
- [81] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and others, “Large scale distributed deep networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [82] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018.
- [83] C. W. Reynolds, “Flocks, Herds and Schools: A Distributed Behavioral Model,” *SIGGRAPH Comput. Graph.*, vol. 21, pp. 25–34, Aug. 1987. Place: New York, NY, USA Publisher: Association for Computing Machinery.
- [84] U. Pascual, W. M. Adams, S. Díaz, S. Lele, G. M. Mace, and E. Turnhout, “Biodiversity and the challenge of pluralism,” *Nature Sustainability*, vol. 4, pp. 567–572, July 2021.
- [85] M. Cain, “Fertility as an Adjustment to Risk,” *Population and Development Review*, vol. 9, no. 4, pp. 688–702, 1983. Publisher: [Population Council, Wiley].
- [86] M. Pereira, “DSK Refresh (climate-augmented K+S) Model (version 0.4.1),” Jan. 2022.
- [87] G. Dosi, G. Fagiolo, M. Napoletano, A. Roventini, and T. Treibich, “Fiscal and monetary policies in complex evolving economies,” *Journal of Economic Dynamics and Control*, vol. 52, pp. 166–189, Mar. 2015.
- [88] G. Dosi, G. Fagiolo, M. Napoletano, and A. Roventini, “Income distribution, credit and fiscal policies in an agent-based Keynesian model,” *Rethinking Economic Policies in a Landscape of Heterogeneous Agents*, vol. 37, pp. 1598–1625, Aug. 2013.
- [89] G. Dosi, G. Fagiolo, and A. Roventini, “Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles,” *Journal of Economic Dynamics and Control*, vol. 34, no. 9, pp. 1748–1767, 2010.
- [90] G. Dosi, M. Napoletano, A. Roventini, and T. Treibich, “Micro and macro policies in the Keynes+Schumpeter evolutionary models,” *Journal of Evolutionary Economics*, vol. 27, pp. 63–90, Jan. 2017.
- [91] W. Nordhaus, “Estimates of the Social Cost of Carbon: Concepts and Results from the DICE-2013R Model and Alternative Approaches,” *Journal of the Association of Environmental and Resource Economists*, vol. 1, pp. 273–312, Mar. 2014. Publisher: The University of Chicago Press.
- [92] K. G. Abraham and J. Mallatt, “Measuring Human Capital,” *Journal of Economic Perspectives*, vol. 36, pp. 103–30, Aug. 2022.

- [93] J. M. Warren, P. J. Hanson, C. M. Iversen, J. Kumar, A. P. Walker, and S. D. Wullschlegel, “Root structural and functional dynamics in terrestrial biosphere models – evaluation and recommendations,” *New Phytologist*, vol. 205, no. 1, pp. 59–78, 2015. _eprint: <https://nph.onlinelibrary.wiley.com/doi/pdf/10.1111/nph.13034>.
- [94] J. Goudriaan and P. Ketner, “A simulation study for the global carbon cycle, including man’s impact on the biosphere,” *Climatic Change*, vol. 6, pp. 167–192, June 1984.
- [95] S. Zaehle, S. Sitch, B. Smith, and F. Hatterman, “Effects of parameter uncertainties on the modeling of terrestrial biosphere dynamics,” *Global Biogeochemical Cycles*, vol. 19, no. 3, 2005. _eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2004GB002395>.
- [96] V. Smil, *The Earth’s biosphere: Evolution, dynamics, and change*. Mit Press, 2003.
- [97] E. A. Ripley and R. E. Redmann, *Environmental effects of mining*. CRC Press, 1995.
- [98] G. Bridge, “CONTESTED TERRAIN: Mining and the Environment,” *Annual Review of Environment and Resources*, vol. 29, no. 1, pp. 205–259, 2004. _eprint: <https://doi.org/10.1146/annurev.energy.28.011503.163434>.
- [99] R. Nagendran, “Chapter 24 - Agricultural Waste and Pollution,” in *Waste* (T. M. Letcher and D. A. Vallero, eds.), pp. 341–355, Boston: Academic Press, 2011.
- [100] M. K. Hill, *Understanding Environmental Pollution: A Primer*. Cambridge University Press, 2 ed., 2004.
- [101] S. Managi and P. Kumar, *Inclusive wealth report 2018*. Taylor & Francis, 2018.
- [102] H. Oeschger, U. Siegenthaler, U. Schotterer, and A. Gugelmann, “A box diffusion model to study the carbon dioxide exchange in nature,” *Tellus*, vol. 27, no. 2, pp. 168–192, 1975. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.2153-3490.1975.tb01671.x>.
- [103] L. Weaver and N. Tao, “The Optimal Reward Baseline for Gradient-Based Reinforcement Learning,” 2013. _eprint: 1301.2315.
- [104] U. Nations, “System of Environmental-Economic Accounting— Ecosystem Accounting (SEEA EA),” *The United Nations (SEEA)*.
- [105] P. Ehrlich and J. Holdren, “Impact of population growth,” *Population, resources, and the environment*, vol. 3, pp. 365–377, 1972. Publisher: US Government Printing Office Washington, DC.
- [106] W. D. Nordhaus, “An optimal transition path for controlling greenhouse gases,” *Science*, vol. 258, no. 5086, pp. 1315–1319, 1992. Publisher: American Association for the Advancement of Science.
- [107] M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter, V. Dakos, H. Held, E. H. Van Nes, M. Rietkerk, and G. Sugihara, “Early-warning signals for critical transitions,” *Nature*, vol. 461, no. 7260, pp. 53–59, 2009. Publisher: Nature Publishing Group.
- [108] W. Steffen, K. Richardson, J. Rockström, S. E. Cornell, I. Fetzer, E. M. Bennett, R. Biggs, S. R. Carpenter, W. De Vries, C. A. De Wit, and others, “Planetary boundaries: Guiding human development on a changing planet,” *Science*, vol. 347,

- no. 6223, p. 1259855, 2015. Publisher: American Association for the Advancement of Science.
- [109] W. Steffen, J. Rockström, K. Richardson, T. M. Lenton, C. Folke, D. Liverman, C. P. Summerhayes, A. D. Barnosky, S. E. Cornell, M. Crucifix, and others, “Trajectories of the Earth System in the Anthropocene,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 33, pp. 8252–8259, 2018. Publisher: National Acad Sciences.
- [110] K. J. Arrow, P. Dasgupta, and K.-G. Mäler, “The genuine savings criterion and the value of population,” *Economic theory*, vol. 21, pp. 217–225, 2003. Publisher: Springer.
- [111] M. Hoel and T. Sterner, “Discounting and relative prices,” *Climatic Change*, vol. 84, no. 3-4, pp. 265–280, 2007. Publisher: Springer.

Appendix A

The Dystopian Schumpeter meeting Keynes (DSK) Model

A.1 Capital good firms

Capital good firms are each characterised by a set of 6 coefficients per machine vintage τ in the DSK model, $A_{i,\tau}^{\{L,EF,EE\}}$ and $B_{i,\tau}^{\{L,EF,E\}E}$. The A coefficients are characteristics of the machines that affect the output of the consumption good firms using them to create output. The B coefficients correspond to the manufacturing process of the capital good firms, and its efficiency, cost, and pollution. The indices of the coefficients correspond to labour productivity (L), energy efficiency (EE), and environmental friendliness (EF). These coefficients are set for every vintage of machine produced at each unit of time. The cost incurred at producing machinery at time t , so $\tau = t$ for firm i , given a market wage $w(t)$ and price of energy $c_{en}(t)$, would be:

$$c_i^{cap}(t) = \frac{w(t)}{B_{i,t}^L} + \frac{c_{en}(t)}{B_{i,t}^{EE}}$$

Finally, the environmental friendliness, $B_{i,t}^{EF}$ determine how much emissions firm i produces as a result of its activities.

