European Lighthouse of AI for Sustainability

Deliverable number 1.1
Date: 1.1

First release of methods for Sustainable AI

Project Title	ELIAS - European Lighthouse of AI for Sustainability
Contract No.	101120237
Start of Project	1 September 2023
Duration	48 months

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101120237.

Funded by
the European Union

Deliverable title	First release of methods for Sustainable AI
Deliverable number	D1.1
Deliverable version	1.0
Previous version(s)	-
Contractual date of delivery	August 31, 2024
Actual date of delivery	August 30, 2024
Deliverable filename	ELIAS_D1.1.pdf
Nature of deliverable	Report
Dissemination level	Public
Number of pages	98
Work Package	WP1
Task(s)	T1.1, T1.2, T1.3, T1.4
Parner responsible	UvA
Author(s)	Jan-Willem van de Meent (UvA), Oscar Pellicer, Jorge Vicent Servera (UVEG), Claire Robin, Nuno Carvalhais (MPG), Christian IGel (UCPH), Marius Leordanu (UPB), Saso Dzeroski (JSI), ERfan Mirzai, Massimiliano Pontil (IIT), John Violios, Symeon Papadopoulos, Yiannis Kompatsiaris (CERTH), Florence d'Alché-Buc, Enzo Tartaglione (IPP)
Editor	Jan-Willem van de Meent (UvA)
Project Officer	Evangelia Markidou

Abstract	This document presents the initial outcomes of the research on AI methods for Sustainable Innovation, Climate Modeling, as well methods that can reduce the energy requirements associated with computation in AI. We discuss ongoing work on use case definition (T1.1), methods for AI-accelerated design (T1.2), AI methods for data-driven modeling of complex physical systems (T1.3), AI methods for fast approximation of expensive scientific computation (T1.4), and methods for reducing energy requirements of computation (T1.5).
Keywords	Artificial Intelligence, Sustainable Technologies, Hybrid models, Surrogate Models, Energy-efficient AI.

Copyright

Contributors

NAME	ORGANIZATION
Jan-Willem van de Meent (j.w.vandemeent@uva.nl)	UvA
Oscar Pellicer (oscar.pellicer@uv.es)	UVEG
Jorge Vicent Servera (jorge.vicent@uv.es)	UVEG
Claire Robin (crobin@bgc-jena.mpg.de)	MPG
Nuno Carvalhais (ncarvalhais@bgc-jena.mpg.de)	MPG
Christian Igel (igel@di.ku.dk)	UCPH
Marius Leordeanu (leordeanu@gmail.com)	UPB
Saso Dzeroski (saso.dzeroski@ijs.si)	JSI
Erfan Mirzai (erfunmirzaei@gmail.com)	IIT
Massimiliano Pontil (massimiliano.montil@iit.it)	IIT
John Violos (violos@iti.gr)	CERTH
Symeon Papadopoulos (papadop@iti.gr)	CERTH
Yiannis Kompatsiaris (ikom@iti.gr)	CERTH
Florence d'Alché-Buc (florence.dalche@telecom-paris.fr)	IPP
Enzo Tartaglione (enzo.tartaglione@telecom-paris.fr)	IPP

Peer Reviews

NAME	ORGANIZATION
Georg Martius	MPG
Nicola Gatti	Polimi

Revision History

Version	Date	Reviewer	Modifications
0.1	12/06/2024		Initial version, table of contents and sections, sent to partners for contributions
0.2	19/07/2024		Integrated contributions from UVEG, MPG, UCPH, UPB, IIT, JSI, IPP, CERTH. Added introduction and executive summary. Sent out to internal reviewers.
1.0	30/08/2024		Release version. Integrated comments from internal reviewers

Table of Abbreviations and Acronyms

Abbreviation	Meaning
AI	Artificial Intelligence
cDFT	Classical Density Functional Theory
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DFT	Density Functional Theory
DNN	Deep Neural Network
FLOP	Floating Point Operation
FMT	Fundamental Measure Theory
FPGA	Field Programmable Gated Array
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
GSA	Global Sensitivity Analysis
KD	Knowledge Distillation
mAh	Milliampere-hour
MC	Monte Carlo
MD	Molecular Dynamics
MOF	Metal-Organic Framework
ReLU	Rectified Linear Unit
RTM	Radiative Transfer Models
SIMD	Single Instruction Multiple Data
TPU	Tensor Processing Unit
Wh	watt-hours

Contents

1 Executive Summary

This document reports research outcomes for Work Package 1 of the European Lighthouse of AI for Sustainability (ELiAS). The focus in this work package is on the development of AI methodology for (1) monitoring and forecasting of climate and ecology, as well as (2) development of technologies that make more efficient use of energy and materials, place a lower burden on the ecosystem, and are compatible with a long-term sustainable economic growth.

This interim report summarizes scientific output during the first 12 months of the project. We are happy to report that research efforts at partnering organizations are well under way and continuing to gain momentum. This is evident from the range of results that we present in this deliverable, which comprise contributions from 7 project partners. This document comprises sections on 5 distinct tasks.

Task 1.1: Use Case Requirements and Definition (lead: MPG). This task focuses on the definition and setting up the three use cases that are associated with WP1, which are intended to demonstrate serve as a test-bed for new AI technologies towards a sustainable planet. The wide scope of WP1's use cases carries a large diversity both related to data characteristics as well as to modelling approaches and frameworks. As such, the challenges pertain much to the particularities of each use case and to bridging them across teams to maximize collaboration.

In this task, the three different use cases are: UC1, AI for Building Optimization (team: RB, RBHU), which works towards the development of AI methods for optimization of energy usage; UC5, AI for Forecasting Vegetation State (team: MPG, UVEG, UCPH, UPB), which focuses on utilizing AI to predict the effects of climate extremes on terrestrial ecosystems using remote sensing and in-situ data; andUC6, Open Materials Discovery Competition (team: UvA, MPG), which will implement a competition on AI methods for materials discovery.

In this report, we summarize the progress on the development of dataset relating to energy usage in buildings (UC1). We further report on the active ongoing collaboration between partners working in UC5, as well as on the results of an initial publication and related software release (UC5). We further describe the initial stages for organizing the competition on AI methods for materials discovery, which will involve partnering with a several organizations outside of the ELiAS project, as well as participants in the project (UC6).

Task 1.2: AI for Accelerating the Design of Sustainable Technologies (lead: UvA). This task focuses on the development of AI methods for design problems. This involves inverse reasoning to determine the design parameters of a system, which is often difficult to model that achieve the desired characteristics. A particular class of design problems that are of specific interest are materials design problems, which are relevant to the development of a range of sustainable technologies, including materials for energy storage and carbon capture.

This task is closely connected to Task 1.3, which focuses on data-driven modeling of physical systems, as well as Task 1.4, which focuses on the development of methods for fast approximate computation. The reason for this is that data-driven and computationally efficient models are often a pre-requisite for solving any design problem, since solving such a problem requires computationally screening a large set of candidate designs.

In this report, we summarize contributions from the *JSI* on the development ML methods for materials design based on semi-supervised classification, multi-target regression, and multi-objective optimization. As part of Task 1.4, we additionally report contributions from the *UvA* on fast approximate methods for modeling materials, with the eventual intended use case of screening metal-organic frameworks for carbon capture applications.

Task 1.3: AI for Data-driven Modelling of Physical Systems (lead: UVEG). This task focuses on the use of AI to develop models for physical systems that can be learned from data. Simulation-based approaches, often formulated in terms of differential equations, have historically formed the backbone of predictive modelling in science and engineering. Unfortunately, in many problems related to modeling physical systems, including our climate, we lack knowledge of key parameters, or lack even a detailed model of the underlying dynamics (e.g., when modelling feedback loops in climate science). Work in this task addresses this limitation by developing hybrid models, which can incorporate partial knowledge of a physical systems, whilst also defining a flexible class of models that can be tuned in a data-driven manner.

In this report, we present contributions from UVEG on the development of explainable emulators for atmospheric radiative transfer models, contributions from IIT on learning dynamical systems from data via Koopman/Transfer operators, and initial work from UPB towards sustainable medical AI technologies.

Task 1.4: AI for Fast Approximation of Scientific Computations (lead: JSI). This task focuses on the development of AI-based methods that can provide fast approximations to numerically intensive scientific calculations. A fundamental challenge in the application of AI to both climate modeling and the design of new technologies is that we are often not able to simulate systems at the scale that is needed. In climate modelling, one must resolve all physical processes and complex land-atmosphere-ocean interactions in 3D grid data. Likewise, materials design problems often require computations in statistical physics or quantum chemistry, e.g., those based on density functional theory, which often need to be repeated for many candidate materials.

In this report, we present work from the *UvA* on neural methods for classical density functional theory, which can be used to perform calculations in mesoscopic systems that are orders of magnitude faster than equivalent calculations based on Monte Carlo or molecular dynamics. This work is a first step towards new methods for materials design (Task 1.2), by providing a fast method for screening candidate materials. A second contribution related to this topic comes from *JSI*, which reports methods for estimating the energy of bismuth atom configurations by learning force fields using machine learning methods.

Task 1.5: Reducing the Energy Requirements of Computation (lead IPP). This task focuses on lowering the energy costs associated with computations in AI. This work addresses a growing demand for more frugal approaches to AI, which in recent years has seen a rapid rise in requirements for computing, which has in turn given rise to energy requirements and associated emissions. Paths towards addressing these challenges include training models with less data, hereby reducing the computational demands during training, and using distillation techniques to learn smaller models from bigger models, hereby reducing the computation and energy requirements during deployment.

In this report, we present work from *IPP* on both low-rank approaches to structured prediction, which reduces computational requirements, and on iteratively removing irrelevant layers from large models. *IDEAS NCBR* presents work on zero-waste methods for continual learning, specifically methods based on selective ensembles of experts and methods for exemplar-free continual learning, which both serve to mitigate catastrophic forgetting. *IDEAS NCBR* further contributes methods for federated contrastive learning, which reduce the total bandwidth needed for computation. We also present contributions from *CERTH*, which focus reducing test-time computation by way of knowledge distillation. Finally *HPI* contributes methods for low-precision Gaussian process regression, which serve to decrease power consumption during computation.

2 Task 1.1: Use Case Requirements and Definition

Contributing Partners: MPG, RB, RBHU, UvA, UVEG, UCPH, UPB.

2.1 Overview

The present task focuses on setting up for development the use cases demonstrating artificial intelligence (AI) solutions for environmental sustainability:

- The Use Case 1, **AI for Building Optimization**, explores the application of smart control systems for energy management in buildings. This involves the development of usage control algorithms and the prediction of energy consumption patterns based on historical data and meteorological information. The primary objective is to enhance energy efficiency in built environments.
- The second use case, **AI for Forecasting of Vegetation State**, aims to leverage AI technologies to predict the impacts of climate extremes on ecosystems. This use case integrates remote sensing data with reanalysis meteorological data to develop robust predictive models for vegetation dynamics under various environmental stressors.
- The third use case, **Open Materials Discovery Project**, is designed as a collaborative competition to advance AI methodologies for novel materials discovery. This initiative seeks to accelerate the identification and development of sustainable materials through innovative AI approaches.

Through a series of meetings and discussions, we have carefully delineated the challenges, requirements, and objectives for each use case. This comprehensive approach ensures that each use case is well-defined and aligned with the overarching goal of promoting environmental sustainability through AI-driven solutions.

2.2 Use Case 1: AI for Building Optimization

This section provides an overview about the progress of the data collection for the UC1 Building Optimization at D1.1, D2.1, D3.1 and D5.1. The data collected for this use case will be used to develop AI methodologies to overcome the challenges faced in the building sector.

2.2.1 Digital transformation of the building sector

The digital transformation of the building sector is already starting to have a significant impact on the way buildings are designed, constructed, and operated. Applications of modern computer-aided design tools give architects and engineers the ability to capture more aspects of the building in a digital form. The building's digital representation can be used for a wide range of unprecedented applications.

However, exploiting this opportunity is not a simple task, as the digital models of the buildings under design often do not contain all the necessary information.

The various parts of the plans, such as the building services, electrical, static plans, and the models prepared by the various disciplines, are prepared separately, and do not always enter the digital model. For example, the building services plans often only contain the plans for the heating, ventilation, and air conditioning systems, but do not contain the electrical, EIB, optical, network wiring plans or the plans for the security systems.

The building management and control systems often have their own separate digital representation displaying the building on HMI screens, but these are also not synchronized with the digital model of the building.

Later the digital model needs to be updated repeatedly to reflect the changes made during the construction and operation of the building. Many of the challenges faced in the building sector are related to the lack of data and the difficulty of collecting and updating the data.

2.2.2 Data collection for the UC1 Building Optimization - RBHU Budapest Campus Building BP201

The building B201 selected for data collection is part of the Bosch Innovation Campus Budapest. The campus is a center for innovation and technology. The modern complex houses various business divisions and functions – it comprises office buildings, research and development facilities, social spaces, a parking garage, and a vehicle testing track.

The ground floor is home to laboratories, test benches, and a workshop, while the five-story office buildings above are spaces for cooperation and exchanging ideas. The campus is an expansion of the Engineering Center Budapest, which plays a significant role in the development of automated and electric vehicle technology. The site is one of the leading research, development, and testing facilities for automotive technology within the Bosch Group. The location within the 10th district of Hungary's capital city is exceptionally convenient, providing good transportation links and easy access to the country's largest airport.

In the past months, we evaluated the potential data sources available for the UC1 Building Optimization use case. One part of the data we collect consists of months of recordings of multimodal time-series data describing hundreds of sensors (e.g., temperature, room occupancy, etc.) distributed over the building, energy consumption for the whole building and relevant parts (e.g., larger machines). Most of the data is already being recorded and will be prepared and enriched in Y1-Y2 by RB and RBHU, with special attention to data privacy protection.

The data is collected using a Data Management System (DMS) that acquires and manages data from various sources such as temperature sensors, humidity sensors, air quality sensors, etc. The DMS system provides access to this data through a WCF (Windows Communication Foundation) service, which exposes methods for retrieving the data. The service can be interfaced programmatically.

A second component of the data is the building information model (BIM) of the building. The BIM model in our case is highly detailed and capable of providing extensive information. The BIM model contains information about the building's elements, such as furniture, walls, ceilings, doors, windows, laboratory machines, and some details about mechanical and electrical components. The BIM model also contains information about the rooms in the building, including parameters such as user, cost center, etc. On the one hand, the BIM model is a valuable source of information for the UC1 Building Optimization use case. On the other hand, the BIM model is a complex data set that requires careful evaluation and processing to extract the relevant information for the use case.

2.2.3 Content of data repository

Using the various data sources available on the RBHU Budapest Campus Building BP201, we aim to create a data repository that will contain the following types of data:

1. Time-series data from sensors: This data will include temperature, humidity, air quality, pressure, flow, energy consumption, valve and damper positions, pump and fan status, control

system outputs, switches and relays status, enthalpy, operation counters, setpoints, control values, alarm and fault indicators.

2. Building Information Model (BIM) data: This data will include detailed information about the building's elements, rooms, furniture, walls, ceilings, doors, windows, laboratory machines, mechanical and electrical components, and other relevant parameters.

The data repository will be structured to facilitate data processing, analysis, and modeling for the UC1 Building Optimization use case. The data will be partially cleaned, preprocessed, and stored in a format that allows for easy access and retrieval. The repository will also include metadata and documentation to provide context and information about the data sources and processing steps.

We also aim to provide cross-references between the different data sources to enable integrated analysis and modeling. For example, we will provide metadata that can be used to connect the sensor data to the corresponding elements in the BIM model to create a comprehensive view of the building's dynamics and energy consumption patterns.

We also aim to create a natural grouping of data based on the several types of equipment the sensors are connected to, the medium they are measuring, and the location of the sensors. This grouping will help in organizing the data and making it easier to analyze and model the building's energy consumption and optimization strategies.

The repository will be hosted on ZENODO, a research data repository, to ensure free access and long-term preservation of the data. The data will be made available to the project partners and the wider research community to foster collaboration and knowledge sharing in the field of AI for building optimization.

The first release of the data repository is planned for month 12 of the project, with subsequent updates and enhancements based on the progress of the project and the availability of new data sources. We aim to provide a comprehensive and well-documented data repository that will support the development of AI methodologies for building optimization and contribute to the project's overall objectives. Contributions from the project partners and the wider research community are welcome to extend and enrich the data repository with additional data sources and insights. The data repository released in the first wave will only contain a small subset of the data collected and will be used for preparatory work.

The final release of the data repository is planned to be finished before the competition and challenges organized in WP5. The final repository's content will be based on the feedback received from the project partners on the first release and additional data will be added to the repository that extends the time period. No additional enrichment of the data is planned for the final release, but the full data set will be made available for the competition and challenges organized in WP5.

2.3 Use Case 5: AI for Forecasting of Vegetation State

2.3.1 Introduction

The UC5, led by **Max Planck Society (MPG)**, brings together multiple universities interested in the project, including the **Universitat de València (UVEG)**, the **University of Copenhagen (UCPH)**, and the **University Politehnica of Bucharest (UPB)**. Numerous meetings and discussions have been organized to facilitate collaboration and progress on this subject.

Global warming is driving significant changes in ecosystems and landscapes worldwide. Rising temperatures are altering various ecological processes, resulting in shifts in plant distribution, phenology, and ecological interactions. Anticipating how vegetation will respond to meteorological changes and extreme events (*vegetation forecasting*) in a climate change context can help mitigate

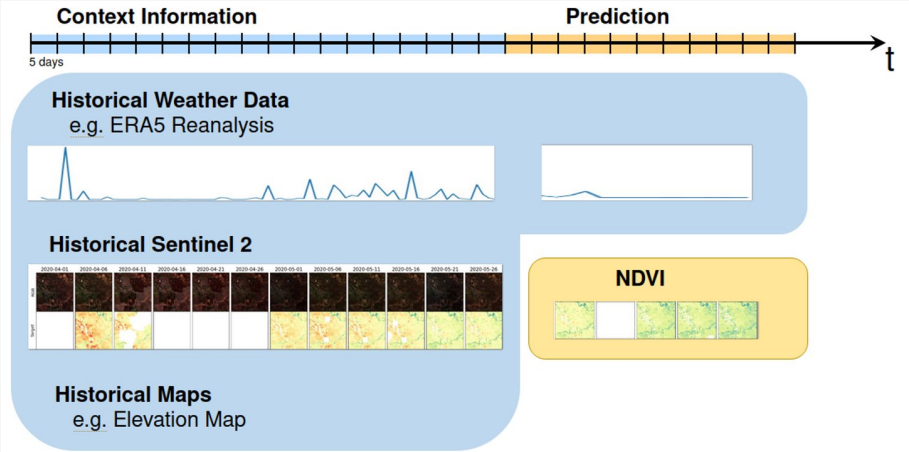

Figure 1. Vegetation forecasting task description. Future vegetation state is predicted with machine learning models from past satellite imagery and past and future weather.

the damage to both human populations and biodiversity in agricultural and natural areas. The forecasting robustness, as in weather, is strongly contingent on lead time, prediction windows, season and geographical context. Additionally, in forecasting vegetation dynamics, changes in vegetation properties themselves depend strongly on the ecosystems development and their inherent resilience and resistance to extreme events. However, and despite understanding of fundamental processes underpinning plant functioning, the impact of extreme events on vegetation at the landscape level is not yet well quantified, as their effects are highly heterogeneous depending on geographic locations, surrounding environment and the history of extreme events in those regions [14], [89]. This complexity poses a challenge in developing accurate vegetation models at landscape level, leading to a growing interest in the development of machine learning models. In a data-driven approach, we define the task as strongly guided video prediction of satellite image time-series [103]. The objective is to forecast a length-K sequence of future vegetation index $(V_{T+1}, \ldots, V_{T+K})$ based on previous length-T sequence satellite imagery $(S_1, \ldots, S_T)$. Predictions are also guided by a set of weather and environmental variables (*e.g.* precipitation, temperature etc.) during context and prediction time steps $(E_1, ..., E_{T+K})$, see Figure 1. Formally, the task is to learn a function f that can be defined as:

$$\hat{V}_{[T+1,\ldots,T+K]} = f(S_{[1,\ldots,T]}, E_{[1,\ldots,T+K]}) \tag{1}$$

2.3.2 Expected deployment

The use case focuses on developing and implementing machine learning methods to predict the impact of climate extremes on vegetation. By leveraging several published Earthnet datasets (https://www.earthnet.tech/), which combine Sentinel-2 data with meteorological variables from the climate reanalysis data ERA5, it avoids the need to develop new, time-consuming datasets. We defined four key gaps crucial to address for understanding vegetation responses to extreme events and supporting efforts to mitigate climate change and protect ecosystems.

2.3.3 Open challenges

First, we propose developing a data-driven definition of extreme events that considers the diversity of vegetation responses to climate extremes and their spatio-temporal extents. We will define classes

for each location and period (Resistant, Impacted, Negative), differentiating various vegetation responses to climate extremes. This approach aims to improve model training and validation by capturing the onset of ecological extremes alongside associated climatic extremes. In addition, it supports handling imbalanced classes and the spatial auto-correlation of geospatial data in machine learning, ensuring a robust estimation our models accuracy on predicting the impacts of climatic extremes on vegetation.

Second, we aim to enhance current state-of-the-art models by using transformer-based approaches, addressing their quadratic time and memory complexity using domain science knowledge to drive the relationship between meteorological and environmental variables. Vegetation forecasting differs from classical video prediction in several key ways that should guide model design. First, the forecasting is spatially static over time; only the intensity of the pixels varies *as trees do not walk*. Second, the temporal dimension is most important; a location's state is influenced by its previous states and long-term weather patterns, such as the slow development and prolonged impacts of drought [71]. Lastly, the surrounding area plays a role but can be captured by a small spatial context, given the very high resolution of the dataset and the understanding that *biogeophysical processes propagate locally*. We propose to develop a transformer-based model with sparse attention: the attention in space is limited to the surrounding pixels and we learn a sparsity function for the attention in time.

Third, we view unseen areas, climate shifts, and novel climate extremes as a data distribution shift problem. We propose to substitute space for time to address the data sparsity issue of climate shift and emphasize estimating the area of applicability of our model using various test sets representing different distribution shifts. We started to explore this direction in a workshop paper listed in the relevant publications section. First, due to the large number of possible distribution shifts, we formulate the problem as *an unlabeled OOD detection task*, meaning the detector does not have access to any samples from the OOD dataset. Second, we will evaluate the OOD detector on several OOD datasets representative of potential distribution shifts, such as new regions, biomes, landcover types, or climates, and estimate the distance to the IID dataset for each OOD dataset using model performance and adversarial validation. Third, we propose to extend the work of [135] by using a ViT auto-encoder with a reconstruction-based pretext task, following the model development done previously. We will adapt this for regression tasks by using a k-NN mean nearest distance approach, following the work of [119] and similar to the method proposed by [101]. We hope to provide a first OOD-detector for regression task in remote sensing.

Finally, we would like to evaluate the reliability of our method using weather forecasts and different climate projections for vegetation impact estimation at short and long term scales. The interest in developing models at different scales stems from the realization that accurately predicting vegetation response to climate extremes is essential for making informed decisions for the development of mitigation and adaptation strategies in a climate change context, contributing to building more resilient and sustainable societies.

2.3.4 Collaborations and future directions

Given the experience that the UCPH and the UPB hold on large models for remote sensing data, and the interest in the domain science, WP1 activities have been supporting the collaboration between the three groups towards the develop this new model while also leveraging previous research on large geospatial models. The realization that foundation models can provide a unique and widely representative perspective on the geographical diversity of vegetation responses to extreme conditions is a potential hypothesis to be tested. Ultimately, the model(s) will be compared against the current state of the art in the domain (recurrent neural networks and transformers-based models), under normal and extreme climatic conditions.

In addition to studying the ability of different foundation models to predict the effect of climate extremes on vegetation, the team at UPB will also address the issue of feature selection across multiple modalities and spatial distributions. This will help us quantify and ultimately better understand how vegetation is influenced by specific climate factors conditioned by geographic context and spatial proximity.

2.3.5 Relevant Publications

- Robin, Claire, Mélanie Weynants, Vitus Benson, Nuno Carvalhais, Marc Rußwurm, and Markus Reichstein. "Spatially Far, Ecologically Close: Evaluating Extrapolation on Vegetation Forecasting Models." in *Machine Learning for Remote Sensing Workshop 2024, at ICLR2024*, .

2.3.6 Relevant Software Release / Datasets

The Earthnet code is available here:

- https://github.com/earthnet2021/earthnet-models-pytorch/
- https://www.earthnet.tech/

2.4 Use Case 6: Open Materials Discovery Competition

The discovery of new materials is important to many sustainability challenges. In the past decade, density functional theory (DFT) based simulation has made significant progress in guiding the experimental discovery of new materials for batteries and carbon capture. However, it remains extremely expensive to discover new materials with brute-force screening. The open materials discovery project is a multi-party collaboration to organise a competition on AI methods for materials discovery. The organisation will be led by the **UvA**, **EKUT**, as well as **Microsoft Research**, which has expressed its intent to participate by way of a letter of support.

We are currently in the initial stages of organizing this partnership, with a tentative goal of organizing a competition at NeurIPS 2025. The current plan is to provide a dataset of unrelaxed structures, for which participants will be asked to predict the top 20 candidates in terms of their match to a set of target properties. We are currently in conversation with Microsoft and will also reach out to other potential partners, such as the Materials Project. The UvA has hired a postdoctoral researcher, who starts in Fall 2024 and will contribute to the organization.