At each timestep, the capital good firms price their machinery using a fixed markup, μ_1 . as $p_i(t) = (1 + \mu_1)c_i(t)$. Capital good firms also "market" their goods to a list of customers, who pick amongst them to get the best priced machinery according to their needs.

Now, each piece of machinery has associated with it coefficients that characterise the cost of using it, $A_{i,\tau}^{\{L,EF,EE\}}$. The cost to a consumption firm j using such machinery from capital good firm i is:

$$c_{i,j}^{cons}(t) = \frac{w(t)}{A_{i,t}^L} + \frac{c_{en}(t)}{A_{i,t}^{EE}}.$$

Therefore, each firm chooses at each t a machine to produce that minimizes the price minus a payback period multiplied by cost. They choose between the machinery available to them at the time, or alternatives that are constructed from the innovation or imitation processes:

$$\min_m p_m(t) - bc_m^{cons}(t) \quad m = \tau, in, im$$

Now each capital good firm can either innovate and improve the efficiency of their machines, or else imitate the closest firm to them in terms of technology. To do so they choose a certain fraction of their past sales to invest in R&D based on a fraction, $\nu \in [0, 1]$ of past sales $RD_i(t) = \nu S_i(t - 1)$, split between innovation and imitation based on a parameter $\xi \in [0, 1]$, $IN_i(t) = \xi RD_i(t)$ and $IM_i(t) = \xi RD_i(t)$. These investments create a probability of change on each of the 6 major coefficients characterizing the capital good firms. The investments in innovation and imitation are fed into a Bernoulli random variable with parameters $\theta_i^{IN}(t) = 1 - e^{-\xi_1 IN_i(t)}$, and $\theta_i^{IM}(t) = 1 - e^{-\xi_1 IM_i(t)}$. If innovation or imitation are successful, then improvements to the 6 coefficients are drawn from a beta distribution with support defined by static input parameters.

Finally, the profit for a capital good firm is defined as:

$$\Pi_i(t) = p_i(t)Q_i(t) - c_i^{cap}(t)Q_i(t) - RD_i(t)$$

$Q_i(t)$ are the number of orders made for capital good firm i from all of its consumption good clients.

A.2 Consumption good firms

Consumption firms use machines to produce output which is then consumed by the labour force. Consumption firms determine their desired level of produced based on a forecast of demand, $D_j^e(t) = f(D_j(t - 1), D_j(t - 2), \dots)$. The desired level of production also takes into consideration the desired inventory and the remaining stock from previous periods. In this model, the forecast demand is set to the last period's actual demand.

Their forecast level of production needed is then constructed using the desired inventory $N_j^d(t)$ and the actual inventory $N_j(t - 1)$ as:

$$Q_j^d(t) = D_j^e(t) + N_j^d(t) - N_j(t - 1)$$

The amount of output a firm can produce is limited based on their capital stock, $K_j(t)$. Capital stock is the sum of the outputs of all the machinery that the firm owns, where each machinery's contributions to capital are normalized to 1. It also computes the level of capital needed to fulfill its desired production, $K_j^d(t) (= Q_j^d(t))$, and the expected investment it needs to make in new machinery as $EL_j(t) = K_j^d(t) - K_j(t)$.

Now, machinery is replaced either when it is older than a threshold η , or when new technology becomes relatively cheap, i.e. then $p^*(t) - b(c_{j,\tau}^{cons}(t) - c^*(t)) \leq 0$. This replacement expense is added to the expected investment for new machinery, producing the overall expected investment $I_j(t)$.

The pricing of consumption food firms is determined based on competitive dynamics, where firms controlling more market share use their power to price goods higher, but the market is cleared using the lowest price goods first, acting as a natural inhibitor to monopolies developing.

The competitiveness of a firm is $E_j(t) = -\omega_1 p_j(t) - \omega_2 l_j(t)$, where $l_j(t)$ is the amount of demand unfilled in the last period. $\omega_{1,2}$ are simulation constants. The average competitiveness at the industry level is $\bar{E}(t)$, and is the competitiveness of each agent weighted by its market share. The current market share then evolves according to:

$$f_j(t) = f_j(t-1) \left(1 + \xi \frac{E_j(t) - \hat{E}(t)}{\hat{E}(t)} \right)$$

Now that the market share has been established, we can calculate the markup consumption firms charge on their goods. Like capital good firms, consumption good firms apply a markup on their cost of production, but unlike capital good firms the markup is variable: $p_j(t) = (1 + \mu_j(t))c_j(t)$, where the markup is:

$$\mu_j(t) = \mu_j(t-1) \left(1 + \nu \frac{f_j(t-1) - f_j(t-2)}{f_j(t-2)} \right)$$

Now consumption good firms have the ability to access capital markets. They each retain a metric of their net wealth, $NW_j(t)$, and finance all investments using a combination of their wealth and debt. Up to $c \in (0, 1)$ is financed from net wealth, and the remainder from debt, up to a maximum debt to sales ratio.

$$Deb_j(t) = Deb_j(t-1) + (1-c)I_j(t)$$

Finally, the profits that each consumption good firm earns is:

$$\Pi_j(t) = p_j(t)Q_j(t) - c_j(t)Q_j(t) - rDeb_j(t)$$

The net wealth is similarly updated as:

$$NW_j(t) = NW_j(t-1) + \Pi_j(t) - rDeb_j(t) - cI_j(t)$$

If the net wealth of the firm reaches 0, they are considered bankrupt and are replaced by a new firm with initial characteristics coming from probability distributions.

When the net wealth of a consumption good firm reaches 0, it is treated as bust and is instantly replaced by another firm with an initial net wealth drawn from a distribution.

A.3 Labour

The labour market is modelling as one environmental agent that contains the state of the entire labour force. The wage level is static across all works and is set using the following formula:

$$w(t) = w(t-1) \left[1 + \psi_1 \frac{\Delta \hat{A}B(t)}{\hat{A}B(t-1)} + \psi_2 \frac{\Delta cpi(t)}{cpi(t-1)} + \psi_3 \frac{\Delta U(t)}{U(t-1)} \right]$$

All wages (and governmental subsidies if any) are consumed through purchasing goods from the consumption firms. The workforce consists of a supply of labourers, L^S . Similarly, a demand for labourers, L^D is created by the consumption and capital good industries. This labour force in the DSK model changes with a constant positive growth rate.

The labour productivity is the total output of the economy, which is the sum of production across capital and consumption good firms, divided by the labour demand. The labour demand is calculated by dividing the total output of all firms employing workers by their labour productivity coefficients.

A.4 The banking sector

Like the labour market, the banking sector is a single environmental agent who's sole responsibility is to determine the level of interest rate in the economy that the firms pay on their debt. The rate is determined by the Taylor rule based on a target level of interest rate r_T , target inflation rate cpi_T , and target unemployment U_T :

$$r_t = r_T + \gamma_{cpi}(cpi(t) - cpi_T) + \gamma_U(U(t) - U_T)$$

There is a further constraint that the banking sector puts on loans. In addition to the maximum leverage ratio at a firm level, the total amount of loans given also cannot exceed the net wealth of all (consumption good) firms in the simulation, $MTC(t) = \sum_j NW_j(t)$.

A.5 Energy industry

The energy industry again consists of a single environmental agent controlling a number of power plants. The power plants supply electricity to the capital and consumption good industries. Power plants can be either dirty, or green.

Here we can start by stating the profit of the energy industry and then drilling into its components:

$$\Pi_e(t) = p_e(t)D_e(t) - c_e(t)D_e(t) - I_e(t) - RD_e(t)$$

Here $p_e(t)$ and $c_e(t)$ are the price charged for energy, and the cost of generating it. $D_e(t)$ is the demand, and $I_e(t)$ and $RD_e(t)$ are the expansion/replacement costs and the R&D expense costs respectively.