3 Task 1.2: AI for Accelerating the Design of Sustainable Technologies

3.1 Overview

Development of new sustainable technologies involves solving challenges to identify and optimize technologies in a manner that accounts for functional criteria, and more broadly for criteria relating to life cycle analysis and environmental impact. From a technical point of view, a key challenge is that we often need to consider a very large space of possible designs in problems where we can only simulate a limited of designs, and experimentally validate an ever smaller subset. This requires development of methods that appropriately balance exploration and exploitation. The aim of this task is to develop such methods, with a partical focus on methods that are applicable to the design of functional materials, such as those used in energy production and storage.

In Section 3.2, we report initial results for work on ML methods for materials design, focusing for the moment on the specific case of foamed glass materials. This work addresses two question relating to design problems in which we wish optimize materials according to multiple criteria. The first is how we can design semi-supervised learning (SSL) methods that can predict multiple target properties from partially labeled data. The second question is how we can design optimization methods that incorporate mutliple criteria for optimization.

The work in this task in closely related to work in T1.4, which focuses on the development of fast methods for approximation of scientific computations. There are many settings where we can perform accurate simulations of individual candidate designs, but where simulating all possible candidate designs is simply not feasible. In such settings, a pre-requisite for AI-based optimization methods is that we can develop fast surrogate methods that allow us to eveluate a much larger set of candidate designs. We return to this topic in Section 5.2, where we discuss fast simulation methods that combine deep learning with classical density functional theory. Our eventual goal is to use these methods for AI-driven discovery of materials for carbon capture.

3.2 Machine Learning for Materials Design

Contributing partners: JSI

3.2.1 Introduction

We address the problem of materials design, in particular the design of foamed glass materials. Understanding the evolution of the structure of foamed glass materials during the direct foaming process is essential for successful development of novel materials with desired porous structure. However, due to the complexity of the process, this remains a challenging task. We have applied machine learning methods to predict properties of a foamed glass from the parameters of the direct glass foaming process, to provide insight into the process.

Two directions were pursued, both looking at multiple properties of foamed glasses. Along one of these directions, we explore the use of fully labeled, as well as unlabeled data during learning through semi-supervised learning. This means that we can use data about samples of materials (foamed glasses) that have been fully characterized (all properties measured), as well as data about samples that have been only partially characterized (some, but not all properties, measured).

Along the other direction, we explore a combination of machine learning and optimization to identify potential new materials with desired properties. We build multi-target prediction models that predict materials properties from foamed glass composition and parameters of the synthesis

process. We then use multi-objective optimization approaches to identify combinations of compositions and synthesis parameters that are expected to yield materials with desired properties.

3.2.2 Semi-supervised multi-label classification for materials design

We used multi-label classification to predict properties of foamed glass materials. The foamed glass samples were prepared via direct foaming with varying process parameters. In particular, eight parameters are varied. These include content parameters (water glass content, carbon black content, Mn_3O_4 content, K_3PO_4 content) and processing parameters (furnace temperature, foaming time, drying and mixing).

The properties of foamed glass samples were measured by using the Archimedes principle. In particular, five properties were measured. These include the apparent and pycnometric density, along with the overall, closed and open porosities. Each of these can have a low or a high value and is treated as a binary label for the purpose of predictive modelling.

Machine learning methods for multi-label classification were used to learn models that simultaneously predict the five material properties from the eight material composition and processing parameters. The data has 165 examples, where closed and open porosity values are missing for 41 foamed glass samples. This means that the data are partially labeled and calls for the use of semi-supervised learning (SSL).

We use predictive clustering trees for multi-label classification (MLC), as well as ensembles (random forests) thereof, both in supervised and in semi-supervised manner. Using random forests for MLC allows us to also estimate variable importance, both in the supervised and semi-supervised case. Finally, we also use tree ensembles in self-training mode for SSL.

The predictive models built have good predictive power (as measured via AUC), where ensembles have better predictive power than single trees. The use of incompletely labeled data in semi-supervised learning mode improves the predictive power, with the self-training approach performing best. The material composition (and in particular carbon black content) is more important than the processing parameters, of which furnace temperature is the most important. Theses insights will be useful in the design of new foamed glasses with desired properties.

3.2.3 Combining multi-target regression and multi-objective optimization for materials design

We next address the problem of multi-criteria foamed glass design more directly, by using a combination of machine learning and optimization (MOO) approaches. In particular, we use approaches for multi-target regression, on one hand, and multi-objective optimization approaches, on the other hand. We consider two properties of foamed glasses.

The design problem is to obtain foamed glass with high closed porosity (ϵ_{cl}) and low apparent density (ρ_{app}), which can be formulated an as optimization problem with two conflicting objectives. Therefore, our aim is to find the combination of foamed glass composition (in terms of four components) and parameters of the synthesis (foaming) process that will simultaneously optimize the two objectives. We use machine learning methods to predict (two) properties of foamed glass from (nine) parameters of the direct glass foaming process. The latter include composition parameters (water glass content, carbon black content, Mn3O4 content, K3PO4 content) and processing parameters (furnace temperature, heating rate, foaming time, drying and mixing).

The two properties to predict are ϵ_{cl} and ρ_{app}. We build multi-target regression models for this purpose. These include neural networks (multi-layer perceptrons). The data from which the models are built has 124 examples, i.e., has been collected for 124 foamed glass samples.

The learned model then represent a two-dimensional objective function, which is taken as input for multi-objective optimization (MOO) methods. The MOO methods then identify combinations of inputs (parameter of the glass foaming process) that yield non-dominated solutions on the Pareto front. These either have the lowest ρ_{app} for a given ϵ_{cl} or the highest ϵ_{cl} for a given ρ_{app}. All of these represent potential solutions to the multi-objective material design process, among which the user (materials science researcher) can choose.

3.2.4 Next steps

We are already working on extending the above work in several directions. First, the multi-target regression (MTR) models used in combination with multi-objective optimization were built in fully supervised mode, using 124 of the 165 available material samples. The natural next step is to use SSL to learn the MTR models. Second, we would like to test the suggested new materials proposed by the combination of ML and MOO: Our materials science collaborators are willing to synthesize several sample and characterize their properties. Finally, we would like to extend the SSL approaches to perform active learning and close the loop of scientific exploration.

3.2.5 Relevant publications

There are no as yet published papers or completed drafts submitted for publication describing the above work. A draft paper is in preparation and will be reported on in the next periodic reports. However, the work has been described in three presentations (one oral and two poster):

- Sintija Stevanoska, Uroš Hribar, Jurica Levatic, Sašo Džeroski. Semi-supervised multi-label classification for materials design. *Workshop on new developments in automated learning and reasoning*, Leuven, 7th-8th Feb 2024.
- Sintija Stevanoska, Uroš Hribar, Jurica Levatic, Sašo Džeroski. Semi-supervised multi-label classification for materials design. *4th Nobel Turing Challenge Initiative Workshop*, Tokyo, 13th-14th Feb 2024.
- Christian L. Camacho-Villalón, Sintija Stevanoska, Uros Hribar, Matjaž Spreitzer, Jakob König, Sašo Džeroski. Multi-criteria foamed glass design with multi-target regression and multi-objective optimization. *4th Workshop on Machine Learning Modalities for Materials Science*, Ljubljana, 13th-17th May 2024.).

4 Task 1.3: AI for Data-Driven Modelling of Physical Systems

4.1 Overview

This task focuses on the development of hybrid methods, which combine traditional modeling techniques that rely on our knowledge of the physical principles of the underlying system, which flexible parameterization of AI-based systems that can be trained on data. There is an incredible diversity of possible approaches in this space, and this report reflects the diversity of approaches that are currently being pursued in the ELiAS network.

In Section 4.2, we present work that explores the application of statistical methods to enhance the physical interpretability of emulators for atmospheric radiative transfer models. On the one hand, feature selection methods identify and select the relevant features in the input space that impact the model outputs. On the other hand, multifidelity is used to improve the accuracy and runtime of emulators.

In Section 4.3, the we present work that applies the Koopman operator regression framework to break down complex dynamical systems into simpler, coherent structures, thus facilitating the development of physically-informed machine learning models for dynamical systems. The proposed methodology has numerous applications, including fluid dynamics, molecular kinetics, and robotics.

In Section 4.4, we present new data-driven methods for medical image applications, specifically methods for pulmonary embolism detection, with an eventual aim of improving a capability to develop a sustainable society, in which medical AI can better protect and improve our health, while reducing energy consumption and human effort.

In Section 4.5, we present probabilistic grammars for modeling dynamical systems from coarse, noisy, and partial data. Probabilistic grammars are a novel method inferring ordinary differential equations, integrating data and knowledge-driven approaches to automate the modeling of dynamical systems.

4.2 Explainable emulator for atmospheric radiative transfer models

Contributing partner: UVEG

4.2.1 Introduction and methodology

Introduction. Statistical regression methods are widely used in remote sensing applications, such as classification, biophysical parameters retrieval, and emulation [22], [29]. These methods offer numerous advantages, including accuracy, adaptability, and computational efficiency. Nevertheless, the mathematical implementation and hyperparameters of the underlying machine learning algorithm tend to lack physical interpretability. This opacity hampers our ability to understand how the predictions are generated. Physics-aware machine learning is an approach that addresses this issue by incorporating physics knowledge into machine learning models [51], [66], [112], [136]. This has the potential to enhance the accuracy, reliability, and performance of statistical regression models while providing a degree of model explainability that helps us to better understand the relationships between the input and output variables. Incorporating physical knowledge into a statistical regression method can be done in various ways, such as using physics-based features [12], implementing physical constraints into the model [102], or generating training datasets with physical models [29], [70].

We explore a combination of these methods to enhance the explainability and physical interpretability of emulators for atmospheric radiative transfer models (RTM). These models describe

the absorption, scattering, and emission of radiation by atmospheric constituents based on physical principles and given a configuration of optical properties and geometric conditions. Atmospheric RTMs are used in a variety of applications, such as remote sensing of the atmosphere, numerical weather prediction, and atmospheric correction of satellite data. Due to their increasing complexity and associated computational burden, these models are rarely directly used in operational applications. A common solution is to interpolate look-up tables (LUT) of pre-computed RTM simulations. However, the LUT size increases exponentially, implying an increasing time to generate it and stringent RAM requirements. This is particularly challenging for hyperspectral Earth Observation satellite missions. With several hundred spectral channels, the data volume of LUTs will increase by one or two orders of magnitude with respect to current multispectral missions, limiting the applicability of state-of-the-art atmospheric correction algorithms. Statistical regression models known as *emulators*, or surrogate models, have been proposed as an alternative to LUT interpolation [7], [31], [122]. Emulation approximates the behaviour of a deterministic model at a fraction of its runtime, reducing the LUT size and interpolation errors.

Two methods are explored to enhance the physical interpretability of RTM emulators: (1) feature selection, and (2) multi-fidelity. Supervised feature selection can be used to construct physics-aware statistical regression models. Feature selection is a technique that aims at reducing the number of input variables in a model by selecting the most relevant ones based on their impact on the model outputs [18], [100]. There are three main mechanisms for feature selection: (1) wrapper, (2) filter, and (3) intrinsic (or embedded) methods. The most important sub-class of the wrapper methods is called stepwise methods: they create multiple regression models varying the subset of selected features and choosing the most accurate one [18]. Filter methods use statistical techniques to assess the relationship between input and output variables [19]. Intrinsic methods refer to statistical regression algorithms that automatically perform feature ranking and selection during model training. Examples of intrinsic methods include automatic relevance determination (ARD) applied to Gaussian Processes (GPs) [2], [8], neural networks [91] and random forests [5]. All these approaches have been widely used in remote sensing applications [28], [37], [51], [144]. In this research activity, we developed a wrapper feature selection method to enhance the physical awareness and explainability of emulators of atmospheric RTMs. Our method sequentially ranks relevant features as they minimize a cost function and determines the optimal number of features through the spectral information criterion (SIC) [133]. This study derived global sensitivity analysis (GSA) from the feature selection method, identifying the most relevant input features of an RTM. Moreover, the method was applied to embed physics information into emulators by selecting only the relevant features affecting the input-output relationships.

To further enhance the accuracy of model predictions whilst reducing runtime, multi-fidelity methods have been developed [4]. These methods combine limited simulations of an accurate but computationally expensive model (high-fidelity) with a larger simulation dataset from a fast but less accurate model (low-fidelity) [61]. By merging various fidelities, multi-fidelity methods correct computationally cheap models so that the outputs resemble those of more accurate models. In the context of atmospheric RTMs, multi-fidelity methods have been implemented in MODTRAN's scaled-DISORT method [39] and more recently in the Cluster Low Streams Regression (CLSR) method [88]. These implementations rely on simplistic approximations or have limited applicability. The scaled-DISORT method scales MODTRAN's 2-stream simulations using a scaling factor obtained from DISORT simulations at fewer wavelengths and interpolates linearly for all remaining wavelengths. Conversely, the CLSR method is only applied on a small spectral range around gaseous absorptions (e.g. O2-A band).

Considering the above limitations, our objectives are: (1) to improve the runtime and accuracy of atmospheric RTM emulators by using multi-fidelity methods, and (2) to implement a generic

approach that is valid for multiple RTM and covers the full spectral range (400-2500 nm).

Explainability through feature selection. The proposed method ranks the d features of the input space taking into account their impact on the output of a deterministic model $g(\mathbf{x})$ by using a *forward selection* method [54], [118]. This method consists of adding recursively variables minimizing a cost function χ_k that measures the difference between $g(\mathbf{x})$ and $g_k(\mathbf{z}^{(k)})$. Here $g_k(\mathbf{z}^{(k)})$ represents a *partial* model that uses $k \leq d$ input features. The method starts searching for the most significant single variable model (in terms of the cost function value), i.e., considering a *partial* model with only one feature ($k = 1$). This search is repeated considering a *partial* model with $k = 2$ variables, re-estimating the model for each pair of variables, including and keeping the previously selected variable. This procedure is repeated until reaching a complete model of $k = d$ variables thus, in practice, providing a sequence of variables that will be the final ranking. Since the forward selection method is based on minimizing the cost function χ_k, the produced ranking depends directly on the model $g(\mathbf{x})$. Namely, the *importance* associated with each feature is related to the output that we are analyzing. In this work, we used the L_1 and L_2 norms (and their relative counterparts) for the cost function χ_k.

The forward selection method has two main drawbacks. The first drawback is related to the definition of the *partial* model $g_k(\mathbf{z}^k)$ as a reduced version of $g(\mathbf{x})$. The *partial* model $g_k(\mathbf{z}^k)$ takes values in a smaller space than $g(\mathbf{x})$ since $\mathbf{z}^k$ has a smaller dimension than $\mathbf{x}$ (except for the last iteration, where they have the same length). Therefore, in order to obtain a *partial* model, we should integrate all the features that are not included in $\mathbf{z}^k$. As a consequence, the partial models are analytically unknown. The second drawback is that the model $g(\mathbf{x})$ is computationally slow in real-life scenarios (e.g., atmospheric RTMs [39]). Thus, the forward selection method would be impractical given the many simulations needed. To overcome these two drawbacks, we use instead a regression function $\widehat{g}_k(\mathbf{z}^{(k)}) : \mathbb{R}^k \rightarrow \mathbb{R}$ that approximates $g(\mathbf{x})$ with a much faster run time. Given a specific choice of the elements z_{ij} in $\mathbf{z}^{(k)}$, $\widehat{g}_k(\mathbf{z}_i^{(k)})$ is obtained from the regression $\mathbf{z}_i^{(k)} \rightarrow y_i$, thus linking a subset of all possible features to the output y. The specific regression method employed has an impact on the ranking of features given that the cost function depends directly on $\widehat{g}_k(\mathbf{z}_i^{(k)})$. However, if the regression method used is accurate enough and the parameters (or hyper-parameters) are well-tuned, the obtained ranking should not change substantially for a change of the regression function. Although emulators are a priori best suited due to their higher flexibility and accuracy [122], they need to be re-trained every time a new feature is added. This makes them slow for the forward selection method. Instead, a well-designed parametric model can capture the main dependencies of $g(\mathbf{x})$ while being fast to "train" (e.g., through least-squares fitting) and run. In the application discussed in this study (i.e., atmospheric RTMs), a d-dimensional 2^{nd} degree polynomial fitting was found as a pragmatic solution given the smooth dependencies of the output spectral data (e.g., transmittance) to the input atmospheric and geometric features.

In the context of **GSA**, the forward selection method provides a direct way to compute sensitivity indices (SI) using the cost function error magnitude with k features, $V(k)$. We defined the SI (ranging from 0% to 100%) for each ranked feature in $\mathbf{z}^{(d)}$ as follows:

$$SI(k) = \frac{100 \cdot [V(k) - V(k-1)]}{\sum_{i=1}^{d} [V(i) - V(i-1)]} \tag{2}$$

Indeed, the decrease in the error magnitude $V(k)$ will be the highest in the first feature of the ranking (k=1) and the lowest in the last feature (k=d). For multi-output models, applying the

forward selection method as a regression-based GSA algorithm is reduced to using the single-output version and looping over each output dimension.

In the context of **emulation**, feature selection was applied to include only the relevant input features when training and running a statistical regression algorithm, thus making the emulator model more accurate and interpretable. Without loss of generality, we used GP emulators [145] and we proposed two options to use feature selection. The first option is to apply feature selection directly on the multi-output data so that, instead of $\mathbf{x}_i \rightarrow \mathbf{g}(\mathbf{x}_i)$, the emulator does now the regression $\mathbf{z}_i^{(k)} \rightarrow \mathbf{g}(\mathbf{x}_i)$. In practice, this option is equivalent to a GP emulator with a Gaussian kernel (see equation 3) with only two hyper-parameters (θ_f and θ_l) but with a reduced number of input dimensions.

$$k(\mathbf{x}, \mathbf{x}^*) = \theta_f^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x}^*)^\top(\mathbf{x}-\mathbf{x}^*)}{2\theta_l^2}\right) \tag{3}$$

The second option uses the sensitivity index, $SI(k)$, to adjust the influence of features within the regression model. This is achieved by defining the Mahalanobis Gaussian kernel as:

$$k(\mathbf{x}, \mathbf{x}^*) = \theta_f^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{x}^*)^\top W(\mathbf{x}-\mathbf{x}^*)}{2\theta_l^2}\right), \tag{4}$$

where W is a $d \times d$ diagonal matrix with values $W_{kk} = SI(k)^2$ for k=1 to d. This equation 4 can be re-written as:

$$k(\mathbf{x}, \mathbf{x}^*) = \theta_f^2 \exp\left(-\sum_{k=1}^{d} \frac{W_{kk}(x_k - x_k^*)^2}{2\theta_l^2}\right), \tag{5}$$

which is equivalent to the ARD-Gaussian kernel in equation 6:

$$k(\mathbf{x}, \mathbf{x}^*) = \theta_f^2 \exp\left(-\sum_{k=1}^{d} \frac{(x_k - x_k^*)^2}{2\theta_{l,k}^2}\right), \tag{6}$$

where $\theta_{l,k}^2 \equiv \theta_l^2 / W_{kk}$. The distinction between the ARD and Mahalanobis kernels lies in the fact that the ARD version utilizes one scale-length hyper-parameter for each input dimension, whereas the Mahalanobis kernel employs a single hyper-parameter. Both kernels provide a degree of model explainability, wherein less relevant features are characterized by higher values of their associated scale lengths, resulting in reduced impact on the model.

Multi-fidelity modeling. Multi-fidelity emulation is an approach to modeling complex systems using multiple layers of approximation. The method builds upon simpler models, progressively adding layers to increase accuracy. Mathematically, a multi-fidelity model can be expressed through the equation $\widehat{\mathbf{g}}_t(\mathbf{x}) = c \cdot \widehat{\mathbf{g}}_{t-1}(\mathbf{x}) + \boldsymbol{\delta}_t(\mathbf{x})$, where $\widehat{\mathbf{g}}_t$ and $\widehat{\mathbf{g}}_{t-1}$ are two subsequent fidelity layers of the model executed at the input conditions in $\mathbf{x}$, c is a scaling factor of the lower fidelity later, and $\boldsymbol{\delta}_t(\mathbf{x}) \in \mathbb{R}^b$ models the bias between two fidelities.

In our implementation, the lowest-fidelity model, $\widehat{\mathbf{g}}_0(\mathbf{x})$, is a 2$^{\text{nd}}$ order polynomial fitting the training data. The polynomial model is not only fast predicting new outputs but also representative of the main trends that describe the dependencies between input and output spectral data of atmospheric RTMs (mainly exponential, cosine, and power functions). For the first higher fidelity layer, we evaluate the polynomial on the training dataset to get $\widehat{\mathbf{g}}_0(\mathbf{x}_i)$ ($i = 1$ to n) and calculate the difference between the training data and the predictions by the lowest fidelity model, i.e., $\boldsymbol{\delta}_1(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i) - \widehat{\mathbf{g}}_0(\mathbf{x}_i)$ ($i = 1$ to n). We then construct a new training dataset $\{\mathbf{x}_i, \boldsymbol{\delta}_1(\mathbf{x}_i)\}_{i=1}^n$

to train a GP emulator $\widehat{\boldsymbol{\delta}}_1(\mathbf{x})$ to approximate $\boldsymbol{\delta}_1(\mathbf{x})$. Accordingly, the first higher fidelity layer is $\widehat{\mathbf{g}}_1(\mathbf{x}) = \widehat{\mathbf{g}}_0(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_1(\mathbf{x})$. This process is repeated for a user-defined number of layers n_l. For example, for the second layer, we apply a GP emulator of the previous layer to construct a new training dataset $\{\mathbf{x}_i, \boldsymbol{\delta}_2(\mathbf{x}_i)\}_{i=1}^n$, where $\boldsymbol{\delta}_2(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i) - \widehat{\mathbf{g}}_1(\mathbf{x}_i)$. This new dataset is then used to train another GP emulator $\widehat{\boldsymbol{\delta}}_2(\mathbf{x})$ to approximate $\boldsymbol{\delta}_2(\mathbf{x})$. The prediction for a 2-layers multi-fidelity emulator would be $\widehat{\mathbf{g}}_2(\mathbf{x}) = \widehat{\mathbf{g}}_1(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_2(\mathbf{x}) = \widehat{\mathbf{g}}_0(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_1(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_2(\mathbf{x})$. The multi-fidelity process is summarized in the following pseudo-code:

Algorithm 1 Multi-fidelity GP emulator (*training*)

Require: Training dataset
  Fit 2nd degree polynomials for the data $\{\mathbf{x}_i, \mathbf{g}(\mathbf{x}_i)\}_{i=1}^n$: $\widehat{\mathbf{g}}_0(\mathbf{x})$
  for t=1 **to** n_l **do**
    - $\{\widehat{\mathbf{g}}_{t-1}(\mathbf{x}_i)\}_{i=1}^n \leftarrow$ Run $t-1$ fidelity layer on training data
    - Calculate $\boldsymbol{\delta}_t(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i) - \widehat{\mathbf{g}}_{t-1}(\mathbf{x}_i)$ for $i = 1$ to n
    - $\widehat{\boldsymbol{\delta}}_t(\mathbf{x}) \leftarrow$ Train emulator considering the pairs input-outputs $\{\mathbf{x}_i, \boldsymbol{\delta}_t(\mathbf{x}_i)\}_{i=1}^n$
    - Set $\widehat{\mathbf{g}}_t(\mathbf{x}) = \widehat{\mathbf{g}}_{t-1}(\mathbf{x}) + \widehat{\boldsymbol{\delta}}_t(\mathbf{x})$
  end for
Ensure: Emulators and polynomial fitting: $\{\widehat{\boldsymbol{\delta}}_t(\mathbf{x})\}_{t=1}^{n_l}$, $\widehat{\mathbf{g}}_0(\mathbf{x})$

This layered approach allows the emulator to capture increasingly fine details of the system behavior. The final prediction is the sum of the lowest fidelity model and all the difference models, $\delta(\mathbf{x})$, from higher fidelity layers. The number of layers is user-defined, allowing for a balance between accuracy and computational efficiency as each new layer increase runtime while improving the model's accuracy.

4.2.2 Experiments

Explainability through feature selection. We first analyzed the influence of the cost function on the performance of the feature ranking method. The $V(k)$ curve was calculated at four selected wavelengths and with four cost functions. The values of $V(k)$ are normalized for their corresponding values at k=1 to better compare the results from each cost function. In Figure 2 (left) we observe a similar behavior of the normalized $V(k)$ curves for all the tested cost functions. In all cases, $V(k)$ shows a decreasing trend with the biggest decrease happening when the number of features is increased from k=1 to k=2. All curves show a nearly flat behavior after 4 to 6 features, indicating that additional features only contain residual information.

To complement these results, we conducted a regression-based GSA using the $V(k)$ curves to calculate Sensitivity Indices (SI). Figure 2 (right) depicts the GSA results for E_{dif} across various wavelengths and cost functions. Each color in the figures corresponds to an input feature and the bar size reflects its importance (SI). We observe that all cost functions get similar GSA results. Two main features (SZA ■ and AOT ■) are always identified as the key drivers for all wavelengths with a SI of 30% to 80% for AOT and 15% to 60% for SZA. The relative norms show more sensitivity to secondary features (surface elevation ■, α ■, and g ■) with an SI ranging from 5% to 20% depending on the wavelength. The results from the various wavelengths indicate that the forward selection method is sensitive to the expected relevant features. For example, at 761 nm the method shows a higher relevance of the surface elevation, and at 940 nm it is sensitive to CWV ■. The RAA ■ does not show any influence in the GSA results, in line with the expected for nadir-view simulations.

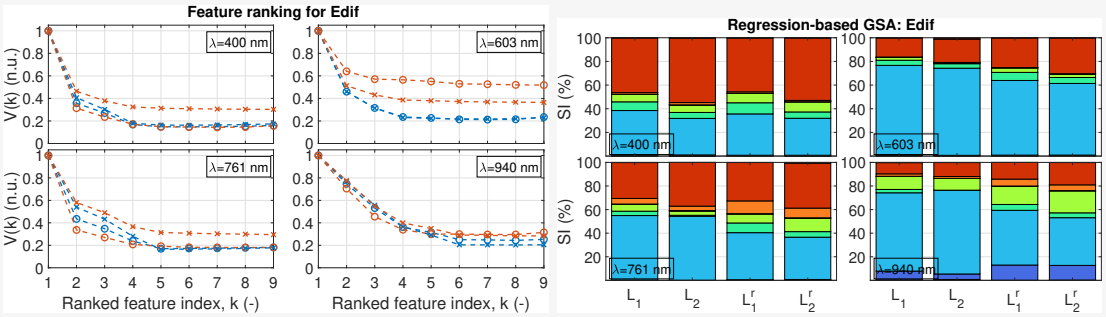

Figure 2. Left: Curve $V(k)$, normalized to 1 at k=1, at various wavelengths (λ) and cost functions: L_1 (blue) and L_2 (red). Markers indicate absolute, o, and relative, x, norms. Right: GSA results for E_{dif} and each tested cost function. L_p^r refers to the relative L_p norm.

We analyzed how feature selection impacts the accuracy of an emulator. For that, we compared the performance of four GP emulator configurations: Gaussian kernel with and without feature selection (#1 and #2 respectively), ARD-Gaussian kernel without feature selection (#3), and Mahalanobis Gaussian kernel with feature selection (#4). In Figure 3 we show the mean relative error (MRE) achieved by these four emulators against a reference test dataset (see [152] for further information). All emulators show similar spectral behavior of the MRE, where the higher values correspond to spectral regions with lower surface reflectance values, as expected given the nature of the relative error. In addition, the errors are higher inside gaseous absorption (mainly H_2O and O_2) due to divisions by nearly zero during the inversion of surface reflectance. Moreover, the MRE values tend to be higher towards shorter wavelengths (<500 nm) due to the impact of aerosol scattering. In terms of accuracy, the highest errors are obtained with the basic GP emulator (configuration #1). Feature selection (configuration #2) improves the results by 0.1% to 0.2% depending on the wavelength. The only exception is inside the O_3 absorption (530-640 nm), where the GP emulator with feature selection obtains the highest errors (1.5%). Applying the feature ranking through the Mahalanobis kernel (configuration #4), the results are improved by nearly a factor 2 in the visible spectral range (400-700 nm) and remain the same in the rest of the spectral range. The emulator with an ARD-Gaussian kernel (configuration #3) achieves the lowest errors in all wavelengths with MRE values 0.2% to 0.6% outside of absorption bands.

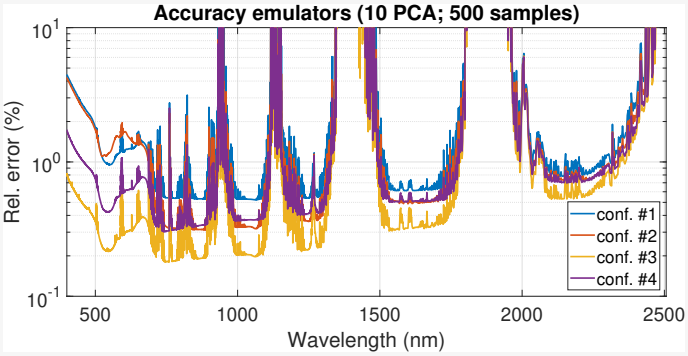

Figure 3. Spectral MRE (in %) for various emulators with and without feature selection (see legend).

Tab. 2 summarizes the accuracy results from the three GP emulator configurations, along with their runtimes when predicting 10000 samples. These results demonstrate that feature selection

enhances the accuracy of a GP emulator with a Gaussian kernel by 0.2%, all while maintaining a negligible increase in runtime. Nonetheless, the ARD-Gaussian GP emulator exhibits superior accuracy, albeit at the cost of longer runtime. The Mahalanobis Gaussian kernel (configuration #4) requires as much runtime as the ARD-Gaussian GP emulator but with nearly twice the value of MRE_λ.

Table 2. Accuracy (MRE_λ, in %), runtime (in s), and explainability for the prediction of 10000 samples using four GP emulator configurations.

Config.:	**#1**	**#2**	**#3**	**#4**
MRE_λ (%)	1.8	1.6	0.5	0.9
Runtime (s)	1.4	1.5	2.3	2.3
Explainability	Low	Medium	Medium	High

Multi-fidelity modeling. The proposed multi-fidelity approach was used to train and validate emulators with various configurations, varying the number of layers and PCA components. A full analysis of the validation results is presented in [145] but a summary of the main findings is described within the next paragraphs.

The performance of the multi-fidelity GP emulators was first analyzed as a function of the number of PCA components (n_c) and multi-fidelity layers (n_l) using a training database of 500 samples. Their accuracy is compared by plotting the spectral mean relative errors (MRE) (Figure 4) (only the results with n_c=7 are shown here). Generally, all results show similar spectral behavior with relative errors increasing inside the deep H_2O and O_2 absorption bands. These results indicate that increasing the number of PCA components reduces the errors in the predicted spectral data. This is particularly observed in the case of the simple (i.e., no multi-fidelity, n_l=0) GP emulator, where the errors are reduced by a factor ∼5 when passing from three PCA components to 15 components. The higher errors associated with a low number of PCA components (n_c=3) are somewhat compensated by adding extra layers in the multi-fidelity GP emulators. This improvement seems to reach a saturation limit after n_l=3 layers. However, this lower limit in the MRE values is achieved with fewer fidelity layers when adding more PCA components. In the extreme case of n_c=15, an emulator of only one fidelity layer obtains the lowest MRE values. This error is still lower than with a simple GP emulator in the 400-1100 nm spectral range. It is also observed that the addition of new fidelity layers reduces the values of MRE differently depending on the spectral range and the number of PCA components. For instance, with n_c=7, passing from n_l=1 to n_l=2 only reduces the MRE for wavelengths above 1000 nm.

The results of the experiment indicated that increasing the number of PCA components and multi-fidelity layers reduce prediction errors. However, adding extra layers in a multi-fidelity emulator seems to reduce faster the error values than adding additional PCA components. The lowest error value of 0.4% is achieved with three multi-fidelity layers regardless of the number of PCA components. This error value can also be achieved with as little as 5 PCA components and three layers or 7 PCA components and two layers. The analysis was done in terms of prediction runtime. The runtime varies linearly as a function of n_c and n_l. As expected, the fastest emulator is also the simplest (n_c=3 and n_l=0) and calculates the prediction in 1.5 s. On the contrary, the slowest emulator is also the most complex (n_c=15 and n_l=5) with a runtime of 18 s. However, a balanced emulator consisting of 5 PCA components and three multi-fidelity layers has a runtime of 5 s and achieves the same accuracy (0.4%) as the most complex emulator. Since the combination of the number of PCA components and multi-fidelity layers can compensate one another and reach similar accuracy and performance, we plot in Figure 5 a bar chart with the product of runtime and

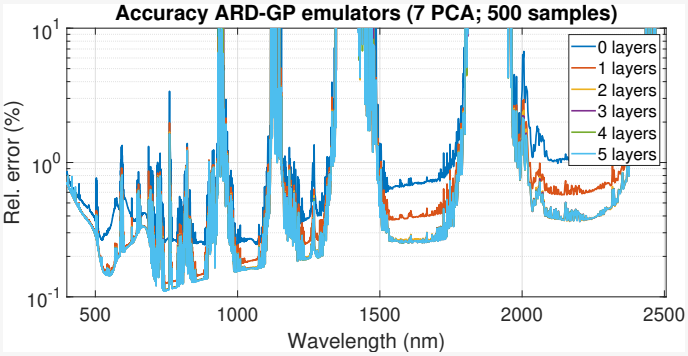

Figure 4. Spectral MRE (in %) for the multi-fidelity emulators (from 0 to 5 layers) for 7 PCA components. Training dataset size: n=500.

prediction error. Indeed, we seek an emulator that obtains the lowest errors with a competitive runtime. This figure shows that the most balanced emulators are achieved with n_l=1 multi-fidelity layers regardless of the number of PCA components. Among them, the best balanced emulator is achieved with 5 PCA components and one multi-fidelity layer, which results in an error of 0.54% and a runtime of 1.86 s. To achieve the same accuracy-time performance as the fastest emulator (n_c=3 and n_l=0), an emulator with n_c=3 and n_l=4 (or with n_c=7 and n_l=3) should be considered. That is, the gain in accuracy (1.8% to 0.4%) compensates for the increase in runtime (1.5 s to 6.3 s),

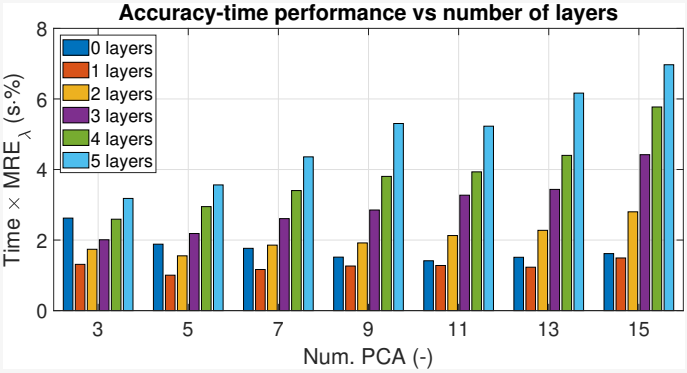

Figure 5. Product of prediction error and runtime as a function of the number of PCA components and multi-fidelity layers. Training dataset size: n=500.

Based on these findings, we fixed the number of PCA components to n_c=5 and studied the impact of training dataset size on accuracy and performance (see results in [145]).The experiment indicated that, in all emulator configurations, the prediction errors decrease when increasing the size of the training dataset. However, this reduction is driven by the number of fidelity layers. With a small training dataset of n=100 training samples, the simple GP emulator (i.e., n_l=0) obtains the lowest MRE compared to any multi-fidelity emulator. The increase of the training dataset size to n=500 reduces the MRE results of this simple emulator by nearly a factor of 10 in the visible spectral range without major improvements at longer wavelengths (>1500 nm). Further increase in the training dataset size (n=1000) does not improve the emulator accuracy. This situation is reversed with emulators of one or more layers after 500 training samples. In these configurations,

the MRE values are lower than with n_l=0 for all wavelengths. When increasing n to 1000, no further improvements are observed at wavelengths above 1500 nm in the case of n_l=1. Yet, adding extra layers allows the emulators to reduce error values further by exploiting the data from larger datasets.

4.2.3 Conclusions

In the research activities presented in Sections 4.2.1 and 4.2.2, we have made progress in the development of physics-aware emulators for atmospheric RTMs.

In the first study, we proposed a feature selection method to enhance the explainability of emulators by integrating physics knowledge through statistical regression. We presented two approaches: (1) direct feature selection on spectral data and (2) constructing a Mahalanobis Gaussian kernel on the GP regressor for each PCA component. Our results show that this physics-aware feature selection improves the accuracy of emulators by 0.2% and enhances model explainability, although the ARD-Gaussian kernel GP emulator still achieved the highest overall accuracy. Despite modest accuracy gains, this method could overcome the limitations of an ARD-Gaussian GP regression in high-dimensional input spaces without requiring dimensionality reduction.

In the second study, we implemented multifidelity methods to boost the accuracy and performance of atmospheric RTM emulators. We analyzed key configuration parameters and found that multifidelity significantly reduces prediction errors, outperforming simple GP emulation. By doubling the training samples, prediction errors decreased by about 50%, though runtime also doubled. We recommended an emulator with one fidelity layer, five PCA components, and 500 training samples, achieving a prediction error of 0.56% in 1.86 seconds for 10,000 samples.

Both methodologies can be combined, reaching efficient and physics-aware emulators. However, there is still room for improvement to achieve fast and more accurate emulators that could potentially be used for operational satellite data processing. These strategies include using active learning techniques, and optimized implementation of emulators to improve the accuracy and performance of atmospheric RTM emulators, enabling their widespread use in operational satellite data processing. Physics-awareness can be further enhanced through the use of *symbolic regression* methods into so-called *hybrid modeling*. Our ongoing research extends our previous work on enhancing emulator interpretability through feature selection in model inputs to now focus on feature selection in model outputs. Previously, we improved emulator interpretability by selecting only relevant input variables. Now, we aim to apply a similar principle to model outputs using symbolic regression methods, specifically LASSO [131]. This approach applies physics-based dimensionality reduction and feature selection to the output space. By expressing data with a semi-empirical parametric model and using LASSO, we identify the most relevant interpretable model parameters, which then serve as new variables for training a statistical regression model. This method effectively replaces our previous use of PCA for dimensionality reduction with a more physically interpretable approach. Symbolic regression, which assumes physical laws can be described by sparse and algebraic input-output relationships, offers the ability to discover mathematical models from data patterns. While various symbolic regression methods exist, we are focusing on LASSO for its simplicity, efficiency, and scalability in handling larger input spaces. This approach bridges our previous work with a new focus on output space, offering a comprehensive strategy for creating physically interpretable emulators that are efficient in both input and output representations.

4.2.4 Relevant Publications

- J. Vicent Servera, L. Martino, J. Verrelst and G. Camps-Valls, "Multifidelity Gaussian Process Emulation for Atmospheric Radiative Transfer Models," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1-10, 2023, Art no. 5519210.
- J. Vicent Servera, L. Martino, J. Verrelst, J. P. Rivera-Caicedo and G. Camps-Valls, "Multi-output Feature Selection for Emulation and Sensitivity Analysis," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1-11, 2024, Art no. 5506411.

4.2.5 Relevant Software Releases / Datasets

Atmospheric RTM data for training emulators. Several datasets of spectral atmospheric transfer functions (i.e. path radiance, transmittances, spherical albedo) simulated with MODTRAN6 atmospheric radiative transfer model are publicly available in the Zenodo repository doi.org/10.5281/zenodo.7826005. The simulations are stored in hdf5 files using the Atmospheric Look-up table Generator (ALG) toolbox (https://doi.org/10.5194/gmd-13-1945-2020). Each dataset has an associated `.xml` file that includes the configuration of MODTRAN6 executions. All datasets include the input atmospheric/geometric variables summarized in Table 3. Each dataset file has a random distribution (based on latin hypercube sampling) of these input variables with varying numbers of points (e.g. `train500.h5` contains 500 samples). The reference dataset contains 10000 samples and was used as a reference for evaluating Gaussian Processes emulators.

Input Variables	Units	Min	Max
O3 column concentration	atm-cm	0.25	0.45
Columnar Water Vapor	g/cm^2	0.2	4
Aerosol Optical Thickness	-	0.04	0.6
Asymmetry parameter	-	0.5	0.85
Angstrom exponent	-	0.1	2
Single Scattering Albedo	-	0.8	1
Surface elevation	km	0	2.5
Solar Zenith Angle	deg	0	70
Relative Zenith Angle	deg	0	180

Table 3. Input variables and range for the MODTRAN6 training/testing datasets.

RTM emulation software tool. To develop the physics-aware emulator strategies described in the previous sections, we have been developing a series of tools to automate the configuration, training, and validation of various emulators and facilitate their intercomparison. The ALG toolbox [83] was extensively used to generate data sets for the training and validation of emulators such as those available in the Zenodo repository (see above). The ALG toolbox was expanded with an *Emulation Tool* that facilitates the configuration and training of various emulators, as well as their validation against reference data. Within the ALG toolbox, the emulator function (`algEmulator`) was implemented. This function includes all the functionalities described in the above paragraphs, including dimensionality reduction, training of GP models, emulation/prediction, feature selection, and global sensitivity analysis. The ALG tool can be freely downloaded from `www.artmotoolbox`

.com. The emulator function is accessible from https://github.com/jorviser/AlgEmulator for standalone use.

4.2.6 Relevant Use Cases

One relevant example in which the proposed atmospheric RTM emulators are being implemented by the authors is in the context of atmospheric correction of satellite data. Atmospheric correction aims at converting the top-of-atmosphere (TOA) radiance signal measured by a satellite instrument into surface reflectance by compensating the effects of scattering and absorption occurring in the Earth's atmosphere. After atmospheric correction, the derived surface reflectance data is used to retrieve geophysical properties for applications such as vegetation monitoring or water quality. In general, accurate atmospheric correction algorithms rely on RTM to derive the atmospheric composition and retrieve the surface reflectance. However, the computational burden and memory requirements to run an RTM make their use impractical in routine data processing chains such as atmospheric correction. In this context, we have implemented RTM emulators as an accurate and fast alternative over traditional LUT interpolation methods.

An example of such use is in the frame of the ESA/NASA excercise called ACIX-III, which aims at intercomparing atmospheric correction algorithms with focus on the hyperspectral satellite missions EnMAP and PRISMA[1]. Both satellite instruments offer similar characteristics: a spectral coverage in the 400-2500 nm range with nearly 240 spectral bands of ∼10 nm resolution. The input top-of-atmosphere radiance product has a size of roughly 30×30 km^2 with a spatial resolution of 30 m, resulting in an image of around 1 million pixels. One of the algorithms participating in ACIX-III, the MAGAC algorithm, makes extensive use of the developed emulators in [145], [152] and related software tools [83]. Without going into out-of-the-scope details, MAGAC retrieves water vapor through a differential absorption technique dubbed APDA [3] and aerosols with a state-of-the-art optimal estimation algorithm inspired on ISOFIT [59]. The retrieval of surface reflectance inverting the radiative transfer equation analytically similar as done in the Sentinel-2 sen2cor package [44].

4.3 Learning Dynamical Systems via Koopman/Transfer Operator

Contributing partner: IIT

Dynamical systems provide a mathematical framework for describing the evolution of state variables over time. In numerous applications, these models often represented by unknown nonlinear differential equations (which may be ordinary or partial, and possibly stochastic) require data-driven techniques to characterize the system and predict future states. Theoretical aspects of dynamical systems are well-documented in the literature. Our initial observation is that, despite the well-established nature of data-driven algorithms for reconstructing dynamical systems, their connection to statistical learning remains largely unexplored. Our overarching goal is to bridge these two important research areas and to establish a solid theoretical foundation for data-driven methods, ensuring statistical guarantees and a comprehensive framework for learning dynamical systems.

In recent years, researchers have stressed the importance of developing physically-informed machine learning models that prioritize interpretability and foster physical insight and intuition, see for example [97] and its references. To learn and interpret nonlinear dynamical systems, the Koopman operator regression framework is highlighted in these works, e.g. [34], [111]. A key element of this approach is the Koopman Mode Decomposition (KMD), which breaks down complex

[1]More details about the PRISMA and EnMAP missions cand be found at www.asi.it/en/earth-science/prisma/ and www.enmap.org/.

dynamical systems into simpler, coherent structures. When the Koopman operator is learned from data using ordinary least squares, the resulting estimated KMD is referred to as Dynamic Mode Decomposition (DMD) [17]. Koopman operator estimators and their modal decomposition have numerous applications, including fluid dynamics, molecular kinetics, and robotics.

4.3.1 Technical Description

First, let's briefly review the basic concepts related to Markov chains and Koopman operators. Let $\mathbf{X} := \{X_t : t \in \mathbb{N}\}$ be a family of random variables with values in a measurable space $(\mathcal{X}, \Sigma_{\mathcal{X}})$, known as the state space. We call $\mathbf{X}$ a *Markov chain* if $\mathbb{P}\{X_{t+1} \in B \,|\, X_{[t]}\} = \mathbb{P}\{X_{t+1} \in B \,|\, X_t\}$. Furthermore, we call $\mathbf{X}$ *time-homogeneous* if there exists $p : \mathcal{X} \times \Sigma_{\mathcal{X}} \to [0, 1]$, referred to as the *transition kernel*, such that for every $(x, B) \in \mathcal{X} \times \Sigma_{\mathcal{X}}$ and every $t \in \mathbb{N}$,

$$\mathbb{P}\{X_{t+1} \in B | X_t = x\} = p(x, B).$$

A large class of Markov chains includes those that possess an *invariant measure* π satisfying $\pi(B) = \int_{\mathcal{X}} \pi(dx)p(x, B)$ for $B \in \Sigma_{\mathcal{X}}$. The Koopman operator returns the expected value of observables of the system in the future, given the present, and estimators of this operator are used to estimate its spectral decomposition, leading to the estimation of KMD. For a *time-homogeneous* Markov chain with an invariant (stationary) distribution π, the (stochastic) *Koopman operator* $A_\pi : L^2_\pi(\mathcal{X}) \to L^2_\pi(\mathcal{X})$ is defined as

$$A_\pi f(x) := \int_{\mathcal{X}} p(x, dy) f(y) = \mathbb{E}\left[f(X_{t+1}) | X_t = x\right], \quad f \in L^2_\pi(\mathcal{X}), x \in \mathcal{X}. \tag{7}$$

In many practical cases, A_π is unknown, but data from one or more system trajectories are available. The main reason for using the (stochastic) *Koopman operator* in dynamical systems is that its linearity can be exploited to compute a spectral decomposition. In many situations, especially for compact Koopman operators, there exist scalars $\mu_i \in \mathbb{C}$, known as Koopman eigenvalues, and observables $\psi_i \in L^2_\pi(\mathcal{X}) \setminus \{0\}$, known as Koopman eigenfunctions, such that $A_\pi f_i = \mu_i f_i$. The dynamical system can then be decomposed into a superposition of simpler signals that can be used for various tasks such as system identification and control. This is particularly elegant when A_π is compact, as for every observable $f \in L^2_\pi(\mathcal{X})$, there exist corresponding scalars $\gamma_i^f \in \mathbb{C}$, known as Koopman modes of f, such that

$$A^t_\pi f(x) = \mathbb{E}[f(X_t) \,|\, X_0 = x] = \sum_{j \in \mathbb{N}} \mu_j^t \gamma_j^f f_j(x), \quad x \in \mathcal{X}, t \in \mathbb{N}. \tag{8}$$

This formula is known as *Koopman Mode Decomposition* (KMD). It decomposes the expected dynamics observed by f into *stationary* modes γ_j^f that combine with *temporal changes* governed by eigenvalues μ_j and *spatial changes* governed by the eigenfunctions f_j[34].

KMD is closely related to the general theory of spectral decomposition for bounded linear operators, specifically the Riesz decomposition theorem. The KMD of a compact self-adjoint Koopman operator can be stated as

$$A^t_\pi f(x) = \mathbb{E}[f(X_t) \,|\, X_0 = x] = \sum_{j \in \mathbb{N}} \mu_j^t \langle f_j, f \rangle f_j(x), \quad f \in L^2_\pi(\mathcal{X}), x \in \mathcal{X}, t \in \mathbb{N}. \tag{9}$$

An operator regression learning framework was proposed to estimate the Koopman operator on $L^2_\pi(\mathcal{X})$ in a reproducing kernel Hilbert space $\mathcal{H}$ with an associated feature map $\phi : \mathcal{X} \to \mathcal{H}$. Given

a dataset of consecutive states $\mathcal{D}_n := (x_i, y_i)_{i=1}^n$, estimators minimize the mean square error (MSE) risk

$$\widehat{\mathcal{R}}(G) := \frac{1}{n}\sum_{i\in[n]}\|\phi(y_i) - G^*\phi(x_i)\|^2, \tag{10}$$

where $G \in \mathrm{HS}(\mathcal{H})$, the space of Hilbert-Schmidt operators acting on $\mathcal{H}$. Minimizing the Tikhonov regularized risk leads to the kernel Ridge regression (KRR) estimator $\widehat{G}_\gamma = \widehat{C}_\gamma^{-1}\widehat{T}$, expressed via the *input* and *cross* empirical covariances

$$\widehat{C} = \frac{1}{n}\sum_{i\in[n]}\phi(x_i)\otimes\phi(x_i), \quad \text{and} \quad \widehat{T} = \frac{1}{n}\sum_{i\in[n]}\phi(x_i)\otimes\phi(y_i),$$

where $\widehat{C}_\gamma := \widehat{C} + \gamma I_{\mathcal{H}}$.

In practice, dynamical systems are only observed, and neither A nor its domain $\mathcal{F} = L^2_\pi(\mathcal{X})$ are known, presenting a significant challenge for learning them from data. The most common algorithms aim to learn the action of $A : \mathcal{F} \to \mathcal{F}$ on a predefined Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, which forms a subset of functions in $\mathcal{F}$ [13]. This allows, via the kernel trick, to frame the problem of learning the restriction of A to $\mathcal{H}$, $A_{|\mathcal{H}} : \mathcal{H} \to \mathcal{F}$, through empirical risk minimization. However, recent theoretical advances have shown that such algorithms are statistically consistent only to $P_{\mathcal{H}}A_{|\mathcal{H}}$, where $P_{\mathcal{H}}$ is the orthogonal projection onto the closure of $\mathcal{H}$ in $\mathcal{F}$. The projection $P_{\mathcal{H}}$ confines the evolved observables back inside $\mathcal{H}$, thus, generally altering the dynamics of the system. Therefore, to accurately learn the dynamics, two main requirements on $\mathcal{H}$ are necessary: i) $A_{|\mathcal{H}}$ must approximate A well, meaning $\mathcal{H}$ must be sufficiently large relative to the domain of A; ii) the difference between the projected restriction and the true one, i.e., the approximation error $\left\|\left[I - P_{\mathcal{H}}\right] A_{|\mathcal{H}}\right\|$, needs to be minimal.