The DSK model assumes that green energy is produced at a 0 running cost, whereas dirty power plants generate energy at a cost of $c_{de}(t) = \frac{p^*(t)}{A_{de}^\tau}$, where $p^*(t)$ is the cost of the commodity input and A_{de}^τ is the thermal efficiency of the power plant. Dirty power plants also emit em_{de}^τ , whereas green power plants have no emissions.

The energy sector first satisfies any energy demand using green power plants, and then it switches on dirty power plants. In the DSK model, the energy demands are normalized to one, and so the capacity is simply the number of existing power plants. Once again, if the amount of energy demanded is less than available capacity, $K_e(t)$, then this triggers a net investment $EI_e(t) = D_e(t) - K_e(t)$.

Now, notably the dirty power plants are assumed to have a 0 cost relative to green plants, which have a cost of $IC_{ge}(t)$. The energy sector chooses to build green power plants to cover this excess capacity whenever the (current) cost of the green plant is less than the (lowest) discounted running cost of the dirty plants, $IC_{ge}(t) \leq b_e \hat{c}_{de}(t)$. This is largely dependent on commodity prices.

Finally, the energy sector also invests in R&D. Similar to the capital good sector, they put a portion of their sales into R&D: $RD_e(t) = \nu_e S_e(t-1)$. A certain fraction of that investment gets put into green innovation $IN_{ge}(t) = \xi_e RD_e(t)$, and the rest into innovation for dirty plants, $IN_{de}(t) = (1 - \xi_e) RD_e(t)$. The probability of successful innovation is determined identically to the capital goods industry.

Successful innovation in the green sector decreases the cost of green power plants, $IC_{ge}(t) = IC_{ge}(t-1)(1 - x_{ge})$, where x_{ge} is sampled from a beta distribution. Successful innovation in the dirty energy sector increases the thermal efficient, $A_{de}^\tau = A_{de}^{\tau-1}(1 + x_{de}^A)$, and decreases its emissions $em_{de}^\tau = em_{de}^{\tau-1}(1 - x_{de}^{em})$.

A.6 Climate box

Ultimately the goal of the climate box is to compute the ocean temperature, T_m , so that climate shocks can be computed as a function of T_m . Temperature is updated as a function of 2 key variables, $F_{CO_2}(t)$, the radiative forcing effect, and $T_d(t-1)$, the temperature of the deep layers of the ocean. $T_d(t)$ is computed as $T_d(t) = T_d(t-1) + c_4(\sigma_{md}[T_m(t-1) - T_d(t-1)])$. Here, σ_{md}, c_4 are constants. Finally, the temperature of the mixed layer $T_m(t)$ is determined as:

$$T_m(t) = T_m(t-1) + c1\{F_{CO_2} - \lambda T_m(t-1) - c_3[T_m(t-1) - T_d(t-1)]\} \quad (\text{A.1})$$

This temperature is then used as a parameter in the distribution generating climate shocks. The radiative forcing effect is calculated as a function of the stock of carbon in the atmosphere, $C_a(t)$:

$$F_{CO_2}(t) = \gamma \log\left[\frac{C_a(t)}{C_a(0)}\right]$$

The procedure modelling anthropogenic climate change is as follows:

1. The energy, capital goods and consumption goods sectors produce carbon emissions in t :

$$em(t) = \sum_\tau \sum_t (em_e(t, \tau) + em_{cons}(t, \tau) + em_{cap}(t, \tau))$$

2. These emissions affect the atmospheric carbon stock to a disequilibrium level:

$$\hat{C}_a(t) = C_a(t-1) + em(t)$$

3. The biosphere and deep ocean layers absorb some of this excess carbon:

- (a) Biosphere:

$$NPP(t) = NPP(0)(1 + \beta_C \log\left[\frac{\hat{C}_a(t)}{C_a(0)}\right])(1 + \beta_{T_1} T_m(t-1))$$

(b) Ocean mixed layer (disequilibrium):

$$\hat{C}_m(t) = C_m(0)[1 - \beta_{T_2} T_b(t-1)] \left(\frac{\hat{C}_a(t)}{C_a(0)} \right)^{\frac{1}{\xi_0 + \delta \log\left[\frac{C_a(t-1)}{C_a(0)}\right]}}$$

(c) Ocean deep layer:

$$C_d(t) = C_d(t-1) + k_{eddy} \frac{\left[\frac{C_m(t-1)}{d_m} - \frac{C_d(t-1)}{d_d} \right]}{d_{md}}$$

(d) Ocean mixed layer (equilibrium):

$$C_m(t) = \hat{C}_m(t) - \Delta C_{md}(t)$$

4. Finally we compute the equilibrium atmospheric carbon content as:

$$C_a(t) = \hat{C}_a(t) - \Delta NPP(t) - \Delta C_m(t) - \Delta C_{md}(t)$$

5. From this we compute the radiative forcing, $F_{CO_2}(t)$, and finally $T_m(t)$ using equation A.1.

Once we have the mixed layer temperature, we can then draw climate shocks for each of the 6 machine parameters for capital good firms, as well as the thermal efficiency and emissions of dirty power plants, and can destroy inventories of consumption good firms (e.g. through destroying crops).

The shocks are drawn from a beta distribution over $(1, 1)$ with parameters $a(t) = a_0[1 + \log T_m(t)]$, and $b(t) = b_0 \frac{\sigma_{10y}(0)}{\sigma_{10y}(t)}$.

Appendix B

Criticisms of and issues with the DSK model

Presented here are some of the key criticisms and shortcomings of how the model is presented in the paper, along with how it is resolved in the VDSK approach developed in chapter 5. Where the resolution is taken from the open source implementation of the DSK refresh model, it is cited.

Section	Problem	Resolution in VDSK
General 1	The parameter values given in the appendices of all reviewed papers is incomplete and omits critical information.	Take initial values from the open source implementation where possible [86].
General 2	The authors give values for the initial number of capital good firms (50), consumption good firms (200) and workforce (3000) but no detail on the initial conditions of the agents in terms of their initial inventory, capital, etc. How these initial conditions are set have a big impact on the simulation dynamics.	The initial distribution of capital and wealth is computed to clear the market (see chapter 5 for details).
General 3	Non-zero tax rates are mentioned across the papers and its also mentioned that this is used to pay unemployment benefits, but then the level of the unemployment benefit is treated as a fixed external parameter.	Pay unemployed workers up to a minimum of the unemployment wage and taxes collected, $\hat{w}_U = \min(Taxes, w_U L^S) / L^S$
General 4	Initial prices and commodity prices are not specified.	

General 5	<p>No indication is given of the machine parameters, $A B_{L EE EF}$. This is troublesome for multiple reasons:</p> <ol style="list-style-type: none"> 1. Since prices are a linear combination of the cost of energy and the prevailing wage, $c_{en}, w(t)$, setting $A B_{L EE}$ to 1 sets initial prices to 2, making it hard to clear the market. 2. Atmospheric carbon is measured in gigatons of carbon, but $A B_{EF}$ is just a dimensionless parameter. It is impossible to know how to calibrate it such that emissions increase in a realistic way without a calibrated value available. 	<p>$A B_{EF}(0)$ is set to 40, $A B_{EE}(0)$ is set to 10, and $A B_L(0)$ to 1 as per Pereira [86].</p>
General 6	<p>The paper talks about parameters that are not even specified in any equation or supporting paper, such as β_{TM}.</p>	<p>Make best guesses as to what parameters the authors were actually referring to, such as β_T.</p>
Cons/Cap Good Firms 1	<p>There is no indication of how labour is split or demanded across firms in the economy, or how much labour is needed per unit of output. The open source model allocates workers to each firm using a complex set of rules, including having specific workers for R&D, which are mechanisms not mentioned in the papers.</p>	<p>We prorate labour supply across firms based on the previous step's demand for goods from the firm.</p>
Cons/Cap Good Firms 2	<p>Capital good firms have no net wealth or costs that could exceed the profit they make in any period since R&D costs are a fraction of sales and adopting new innovations have no cost. However, the paper mentions that firms with 0 market share/net wealth across either industry go bust and are replaced, which is not possible for capital good firms as specified in the paper.</p>	<p>Implement agents as specified in the paper.</p>
Cons/Cap Good Firms 3	<p>No mechanism is explicitly given for how consumers choose consumption firms, and how consumption good firms choose among their suppliers. There is an implicit suggestion that consumers prioritize firms by price until inventories are cleared, but this is not anywhere stated.</p>	<p>Allocate consumer demands in proportion to $1/\sqrt{price}$.</p>

Labour 1	There is no indication of how labour is split or demanded across firms in the economy, or how much labour is needed per unit of output. The open source model allocates workers to each firm using a complex set of rules, including having specific workers for R&D, which are mechanisms not mentioned in the papers.	We prorate labour supply across firms based on the previous step's demand for goods from the firm.
Labour 2	In wage updates, there is no indication of how to calculate the productivity of the economy, or the CPI, needed for wage updates.	Using a similar approach to Pereira [86] we calculate productivity as the sum of produced goods across firms, and CPI as the average capital good, consumption good, and energy price, detailed in chapter 5.
Energy Sector 1	There is no mention of net wealth for energy firms, but they still make investments in green energy. These can't be unlimited and only a function of dirty machinery cost, as the model currently implies.	Keep track of net wealth and use it for green energy investments. When net wealth runs out build dirty power plants instead.
Energy Sector 2	There is no limit to how much energy capacity can be added in a step, which is unrealistic.	Limit capacity addition to 10% per quarter [from [86]].
Energy Sector 3	Lamperti et al do not detail how the lack of available energy capacity limits the output of consumption good or capital good agents in any way. In the presented model, lack of energy capacity does not constrain the output of consumption/capital good firms.	Limit output of consumption/capital good firms based on energy capital allocation computed using their previous step's energy demand: $K_{actual} = \min(K_{installed}, E_{allocated} \times C_E)$ where $E_{allocated} = K_{actual}^{energy} \times Q/Q_{total}$.
Energy Sector 4	There is no mention of the parameters of the distribution from which energy innovations are drawn in any published papers.	Use the same (a, b) parameters as consumption/capital good firms but with supports rescaled to the domains mentioned in the paper.
Energy Sector 5	The initial commodity price and how it changes is not specified.	As per Pereira [86] the commodity price is set to a constant of 1. However, this misses very important dynamics that can result from how commodity prices affect investment in green technology.
Climate Box 1	Initial carbon in the mixed ocean layer is not specified.	We compute it in accordance to Pereira [86].

Climate Box 2	The explicit procedure to actually calculate temperature updates is actually not specified at all other than very high level descriptions not sufficiently detailed to actually carry out calculation. Furthermore, the open source implementation by Pereira [86] uses an entirely different climate model to DICE-2013R [91].	Through trial and error arrived at the procedure described in chapter A.
Climate Box 3	The paper mentions that shocks and agents are randomly matched, but there is no mechanism to do the matching, e.g. through a Bernoulli random variable.	Apply individually sampled random shocks to each agent.
Climate Box 4	How shocks are actually applied to the agent parameters is not specified anywhere in the papers. It mentions that shocks are applied to infrastructure parameters and inventories but isn't explicitly clear about how or where it is applied, or even how it is sampled beyond from a heavy tailed beta distribution.	
Climate Box 5	The shocks are drawn from a beta distribution using the 10 year standard deviation in T_m , which isn't determined until 10 years into the simulation. Pereira simply doesn't calculate the climate box for 10 years to deal with this.	Set an initial value and let it update as new temperature readings come in.
Climate Box 6	The update equation for T_m is wrong. The efficiency coefficient λ that calculates the feedback from the current temperature is taken by Lamperti et al to be 2.9. However, this is incorrect; in DICE-2013R 2.9 is the temperature increase from a doubling of atmospheric carbon, and λ is actually $1/2.9$.	We correct it to $1/2.9$.
Climate Box 7	The application of the DICE-2013R model seems flawed. Looking at equation A.1 for T_m , the temperate increases if: $F_{CO_2} > \lambda T_m(t-1) - c_3[T_m(t-1) - T_d(t-1)]$. However, F_{CO_2} is in a log scale, and the starting value based on current atmospheric carbon as 830 and initial atmospheric carbon of 590 is 0.8. As seen in figure 5.1 this causes temperatures to plummet in the model at the start of the period. Even correcting for λ , this initial plummet exists.	Pereira corrected this phenomenon by not allowing the climate box to run for the first 10 years. We allow it to run but recognize this as a flaw in the model.
Banking Sector 1	The interest rate is said to update using the Taylor rule, but the coefficients for the Taylor rule equation are never given.	Use commonly accepted values in literature [12].

Appendix C

Dasgupta on the economics of biodiversity

C.0.1 Introduction

Dasgupta's model is built on a so-called "bounded economy" model; it is built to operate in the regime where the biosphere isn't tipped past an unsustainable point, thereby destroying life as we know it. This creates bounds on what the economy can do as a function of the interaction with the biosphere, even when including technological change (which is novel, since technological innovation made economies able to be unbounded in size in traditional economics).

Dasgupta's report studies the structure of the economy at an abstract level, and attempts to draw together the camps of economic exploitation and conservation through economic arguments about the value of biodiversity and a strong biosphere. It argues that we fundamentally mis-value the biosphere due to our economic short-termism, our reliance on technological innovation and our assumption that it will fix all future problems, and misspecification in our national accounting identities. In particular, he argues that nature should be treated as a form of capital, akin to human or produced capital.

Dasgupta argues that not only is growth bounded, but so is the total size, using planetary resources only. His argument is that due to the laws of entropy, it isn't possible to convert all our waste back into products that make no further demands on the biosphere, so at some point we will reach a peak capacity where we are using all available services and will be unable to grow. Dasgupta further argues that inclusive wealth, not GDP, should be used as a measure of economic progress, and this wealth should include natural capital as well. It is possible to generate more GDP in the short run by unsustainably using your assets, decreasing future wealth but boosting current GDP. This is because GDP is a flow - it is the produced output of an economy, not its stock (i.e. assets).

An illustration of changes in wealth from Managi and Kumar [101] is given below in figure C.1.¹ Here are the definitions of the various measures:

1. Produced Capital: this is the value of all physical, non-human capital that we rely on, such as land, machinery, physical infrastructure, housing, etc.