When $\mathcal{H}$ is an infinite-dimensional universal RKHS, both requirements are met, i.e., $\mathcal{H}$ is dense in $\mathcal{F}$ and the approximation error is zero, leading to an arbitrarily good approximation of dynamics with sufficient data. Nonetheless, another critical issue arises as the norms on the a-priori chosen $\mathcal{H}$ and the unknown $\mathcal{F}$ do not coincide, since the latter depends on the process itself. This metric distortion phenomenon has been recently identified as the cause of spurious estimation of the spectra of A, limiting the utility of the learned transfer operators. Even if A is self-adjoint, meaning the eigenfunctions are orthogonal in $\mathcal{F}$, the estimated ones will not be orthogonal in $\mathcal{H}$, resulting in spectral pollution. This motivates the additional requirement that iii) $\mathcal{H}$ is a subspace of $\mathcal{F}$, i.e., both spaces share the same norm.

In summary, the ideal $\mathcal{H}$ is the leading invariant subspace of A, corresponding to the largest (in magnitude) eigenvalues of A. This subspace $\mathcal{H}$ achieves zero approximation error, eliminates metric distortion, and best approximates (in the dynamical system sense) the operator A. Since any RKHS $\mathcal{H}$ is entirely described by a feature map, learning a leading invariant subspace $\mathcal{H}$ from data is fundamentally a representation learning problem.

We begin by formalizing the problem of learning a good finite-dimensional representation space for A, and then address the same for the generator $\mathcal{L}$. Our approach is inspired by the following upper and lower bounds on the approximation error, a direct consequence of the norm change from $\mathcal{H}$ to $\mathcal{F}$,

$$\left\|\left[I - P_{\mathcal{H}}\right] AP_{\mathcal{H}}\right\|^2 \lambda^+_{\min}\left(C_{\mathcal{H}}\right) \leq \left\|\left[I - P_{\mathcal{H}}\right] A_{|\mathcal{H}}\right\|^2 \leq \left\|\left[I - P_{\mathcal{H}}\right] AP_{\mathcal{H}}\right\|^2 \lambda_{\max}\left(C_{\mathcal{H}}\right),$$

where $C_{\mathcal{H}}$ is the covariance operator on $\mathcal{H}$ w.r.t. the measure π, while $\lambda^+_{\min}$ and $\lambda_{\max}$ are the smallest and largest non-null eigenvalues, respectively. Note that the norms on the hypothetical domain $\mathcal{H}$ and true domain $L^2_\pi(\mathcal{X})$ coincide if and only if $C_{\mathcal{H}} = I$, in which case equalities hold in the mentioned equation and the approximation error becomes $\left\|\left[I - P_{\mathcal{H}}\right] AP_{\mathcal{H}}\right\|$.

When the operator A is known, the latter quantity can be directly minimized by standard numerical algorithms for spectral computation to find invariant subspaces. Unfortunately, in our

stochastic setting, A is unknown since we cannot compute the conditional expectation. To overcome this issue, we propose a learning approach to recover the invariant space $\mathcal{H}$, which is rooted in the singular value decomposition, holding under the mild assumption that A is a compact operator. The main idea is that the subspace made of the leading r left singular functions of A serves as a good approximation of the desired leading invariant subspace of A. Namely, due to the orthonormality of the singular functions, we have that $C_{\mathcal{H}} = I$ and $P_{\mathcal{H}}A$ becomes the r-truncated SVD of A, that is, its best rank-r approximation. Therefore, according to the previous equation, the approximation error is at most $\sigma_{r+1}(A)$, which can be made arbitrarily small by increasing r. Moreover, the distance of the subspace of left singular functions to the desired leading invariant subspace is determined by the "normality" of A. If the operator A is normal, that is $AA^* = A^*A$, then both its left and right singular spaces are invariant subspaces of A, resulting in zero approximation error regardless of r. This leads us to the following optimization problem

$$\max_{\mathcal{H},\mathcal{H}' \subset L^2_\pi(\mathcal{X})} \left\{ \|P_{\mathcal{H}} A P_{\mathcal{H}'}\|^2_{\mathrm{HS}} \mid C_{\mathcal{H}} = C_{\mathcal{H}'} = I, \dim(\mathcal{H}) \leq r, \dim(\mathcal{H}') \leq r \right\}.$$

Using the application of Eckart-Young-Mirsky's Theorem, we can show that the desired representation space $\mathcal{H}$ can be computed by solving the above problem. Note that, in general, the auxiliary space $\mathcal{H}'$ is needed to capture the right singular functions, while if we have prior knowledge that A is normal, without loss of generality, one can set $\mathcal{H} = \mathcal{H}'$ in the above equation.

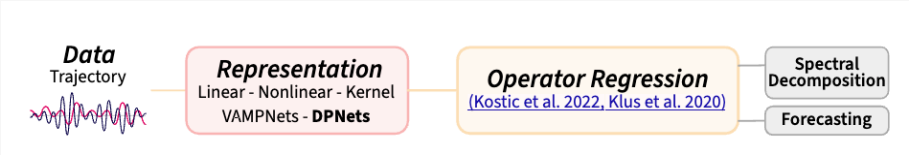

Figure 6. Pipeline for learning dynamical systems. DPNets learn a data representation to be used with standard operator regression methods. In turn, these are employed to solve downstream tasks such as forecasting and interpreting dynamical systems via spectral decomposition.

In Summary, in this project, we synergistically combine both kernel and DNN paradigms: initially, DNNs are utilized to learn an invariant representation that fully captures the system dynamics. This representation is then forwarded to kernel-based algorithms for the actual transfer operator regression task. This general framework is illustrated in the above figure. Our method, termed Deep Projection Networks (DPNets), addresses the challenge of providing good representations to the operator regression algorithms. It can be formulated as an optimization problem over neural networks and can benefit from a differentiable and numerically well-conditioned score functional, enhancing the stability of the training process.

4.3.2 Relevant Publications

- Kostic, V. R., Novelli, P., Grazzi, R., Lounici, K., Pontil, M. (2024, May). Learning invariant representations of time-homogeneous stochastic dynamical systems. In ICLR 2024.

4.3.3 Relevant Software Releases / Datasets

We developed a Python library for learning the Koopman/transfer operator. We aimed to make it user-friendly and convenient by including documentation and several examples across different fields. You can find the documentation webpage for the library at https://kooplearn.readthedocs.io/latest/, and the GitHub repository at https://github.com/Machine-Learn

ing-Dynamical-Systems/kooplearn. Additionally, for the specific paper "Learning invariant representations of time-homogeneous stochastic dynamical systems - ICLR 2024," you can use the GitHub repository at https://github.com/pietronvll/DPNets.

4.3.4 Relevant Use Cases

The tool outlined in previous sections has a wide range of potential applications, including energy forecasting, epidemiology, finance, atomistic simulations, fluid dynamics, weather and climate forecasting, neuroscience, and many other fields.

4.4 Towards Sustainable Medical AI Technologies: Anatomically Aware Dual-hop Learning for Pulmonary Embolism Detection

Artificial Intelligence has the capacity to significantly improve the ability of doctors to detect and recognize various medical conditions and diseases, through the development of intelligent systems and instruments in a vast array of medical domains. Being able to learn efficiently from large quantities of medical data can significantly increase both the accuracy and speed of diagnosis at a highly reduced human and financial cost. This affects, either directly or indirectly, our capability to develop a sustainable society, in which medical AI can better protect and improve our health, while reducing energy consumption and human effort.

In our recent work, we developed such an efficient learning system for addressing one of the major health concerns of our society, which is the early detection and subsequent efficient treatment of pulmonary embolism. Pulmonary embolisms (PEs), manifesting as a blood clot (thrombus) in the pulmonary arteries, represent a major health concern, having a high rate of incidence and mortality, representing globally the third most frequent cardiovascular syndrome, trailing only myocardial infarction and stroke [23]. Pulmonary embolisms affect between 39-115 per 100 000 individuals, while the closely related deep vein thrombosis affects 53-166 per 100 000 individuals [79], causing up to 300 000 deaths per year in the US alone [38]. This situation is likely to be exacerbated by the correlation with previous Covid19 infections [116], and the rising tendency of PE incidence observed in longitudinal studies [79].

Of the reported deaths, 34% happen suddenly, or within a few hours after the acute event, i.e., before a treatment can take affect or even be initiated [11]. Hence, PE diagnosis is a time critical procedure. Thus, given the gravity and urgency of PEs, together with the rising workload of hospitals [20], an approach for the triage and prioritization of patients, which is both fast and accurate, is deemed necessary.

The gold standard for diagnosing pulmonary embolisms is the CT pulmonary angiogram [21], a medical imaging modality which sets the task of pulmonary embolism detection in the realm of modern computer vision with deep neural networks. Deep neural networks in general, and convolutional neural networks (CNNs) in particular, are well known for their pattern recognition and detection capabilities in the vision domain. Such models have been shown to work well with medical imaging, achieving great results on CTs, for tasks such as Chronic obstructive pulmonary disease [96], Covid-19 detection [81] or intracranial hemorrhage [47]. Accurate results are also reported on other imaging modalities, such as radiography [68] and magnetic resonance imaging (MRI) [60]. However, despite the strong recent success of deep learning and computer vision in various medical image analysis tasks, for Pulmonary Embolism detection there are few works published recently [115].

Given the need for reducing the workload in hospitals, and the strong previous results obtained by CNNs in the space of disease detection and classification using medical imaging, in this study we design an image processing system, which starts with the detection of specific anatomical

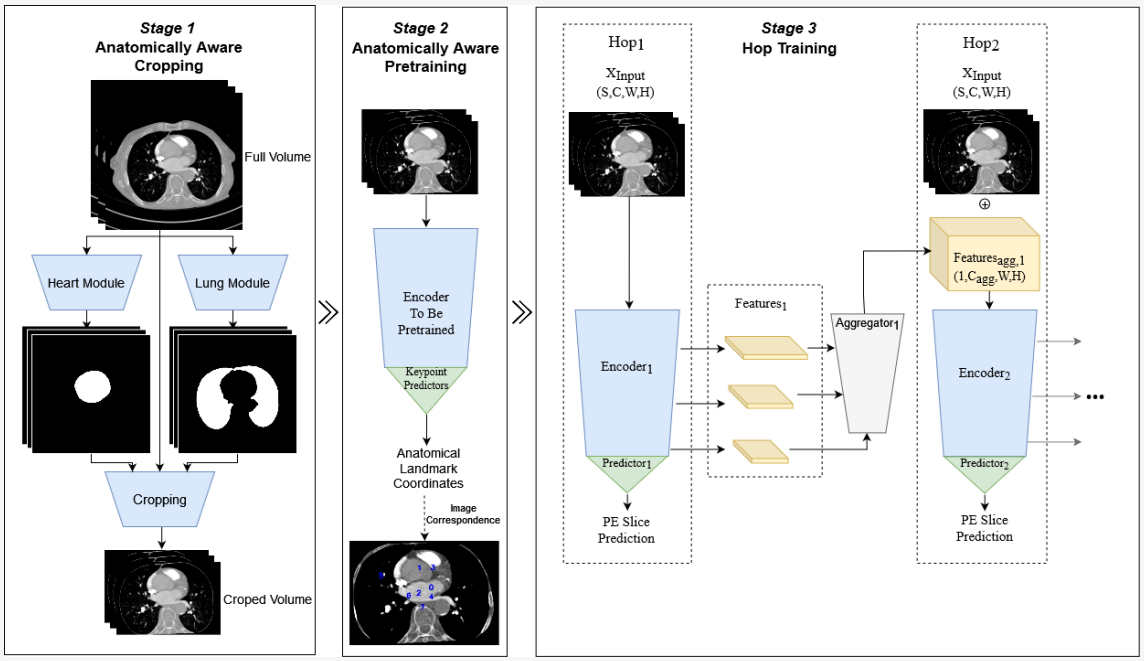

Figure 7. Proposed workflow: each stage represents one of the contributions. ***Left****: Anatomically Aware Segmentation and Cropping, through which data is specialised for PE detection.* ***Middle****: Anatomically Aware Pretraining on the related task of Anatomical Landmark Detection, through which the model is primed for our task of Pulmonary Embolism detection.* ***Right*** *Hopped training, through which model predictions are refined, over two hops of neural processing.*

structures and continues with a two-phase (initial followed by refined) detection of pulmonary embolisms. Each component in the pipeline is vital for the observed performance, as demonstrated below in the thorough theoretical and experimental analysis and validation.

Main Contributions. We introduce an efficient and highly-accurate deep neural architecture for Pulmonary Embolism detection, with state-of the art performance, which comprises three different phases, along three independent axes, which prove to be necessary for an accurate performance. They constitute the main contributions of your approach:

1. **First phase:** anatomically aware masking and cropping of lung and heart regions. Deep neural modules trained on physiological information for segmenting lung and heart regions are used to segment only the relevant information with respect to PE detection.

2. **Second phase:** anatomically aware pretraining. Relevant features are pretrained on the task of localizing specific anatomical keypoints, before starting the PE learning phase.

3. **Third phase:** dual-hop architecture for PE detection. The dual-hop architecture performs classification in two-stages. The first stage performs an initial evaluation, and the second stage, having access to the initial input as well as the output of the first stage, is able to produce a more accurate, refined prediction.

From an experimental point of view, we show that each component brings an important boost in performance, while the overall system achieves state of the art results compared to the recently published methods.

4.4.1 Discussion and conclusions

The three phases we porpose, in essence, follow the intuitive normal steps in which a doctor performs the diagnosis based on medical images: (1) focus of attention on the region of interest, (2) use of rich previously learned knowledge of anatomy, and (3) a rigorous pathological examination during several cycles of inspection at different levels of detail.

We perform extensive experiments on highly relevant datasets **RsnaPEDataset**, which demonstrate the effectiveness of each of these phases, with significant quantitative improvements over strong baselines and recent state of the art, in the big data regime. Besides the demonstrated results on a specific and highly important medical problem, the three mechanisms introduced in this paper also constitute a more general proof of concept, which could open the door for similar approaches in other medical analysis tasks. Such highly effective methods could push the techology towards medical systems, which could significantly contribute to a sustainable and healthy society, by improving efficiency and accuracy of medical diagnosis, while reducing energy consumption and human effort.

4.4.2 Relevant publications

Condrea, Florin, Saikiran Rapaka, Lucian Itu, Puneet Sharma, Jonathan Sperl, A. Mohamed Ali, and Marius Leordeanu. "Anatomically aware dual-hop learning for pulmonary embolism detection in CT pulmonary angiograms." Computers in Biology and Medicine 174 (2024): 108464.

4.5 Probabilistic grammars for modeling dynamical systems from coarse, noisy, and partial data

Contributing partner: JSI

4.5.1 Introduction

Ordinary differential equations (ODEs) are a widely used formalism for the mathematical modeling of dynamical systems, a task omnipresent in scientific domains. We introduce a novel method for inferring ODEs from data, which extends ProGED, a method for equation discovery that allows users to formalize domain-specific knowledge as probabilistic context-free grammars and use it for constraining the space of candidate equations. The proposed method thus integrates data and knowledge-driven approaches to automated modelling of dynamical systems.

The extended method can discover ODEs also from partial observations of dynamical systems, where only a subset of state variables can be observed. To evaluate the performance of the newly proposed method, we perform a systematic empirical comparison with alternative state-of-the-art methods for equation discovery and system identification from complete and partial observations. The comparison uses Dynobench, a set of ten dynamical systems that extends the standard Strogatz benchmark.

We compare the ability of the proposed method and several competitior methods to reconstruct the known ODEs from synthetic data simulated at different temporal resolutions. We also consider data with different levels of noise, i.e., signal-to-noise ratios. The improved ProGED compares favourably to state-of-the-art methods for inferring ODEs from data regarding reconstruction abilities and robustness to data coarseness, noise, and completeness.

4.5.2 The proposed methodology for automated modelling

ProGED discovers equations by following the generate-and-test paradigm. In the generate phase, ProGED addresses the task of structure identification, in which candidate equations (structures) are constructed. The test phase performs parameter estimation, in which the values of unknown numeric parameters in the equations are fit to data. Among a large number of tested equations, ProGED chooses the ones with the lowest error-of-fit. ProGED composes candidate equations from algebraic expressions, sampled from a probabilistic context-free grammar (PCFG).

A context-free grammar (CFG) is defined by the tuple $(\mathscr{T}, \mathscr{N}, \mathscr{R}, S)$. When defining arithmetic expressions, the set of terminal symbols $\mathscr{T}$ consists of symbols representing variables (e.g., x, y), operators or functions (e.g., $+, \cdot, \sin$), and constant parameters (c). The nonterminal symbols in $\mathscr{N}$ do not appear in expressions, but represent higher-level concepts in the language of mathematics, such as polynomials, monomials or terms. The set $\mathscr{R}$ contains production (rewrite) rules $A \to \alpha_1 \dots \alpha_k$, where $A \in \mathscr{N}$ and $\alpha_i \in \mathscr{N} \cup \mathscr{T}$. A production rule specifies how to replace a particular nonterminal symbol with a string of nonterminal and terminal symbols. In a probabilistic context-free grammar, each production rule is assigned a probability, so that the probabilities of all production rules with the same nonterminal symbol on the left-hand side (LHS) sum up to 1. An example grammars that was used in our experiments are shown in Table 4.

The generation of a random expression with a PCFG begins from a string (starting symbol) S and proceeds by successively applying production rules to the string until only terminal symbols remain. Whenever more than one rule applies, we randomly choose a rule according to the probabilities. The final result of one instance of the sampling process is an arithmetic expression, which we transform to its canonical form by using the symbolic mathematics engine SymPy.

Besides acting as a generator of expressions, a grammar is a powerful way of encoding background knowledge. Note that a PCFG defines a probability distribution over the space of candidate expressions, which allows the user to impose an inductive bias by manipulating the production probabilities. For example, we can manipulate the complexity of generated equations through the probabilities of recursive productions, or express a bias towards trigonometric functions by raising their respective probabilities. In the absence of background knowledge, we can use a universal grammar for generating an arbitrary expression, composed of the four basic operations $(+, -, *, /)$, as well as arbitrary functions.

After generation, a candidate equation contains generic constants (denoted by c), the values of which must be fitted to data. Since the equations are, in general, non-linear in their parameters, a universal optimization algorithm is used to minimize the error-of-fit to the data, which can be computationally demanding, but is more flexible than approaches based on linear regression. ProGED uses the differential evolution (DE) algorithm for numerical optimization to fit the values of the constants in an expression to the provided data.

Numeric differentiation and algebraic equation discovery. The task of discovering ODEs can be transformed into a task of discovering algebraic equations by numerically calculating the derivatives of the state variables. These time derivatives are then considered as dependent variables and placed in the LHS of algebraic equations to be discovered. In that case, we estimate the parameter values by minimizing the difference L between the observed (calculated) and predicted values of the time derivatives of the state variables,

$$L = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\dot{u}(t_i) - \hat{\dot{u}}(t_i))^2}, \tag{11}$$

where $\dot{u}(t_i)$ represents the value of the time derivative of variable u at time t_i, numerically calculated from observed data, $\hat{\dot{u}}$ represents the corresponding predicted value, obtained by evaluating the

candidate equation, and n is the number of observed time points.

This simple transformation is commonly used by ODE discovery approaches. However, its use is problematic if we deal with coarsely sampled measured values of the system variables and high levels of noise. Also, this approach is only possible when the measurements of all state variables are readily available, i.e. the system at hand is fully observable.

Discovery of differential equations with direct simulation. To address the limitations of numerical differentiation, we introduce an approach to ODE discovery based on simulating differential equations. During each step of parameter estimation, we must compute the error of the candidate equation with a given set of parameter values. To obtain this, we solve the initial-value problem by performing a full simulation of the system of ODEs, using the LSODA algorithm implemented in the function `odeint` from the SciPy library. We define the error as the root-mean-squared error of the simulated trajectory, with respect to the true trajectory of the observed variables. In other words, we minimize the error

$$L = \sqrt{\frac{1}{|\mathcal{U}_{\text{obs}}| \cdot n} \sum_{u \in \mathcal{U}_{\text{obs}}} \sum_{i=1}^{n} (u(t_i) - \hat{u}(t_i))^2}, \tag{12}$$

where $\mathcal{U}_{\text{obs}}$ is the set of all *observed* variables, $u(t_i)$ represents the observed value u at time t_i, $\hat{u}(t_i)$ represents the corresponding simulated value (i.e., the value obtained by simulating the candidate equation), and n is the number of observed time points.

Background knowledge. ProGED uses Monte-Carlo sampling to search the space of possible systems of ODEs. Since this search is not guided, the method works best when the search space is as constrained as possible. Generally, the search space is constrained based on various types of background knowledge, from general modeling principles, such as the parsimony principle, to domain-specific or even problem-specific knowledge. ProGED employs PCFGs as a robust and powerful framework for expressing different types of background knowledge and imposing both hard constraints (through the PCFG structure) and soft constraints (through production probabilities).

In this study, we aim to discover models of dynamical systems, which already provides some background knowledge in itself. To demonstrate how background knowledge can be expressed with PCFGs, we detail some of the background knowledge for modeling dynamical systems and design a grammar that expresses the general knowledge as follows:

1. A dynamical system is described by a system of 1st-order ODEs,
2. the right-hand side of each ODE is a linear combination of terms,
3. since each term has its own multiplicative numerical constant, there is no need to include the subtraction operation,
4. terms are most commonly low-order monomials of state variables,
5. rarely, trigonometric functions appear, in particular sine and cosine,
6. the arguments of trigonometric functions tend to be linear functions of state variables and/or time,
7. rarely, terms can be rational functions,
8. the terms in the numerator and denominator of rational functions tend to be low-order polynomials,

9. numerical constants can appear in trigonometric and rational functions, making the expressions nonlinear in parameters.

We present a PCFG that expresses this background knowledge for modeling dynamical systems in Table 4. The grammar has three production rules for the starting symbol E (*expression*), which generate the sum of a number of terms, which are either ordinary terms derived by T, or rational functions in the form of T/D. The two productions for D (*denominator*) derive the denominator of a rational function as a sum of terms. The productions for T (*term*) generate a monomial, composed of variables and/or trigonometric functions. The productions for R and M derive the trigonometric factors as the sine or cosine of a product of state variables. Finally, the production rules for V generate state variables. The presented grammar generates expressions, following the domain knowledge above. A system of ODEs is composed by independently generating an expression for the right-hand side of each ODE in the system.

Table 4. An example of a generic grammar for mathematical expressions appearing in models of dynamical systems, used to generate the right-hand sides of ODEs. Production rules with the same nonterminal on the left-hand side are separated by a vertical line (|). The probability of each production rule is given in square brackets. The grammar can generate expressions (E) involving linear combinations of multiplicative terms (T), composed by multiplying variables (V) or trigonometric terms (R). Trigonometric terms include sines and cosines of monomials (M) of the state variables (V). Moreover, the grammar can generate expressions involving the division of linear combinations of multiplicative terms (T) with terms in the denominator (D), where D can be a linear combination of terms T. The example grammar refers to a system with two state variables, x and y, but can be easily extended to an arbitrary number of state variables by adding new production rules to the nonterminal V.

$$
\begin{aligned}
\mathscr{N} &= \{E, D, T, R, M, V\} \\
\mathscr{T} &= \{+, *, /, \sin, \cos, (,), c, x, y\} \\
\mathscr{R} &= \\
E &\to E + T\ [0.6] \mid T/(D)\ [0.15] \mid T\ [0.25] \\
D &\to D + T\ [0.5] \mid T\ [0.5] \\
T &\to T * V\ [0.3] \mid T * R\ [0.1] \mid c\ [0.6] \\
R &\to \sin(M)\ [0.5] \mid \cos(M)\ [0.5] \\
M &\to M * V\ [0.5] \mid c\ [0.5] \\
V &\to x\ [0.5] \mid y\ [0.5] \\
S &= E
\end{aligned}
$$

The choice of probabilities in the PCFG warrants discussion as well, since it allows us to express soft constraints on the space of equations and decide the level of parsimony of generated expressions. In the grammar in Figure 4, we set the probability of recursion in the rules for E relatively high (0.6), in order to generate expressions with several terms. On the other hand, the probability of recursion in the production rules for T is lower ($0.3 + 0.1 = 0.4$), as we prefer lower-order terms. We set the probabilities of rational functions (0.15) and trigonometric functions (0.1) low to reflect their relative rarity in models of general dynamical systems. We set the probabilities in the presented grammar based on past experience with modeling dynamical systems, as well as by generating samples of random expressions and observing their properties.

The presented grammar is designed to constrain the space of expressions as much as possible, while still describing most dynamical systems. Nevertheless, some dynamical systems may be described by systems of ODEs that fall outside the space described by this grammar, which would preclude their discovery with ProGED. This risk is common when using hard constraints in equation discovery. On the other hand, the presented grammar still generates many types of expressions, including the entire class of rational functions.More limited grammars can be designed to describe more specific types of dynamical systems by altering the grammar rules and their probabilities.