¹Natural capital is notoriously difficult to measure. The field of ecosystem accounting tries to address measurement challenges using proxy measures; a UN report covers the challenges and approaches in detail here: [104]

Figure 4.8 Global Wealth Per Capita, 1992 to 2014

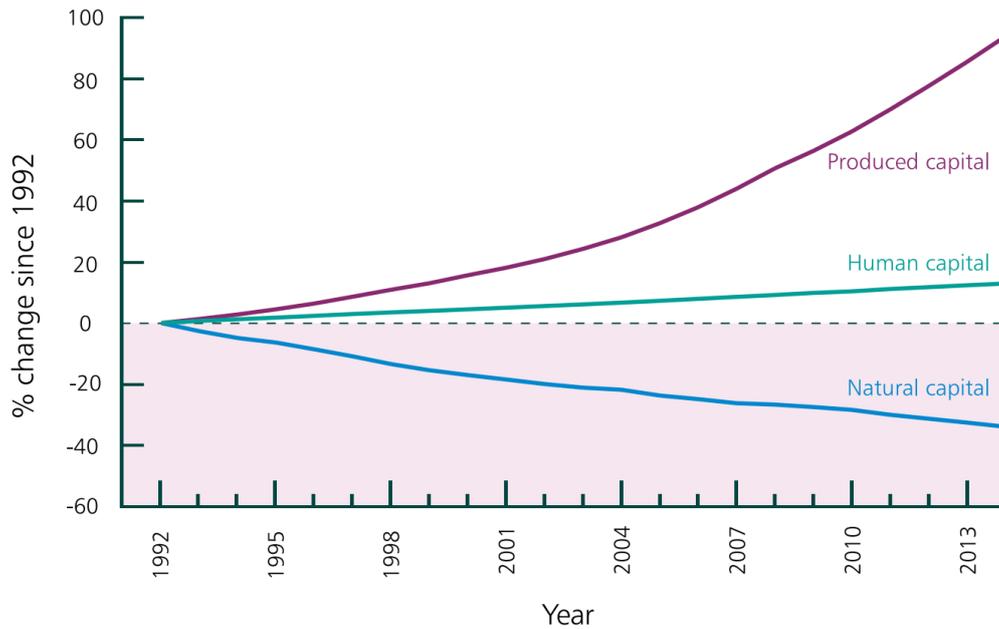


Figure C.1: Changes in global wealth per capita [101]

2. Human Capital: this includes items such as health (i.e. the marginal ability to produce more during their lifetimes), income, etc.
3. Natural Capital: stock in fossil fuels, metals and minerals, fertile land, forests, etc.

C.0.2 The Impact Inequality

A lot of Dasgupta's economic arguments and rationale for the model stem from the Impact Inequality [1]. This inequality models the difference between economy's the demand for services from the biosphere, and the biosphere's ability to supply them. One key aspect of this is that the demand can be decomposed into 2 major components: harvesting nature's goods for consumption/production, such as fish, etc., and using nature as a sink for our waste. Recognizing these two factors is important in allowing us to see what kind of policy levers there are to pull to steer the economy to a better balance.

Say that we can look at the global economy as the number of people N multiplied by the GDP per capita, y , so $Y = Ny$. Let's define two additional factors corresponding to our 2 uses of nature:

1. α_x representing the efficiency with which we convert biosphere goods into GDP
2. α_z , which is the extent to which the biosphere is affected by our waste

We can define $X = NY/\alpha_x$ as the value of what we extract from the atmosphere, and $Z = Ny/\alpha_z$ the demand we make on the biosphere as a sink for our pollution. Then, a proxy measure of our global ecological *impact* from our aggregated economic activity is:

$$Ny/\alpha = X + Z = Ny/\alpha_x + Ny/\alpha_z \quad (\text{C.1})$$

Dasgupta "heroically" assumes we can measure the entire accounting value of the biosphere into one scalar S , and we can represent the regeneration of the biosphere as a

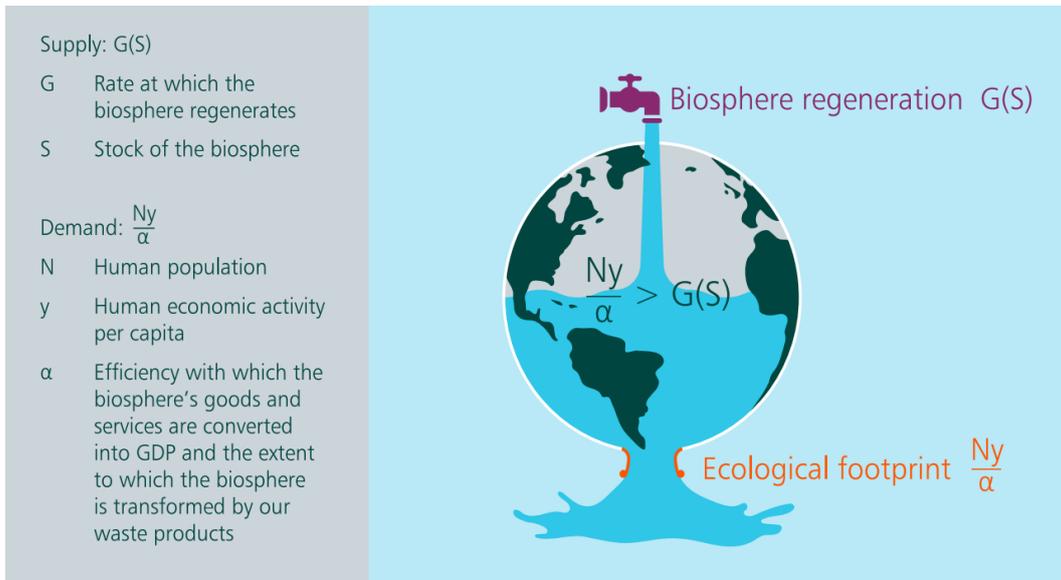


Figure C.2: The Impact Inequality: natural biosphere regeneration is offset by humanity's ecological footprint (a function of human population, economic activity, and our technology). Note the feedback loop between $G(S)$ and our ecological footprint - $G(S)$ is not a constant offset but degrades as we deplete S , our biosphere. Figure from [1].

function $G(S)$. The balance between the biosphere regeneration rate, $G(S)$, and the ecological footprint of our use of the biosphere, Ny/α , is the so-called Impact Inequality. Note that this assumes that to pollute has the same impact as conservation. A concise illustration of the impact inequality is given in C.2. The Impact Inequality is:

$$\frac{Ny}{\alpha} - G(S) = \frac{Ny}{\alpha_x} + \frac{Ny}{\alpha_z} - G(S) > 0 \quad (\text{C.2})$$

This concept of ecological "impact" comes from classic ecological economics literature, where Ehrlich and Holdren [105] proxied the ecological impact of humanity in terms of a set of factors such as population, affluence and technology. The direction of the inequality is currently that human ecological impact exceeds the biosphere's ability to regenerate. Wackernagel and Beyers [21] report that the ratio of Ny/α related to $G(S)$ increased from 1 in 1970 to 1.7 in 2019. The rate at which α needs to grow to offset our impact on the biosphere is 10%, vs. a historical average of 3.5% as reported by Managi and Kuamr [101]. I.e. we need a whole lot more efficiency in how we treat waste and in our use of natural products to reach a balance.

Note that there is also empirical evidence that ecological footprint is an increasing function of income, i.e. $y_i/\alpha_i < y_{i+1}/\alpha_{i+1}$ with i representing an ordering over income. This is ominous, since it suggests that egalitarian income leads to larger ecological footprints, which has terrifying moral implications.

C.0.3 The model

This model starts by looking at aggregate demand of the economy, whose output, Y (a scalar), is either consumed C or invested I . This model can be applied at any level, for instance Y could represent the economic value of just a single good. Four categories of capital are considered: K , produced capital (e.g. infrastructure, goods, etc.), H , human capital (labour, health, education), A , available knowledge (productivity and innovation),

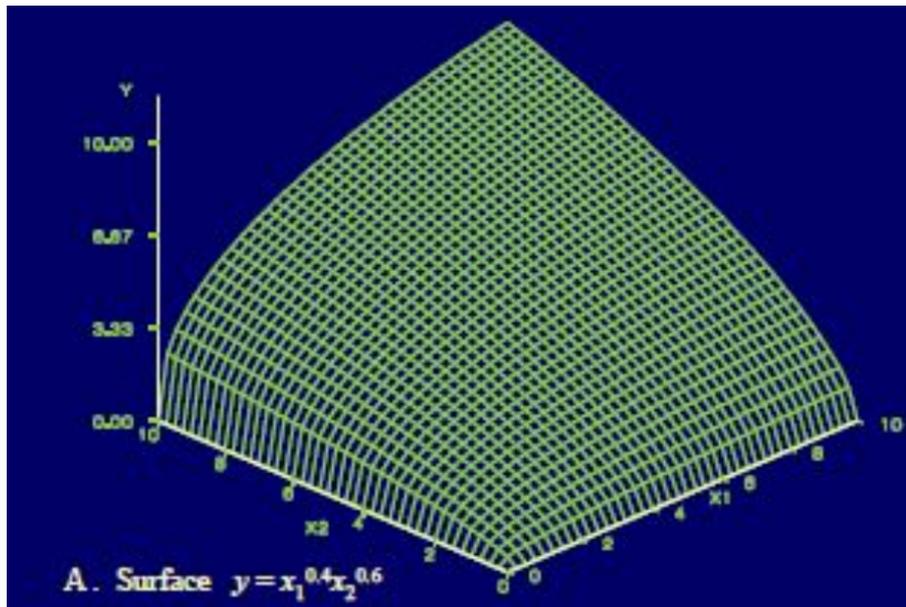


Figure C.3: Example of how the Cobb-Douglas form models economic output as a combination of 2 sources of capital. Note the constant returns to scale as well as marginally decreasing output. Figure from [1].

and natural capital. Natural capital is broken out into 2 components: R , provisioning services provided by the biosphere, such as wood, metals, crops, fish, etc., and S , which are regulating services, like the nitrogen cycle, climate regulation, soil regeneration, etc. In continuous time, the Dasgupta model presents the output, in a Cobb-Douglas-like form, as:

$$Y(t) = A(t)S(t)^\beta K(t)^a H(t)^b R(t)^{1-a-b} \quad (\text{C.3})$$

$$a > 0, b > 0, (1 - a - b) > 0, \beta > 0$$

The condition of the exponents summing to 1 except for β comes from production being subject to constant returns of scale, i.e. doubling the amount of K, H and R leads to a doubling of Y all else equal. This is a standard assumption in economics and one that Dasgupta uses. This assumption can be challenged on obvious grounds: its not clear that producing twice the amount of timber would automatically produce twice the output of the furniture industry, even with twice the machinery and human capital, given increased administrative costs, etc. but can be taken as stated for now. The exponents also act as a measure of elasticity - i.e. if you were to increase R by 1%, the output Y would increase by $(1 - a - b)\%$.

Note that this kind of model is called the Cobb-Douglas form after economists who developed and tested it against empirical evidence of how 2 or more capital inputs produces economic output. The relationship between the 2 classical capital inputs with typical exponent values is shown in figure C.3.

C.0.4 Total Factor Productivity (TFP)

TFP is the ratio between the total production of an economy, and the estimated "capital inputs" - i.e. labour, natural resources, etc. This measures how much of growth is explained by innovation and our ability to produce more from existing capital, rather

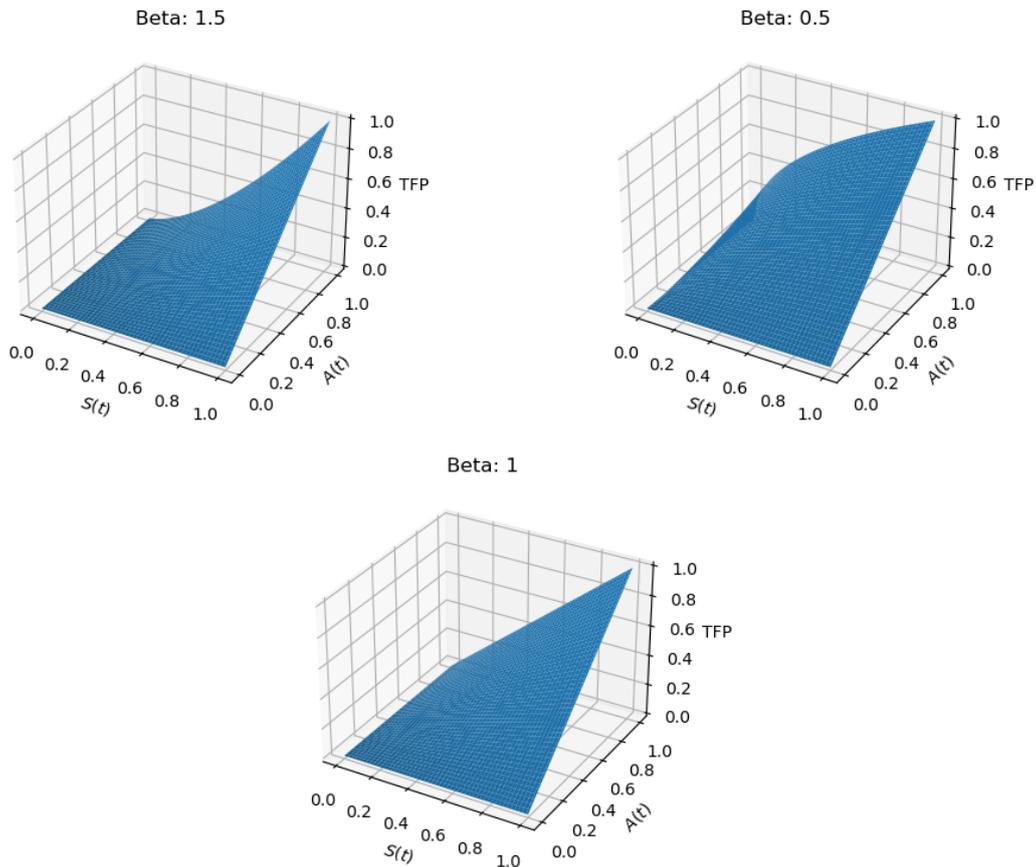


Figure C.4: How different estimates of β affect the TFP estimate

than from just growth in capital. In this model, the Total Factor Productivity (TFP) is given as:

$$TFP = A(t)S^\beta(t) \quad (C.4)$$

This is also a key innovation in Dasgupta's report: the TFP is not just from human knowledge and ability ($A(t)$, e.g. from using computers, or machinery, or the assembly line) but also from the biosphere's regulating services S , when $\beta > 0$. Traditionally biosphere services aren't included in productivity measures. This is already a critical feedback loop, since as S lessens, "productivity" declines even potentially with increasing A and increasing capital.

We can see the effect of $S(t)$ and $A(t)$ on TFP for different values of β in figure C.4. When $\beta > 1$, TFP increases superlinearly with $A(t)$, and for a fixed amount of services provided by the biosphere, increased innovation produces greater-than-linear increases in economic output. The converse is true if $\beta < 1$. In current estimates of TFP, β is assumed to be 0 since services rendered by the biosphere are ignored. This is a key accounting flaw flagged by ecological economists [106] [12] [18] in how traditional accounting measures like GDP completely ignore our climate and biosphere, despite them being critical to the success of the species. By excluding the degradation of natural capital from our wealth and GDP calculations, we overestimate our actual wealth, and so incentive economic agents to degrade the biosphere further and faster to maximize GDP.

C.0.5 Human impact on the biosphere

Human impact on the biosphere is the sum of the resources we take from it, and the waste we send back to it. We saw in equation C.3 that the capital from resources extracted from the biosphere is $R(t)$. So, in terms of equation C.2, human impact is $R(t) + Y(t)/\alpha_z$, and so Dasgupta takes $R(t)$ to be $Y(t)/\alpha_x$.

Here Dasgupta makes a set of very significant assumptions about the model that merit some discussion. Firstly Dasgupta assumes that human knowledge, A , contributes both to Y and α_z , i.e. $\alpha_z = \alpha_z(A)$. Dasgupta argues that α_z must be bounded above to some value α_z^* . Dasgupta argues that if α_z can go to infinity, then Y/α_z goes to 0 for finite Y , which says that with enough technological progress, we can break free of the biosphere entirely (whilst still remaining terrestrial). If α_z and Y go to infinity at the same rate, then that's equivalent to us saying we can make any demands on our biosphere without experiencing a breakdown. If α_z increases more slowly than Y , then Y/α_z is unbounded, so Dasgupta argues "we may as well assume α_z is bounded".