Table 5. Description of ten dynamical systems, including their model (system of ODEs), the parameters we have chosen for each system, and the number of terms present in each ODE.

System name	System model	Chosen parameters	# Terms
bacterial respiration (*bacres*)	$\dot{x} = B - x - \frac{xy}{qx^2+1}$ $\dot{y} = A - \frac{xy}{qx^2+1}$	$A = 10, B = 20,$ $q = 0.5$	(3, 2)
bar magnets (*barmag*)	$\dot{x} = K\sin(x-y) - \sin(x)$ $\dot{y} = K\sin(y-x) - \sin(y)$	$K = 0.5$	(2, 2)
coupled phase oscillator (*cphase*)	$\dot{x} = W(t) + A\sin(x) + B\sin(y)$ $\dot{y} = C\sin(x) + D\sin(y) + E$	$W(t) = 2-$ $-0.5\sin(2\pi \cdot 0.0015t),$ $A = B = 0.8, C = 0,$ $D = 0.6, E = 4.53$	(4, 2)
glider	$\dot{x} = -\sin(y) - Dx^2$ $\dot{y} = -\frac{\cos(y)}{x} + x$	$D = 0.05$	(2, 2)
Lorenz oscillator (*lorenz*)	$\dot{x} = \sigma(y-x)$ $\dot{y} = x(\rho - z) - y$ $\dot{z} = xy - \beta z$	$\sigma = 10,\ \rho = 28,$ $\beta = \frac{8}{3}$	(2, 3, 2)
Lotka-Volterra (*lv*)	$\dot{x} = x(A - x - By)$ $\dot{y} = y(C - x - y)$	$A = 3, B = 2,$ $C = 2$	(3, 3)
Van der Pol (*vdp*)	$\dot{x} = y$ $\dot{y} = -x - \mu(x^2-1)y$	$\mu = 2$	(1, 3)
predator-prey (*predprey*)	$\dot{x} = x(b - x - \frac{y}{1+x})$ $\dot{y} = y(\frac{x}{1+x} - ay)$	$b = 4, a = 0.075$	(3, 2)
shear flow (*shearflow*)	$\dot{x} = \cot(y)\cos(x)$ $\dot{y} = (\cos^2(y) + A\sin^2(y))\sin(x)$	$A = 0.1$	(1, 2)
Stuart-Landau (*stl*)	$\dot{x} = ax - \omega y - x(x^2+y^2)$ $\dot{y} = \omega x + ay - y(x^2+y^2)$	$a = 1, \omega = 3$	(4, 4)

4.5.3 Experimental evaluation

We conduct a detailed experimental evaluation of the developed approach on ten dynamical systems, with small and large datasets. We consider three levels of noise (no noise, low noise and high noise). The experimental evaluation is performed separately under full and partial observability.

We compare ProGED to four major competitors, DSO and SINDY for full observability, GPoM and L-ODEfind for partial observability. We use three different performance measures to compare the different ODE discovery methods.

Data. The benchmark we used for evaluation contains trajectories of ten dynamical systems, which describe the temporal evolution of the state variables of each system. Models of the ten dynamical systems are shown in Table 5. Seven out of the ten systems come from the Strogatz benchmark, while three are introduced by ourselves. The data we have created for the purpose of the evaluation are described in the section on SW and data release (Dynobench benchmark).

Dynobench includes coarse data of only 100 time points sampled at 0.1 Hz (called "small data"). Additionally, we created densely sampled data with 2000 time points, sampled at 0.01 Hz (called "large data"). Each system of ODEs was simulated four times, each time with a different set of initial values of the state variables. We also evaluated the methods on clean (noise-free) data, noisy data with a 30 dB signal-to-noise ratio (SNR), where the signal was a thousand times stronger than the noise, and data with a 13 dB SNR (signal twenty times stronger than noise).

Background knowledge. The use of background knowledge is an important aspect of equation discovery, but can be difficult to take into account in comparative benchmark experiments. To facilitate its use, we equip the problems in Dynobench with explicit background knowledge. We classify each of the ten dynamical systems into one of three classes that describe its type and define background knowledge for each of the three classes (see the Appendix of the paper):

1. **State oscillators and population models** include systems *lv, predprey, vdp, stl, lorenz* (see Table 5). The right-hand sides of the ODEs for these systems are polynomials, most commonly composed of two or three terms. The terms are monomials of the state variables, typically of order 1-3. An exception that often appears in population models is the Monod term (fraction of the form $\frac{V}{V+c}$. Interaction terms (terms involving two or more state variables) often appear in the same form in more than one ODE in the systems of ODEs. systems, To find ODEs for this group of systems we created a grammar for state oscillators.

2. **Phase oscillators** include the systems *cphase* and *barmag*. Phase oscillators are oscillatory systems, which we observe through the phases of their state variables, instead of the state variables themselves. Consequently, the right-hand sides of the ODEs are linear combinations of trigonometric functions. Sine and cosine are the most common, but other trigonometric functions may appear as well. The functions may be phase-shifted, which is realized by an additive constant in the function's argument. The arguments of the trigonometric functions tend to be linear functions of the phase variables.

3. Finally, **general dynamical systems** include *bacres, glider* and *shearflow*. Systems that do not fall into any of the previous two classes tend to follow the general background knowledge for modeling dynamical systems and the grammar shown in Figure 4.

Full and partial observability. To assess the system identification performance of ProGED on *fully observable systems*, we compared it to the DSO and SINDy methods on the Dynobench benchmark. When evaluating each method, we have done two experiments. First, we tried to use as much of the available background knowledge as possible and second, we left the model search space unconstrained. The first experiment was named *constrained* and second *unconstrained*. The configuration of each method is described in Section 4.3 of the paper.

In the *partial observability* scenario, we compared ProGED with two other methods, L-ODEfind and GPoM, which are capable of handling partially-observed dynamical systems, unlike DSO and SINDy. We chose to test on only the *vdp* model, which was best reconstructed under full observability, as the identification under partial observability is a challenging task. Moreover, the *vdp* model

includes only terms with linearity in parameters and can thus be reconstructed by L-ODEfind and GPoM. To create a scenario with partial observability, we simply removed the time-series data for one of the two state variables, depending on which one was observed, resulting in incomplete knowledge of the system's dynamics. We tested the performance of each method by varying the level of data coarseness and noise. The particular settings of each method are described in Section 4.4 of the paper.

Performance measures. We compared the results of the different methods with three metrics that quantify either the accuracy of the reconstruction or the complexity of the resulting expression. The primary metric that was used for model selection, was the *trajectory error*, calculated on train, validation and test sets. We calculate it, for a given state variable u, by using the relative-root-mean-square-error

$$TE_u = \sqrt{\frac{\sum_{i=1}^{n}(u(t_i) - \hat{u}(t_i))^2}{\sum_{i=1}^{n}(u(t_i) - \overline{u})^2}}, \tag{13}$$

where $u(t)$ and $\hat{u}(t)$ denote the simulated values of the state variable u at time point t, computed using the true model and the reconstructed model, respectively. The $\overline{u}$ is the mean value of $u(t)$ in the data obtained by simulating the true model. We define the total trajectory error as the sum over all the state variables u in the system of equations, $TE = \sum_u TE_u$. Lower trajectory error means better reconstruction of systems' dynamics.

The second metric we used was the *normalized term difference* (TD), defined as the sum of the number of missing terms and the number of wrong terms in the reconstructed system, divided by the number of true terms:

$$TD_u = \frac{N_{u,\text{missing}} + N_{u,\text{wrong}}}{N_{u,\text{true}}} \tag{14}$$

Here, $N_{u,\text{missing}}$ is the number of terms that are missing from the reconstructed differential equation for the state variable u in the system of equations. The number of wrong terms $N_{u,\text{wrong}}$ is the number of terms that are not present in the true differential equation and the $N_{u,\text{true}}$ is the number of true terms. We define the *term* equivalently to a summand, which is an individual part of the expression, separated by addition or subtraction. Again, the total TD was calculated as the sum over all the state variables u in the system of equations, $TD = \sum_u TD_u$.

The third measure of the reconstruction success was the *normalized complexity* (NC), which is calculated as the number of nodes in the expression tree of a reconstructed equation divided by the number of nodes in the expression tree of a true system's equation:

$$NC_u = \frac{N_{u,\text{nodes in reconstr}}}{N_{u,\text{nodes in true}}}. \tag{15}$$

The NC of a system of equations was considered as the sum of the NC_u of the individual equations of state variables u in the model. NC is best at 1, where the complexity of the true and reconstructed model are equal. We included NC to complement TD. The NC metric is commonly used in research due to its straightforward calculation. Nevertheless, TD provides a more informative measure of accurate reconstruction. Even if NC is equal to 1, TD can still take nonzero values, implying the existence of erroneous terms in the reconstruction models. However, when TD is equal to 0, NC should also be 1.

The performance of the methods was statistically evaluated in terms of their trajectory error for each system by using a critical difference diagram. The critical difference measure includes the

Friedman test with corresponding post-hoc Wilcoxon tests for pair-wise comparison between the methods. Statistical results were corrected for multiple comparisons using Holm's method. We performed the statistical evaluation separately on six dependent configurations, consisting of pairs of the three SNR values and the two data types (small and large data). The statistical tests were applied in both, the constrained and unconstrained model space under full observability.

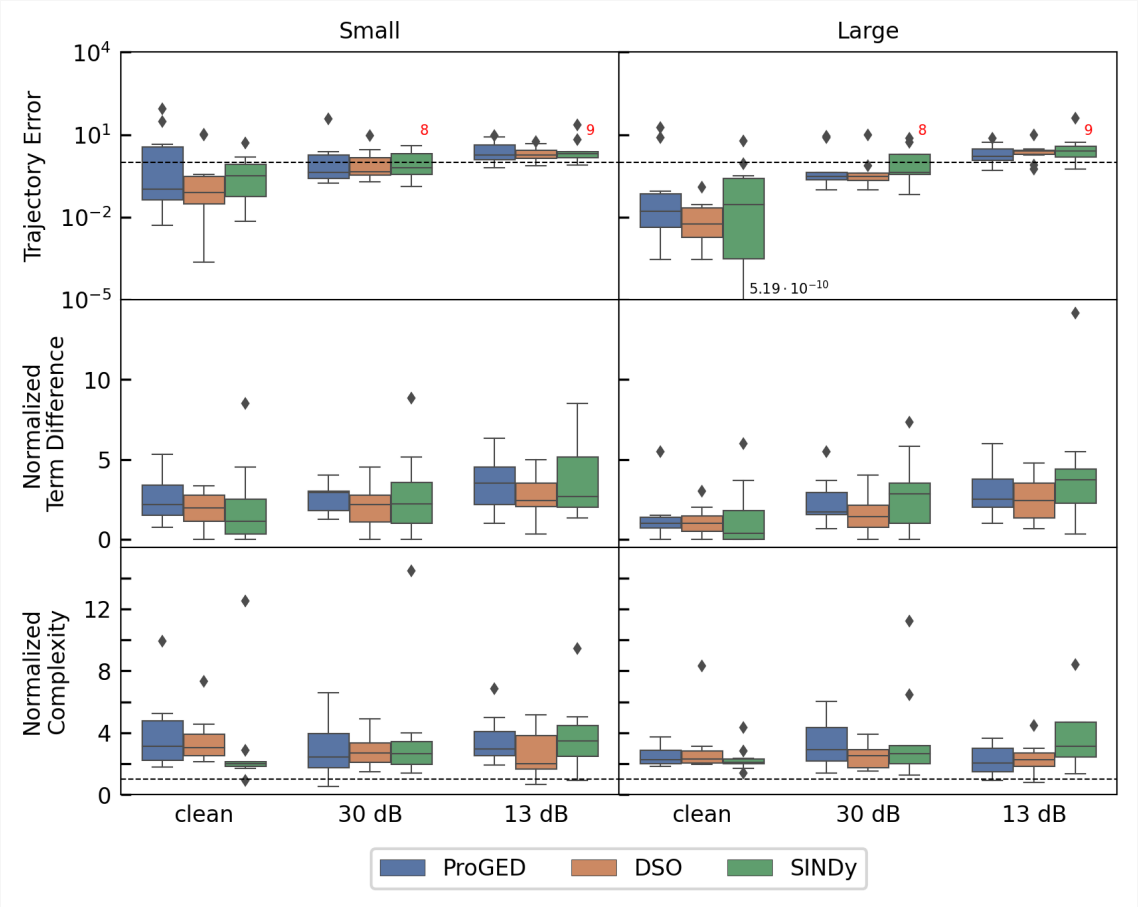

Figure 8. Comparison of the trajectory error (top row), normalized term difference (middle row) and normalized complexity (bottom row) for the ProGED, DSO and SINDy methods, colour-coded as shown in the legend. The left-hand side plots correspond to results on the small data and the right-hand side plots to results on the large data. Each subplot compares performance on data with different noise/SNR levels: clean data, 30 dB and 13 dB. Each boxplot represents the distribution of performance over the ten dynamical systems. The black diamonds represent outliers. The red numbers denote the number of successful identifications when some out of the 10 system identifications failed (only in SINDy). Note that TE is shown on a logarithmic scale. The dashed horizontal line at $TE = 1$ represents the threshold, which can be seen as a minimum requirement for good performance. The line at $NC = 1$ represents the optimal complexity value. The optimal value for TD is 0. One value for SINDy was too small to fit on the plot, so the boxplot was truncated and the value is instead displayed numerically at the end of the whisker.

4.5.4 Results and discussion

The comparative evaluation of ProGED and its comparison to competitors involved the reconstruction of known ODEs from given data of various levels of coarseness, noise, and observability.

This allowed the assessment of the methods' robustness, as well as their strengths and limitations related to their applicability to practical tasks of modeling dynamical systems from data. The methods were tested on an updated and extended version of the Strogatz benchmark that consists of ten dynamical systems. Three different performance measures were considered.

We first discuss the resuts of the evaluation of ProGED and its comparison with competitors for the **full observability scenario**. These results are summarized (for models spaces constrained by the specified grammars) in Figure 8. Each of the three panels of the Figure corresponds to one of the performance measures.

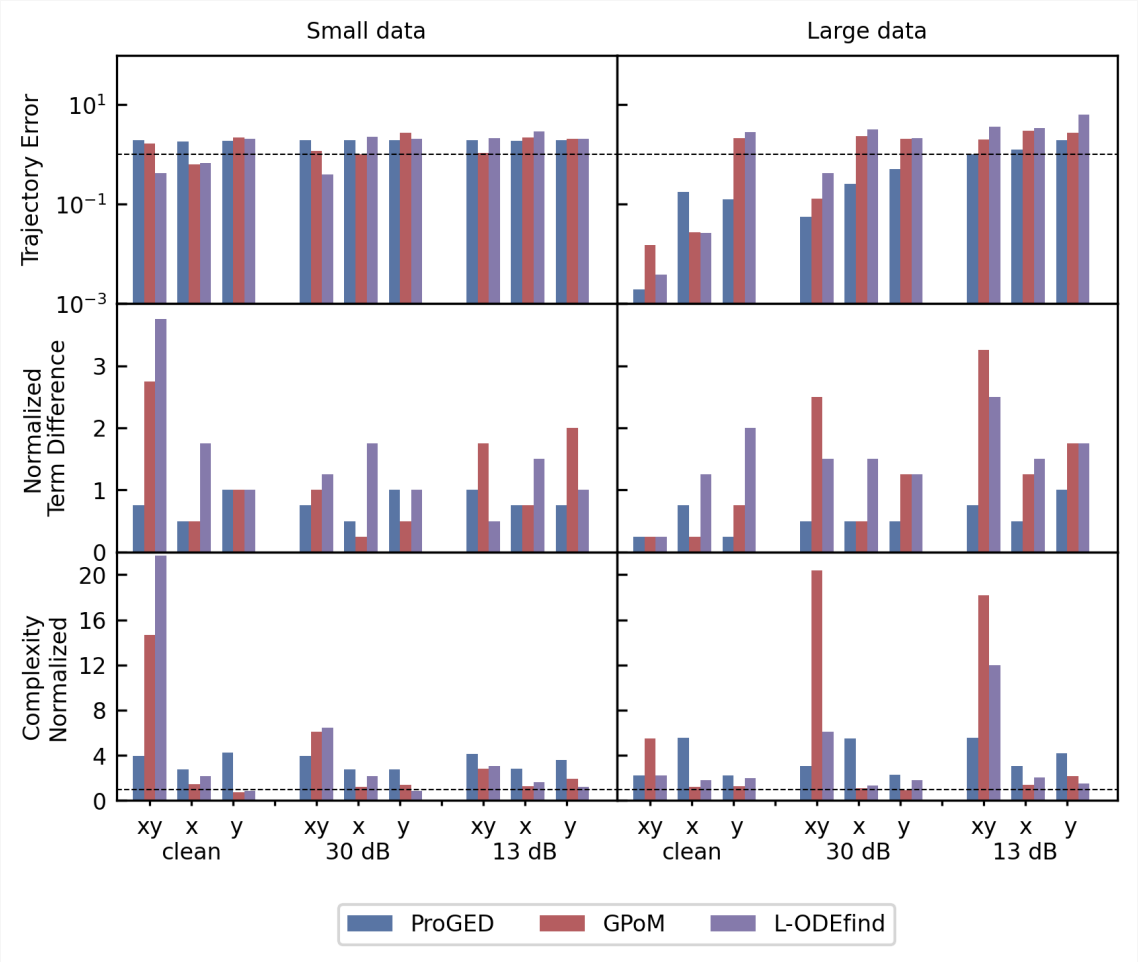

Figure 9. Comparison of the performance of three ODE discovery methods under partial observability, with each method colour-coded according to the legend. The top row displays the trajectory error (TE) on a logarithmic scale, the middle row the term difference (TD) and the bottom row the normalized complexity (NC). The graphs are arranged into two columns based on data length, as indicated by the subtitles. On the x-axes, the results are hierarchically grouped by SNR (clean, 30 dB, 13 dB) and by the observability modes: full observability (labelled as xy), observability in only the x variable (labelled as x), and observability in only the y variable (labelled as y). Since only one system (vdp) was tested in the partially observed setting, error bars are not included. The dashed horizontal line at $TE = 1$ represents the threshold/ minimum requirement for good performance, while the line at $NC = 1$ represents the optimal complexity value. The optimal value for TD is 0.

The results show that the reconstruction on clean, large data as expected outperformed all the other configurations (see the first group of three boxplots in the top right subplot). All three methods had a median TE of around 10^{-2} in this configuration, as well as the lowest TD, meaning that they made the least reconstruction mistakes. This is because noise-free, higher-resolution data is an ideal scenario, especially for methods that rely on numerical differentiation.

Both the presence of noise and the data size/granularity notably influenced the methods' performance. A comparison of the TE on the clean, small data with the TE on the moderately noisy (30 dB SNR) but large data reveals that moderate noise had a more pronounced effect on TE in all three methods. However, the TD and NC metrics present a somewhat different picture. There were more erroneous terms when provided with small, clean data as compared to large, moderately noisy data. One plausible explanation for this discrepancy is that noise has a more substantial effect on parameter estimation, influencing TE more, as compared to the effect on the actual structure selection algorithm.

The assessments on highly noisy data (13 dB SNR) unsurprisingly yielded the poorest results, irrespective of data size/granularity. The TE was around or above 1 for all methods, and, on average, the reconstructed models contained more than twice the number of incorrect or missing terms compared to the terms in the true model.

ProGED performed best in terms of all three metrics for clean data of both sizes. For data with noise, DSO performed best, but the differences in performance were not significant. SINDY, on the other hand, does perform much worse than both DSO and ProGED.

In the **partial observability** scenario, we considered only the *vdp* system for reconstruction. Three different observability settings were included, where both variables (xy) or only one variable (x and y) were observed. The results are shown in Figure 9. In our discussion, we focus only on the results from the large dataset.

As expected, the reconstruction performance under partial observability decreases with decreasing SNR. On clean, large data, all the methods showed relatively good reconstruction of dynamics under at least one of the two partial observability settings. However, only ProGED was able to reconstruct *vdp* under the moderate SNR of 30 dB. Interestingly, while ProGED performed similarly under both partial observability settings (when either x or y were observed), GPoM and L-ODEfind showed a bigger discrepancy and performed well when only x was observed, but not when only y was observed.

Effects of data coarseness. All the methods performed (more) poorly when applied to a small/coarse dataset containing only 100 data samples, sampled at 0.1 Hz. Conversely, as anticipated, the reconstruction of model dynamics improved significantly when the methods were applied to larger/finer datasets (2000 data points, sampled at 0.01 Hz). We highlight this point given the common usage of benchmarks which include a limited number of data samples: The performance of ODE discovery methods should also be tested on larger data samples. The length and the sampling rate of the large data we used are still well within the range of common sets of measurements in real-world experiments.

Effects of noise. As expected, the measurement noise greatly affects the ability to identify an underlying dynamical system. The evaluated methods had great difficulties when the signal-to-noise ratio (SNR) was the lowest. Interestingly, when the SNR was moderate, noise had relatively more influence on the results under full observability than the influence of data size and coarseness. Under partial observability, data coarseness had a greater influence on the results as compared to the influence of SNR. This was to some degree anticipated, as the methods employing numerical differentiation (used under full observability) would demonstrate a higher drop in performance when exposed to noisy data, as compared to the methods that use simulation of ODEs (which is necessary under partial observability).

Computational complexity. Lastly, while methods that rely on the polynomial structure

(GPoM) or linearity in parameters (SINDy, L-ODEfind) perform fast and sufficiently well for some systems, ProGED and DSO are able to generate and identify much larger spaces of possible ODE models. This comes at the cost of longer running times. The response time of reconstruction experiments can be greatly reduced through extensive parallel implementation, particularly in the case of ProGED. Alternatively, the speed of ProGED can be further increased by using more restricted grammars in practical applications where domain knowledge is available.

4.5.5 Relevant publication

- Nina Omejc, Boštjan Gec, Jure Brence, Ljupčo Todorovski, Sašo Džeroski. Probabilistic grammars for modeling dynamical systems from coarse, noisy, and partial data. *Machine Learning*, 2024. DOI: 10.1007/s10994-024-06522-1.

4.5.6 Software and datasets availability

All methods used in the experiments are available online, in the respective GitHub repositories. Our ProGED SW is available from the Github repository https://github.com/brencej/ProGED. Additionally, we provide our code by which we obtained the presented results. It is located on the Github repositoty: https://github.com/NinaOmejc/symreg_methods_comparison.

We have also released the data we used for evaluation, which contains trajectories of ten dynamical systems, which describe the temporal evolution of the state variables of each system. Seven out of the ten systems come from the Strogatz benchmark, currently widely used for system identification. Due to several limitations and problems encountered with this benchmark, we have improved/ extended it in two ways.

First, we added models of three new dynamical systems: a model of coupled phase oscillators (cphase); the Stuart-Landau (stl) oscillator (a normal form of a Hopf bifurcation); and the Lorenz chaotic oscillator (lorenz), which have often been used in system identification experiments and add some variety to the benchmark set of dynamical systems. The system cphase, for example, introduces time dependency on the right-hand side, a property of non-autonomous systems, while lorenz contains three state variables.

We generated the trajectories by directly simulating the models using the LSODA algorithm, as the data from the Strogatz benchmark did not provide adequate accuracy for certain dynamical systems. Moreover, the Strogatz benchmark contains trajectories with only 100 time points, sampled at a 0.1 Hz rate. The coarse sampling of data can significantly affect the performance of system identification methods. For example, some oscillators do not yet complete their limit cycles during this period, so their main characteristic stays hidden. For these reasons, we have decided to improve upon the Strogatz benchmark and make our extended benchmark available on the digital repository Zenodo under the name Dynobench 10.5281/zenodo.10041312.

Analogously to the Strogatz benchmark, Dynobench includes coarse data of only 100 time points sampled at 0.1 Hz (hereafter called ”small data”). Additionally, we created densely sampled data with 2000 time points, sampled at 0.01 Hz (hereafter called ”large data”). Each system of ODEs was simulated four times, each time with a different set of initial values of the state variables. Python code for the simulation of dynamical systems is included in the Dynobench repository.

Besides clean (noise-free) data, we also provide noisy data with a 30 dB signal-to-noise ratio (SNR), where the signal was a thousand times stronger than the noise, and data with a 13 dB SNR (signal twenty times stronger than noise). We added Gaussian noise $U_{noise}[V] \sim \mathcal{N}(0, \sqrt{P_{noise}[W]})$, with mean $\mu = 0$ and standard deviation $\sigma^2 = \sqrt{P_{noise}[W]}$. The $P_{noise}[dB]$ was calculated based on the power of the signal and the desired SNR, following $P_{noise}[dB] = P_{signal}[dB] - SNR[dB]$.

4.5.7 Next steps

We are currently pursuing several directions for further work. At the top of the list is increasing the number of systems and systems' trajectories used for the evaluation of the approach, which would provide more reliable evaluations and higher statistical power of the tests applied in this context. This is especially true for evaluation under partial observability.

Context free grammars (and their probabilistic version, PCFGs) have limitations in expressing domain knowledge. We are considering the use of probabilistic attribute grammars, which allow us to capture context dependencies. In attribute grammars, symbols can have attributes and production rules can have attribute rules, which impose constraints. We thus need more elaborate sampling algorithm than for PCFGs.