From this, Dasgupta frames an equation for the dynamics of the biosphere using the impact inequality C.2:

$$\frac{dS(t)}{dt} = G(S(t)) - \frac{Y(t)}{\alpha_x} - \frac{Y(t)}{\alpha_z}, \quad \alpha_x > 0, \alpha_z^* \geq \alpha_z > 0 \quad (\text{C.5})$$

Now Dasgupta assumes a functional form for $G(S)$ taken from Scheffer's formulation [107] of the dynamics of fisheries:

$$G(S(t)) = rS(t)\left(1 - \frac{S(t)}{\underline{S}}\right)\left(\frac{S(t) - L}{\underline{S}}\right) \quad (\text{C.6})$$

This functional form says if S were to fall below its safety zone L , the biosphere would collapse. L is given motivation through the idea of "planetary boundaries" discussed in literature such as [5] [108] [109]. In the absence of consumption of natural resources, the biosphere would settle to some carrying capacity, \underline{S} . The graph of this growth rate is given in figure C.5.

C.0.6 The human population

Denote the human population as $N(t)$. Each term in Dasgupta model can then be written as a per person value multiplied by the human population, $H(t) = h(t)N(t)$. Dasgupta stipulates population cannot be controlled directly, but that it can be influenced by investment into human capital. That is making a very strong statement composing of two items:

1. Humans on average have a "target" number of children desired by a household - this is backed up by many empirical studies such as Bongaarts and Cain [85] in rural Bangladesh during times of famine and just after.
2. This target is a function of the investment in human capital we as a society make: better standards of living, more expensive education and health, access to family planning services, lack of need for familial manual labour, all reduce the desired family size.

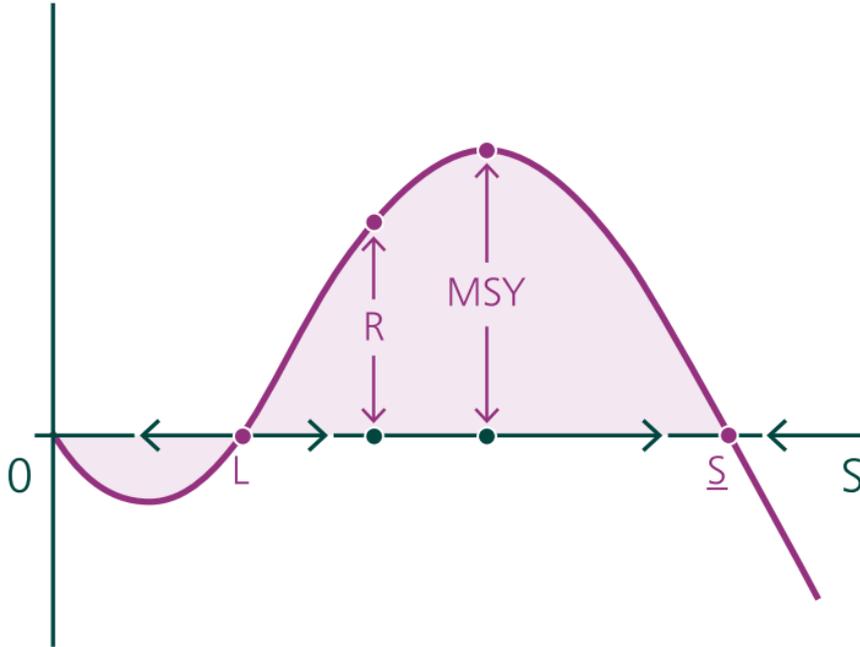


Figure C.5: An illustration of equation C.6 from [1]. Growth rates when stock is below the safety zone is negative - the biosphere is no longer self-sustaining. Growth rates above the carrying capacity would also see negative growth from overcrowding.

To quote Dasgupta: “women’s education and knowledge of, and access to, modern family planning services reduce desired family size“ [1]. Dasgupta notes that more nuance can be applied by making J a function of K as well, with smaller positive K resulting in better diet, hygiene, etc. resulting in a baby boom, and large positive K reducing birth rate, but omits this from his logistic specification of population size, taken from Arrow, Dasgupta, and Mäler [110], in equation C.7.

$$\frac{dN(t)}{dt} = N(t)(J(h) - N(t)), \quad J > 0, \quad dJ(h)/dh < 0 \quad (\text{C.7})$$

C.0.7 Human capital, produced capital, innovation and investment

A canonical decomposition of GDP by national accountants is that it is a sum of consumption and investment. This measure of aggregate demand is the familiar $Y = C + I + G + (X - M)$, where G is governmental expenditure and X, M are exports and imports respectively. Dasgupta states that investment can be broken out into: accumulating capital from produced (I_K) and human (I_H) capital and expenditure on research and development which increases A , (I_A). Aggregate demand can then be expressed as per equation C.8.

$$Y(t) = C(t) + I(t) = C(t) + I_H(t) + I_K(t) + I_A(t) \quad (\text{C.8})$$

Produced capital always depreciates (machinery rusts, buildings wear, entropy rises). Dasgupta assuming it to depreciate with a constant λ and writes the change in produced capital in equation C.9.

$$\frac{dK(t)}{dt} = I_K(t) - \lambda K(t) = Y(t) - C(t) - I_H(t) = I_A(t) - \lambda K(t) \quad (\text{C.9})$$

Innovation is expressed more simply in equation C.10.

$$\frac{dA(t)}{dt} = I_A(t) \quad (\text{C.10})$$

Finally, Dasgupta gives human capital, using the chain rule on $H(t) = h(t)N(t)$, in equation C.11.

$$\frac{dH(t)}{dt} = N(t)\frac{dh(t)}{dt} + h(t)\frac{dN(t)}{dt} = I_H(t) \quad (\text{C.11})$$

C.0.8 Summarizing assumptions

Now that the dynamics of each of the state variables have been defined, all the copious assumptions made in the process of derivation can be summarized:

About the biosphere

1. The biosphere has a natural regeneration rate that can be expressed as a function of its current state: $G(S)$, which can be expressed with a simple closed form function
2. The biosphere's value can be expressed as a single scalar accounting value, S
3. There exist planetary boundaries, or a natural tipping point, where if we push past it, we kill the biosphere
4. There is a carrying capacity at which the biosphere would operate at if we didn't consume any of its resources

About human interactions with the biosphere

1. Humans interact with the biosphere in 2 key ways:
 - (a) to get natural resources such as fish, timber, fossil fuels and metals, and
 - (b) as a store for our waste.
2. Humans cannot completely free themselves of the biosphere's services, there must always be a reliance even with technological advance, as long as we're on this planet. This causes 3 colloraries:
 - (a) Human innovation cannot create a world where we produce waste that doesn't disrupt the biosphere at all
 - (b) Human innovation cannot free us from biosphere services entirely
 - (c) Human innovation cannot make it such that we can do whatever we want without it pushing past the biosphere tipping point
3. Human innovations can affect how efficiently we extract services from the biosphere in terms of maximizing our GDP per unit of resource, and how we efficiently feed our waste back to the environment, in terms of the impact on the biosphere

About the economy

1. We can place a price on all capital, including ineffable things like innovation
2. We have a good understanding of investments in capital
3. Humanity on average has a target number of children which varies as a function of human capital only
4. Several functional forms are assumed, e.g. a logistic form for human population dynamics
5. Produced capital depreciates linearly and at a constant rate

C.0.9 The Utility Maximization Problem

Let's say that we have a consumption per capital as $c(t)$. We want to maximize some utility of the population. Dasgupta defines the a "well-being" utility as a function $u(c(t), S(t))$. The "well-being" element comes from Dasgupta's arguments that the "non-use value" of nature is more important than it's "use value". Non-use value means the value coming from regulating services as well as happiness derived from natural beauty, etc. whereas use value is the value of the resources we get from nature. Using such a utility states that we as humans value the parts of nature that we don't directly consume more than the parts of nature that we do.

Dasgupta tries to use this framework to look at the case when people care more about the quality of the environment as their standard of living rises. This is a well documented empirical observation, as per Hoel and Stermer [111]. Mathematically, Dasgupta notes that that means: $\partial(u_s/u_c)/\partial c > 0$. The total social well-being is $N(t)u(c(t), S(t))$.

Given the economic framework we've described so far, Dasgupta now puts on the hat of a "social evaluator" who is looking at all possible futures and trying to maximise their expected utility, in equation C.12. Dasgupta's social evaluator only cares about the futures where the biosphere is stable and hasn't crossed a the threshold. This evaluator has to pick the best possible future given the initial conditions in its 5 key state variables as $A(0), S(0), K(0), N(0), h(0)$, given that each of the 5 key state variables have to follow the dynamical equations specified above. The social evaluator has the ability to vary 4 key control variables, $C(t), I_H(t), I_A(t)$ and $R(t)$ to achieve the best possible future.

$$V(0) = \int_0^{\infty} u(c(t), S(t))e^{-\delta t} dt, \quad \delta > 0 \quad (\text{C.12})$$

Appendix D

Additional Future Work

This appendix presents additional future work that can be undertaken on SalVO and VDSKB.

D.1 ABMs as timely data flow

Previous work [53] showed that (even using its simpler formalism) this mode of ABM computation is very amenable to the scatter-gather programming model [78], illustrated in figure D.1.

Note that there is a clear issue of timing here. Figure D.1 represents agent partitions across different nodes of a single agent type, which can happen synchronously in our formalism. However, agents of different types, as discussed in §3.2.2, require a DAG dependency graph's execution per step. This is attainable in large-scale data processing frameworks using a timely dataflow [76] computation model for execution, illustrated and described in figure D.2. In this model, each loop context would contain nodes where agents of a single type had been partitioned. The collection of agent type loop contexts is held within one broader loop context that executes the main simulation flow.

D.2 Extensions to VDSK-B

Once the modifications in the previous section have been made, a few possible extensions that we can study from the resulting ABMs are detailed here. There are 4 key avenues that are possibly inter-related we want to explore:

1. Planetary Boundaries: The key planetary boundaries under stress [108] are climate change, biosphere integrity, biochemical flows and land-system change. Out of these, only biosphere integrity and climate change are directly modelled in the Modified DSK formulation proposed above. In this proposal, we should define a "planetary boundary" on temperature and $S(t)$, and initialize various taxation systems that penalize economic participants based on global proximity to the boundary.
2. Green Taxes: various static taxes on carbon emissions and pollution can be implemented and explored for their effect vs. the baseline pathway.
3. Carbon Marks: firms in DSK ultimately try to gain market share, and pay taxes on profits. We could impose a compliance market where firms are limited in how much goods they can produce by the emissions they produce unless they purchase carbon

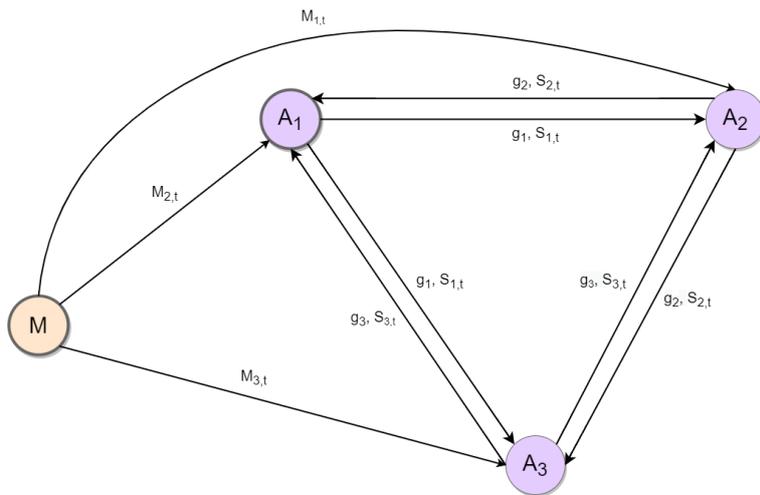


Figure D.1: Scatter-gather applied to state flowing between agent types. Each node is a partition of agents of a single type across nodes. Each agent first gathers state updates from other partitions, and then computes its own state update and emits it. Figure from [53].

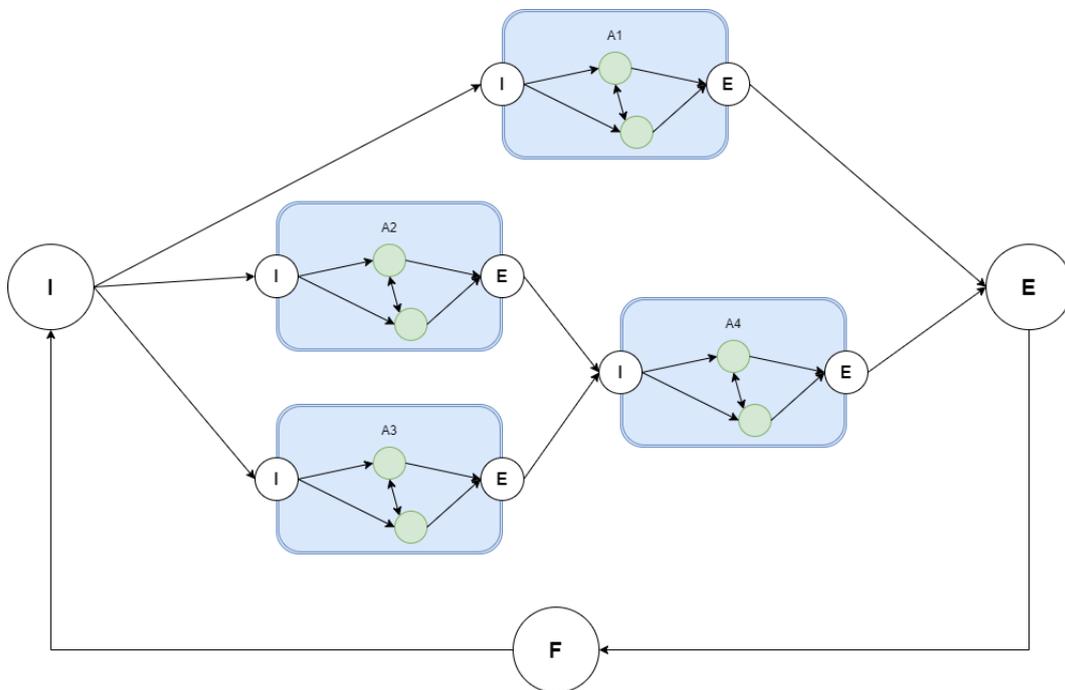


Figure D.2: A timely-dataflow illustration of figure 3.1. Nodes within each loop context compute updates to their state and broadcast it to their sister nodes, before notifying the egress node of completion. The entire flow is itself in a loop context that handles the ABM stepping.

credits, or reduce the emissions through R&D. The prices of carbon credits could potentially be set centrally by the government, and could be a potential optimisation target for Backpropagation based ABM learning. Reducing carbon emissions could directly alter the temperature, which would feed back to $S(t)$ through the growth rate.

4. Biosphere markets: a compliance limitation can be placed on the resources drawn from the biosphere in terms of $R(t)$. Firms can be limited in terms of how much they can produce unless they enter into a compliance market for $R(t)$ and purchase the right to consume more. The price would again be set centrally. This would implicitly influence $S(t)$ by changing the growth-rate, but wouldn't change temperature directly in the Modified DSK model.