Finally, we are considering a sampling procedure of Bayesian nature. Here the probabilities of the rules of the PCFG would be updated as equation structures are sampled and evaluated (after the parameters in the equation structures are fitted). This would allow focusing the search in promising regions of the space of equation structures.

5 Task 1.4: AI for Fast Approximation of Scientific Computations

5.1 Overview

In many computational problems relating to sustainability we face bottlenecks. Often we are able to simulate systems, but we are not able to do so at the scale that is needed to solve the problem at hand. This is a challenge in climate modelling, one must resolve physical processes and land-atmosphere-ocean interactions in 3D grid data. Likewise, materials design problems often require computations in statistical physics such as those based on density functional theory, for a very large set of candidate materials.

This task aims expand the scale at which we can perform scientific computation by developing AI-based surrogate methods that can provide fast approximations to expensive numerical computations. A common challenge in developing ML-based surrogates is that it is generally difficult to create a large training dataset when each data point itself requires an expensive numerical simulation. A key question in developing such methods is therefore how we can train surrogate models that are suitable to large-scale simulation from data from smaller-scale gold-standard simulations.

In the next Section, we present an example of such approaches, in which develop neural methods for classical density functional theory, a class of computationally efficient methods for simulating fluids at mesoscopic scales, which can be trained on data from small-scale Monte Carlo simulations. As noted previously, this work is closely related to the work in T1.2. The eventual use case of these methods is to enable applications in materials design.

5.2 Neural Density Functionals for Materials Design

Contributing partner: UvA

5.2.1 Project Description

Background. Designing new materials often requires screening of a large set of candidate materials. While individual materials can be simulated with Monte Carlo (MC) or molecular dynamics (MD) methods, it is simply not feasible to perform such simulations for all candidate materials. An example is the *screening of metal-organic frameworks (MOFs) for carbon capture applications*. Here the search space of candidate materials is very large; There are over 100k known MOF structures in the Cambridge Structural Database [99]. Evaluating the suitability of these candidate structures therefore requires simulation methods that are several orders of magnitude faster than their MC-based or MD-based counterparts.

Neural Classical Density Functional Theory. Density functional theory (DFT) expresses observable quantities in quantum many body systems in terms of a functional of the electron density. In statistical mechanics, *classical* density functional theory (cDFT), expresses the intrinsic Helmholtz free energy as a functional of a particle density. Simulations based on cDFT are much faster than MC or MD simulations, but to date application of cDFT has been somewhat held back by the fact that the free energy functional can at best be computed approximately. Neural approaches to cDFT sidestep the need for an analytical approximation by learning a parametric approximation of the functional from a dataset of MC simulations.

More concretely, cDFT is a grand-canonical framework that expresses the grand potential $\Omega[\rho]$

as a functional of a particle density $\rho(\mathbf{r})$,

$$\Omega[\rho] = \mathscr{F}[\rho] + \int \mathrm{d}r\rho(\mathbf{r})\left(V_{\mathrm{ext}}\left(\mathbf{r}\right) - \mu\right). \tag{16}$$

Here $\mathscr{F}[\rho]$ represents the intrinsic Helmholtz free-energy functional, $V_{\mathrm{ext}}(\mathbf{r})$ the external potential, and μ the chemical potential.

For a given particle-particle interaction and temperature, the unique density functional $\mathscr{F}[\rho]$ determines the thermodynamic and structural equilibrium properties of a system for any chemical potential and external potential. Within cDFT, it is convention to split the intrinsic free-energy functional into excess contribution and an ideal contribution that can be computed analytically,

$$\mathscr{F}_{\mathrm{exc}}[\rho] = \mathscr{F}[\rho] - \mathscr{F}_{\mathrm{id}}[\rho], \qquad \mathscr{F}_{\mathrm{id}}[\rho] = \frac{1}{\beta}\int \mathrm{d}\mathbf{r}\,\rho(\mathbf{r})\big(\ln\rho(\mathbf{r})\Lambda^3 - 1\big), \tag{17}$$

with $\beta = 1/k_BT$ and Λ the thermal wavelength.

Mathematical proofs exist [1] stating that (i) the equilibrium density profile, denoted here as $\rho_0(\mathbf{r})$, minimizes $\Omega[\rho]$, and (ii) the equilibrium grand potential equals $\Omega[\rho_0]$. Clearly, once $\mathscr{F}[\rho]$ for the system of interest is known, the Euler-Lagrange equation $\delta\Omega[\rho]/\delta\rho(\mathbf{r})|_{\rho_0} = 0$ can be solved to find $\rho_0(\mathbf{r})$ and $\Omega[\rho_0]$. The Euler-Lagrange equation takes the form

$$\rho_0(\mathbf{r}) = \frac{1}{\Lambda^3}\exp\left(\beta\mu - \beta\frac{\delta\mathscr{F}_{\mathrm{exc}}[\rho]}{\delta\rho(\mathbf{r})}\bigg|_{\rho=\rho_0} - \beta V_{\mathrm{ext}}\left(\mathbf{r}\right)\right). \tag{18}$$

This self-consistency relation can be leveraged to find $\rho_0(\mathbf{r})$ through recursive iteration.

While cDFT has tremendous potential as a method for fast simulation of many-body systems at mesoscopic scales, realizing this potential poses challenges. The self-consistency iteration in cDFT often converges in 10–100 steps, which means that equilibrium densities can be computed orders of magnitude faster than in an equivalent MC or MD simulation. However to perform this iteration, we need to compute the functional derivative $\delta\mathscr{F}_{\mathrm{exc}}[\rho]/\delta\rho(\mathbf{r})$ of the excess free energy. Loosely, the excess free energy describes the expected energy associated with two-body and multi-body interactions in a system of particles. This is a quantity that is extremely difficult to compute, even in approximation. Moreover, a new approximation is needed for each type of particle. This means that applying cDFT to a range of application domains has proven difficult in practice.

In recent years, there has been a resurgence of cDFT developments facilitated by machine learning (ML) methods, which employ virtually *exact* thermodynamic and structural data obtained from explicit many-body simulations to learn data-driven representations of the excess free-energy functional $\mathscr{F}_{\mathrm{exc}}[\rho]$. In the classical regime, the first machine-learned cDFTs focused on supercritical Lennard-Jones fluids, for which explicit approximate functional forms for $\mathscr{F}_{\mathrm{exc}}[\rho]$ were fitted to density profiles in external fields obtained from simulations, both for 1D [64] and 3D systems in planar geometry [93]. Recent work, once again leveraging simulations of density profiles in a variety of external potentials, has shown that a neural approximation of the functional derivative $\delta\mathscr{F}_{\mathrm{exc}}/\delta\rho$ for hard-sphere systems outperforms existing approaches based on fundamental measure theory (FMT) [141] in accurately estimating inhomogeneous density profiles.

Pair Correlation Matching. Our contribution to neural methods for cDFT is new objective for training an approximate free energy functional, which we refer to as *pair-correlation matching*. In pair correlation matching, we use MC simulations to obtain an estimate of the direct correlation function $c^{(2)}(\mathbf{r},\mathbf{r}')$, which describes fluctuations relative to a constant equilibrium density in the

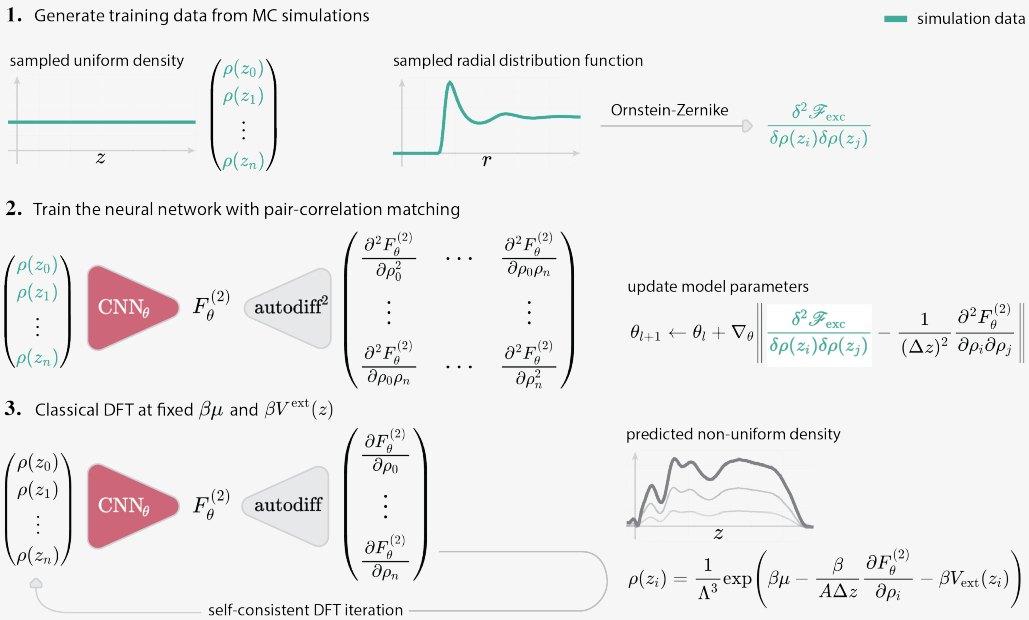

Figure 10. ***1.*** *Bulk densities in planar geometry $\rho(z_i) = \rho_b$ and radial distribution functions $g(r)$ are sampled from Monte Carlo simulations of homogeneous bulk systems of Lennard-Jones particles. Each $g(r)$ is converted to the second functional derivative of the excess free-energy $\delta^2\mathscr{F}_{exc}/\delta\rho(z_i)\delta\rho(z_j)$ by employing the Ornstein-Zernike equation.* ***2.*** *Through automatic differentiation (autodiff^2), the neural functional $F_\theta^{(2)}$ is optimized to fit the Hessian of the model output with respect to input density profiles to $\delta^2\mathscr{F}_{exc}/\delta\rho(z_i)\delta\rho(z_j)$.* ***3.*** *The optimized model can then be applied in cDFT to obtain non-uniform equilibrium density profiles through automatic differentiation (autodiff) and the free energy $F_\theta^{(2)}$ for a system of Lennard-Jones particles subjected to arbitrary external potentials.*

absence of an external potential (i.e. $V_{\text{ext}}(\mathbf{r}) = 0$). The direct correlation function is related to the second functional derivative of the excess free energy through

$$c^{(2)}(\mathbf{r}, \mathbf{r}') = -\beta \frac{\delta^2 \mathscr{F}_{\text{exc}}[\rho]}{\delta\rho(\mathbf{r})\delta\rho(\mathbf{r}')}. \tag{19}$$

In pair correlation matching, we approximate the excess free energy functional $\mathscr{F}_{\text{exc}}[\rho]$ using a neural network $F_\theta(\{\rho_i\})$ that accepts densities $\rho_i := \rho(\mathbf{r}_i)$ on a grid of points $\{\mathbf{r}_i\}$ as inputs. To approximate the second functional derivative, we use automatic differentiaton compute the Hessian with respect to ρ. We can then optimize with respect to the weights θ to match the Hessian of the network to estimates of the correlation function based on MC simulations. In a simplified quasi one-dimensional system, in which the density $\rho(\mathbf{r}) = \rho(z)$ and correlation function $\bar{c}^{(2)}(|z - z'|)$ vary only with a single cooridinate z, this leads to the loss function

$$L(\theta) = \sum_{i,j} \left(\bar{c}^{(2)}(|z - z'|) + \frac{\beta}{A(\Delta z)^2} \frac{\partial^2 F_\theta^{(2)}}{\partial\rho_i \partial\rho_j} \right)^2. \tag{20}$$

Unlike previous ML approaches to cDFT [64], [93], [141], [151], pair correlation matching learns a neural functional directly from radial distribution functions sampled from short simulations of systems with constant density (illustrated in Figure 10). By contrast, many existing approaches

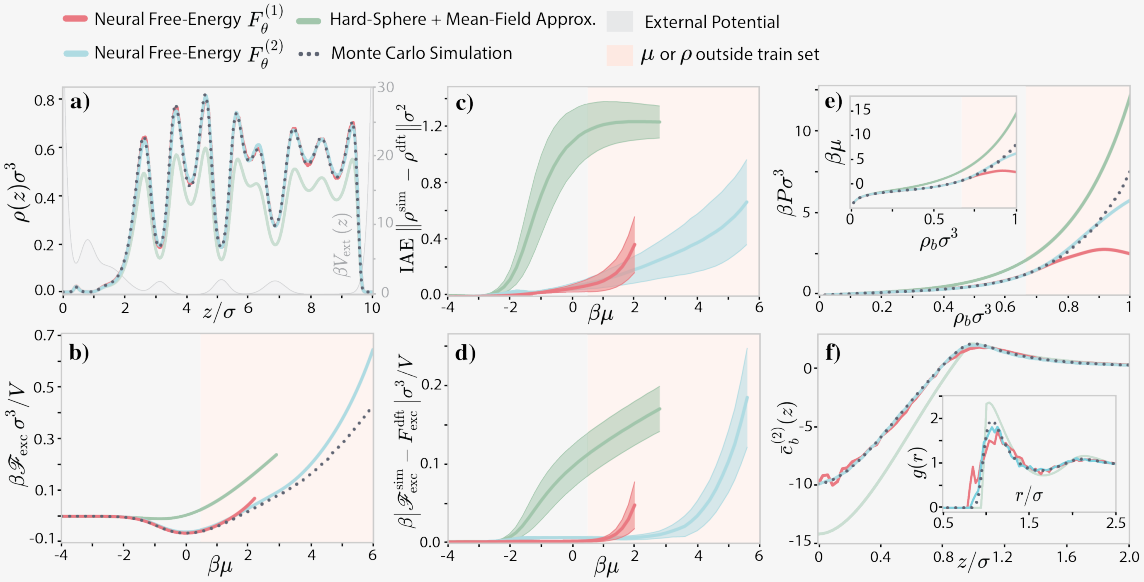

*Figure 11. Evaluation of neural free-energy functionals $F_\theta^{(1)}$ and $F_\theta^{(2)}$, where $F_\theta^{(1)}$ is optimized by matching inhomogeneous one-body densities and $F_\theta^{(2)}$ by pair-correlation matching in the homogeneous bulk. **a)** Density profiles of a Lennard-Jones system in a planar geometry characterized by an external potential (shown in gray) at a chemical potential of $\beta\mu = 0$ obtained from DFT using $F_\theta^{(1)}$ and $F_\theta^{(2)}$ and the mean-field approximation F_{exc}^{MF}, along with the simulated density profile. **b)** Comparison of the free-energy estimates using $F_\theta^{(1)}$, $F_\theta^{(2)}$, and F_{exc}^{MF} for the specific external potential shown in a), for the same chemical potential of $\beta\mu = 0$. **c)** Integrated absolute error (IAE) between density profiles obtained from DFT using $F_\theta^{(1)}$, $F_\theta^{(2)}$, and F_{exc}^{MF}, along with the densities sampled from simulation. **d)** Absolute error of the excess free energy from DFT and simulations. **c)/d)** Data is shown for 150 distinct external potentials, evaluated across the range $-4 < \beta\mu < 6$, with steps of $\Delta\beta\mu = 0.1$. The area of the mean $\pm$ standard deviation is colored. The error is shown up to the point where the DFT iterations stop to converge to a solution within 1000 iterations. **e)** The pressure and the chemical potential obtained from DFT and simulations. **f)** The laterally integrated direct correlation function $\bar{c}_b^{(2)}(z)$ at $\rho_b\sigma^3 = 0.62$ and the radial distribution function $g(r)$ obtained from simulation and DFT.*

have focused matching the first derivative to simulation data, which requires simulations of densities with varying external potentials $V_{\text{ext}}(\mathbf{r})$. As a consequence, pair correlation matching has the potential to substantially improve the data efficiency of neural methods for cDFT.

Results. To evaluate the accuracy of our neural excess free-energy functional $F_\theta^{(2)}$, we compare it to the Van der Waals-like mean-field approximation $F_{\text{exc}}^{\text{MF}}$, which treats the attractions of Lennard-Jones particles as a perturbation on the hard-sphere system, as implemented in PyDFTlj [143]. We use the White-Bear mark II version of FMT for the excess free energy of the hard-sphere system [6]. Additionally, we compare to $F_\theta^{(1)}$, which is a neural functional trained by minimization of the error between $(1/\Delta z)\partial F_\theta^{(1)}/\partial\rho_i$ and $\delta\mathscr{F}_{\text{exc}}/\delta\rho(z_i)$ rather than by pair-correlation matching. This neural functional is trained on a dataset of 800 non-uniform densities, subjected to the same set of chemical potentials as before. By approximating $\delta\mathscr{F}_{\text{exc}}/\delta\rho(z_i)$ by the gradient $(1/\Delta z)\partial F_\theta^{(n)}/\partial\rho_i$ for $n = 1$ and 2, both neural functionals are applied in Picard iterations [10], [30], [45] to obtain DFT estimates for the equilibrium density profiles of inhomogeneous systems according to Eq. (18).

The DFT results for an exemplary external potential at $\beta\mu = 0$ are shown in Figure 11a, where we observe that the neural functionals $F_\theta^{(1)}$ and $F_\theta^{(2)}$ provide similar estimates, both outperforming $F_{\text{exc}}^{\text{MF}}$. For the same external potential, we evaluate the accuracy of DFT estimates for the free energy for a range of chemical potentials $-4 < \beta\mu < 6$ (Figure 11b). We compare with the excess free energy obtained from GCMC simulations through thermodynamic integration. We observe that both neural functionals outperform $F_{\text{exc}}^{\text{MF}}$ within the range of μ values in the training set, exhibiting good agreement with the simulations. The DFT estimates are shown until the DFT iterations diverge and errors become unmanageable. This reveals that $F_\theta^{(1)}$ diverges rapidly when extrapolating beyond the training set, even earlier than $F_{\text{MF}}^{\text{exc}}$. In contrast, $F_\theta^{(2)}$ is capable of converging to a solution far beyond the trained μ range.

For a more detailed comparison for various inhomogeneous systems, we performed separate DFT calculations for 150 distinct external potentials, evaluated across the range $-4 < \beta\mu < 6$. For both the density estimates (Figure 11c) and the free-energy estimates (Figure 11d), we observe excellent agreement between the $F_\theta^{(1)}$ and $F_\theta^{(2)}$ functionals and simulated data. They demonstrate similar performance for μ values within the training set range, and outperform the $F_{\text{exc}}^{\text{MF}}$ functional for all evaluated external potentials (Figure 11). Again, we observe that $F_\theta^{(2)}$ surpasses $F_\theta^{(1)}$ when extrapolating beyond the μ range encountered during training.

Conclusion and Next Steps. Our results suggest that pair correlation matching works much better than one might expect. Using the direct correlation function as training data greatly simplifies dataset generation. We can simply simulate systems at different levels of constant density, and do not have to simulate systems in wide variety of external potentials, as in existing methods. A limitation of this approach is that correlation functions likely encode a particular class of inhomogeneities. This may well not be sufficient to describe all phenomena in a physical systems, so one might expect inaccurate predictions in certain regimes, such as near phase transitions or in systems with very high densities. Yet, the extent to which a functional trained with pair correlation matching can generalize to non-uniform densities is encouraging. The computational expense of simulating systems with a variety of external potentials greatly increases as we move from the quasi one-dimensional systems considered in this work to fully three-dimensional systems. Here pair correlation matching provides an avenue for improving the data efficiency of neural cDFT methods. This suggests that, with further development, these neural cDFT may well become a viable method for fast simulation of complex phenomena, such as carbon capture in MOFs.

5.2.2 Relevant Publications

- J. Dijkman, M. Dijkstra, R. van Roij, *et al.*, *Learning Neural Free-Energy Functionals with Pair-Correlation Matching*, May 2024. DOI: 10.48550/arXiv.2403.15007. arXiv: 2403.15007 [cond-mat]

5.2.3 Relevant Use Cases

This work is relevant to use case 6, the open materials discovery competition.

5.3 Estimating the energy of Bi atoms configurations with machine learning

Contributing partner: JSI

5.3.1 Summary

Due to its interesting properties, bismuth (Bi) is increasingly relevant not just for materials science but also for a range of other applications, including quantum computing. However, simulating amorphous bismuth is computationally expensive because it requires DFT calculations with very large supercells. Empirical force-field calculations would be much faster, yet they would be relevant only if the utilized force field is sufficiently accurate.

Here, we propose a method for parametrizing a force field based on Machine Learning. We trained a predictor (regressor) for Bi atoms configurations on several thousand DFT-calculated configurations, ranging from 2 to 64 Bi atoms. We found it to provide a reasonable estimate of the configuration's energy at only a fraction of the DFT computational cost.

Bulk Bi configurations were generated using DFT (density functional theory) - based ab initio molecular dynamic (AIMD) simulations. The DFT calculations are performed with QuantumESPRESSO (www.quantum-espresso.org). They used the Perdew-Burke-Ernzerhof (PBE) functional, plane-wave kinetic energy cutoffs of 50 Ry (wavefunctions) and 400 Ry (charge density), as well as structural relaxations (clusters) and AIMD (bulk structures).

Predictive models are built by different machine learning methods, including support vector regression and neural networks. Features (descriptors) were generated by using the Dscribe package (https://singroup.github.io/dscribe/latest/). In particular, the following descriptors were used: MBTR (Multi-Body Tensor Representation), SOAP (Smooth Overlap of Atomic Orbitals), Sine Matrix and inverse pairwise distance. The best results were achieved by using support vector regression.

Our method is applicable to Bi configurations consisting of an arbitrary number of atoms and can be extended to estimate other quantities of interest, such as forces. Although these results are still preliminary, they open an avenue towards a practically useful force field for modeling arbitrary Bi structures, including liquid and amorphous Bi.

5.3.2 Relevant publications

There are no as yet published papers or completed drafts submitted for publication describing the above work. However, the work has been described in a poster presentations:

- Tone Kokalj, Matej Petković, Daniel Meljanac, Zoran Levnajić, Juan Jose Palacios, Sašo Džeroski. Estimating the energy of Bi configurations with machine learning. *4th Workshop on Machine Learning Modalities for Materials Science*, Ljubljana, 13th-17th May 2024.).

5.3.3 Next steps

We are pursuing several directions for further work. One is the consideration of additional configurations of Bismuth atoms, i.e., additional training examples (of which we had relatively few). Another direction is the consideration of additional features (descriptors). Finally, we will consider the use of additional machine learning methods.

6 Task 1.5: Reducing the Energy Requirements of Computation

As the demand for AI continues to grow, the computational demand for machine learning solutions is becoming increasingly large. The recent blooming of foundation models, that guarantee generality and reduce deployment efforts through mechanisms like transfer learning and knowledge distillation might apparently look a cheap and effective way, but hide the cost of maintaining these large models even for simpler tasks. This evidently translates into progressively higher energy consumption, posing both practical and conceptual issues.

It is a key challenge to find ways to make AI solutions more energy-efficient. In particular, reducing the memory and computation power (and besides, reducing the environmental impact) of these systems can be divided into three main challenges: training models with little data, using limited computational resources for training these models, and deploying models that save energy.

6.1 Overview

Within this task, partners are contributing in the core challenge of how to reduce energy requirements of computation, tackling the problem from many different perspectives so that the advances can be each other complimentary but aiming at the same goal. We will here below summarize at a glance the contributions per partner.

In Subsection 6.2, IPP proposes the employment of low-rank approaches for structured prediction. This contribution frames itself on one opposite side of deep learning, providing a frugal learning shame with excess risk bounds. The same partner takes as well a complementary approach by making complex CNNs shallow. In Subsection 6.3, IPP proposes EASIER, a simple yet effective strategy to reduce the Deep Neural Network's depth by iteratively removing irrelevant layers from the model. This work frames itself in the typical deployment context where practitioners take off-the-shelf architectures like ResNets and Vision Transformers and deploy for a specific downstream task. In this work, IPP shows that all the computational complexity put at stake is not really necessary, but can be sometimes massively reduced and adapted to the specific task to be solved.

In Subsection 6.4 and Subsection 6.5, IDEAS NCBR proposes its contributions in the context of Zero-waste machine learning, in the contexts of knowledge accumulation and sustainable computer vision for autonomous machines. Related to knowledge accumulation, IDEAS NCBR proposes a Selective Ensemble of Experts for Continual Learning (SEED) that is able to mitigate the issue of catastrophic forgetting, typical in continual learning scenarios, through the employment of experts. Besides, in the same context IDEAS NCBR proposes EFCIL, an Exemplar-Free Continual Learning strategy, where catastrophic forgetting is avoided by using an adversarial sample generation. Related to sustainable Computer Vision for autonomous machines, IDEAS NCBR proposes a federated solution to contrastive learning-based solutions that refuses the total bandwidth needed for the computation, through a Momentum-Aligned Contrastive Split Federated Learning (MonAcoSFL) approach.

In Subsection 6.6, CERTH tackles the issues of both reducing the model's complexity of learned models through a Knowledge Distillation (KD) scheme, where a shallow student is learned, and to make the whole KD scheme computationally efficient. CERTH accomplishes the second ambitious objective by selecting a subset of relevant samples through which performing KD, leveraging on the information content of these. Another parallel contribution from CERTH lies in making inference computation efficient: in Subsection 6.7, a scheme where instead of training one large model an ensemble of two shallower ones is deployed, is proposed. More specifically, "Two Heterogeneous CNN" proposes to impose a differentiation in prediction between these two models, such that the

two focus on more diverse features from the same input. This approach has the dual advantage of both making computation demands lower and ensuring higher diversity in the feature extraction process.

Finally, in Subsection 6.8 HPI employs the notion of low-precision Gaussian process regression to decrease the power consumption of AI. This contribution comes timely, closer to the hardware level, leveraging the knowledge that high-precision error-free computation, in many contexts, is not strictly necessary. Through the employment of Gaussian processes-based models to leverage approximate computing, HPI shows that massive power reduction is indeed possible with negligible impact on the model's performance.

6.2 Low-rank approaches in structured prediction

Contributing partner: IPP

The growing energy requirements for computational tasks, especially driven by AI solutions, pose a core challenge to both environmental sustainability and operational efficiency [82], [124]. Within this context, we propose innovative methods to reduce the energy demands of complex computational tasks. Leveraging the kernel trick in the output space, our research introduces kernel-induced losses as a robust way to define structured output prediction tasks for various output modalities. By integrating these techniques with deep neural networks, which are inherently more expressive and powerful for handling inputs like images and texts, we aim to enhance computational efficiency. Through this work, we demonstrate that it is possible to achieve state-of-the-art results in structured prediction while simultaneously cutting down on energy consumption, making a compelling case for its relevance and importance in sustainable AI development.

6.2.1 Technical Description

Kernel-induced losses provide a principled way to define structured output prediction tasks for a wide variety of output modalities. In particular, they have been successfully used in the context of Input Output Kernel Regression that solves a surrogate vector-valued regression problem where the input and the output spaces are both Reproducing Kernel Hilbert Spaces. At inference time the structured predictive model $\hat{f}$ is obtained by decoding the infinite-dimensional output using kernel trick. While offering SOTA performance on various Structured Prediction problems, these methods suffer from two drawbacks: first, as with other kernel methods, output kernel regression does not scale to large-size datasets, and second, they currently do not have a counterpart in parametric modeling (e.g. neural networks), needed for dealing with images or text for instance. We have developed two approaches to solve these issues.

In "Sketch in, Sketch out: Accelerating both learning and inference for structured prediction with kernels", we proposed to equip these methods with sketching-based approximations, applied to both the input and output feature maps. Sketching is a well-known technique that allows to reduce the computation time and space by leveraging random projections. The originality of this work relies on the application of two random projection operators on the infinite dimensional input and output feature maps, giving rise to the estimator depicted on the right in Figure **??**. We proved excess risk bounds on the original structured prediction problem, showing how to attain close-to-optimal rates with a reduced sketch size that depends on the eigendecay of the input/output covariance operators. We showed that the two approximations have distinct but complementary impacts: sketching the input kernel mostly reduces training time, while sketching the output kernel decreases the inference time. Empirically, our approach was shown to scale, achieving state-of-the-art performance on benchmark data sets where non-sketched methods are intractable.

Table 6. Comparison of Deep Sketched Output Kernel Regression (DSOKR) with different baseline on SMI2Mol test set

	GED w/o edge feature ↓
SISOKR	3.330 ± 0.080
NNBary-FGW	5.115 ± 0.129
Sketched ILE-FGW	2.998 ± 0.253
DSOKR	$\mathbf{1.951 \pm 0.074}$

With our second contribution, "Deep Sketched Output Kernel Regression for Structured Prediction", we tackled the question of how to train neural networks to solve structured output prediction tasks, while still benefiting from the versatility and relevance of kernel-induced losses. We designed a novel family of deep neural architectures, whose last layer predicts in a data-dependent finite-dimensional subspace of the infinite-dimensional output feature space deriving from the kernel-induced loss. This subspace is chosen as the span of the eigenfunctions of a randomly approximated version of the empirical kernel covariance operator obtained by sketching. Interestingly, this approach unlocks the use of gradient descent algorithms (and consequently of any neural architecture) for structured prediction. Experiments on real-world supervised graph prediction problems show the relevance of our method. Table 6 reports numerical results of DSOKR on the emblematic task which consists of predicting a molecule from its symbolic representation SMILE. DSOKR is compared to other structured prediction tools based on the minimization of Optimal Transport losses as well as SISOKR, described above.

6.2.2 Relevant Publications

- El Ahmad, T., Brogat-Motte, L., Laforgue, P., & d'Alché-Buc, F, Sketch in, sketch out: Accelerating both learning and inference for structured prediction with kernels, AISTATS 2024.
- Ahmad, T. E., Yang, J., Laforgue, P., & d'Alché-Buc, F. (2024). Deep Sketched Output Kernel Regression for Structured Prediction. arXiv preprint arXiv:2406.0925, accepted in ECML-PKDD 2024.

6.2.3 Relevant Software Releases / Datasets

- The implementation of our work "Sketch In Sketch Out: Accelerating both Learning and Inference for Structured Prediction with Kernels" can be found at https://github.com/tamim-el/sisokr.
- The implementation of our work "Deep Sketched Output Kernel Regression for Structured Prediction." can be found at https://github.com/tamim-el/dsokr.

6.2.4 Relevant Use Cases

The approach, described in Subsection 6.2, will be applied to the UC6 "Open Materials Discovery Project".

6.3 Layer collapse

Contributing partner: IPP

Despite Deep Neural Networks having demonstrated scalability in terms of model and dataset size, they hinder high computational demands. Indeed, neoteric architectures are made up of millions, or even billions, of parameters, resulting in billions, or even trillions, of FLoating-point OPerations (FLOPs) for a single inference [114]. The development of compression techniques, which constitute an essential means of remedying the resource-hungry nature of DNNs, has marked the research landscape over the past decade. It is well-known that the complexity of the model is intrinsically linked to the generalizability of DNNs [40], and since pre-trained architectures that can be used in downstream tasks tend to be over-parameterized, compression with no (or only slight) performance degradation is in principle possible [120]. To design a more efficient architecture, a set of methods has been proposed, ranging from parameter pruning [26] to the reduction of numerical precision [36]. Nonetheless, few approaches are capable of lessening the number of layers in a DNN. Indeed, removing single parameters or whole filters offers very few if any, practical benefits when it comes to using the model on recent computing resources, such as GPU. Thanks to the intrinsic parallel computation nature of GPUs or TPUs, the limitation on layer size, whether larger or smaller, comes mainly from memory caching and core availability. Indeed, reducing the critical path that computations must traverse [127] would help to relieve the DNN's computation demand, which can be achieved by strategically removing layers.

In the work that follows we have identified a simple yet effective strategy to reduce the depth of deep neural networks, identifying the average state of a given rectifier-activated neuron for the trained task. Given the definition of rectifier activation functions, our approach named after EASIER can find the probability that this neuron uses one of the two regions, and hence can calculate an entropy-based metric per layer. Such a metric is then used to drive the linearization of layers toward neural network depth reduction.

6.3.1 Technical Description

Our proposed approach builds its foundations on top of recent research in the field that has already shown that layers can naturally collapse if their parameters are properly pruned [149]. In order to assess the layer's collapse, we need to observe the output of all the neurons belonging to it, identifying their "state". In ReLU-activated networks, we can identify three of them, depending on the sign of the pre-activation signal (hence, these can be +1 for the ON state, -1 for the OFF, and 0). We consider the "zero" state as a don't care state: whether we are in the ON or in the OFF state, if the pre-activation signal values exactly zero, the output will be zero in both cases. From this, we can easily calculate (over the samples in the dataset) the probability (in the frequentistic sense) of a target i-th neuron in the l-th layer of being in the ON state (and its complementary OFF probability), used then to calculate the entropy for this neuron:

$$\mathcal{H}_{l,i} = -\sum_{s_{l,i}=\pm 1} p(s_{l,i}) \log_2 \left[p(s_{l,i})\right] \tag{21}$$

Given the definition in equation 21, $\mathcal{H}_{l,i} = 0$ (which is the condition for neuron collapse) can be verified in two cases:

- $s_{l,i} = -1 \ \forall j$. In this case, the pre-activation signal is always negative or zero. The output of the i-th neuron is always 0 when for example employing a ReLU.
- $s_{l,i} = +1 \ \forall j$. In this case, the pre-activation signal is always positive or zero. As it belongs to the linear region, the output of the i-th neuron is equal to its input (or very close as in

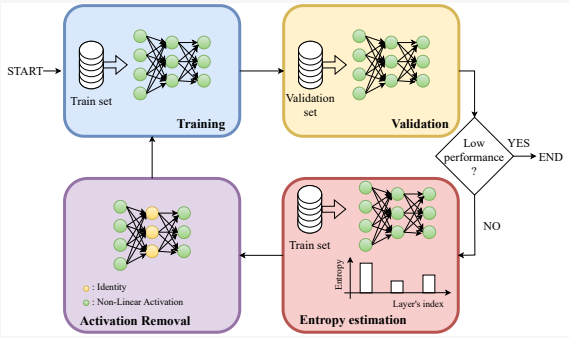

Figure 12. Overview of EASIER. We iteratively train, evaluate, and estimate the entropy on the training set and linearize the lowest-entropy layer of the neural network, until the performance drops.

GeLU). Therefore, since there is no non-linearity between them anymore, this neuron can in principle be absorbed by the following layer.

To estimate one layer's collapse, we can employ the average entropy: for the l-th layer counting N_l neuron it is

$$\widehat{\mathcal{H}}_l = \frac{1}{N_l} \sum_i \mathcal{H}_{l,i}. \tag{22}$$

We would like to have $\widehat{\mathcal{H}}_l = 0$ since we target deep neural networks' depth reduction by eliminating layers with almost zero entropy.

At a glance, we assume that the lowest-entropy layer is the one likely to make the least use of the different regions, or states, of the rectifier. Therefore, the need for a rectifier is there reduced: the rectifier can be linearized entirely. In this regard, we first train the neural network and evaluate it on the validation set. As defined in equation 22, we then calculate the entropy $\widehat{\mathcal{H}}$ on the training set $\mathcal{D}_{\text{train}}$ for all the L rectifier-activated layers, (therefore, the output layer is excluded). We then find the lowest entropy layer and replace its activation with a linear one. To recover the potential performance loss, the model is then fine-tuned using the same training policy and re-evaluated on the validation set $\mathcal{D}_{\text{val}}$. The final model is obtained once the performance on the validation set drops below the threshold δ. An overview on the method is provided in Figure 12.

The approach has been validated through a variety of architectures and datasets. We have selected the architectures ResNet-18, MobileNet-V2, Swin-T and VGG-16, and trained on seven datasets: CIFAR-10, Tiny-ImageNet, PACS and VLCS from DomainBed, as well as Flowers-102, DTD, and Aircraft. All the hyperparameters, augmentation strategies, and learning policies are provided in Appendix, mainly following [137] and [150]. For ResNet-18, MobileNetv2, and VGG-16 all the ReLU-activated layers are taken into account. For Swin-T, all the GELU-activated layers are considered. We show here some results obtained on CIFAR-10 in Figure. Figure 13 shows the test performance (Top-1) versus the number of removed layers for all the considered models on CIFAR-10, achieved with our method EASIER, while Figure 14 studies the impact of EASIER while employing diverse rectifiers. Interestingly, all the models exhibit a similar depth-accuracy trend, regardless of their initial depth, and that except for the ReLU case, all the other rectifiers collapse at a similar depth.

6.3.2 Relevant Publications

- V. Quétu, Z. Liao, and E. Tartaglione, The Simpler The Better: An Entropy-Based Importance Metric To Reduce Neural Networks' Depth, ArXiv preprint arXiv:2404.18949, accepted

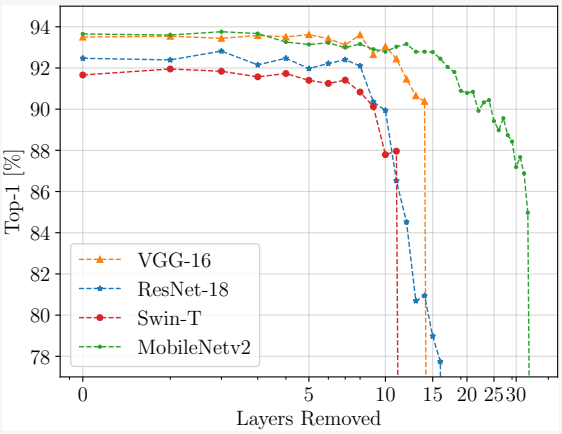

Figure 13. EASIER applied on ResNet-18, VGG-16, Swin-T and MobileNetv2 networks on CIFAR-10. For each model, we gradually remove non-linear layers.

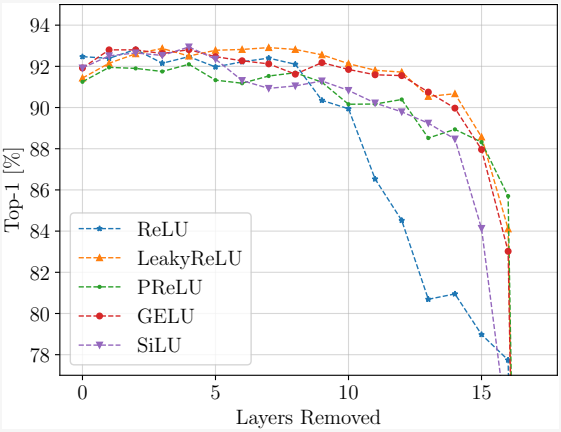

Figure 14. EASIER applied on ResNet-18 on CIFAR-10 with different rectifiers: ReLU, LeakyReLU, PReLU, GELU, and SiLU. Our method is not bound to a specific one and is effective with the most popular.

in ECML-PKDD 2024.

6.3.3 Relevant Software Releases / Datasets

- The Pytorch implementation of our work "The Simpler The Better: An Entropy-Based Importance Metric To Reduce Neural Networks' Depth" can be found at https://github.com/VGCQ/EASIER.

6.3.4 Relevant Use Cases

The developed tool, described in Subsection 6.3, can potentially be applied to any of the proposed use cases, where an overparametrized deep neural network model is employed.

6.4 Continual Machine Learning and Knowledge Accumulation

Contributing partners: IDEAS, NCBR

This research addresses the challenge of catastrophic forgetting in neural networks by developing methods for incremental learning without losing previously acquired knowledge. In zero-waste machine learning, we consider knowledge accumulation and continual learning as key aspects of training neural networks in an efficient way. In this stage of the project, we focus on exemplar-free class-incremental learning (EFCIL) as one of the most challenging settings of continual learning [69], [117], where the method cannot store any exemplars and during the inference the task is unknown, and the network needs to correctly classify the object as one of the encountered class during the continual learning session. In the last few years, multiple methods that focus mostly on catastrophic forgetting have been proposed. In our work, we focus more on efficient knowledge accumulation where the network can be trained from scratch, not from the already pre-trained network or from the first task that has most of the data. We proposed two different EFCIL methods: *Selective Ensemble of Experts for Continual Learning* (SEED) [140] and *Adversarial Drift Compensation* (ADF) [148]. Both tackle the problem of EFCIL and outperform current state-of-the-art methods in different ways in well-established class-incremental learning benchmarks.

6.4.1 Technical description of the work

We consider the exemplar-free class-incremental learning (EFCIL) setup where new classes emerge over time and we are not allowed to store samples from old classes. These classes come in different tasks, one task at a time, and the tasks contain a mutually exclusive set of classes. When training on task t, we have access to current dataset $D_t = \{X_t, Y_t\}$ with images X_t and labels Y_t. The main goal of EFCIL is to learn a model h that correctly classifies the data into classes encountered so far. We use $h_t(x) = \sigma(W_t f_t(x))$, where f_t is the feature extractor parameterized by θ_t learned in task t and W_t is weight matrix of the linear classifier with softmax function σ.

Multiple recent CIL methods that do not store exemplars rely on having a strong feature extractor from the beginning of incremental learning. This extractor is trained on the larger first task, which provides a substantial amount of data (i.e., 50% of all available classes) [63], [125], [139], or it starts from a large pre-trained model that remains unchanged [76], [123] that eliminates the problem of representational drift [85]. However, these methods perform poorly when little training data is available upfront. In Figure 15, we illustrate both CIL setups, with and without the more significant first task. The trend is evident when we have a lot of data in the first task - results steadily improve over time. However, the progress is not evident for the setup with equal splits, where a frozen (or nearly frozen by high regularization) feature extractor does not yield good results. We focus on this more challenging setup as it requires the whole network to continually learn new features (*plasticity*) and face the problem of catastrophic forgetting of already learned ones (*stability*). We proposed two different methods: SEED [140] and ADF [148] that perform well in the small-start setting.

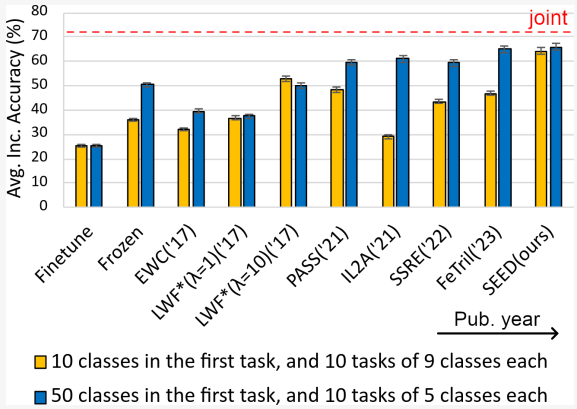

Figure 15. Exemplar-free Class Incremental Learning methods evaluated on CIFAR100 divided into eleven tasks for two different data distributions.

SEED Method. We introduce a novel ensembling method for exemplar-free CIL called *SEED: **S**election of **E**xperts for **E**nsemble **D**iversification*. Similarly to CoSCL and [113], SEED uses a fixed number of experts in the ensemble. However, only a single expert is updated while learning a new task. That, in turn, mitigates forgetting and encourages diversification between the experts. While only one expert is being trained, the others still participate in predictions. In SEED, the training does not require more computation than single-model solutions. The right expert for the update is selected based on the current ensemble state and new task data. The selection aims to limit representation drift for the classifier. The ensemble classifier uses multivariate Gaussian distribution representation associated with each expert (see Figure 16). At the inference time,

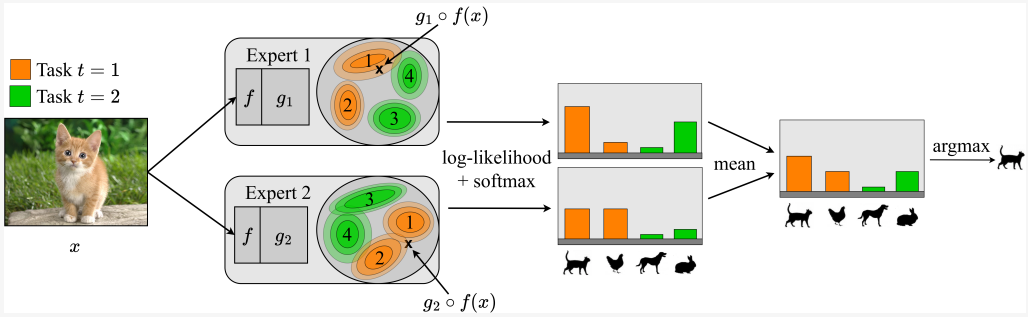

Figure 16. SEED comprises K deep network experts $g_k \circ f$ (here $K = 2$), sharing the initial layers f for higher computational performance. f are frozen after the first task. Each expert contains one Gaussian distribution per class $c \in C$ in his unique latent space. In this example, we consider four classes, classes 1 and 2 from task 1 and classes 3 and 4 from task 2. During inference, we generate latent representations of input x for each expert and calculate its log-likelihoods for distributions of all classes (for each expert separately). Then, we softmax those log-likelihoods and compute their average over all experts. The class with the highest average softmax is considered as the prediction.

Bayes classification from all the experts is used for a final prediction. As a result, SEED achieves state-of-the-art accuracy for task-aware and task-agnostic scenarios while maintaining the high plasticity of the resulting model under different data distribution shifts within tasks.

The core idea of our approach is to directly diversify experts by training them on different tasks and combining their knowledge during the inference. Each expert contains two components: a feature extractor that generates a unique latent space and a set of Gaussian distributions (one per class). The overlap of class distributions varies across different experts due to disparities in expert embeddings. SEED takes advantage of this diversity, considering it both during training and inference.

Architecture. Our approach, presented in Figure 16, consists of K deep network experts $g_k \circ f$ for $k = 1, \ldots, K$, sharing the initial layers f for improving computational performance. f are frozen after the first task. We consider the number of shared layers a hyperparameter (see Appendix **??**). Moreover, each expert k contains one Gaussian distribution $G_k^c = (\mu_k^c, \Sigma_k^c)$ per class c for its unique latent space.

Algorithm. During inference, we perform an ensemble of Bayes classifiers. The procedure is presented in Figure 16. Firstly, we generate representations of input x for each expert k as $r_k = g_k \circ f(x)$. Secondly, we calculate log-likelihoods of r_k for all distributions G_k^c associated with this expert

$$l_k^c(x) = -\frac{1}{2}[\ln(|\Sigma_k^c|) + S\ln(2\pi) + (r_k - \mu_k^c)^T(\Sigma_k^c)^{-1}(r_k - \mu_k^c)], \tag{23}$$

where S is the latent space dimension.

Then, we softmax those values $\widehat{l_k^1}, \ldots, \widehat{l_k^{|C|}} = \text{softmax}(l_k^1, \ldots, l_k^{|C|}; \tau)$ per each expert, where C is the set of classes and τ is a temperature. Class c with the highest average value after softmax over all experts (highest $\mathbb{E}_k \widehat{l_k^c}$) is returned as a prediction for task agnostic setup. For task aware inference, we limit this procedure to classes from the considered task.

Our training assumes T tasks, each corresponding to the non-overlapping set of classes $C_1 \cup C_2 \cup \cdots \cup C_T = C$ such that $C_t \cap C_s = \emptyset$ for $t \neq s$. Moreover, task t is a training step with only access to data $D_t = \{(x, y) | y \in C_t\}$, and the objective is to train a model performing well both for classes of a new task and classes of previously learned tasks ($< t$).

The main idea of training SEED, as presented in Figure 17, is to choose and finetune one expert for each task, where the chosen expert should correspond to latent space where distributions of

new classes overlap the least. Intuitively, this strategy causes latent space to change as little as possible, improving stability.

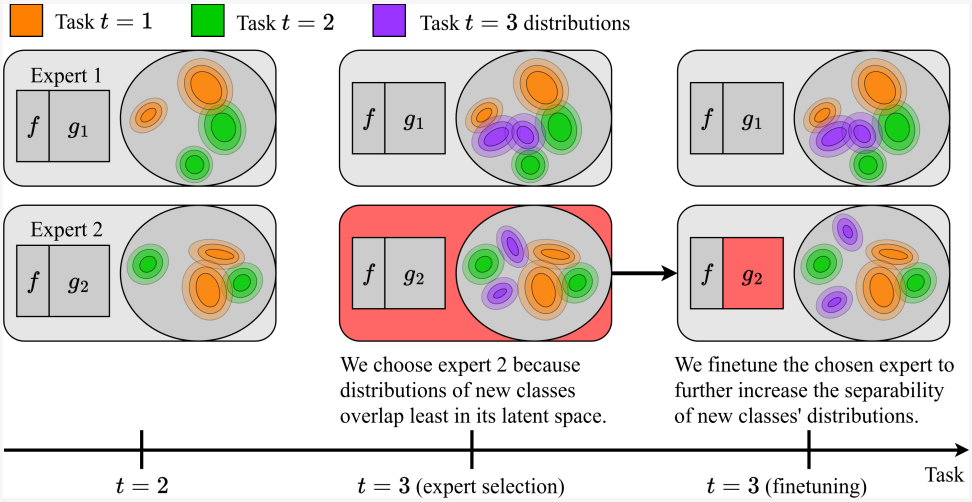

Figure 17. SEED training process for $K = 2$ experts, $T = 3$ tasks, and $|C_t| = 2$ classes per task. When the third task appears with novel classes C_3, we analyze distributions of C_3 classes (here represented as purple distributions) in latent spaces of all experts. We choose the expert where those distributions overlap least (here, expert 2). We finetune this expert to increase the separability of new classes further and move to the next task.

To formally describe our training, let us assume that we are in the moment of training when we have access to data $D_t = \{(x, y)|y \in C_t\}$ of task t for which we want to finetune the model. There are two steps to take, selecting the optimal expert for task t and finetuning this expert.

Expert selection starts with determining the distribution for each class $c \in C_t$ in each expert k. For this purpose, we pass all x from D_t with $y = c$ through deep network $g_k \circ f$. This results in a set of vectors in latent space for which we approximate a multivariate Gaussian distribution $q_{c,k}$. In consequence, each expert is associated with a set $Q_k = \{q_{1,k}, q_{2,k}, ..., q_{|C_t|,k}\}$ of $|C_t|$ distributions. We select expert $\bar{k}$ for which those distributions overlap least using symmetrized Kullback–Leibler divergence d_{KL}:

$$\bar{k} = \underset{k}{\operatorname{argmax}} \sum_{q_{i,k}, q_{j,k} \in Q_k} d_{KL}(q_{i,k}, q_{j,k}). \tag{24}$$

To finetune the selected expert $\bar{k}$, we add the linear head to its deep network and train $g_{\bar{k}}$ using D_t set. As a loss function, we use cross-entropy combined with feature regularization based on knowledge distillation [35] weighted with α: $L = (1-\alpha)L_{CE} + \alpha L_{KD}$, where $\mathcal{L}_{KD} = \frac{1}{|B|} \sum_{i \in B} ||g_{\bar{k}} \circ f(x_i) - g^{old}_{\bar{k}} \circ f(x_i)||$, B is a batch and $g^{old}_{\bar{k}}$ is frozen $g_{\bar{k}}$.

While we use CE for its simplicity and effective clustering [62], it can be replaced with other training objectives, such as self-supervision. Then, we remove the linear head, update distributions of $Q_{\bar{k}}$, and move to the next task.

Due to the random expert initializations, we skip the selection procedure for K initial tasks and omit L_{KD}. Instead, we select the expert with the same number as the number task ($k = t$) and use $L = L_{CE}$. For the same reason, we calculate distributions of new tasks only for the experts trained so far ($k \leq t$). Finally, we fix f after the first task so that finetuning one expert does not affect others.

Tab. 7 presents the comparison of SEED and state-of-the-art exemplar-free CIL methods for CIFAR-100, DomainNet, and ImageNet-Subset in the equal split scenario. We report average

incremental accuracies for various split conditions and domain shift scenarios (DomainNet). We present joint training as an upper bound for the CL training.

SEED outperforms other methods by a large margin in each setting. For CIFAR-100, SEED is better than the second-best method by 14.7, 17.5, and 15.6 percentage points for $T = 10, 20, 50$, respectively. The difference in results increases as there are more tasks in the setting. More precisely, for $T = 10$, SEED has 14.7 percentage points better accuracy than the second-best method (LwF*, which is LwF implementation with PyCIL [105] data augmentations and learning rate schedule). At the same time, for $T = 50$ SEED is better by 15.6%. The results are consistent for other datasets, proving that SEED achieves state-of-the-art results in an equal split scenario. Moreover, based on DomainNet results, we conclude that SEED is also better in scenarios with a significant distributional shift.

Table 7. Task-agnostic avg. inc. accuracy (%) for equally split tasks on CIFAR-100, DomainNet and ImageNet-Subset. The best results are in bold. SEED achieves superior results compared to other methods and outperforms the second best method (FeTrIL) by a large margin.

CIL Method	CIFAR-100 (ResNet32)			DomainNet			ImageNet-Subset
	T=10	T=20	T=50	T=12	T=24	T=36	T=10
Finetune	26.4±0.1	17.1±0.1	9.4±0.1	17.9±0.3	14.8±0.1	10.9±0.2	27.4±0.4
EWC [42] (PNAS'17)	37.8±0.8	21.0±0.1	9.2±0.5	19.2±0.2	15.7±0.1	11.1±0.3	29.8±0.3
LwF* [49] (CVPR'17)	47.0±0.2	38.5±0.2	18.9±1.2	20.9±0.2	15.1±0.6	10.3±0.7	32.3±0.4
PASS [107] (CVPR'21)	37.8±1.1	24.5±1.0	19.3±1.7	25.9±0.5	23.1±0.5	9.8±0.3	-
IL2A [106] (NeurIPS'21)	43.5±0.3	28.3±1.7	16.4±0.9	20.7±0.5	18.2±0.4	16.2±0.4	-
SSRE [125] (CVPR'22)	44.2±0.6	32.1±0.9	21.5±1.8	33.2±0.7	24.0±1.0	22.1±0.7	45.0±0.5
FeTrIL [139] (WACV'23)	46.3±0.3	38.7±0.3	27.0±1.2	33.5±0.6	33.9±0.5	27.5±0.7	58.7±0.2
SEED	**61.7±0.4**	**56.2±0.3**	**42.6±1.4**	**45.0±0.2**	**44.9±0.2**	**39.2±0.3**	**67.8±0.3**
Joint		71.4±0.3		63.7±0.5	69.3±0.4	69.1±0.1	81.5±0.5

We present results for this setting in Tab. 8. For CIFAR-100, SEED is better than the best method (FeTrIL) by 4.6, 4.1, and 1.4 percentage points for $T = 6, 11, 21$, respectively. For $T = 6$ on ImageNet-Subset, SEED is better by 3.3 percentage points than the best method. However, with more tasks, $T = 11$ or $T = 21$, FeTrIL with a frozen feature extractor presents better average incremental accuracy.

We can notice that simple regularization-based methods such as EWC and LwF* are far behind more recent ones: FeTrIL, SSRE, and PASS, which achieve high levels of overall average incremental accuracy. However, these methods benefit from a larger initial task, where a robust feature extractor can be trained before incremental steps. In SEED, each expert can still specialize for a different set of tasks and continually learn more diversified features even with using regularization like LwF. The difference between SEED and other methods is noticeably smaller in this scenario than in the equal split scenario. This fact proves that SEED works better in scenarios where a strong feature extractor must be trained from scratch or where there is a domain shift between tasks.

Figure 18 depicts the quality of each expert on various tasks and their respective contributions to the ensemble. It can be observed that experts specialize in tasks on which they were fine-tuned. For each task, there is always an expert who exhibits over 2.5% points better accuracy than the average of all experts. This demonstrates that experts specialize in different tasks. Additionally, the ensemble consistently achieves higher accuracy (ranging from 6% to 10% points) than the average of all experts on all tasks. Furthermore, the ensemble consistently outperforms the best individual expert, indicating that each expert contributes uniquely to the ensemble.

Table 8. Comparison of CIL methods on ResNet18 and CIFAR-100 or ImageNet-Subset under larger first task conditions. We report task-agnostic avg. inc. accuracy from multiple runs. The best result is in bold. The discrepancy in results between SEED and other methods decreases compared to the equal split scenario.

CIL Method	CIFAR-100			ImageNet-Subset		
	T=6 $\lvert C_1\rvert$=50	T=11 $\lvert C_1\rvert$=50	T=21 $\lvert C_1\rvert$=40	T=6 $\lvert C_1\rvert$=50	T=11 $\lvert C_1\rvert$=50	T=21 $\lvert C_1\rvert$=40
EWC* [42] (PNAS'17)	24.5	21.2	15.9	26.2	20.4	19.3
LwF* [49] (CVPR'17)	45.9	27.4	20.1	46.0	31.2	42.9
DeeSIL [50] (ECCVW'18)	60.0	50.6	38.1	67.9	60.1	50.5
MUC* [80] (ECCV'20)	49.4	30.2	21.3	-	35.1	-
SDC* [85] (CVPR'20)	56.8	57.0	58.9	-	61.2	-
ABD* [104] (ICCV'21)	63.8	62.5	57.4	-	-	-
PASS* [107] (CVPR'21)	63.5	61.8	58.1	64.4	61.8	51.3
IL2A* [106] (NeurIPS'21)	66.0	60.3	57.9	-	-	-
SSRE* [125] (CVPR'22)	65.9	65.0	61.7	-	67.7	-
FeTrIL* [139] (WACV'23)	66.3	65.2	61.5	72.2	**71.2**	**67.1**
SEED	**70.9±0.3**	**69.3±0.5**	**62.9±0.9**	**75.5±0.4**	70.9±0.5	63.0±0.8
Joint		80.4			81.5	

ADF Method. A critical aspect in EFCIL is the *semantic drift* of feature representations [86] after training on new tasks. This results in the movement of class distributions in feature space. Thus, it is crucial to track the old class representations after learning new tasks. While the class-mean in the new feature space can be effectively estimated using Nearest-Mean of Exemplars (NME) [48], [74], it is challenging to estimate it without exemplars. Usually, this drift is minimized with heavy functional regularization, which consequently restricts the plasticity of the network. Another way is to estimate it from the drift of current data, as done in SDC [86] or by augmenting old prototypes using new class features [138], [142]. In this paper, we propose a novel drift estimation method using adversarial examples to resurrect old class prototypes in the new feature space as shown in Fig. 19.

We present a novel and intuitive method - Adversarial Drift Compensation (ADC) to estimate semantic drift and resurrect old class prototypes in the new feature space. Exploiting the concept of targeted adversarial attacks [33], [56], we propose to perturb the new data such that the adversarial images result in embeddings close to the old prototypes. Now, the drift from old to new feature space is estimated using these adversarial samples, which serve as pseudo-exemplars for the old classes. We hypothesize that the pseudo-exemplars behave like the original exemplars in the feature space, and thus we exploit them to measure the drift. This generation of adversarial samples is computationally cheaper and much faster (only a few iterations) compared to data-inversion methods [84] which inverts embeddings to realistic images.

To estimate the drift of old class prototypes after updating the model on new classes, it is desirable to have the exemplars. These exemplars can be passed through the new model to compute the *oracle* prototype position in the new feature space. However, in the exemplar-free setting, we can only access the new data. In order to use the new data to represent the old data, we exploit the concept of targeted adversarial attacks [33], [56] to target one old class at a time and perturb the new data in a way that it serves as a substitute of old data to the model. We perform adversarial attacks on new data to move its embeddings very close to old prototypes in the old feature space.

To estimate the drift of prototype P^k_{t-1} for a target old class k, we obtain $\mathcal{X}^k$ by sampling a

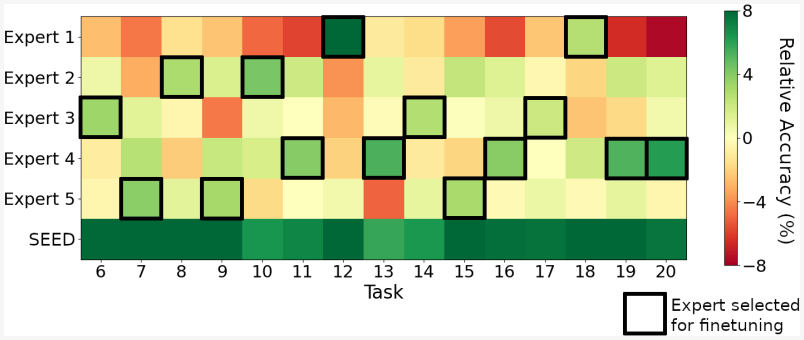

Figure 18. Diversity of experts on CIFAR-100 dataset with $T = 20$ split. The presented metric is relative accuracy (%) calculated by subtracting the accuracy of each expert from the averaged accuracy of all experts. Black squares represent experts selected to be finetuned on a given task. Although we do not impose any cost function associated with experts' diversity, they tend to specialize in different tasks by the design of our method. Moreover, our ensemble (bottom row) always performs better than the best expert, proving that each expert contributes uniquely to the ensemble in SEED.

set of m data points from the current task data X_t which are closest to P^k_{t-1} based on L2 distance between the embeddings of samples in X_t and the prototype P^k_{t-1}. We aim to perturb the samples $x \in \mathcal{X}^k$ and obtain $\mathcal{X}^k_{adv}$ such that the adversarial samples $x_{adv} \in \mathcal{X}^k_{adv}$ are closer to P^k_{t-1} and are now classified to class k using the NCM classifier in the old feature space:

$$k = \underset{y \in Y_{1:t-1}}{\arg\min} \|f_{t-1}(x_{adv}) - P^y_{t-1}\|_2. \tag{25}$$

We propose the following optimization objective by computing the mean squared error between the features $f_{t-1}(x)$ and the prototype P^k_{t-1} as:

$$L(f_{t-1}, \mathcal{X}^k, P^k_{t-1}) = \frac{1}{|\mathcal{X}^k|} \sum_{x \in \mathcal{X}^k} \|f_{t-1}(x) - P^k_{t-1}\|^2_2. \tag{26}$$

In order to move the feature embedding in the direction of the target prototype P^k_{t-1}, we obtain the gradient of the loss with respect to the data $x \in \mathcal{X}^k$, normalize it to get the unit attack vector and scale it by α as follows:

$$x_{adv} \leftarrow x - \alpha \frac{\nabla_x L(f_{t-1}, x, P^k_{t-1})}{\|\nabla_x L(f_{t-1}, x, P^k_{t-1})\|_2} \; \forall x \in \mathcal{X}^k, \tag{27}$$

where $\nabla_x L(f_{t-1}, x, P^k_{t-1})$ is the gradient of the objective function with respect to the data x and α refers to the step size. We perform the attack for i iterations.

Here, the goal is different from conventional adversarial attacks like FGSM and its variants [25], [33], [53], [56] which aim to minimize the perturbation in order to keep the perturbed image visually similar to the real image by having a fixed ϵ-budget generally based on ℓ_2 or ℓ_∞-norm of perturbation. In our case, we do not need to apply such restrictions on the distance between initial and final image, instead, we only clip the perturbed image in the existing range of pixel values. We show in supplementary materials that indeed the generated adversarial images have much higher perturbation. We do observe that our formulation is closer to the ℓ_2-norm based attack as we use ℓ_2 normalization of the gradient vector to obtain a unit perturbation vector which is scaled using the step size.

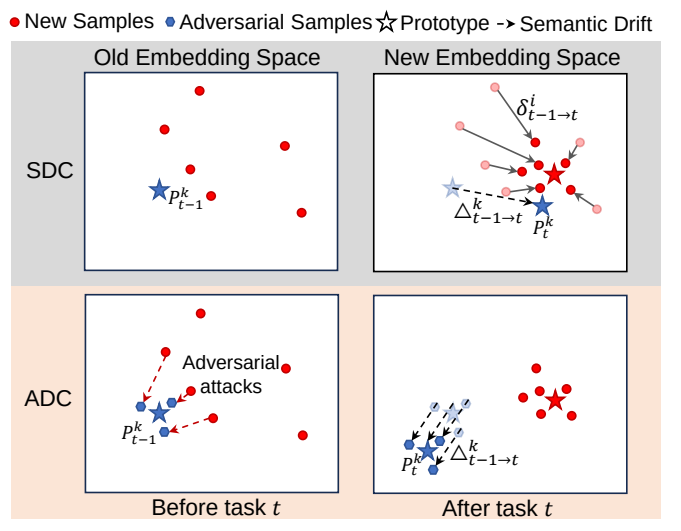

Figure 19. Illustration of Adversarial Drift Compensation (ADC) and SDC [86]. In SDC, the drift $\Delta^k_{t-1\to t}$ is estimated as the average of drift of all new task samples after training on a new task. Instead, we propose to move the new task features close to the old prototype P^k_{t-1} of class k by perturbing the new images using targeted adversarial attacks. The drift of the adversarial samples from old to new feature space is used to resurrect all old prototypes.

Drift Compensation. The adversarial samples when passed through the new feature extractor f_t are expected to lie close to the drifted prototype and hence are used to compute the drift. After generating the adversarial samples for each target class k, we measure the prototype drift as:

$$\Delta^k_{t-1\to t} = \frac{1}{|\mathcal{X}^k_{adv}|} \sum_{x_{adv}\in\mathcal{X}^k_{adv}} (f_t(x_{adv}) - f_{t-1}(x_{adv})), \tag{28}$$

where $x_{adv} \in \mathcal{X}^k_{adv}$ is the set of only those adversarial samples which are classified as the target class k using the NCM classifier. We resurrect the old prototypes by compensating the drift as follows:

$$P^k_t = P^k_{t-1} + \Delta^k_{t-1\to t}. \tag{29}$$

After compensating all old prototypes, we use the NCM classifier in the new feature space for classifying the test samples. Unlike SDC [86], we do not perform weighted averaging based on the distances to the prototype since embeddings from adversarial images are very close to the prototypes and we found no gain by applying this additional weighting scheme.

We observe that methods proposed for the big-start settings of EFCIL are not effective in small-start settings and perform poorly. A simple baseline trained with LwF and using NCM classifier is performing better than most of the existing approaches - SSRE, PASS, FeTrIL and FeCAM in several settings. While SDC improves over NCM, the proposed method ADC outperforms all existing methods in both last task accuracy and average incremental accuracy across all settings in Table 9 and Table 10. ADC outperforms the second-best method SDC by 4.2% on 5-task and by 5.12% on 10-task settings of CIFAR-100 on last-task accuracy. For TinyImageNet, ADC improves over the second-best method by 0.95% on 5-task and by 5.17% on 10-task settings. On ImageNet-Subset, ADC is better by 2.58% on 5-task and by 1.72% on 10-task settings after the last task.

Method	CIFAR-100				TinyImageNet				ImageNet-Subset			
	T = 5		T=10		T = 5		T =10		T = 5		T = 10	
	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
LwF [43]	45.35	61.94	26.14	46.14	38.81	49.70	27.42	38.77	50.88	69.11	37.90	61.60
NCM	53.53	66.35	41.31	57.85	38.69	50.45	26.56	41.04	57.74	71.99	45.86	65.04
SDC [86]	54.94	64.82	41.36	58.02	40.05	50.82	27.15	40.46	59.82	74.10	43.72	65.83
PASS [108]	49.75	63.39	37.78	52.18	36.44	48.64	26.58	38.65	50.96	66.15	38.90	54.74
SSRE [125]	42.39	56.57	29.44	44.38	30.13	43.20	22.48	34.93	40.30	57.57	28.12	45.87
FeTrIL [139]	45.11	60.42	36.69	52.11	29.91	43.99	23.88	36.35	49.18	63.83	40.26	55.12
FeCAM [132]	47.28	61.37	33.82	48.58	25.62	39.85	23.21	35.32	54.18	67.21	42.68	57.45
ADC (Ours)	**59.14**	**69.62**	**46.48**	**61.35**	**41.0**	**50.94**	**32.32**	**43.04**	**62.40**	**74.84**	**47.58**	**67.07**

*Table 9. Evaluation of EFCIL methods on small-start settings. Best results in **bold** and second best results are underlined.*

Method	CUB-200				Stanford Cars			
	T = 5		T=10		T = 7		T =14	
	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
LwF [43]	58.68	71.31	41.96	60.15	45.18	61.14	30.33	49.93
NCM	52.74	67.13	38.47	57.83	42.22	59.06	31.60	51.34
SDC [86]	55.20	68.64	41.63	60.43	45.03	61.75	32.15	53.18
PASS [108]	34.04	49.00	26.37	41.08	20.71	37.13	12.30	25.46
FeTrIL [139]	54.66	67.45	49.09	62.42	36.92	54.09	34.29	50.41
FeCAM [132]	53.47	66.39	51.78	64.97	40.64	56.24	37.50	52.78
ADC (Ours)	**64.46**	**73.49**	**57.97**	**68.91**	**54.86**	**67.07**	**45.07**	**61.39**

*Table 10. Evaluation of EFCIL methods on fine-grained datasets. Best results in **bold** and second best results are underlined.*

We also evaluate the EFCIL methods on the challenging fine-grained datasets of CUB-200 and Stanford Cars. We observe in Table 10 that LwF is a strong baseline here, particularly in the 5-task and 7-task settings and methods like NCM and SDC are not much better than LwF. While PASS performs poorly on both datasets, FeTrIL and FeCAM performs better with FeCAM outperforming the other methods on the 10-task setting of CUB-200 and 14-task setting of Stanford Cars. ADC outperforms the runner-up methods by 5.78% on 5-task setting and by 6.19% on 10-task settings of CUB-200. On Stanford Cars dataset, ADC is better by 9.68% on 7-task setting and 7.57 % on 14-task setting.

Drift estimation quality: We validate through Table 9 and Table 10 that the designed ADC method is giving better accuracy results than the previous SDC method for all datasets. As an additional verification, we check that this method was indeed better than SDC at estimating the old prototypes drift. To do so, we use both SDC and ADC on the same trained checkpoints on CIFAR-100 5-task settings and compare the estimated drift to the true drift computed using old data. We report the results in Fig. 20, where we show the distribution of the estimated drift qualities. One drift per class is estimated and we compute the cosine similarity of the estimated drift to the true drift. We see that for all training tasks, the drifts estimated with ADC are of better quality than the ones estimated with SDC. We observe that some class drift estimations with SDC have negative cosine similarity with the true drift. However, we also see that the estimation quality decreases slightly

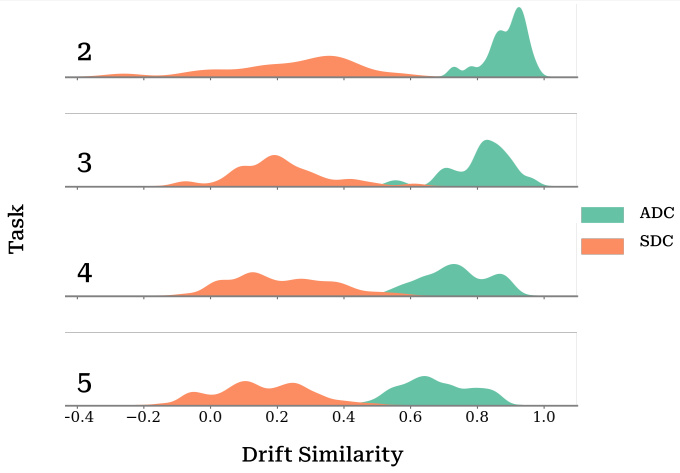

Figure 20. Comparison between the two drift estimation methods SDC [86], and the proposed ADC, on CIFAR-100 (5 tasks). We compute the drift for each class with the two methods and report the distribution of drift estimation quality, measured by computing the cosine similarity between the estimated drift vector and the true drift (obtained using old data), for all previous class prototypes.

for later training tasks. Indeed, as the backbone drifts more and more, it gets harder to estimate the actual drift. The fact that we see this decrease more prominently for ADC might be because the similarities obtained by SDC are already centered around a low-value (0.15) after the second task, whereas the better ADC drift estimation is centered first around 0.9, to then decrease and reach a minimum average of 0.7. This validates that ADC is able to track the movement of the prototypes in the feature space.
Using ADC requires some additional computation to be made in-between each training session. In this section, we provide an estimation of the additional computation required by our method and compare it to the training time of a single task. At the end of each task, our method requires estimating the drift of each stored prototype (1 per old class) and for each of these, compute several adversarial samples starting from available current task samples. As a consequence, the training time of our method scales linearly with the number of classes. For each class, we compute 100 adversarial samples in a single batch and perform 3 training iterations. In order to perform one iteration, we need to compute the gradient of the adversarial loss with respect to the input image, whose cost is equivalent to the one of a normal training backward pass [67]. So, if we denote the number of classes by N_c, and the number of iterations by N_i, we need to perform $N_c \times N_i$ backward passes. In the case of CIFAR-100 and ImageNet-Subset divided into 10 tasks each containing 10 classes, this means an overhead of, $\sum_{t=1}^{9} 10 \times t \times 3 = 1350$ backward passes. In contrast, one new task is trained for 100 epochs with a batch size of 128 (39 batches per epochs with 10 tasks on CIFAR-100), which amounts to 3900 backward passes per task, and two times more for the first task (trained for 200 epochs). In total, our method increases the computational cost by 3.1% in this setting. For the 5-task setting of CIFAR-100 and ImageNet-Subset, it increases by 2.5%.

6.4.2 Relevant publications

1. *Divide and not forget: Ensemble of selectively trained experts in Continual Learning*, Rypeść G., Cygert S., Khan V., Trzciński T., Zieliński B., Twardowski B., The Twelfth International Conference on Learning Representations (ICLR) 2024.

2. *Resurrecting Old Classes with New Data for Exemplar-Free Continual Learning*, Goswami D., Soutif-Cormerais A., Liu Y., Kamath S., Twardowski B., van de Weijer J., The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2024.

6.4.3 Relevant Software Releases / Datasets

- Code of SEED method, implemented in FACIL framework [117].
  - Available on GitHub: https://github.com/grypesc/SEED.
- Code of Adversarial Drift Compensation method implemented using PyCIL framework [146].
  - Available on GitHub: https://github.com/dipamgoswami/ADC.

6.4.4 Relevant Use Cases

The two proposed methods SEED and ADF can be easily applied in all image classification scenarios, where the number of classes grows through time (class-incremental), as well as in scenarios where the new samples are gathered through time (domain-incremental). Both methods will result in no need for training from scratch whenever new data emerge, and continue to accumulate the new knowledge with the same model.

6.5 Sustainable Computer Vision for Autonomous Machines

Contributing partners: IDEAS, NCBR

In recent years, the proliferation of data across various devices and the growing concerns over data privacy have spurred significant interest in Federated Learning (FL). Federated Learning enables multiple parties to collaboratively train machine learning models without the need to centralize their data, thereby preserving privacy and reducing the risk of data breaches. However, traditional Federated Learning methods often assume fully labeled datasets, which is not always practical due to the high cost and expertise required for accurate labeling. This has led to the exploration of Self-Supervised Learning (SSL) within the Federated Learning framework, where models learn useful representations from unlabeled data.

In this stage of our research, we focus on Split Federated Learning (SFL), a variant of Federated Learning that further enhances privacy and reduces computational overhead on client devices. SFL divides the model into two parts: one part is trained on the client devices, and the other part is trained on a central server. This division allows for more efficient use of computational resources and better protection of client data. [46], [134]. In conventional federated learning, communication mainly involves exchanging model parameters, whereas SFL also requires transferring activations and gradients for forward and backward propagation.

Traditional FL methods have shown considerable success in supervised learning tasks, but the assumption of fully labeled datasets is often impractical [46], [94]. Early attempts to integrate SSL with FL, such as FedU and FedEMA, have struggled with data diversity and privacy concerns, limiting their scalability. However, a notable example is the Momentum Contrastive Split Federated Learning (MocoSFL) method, which combines MoCo with SFL, has demonstrated the potential to scale SSL to highly distributed environments with up to 1000 clients [77], [134].

Despite the advancements, MocoSFL and similar methods face limitations, particularly in terms of communication overhead and privacy concerns. The communication of intermediate representations from low layers can increase the risk of data leakage and attacks such as Model Inversion Attack (MIA) [24]. Additionally, the size of these representations can lead to increased communication overhead, making the process less efficient [134] as shown in Figure 21.

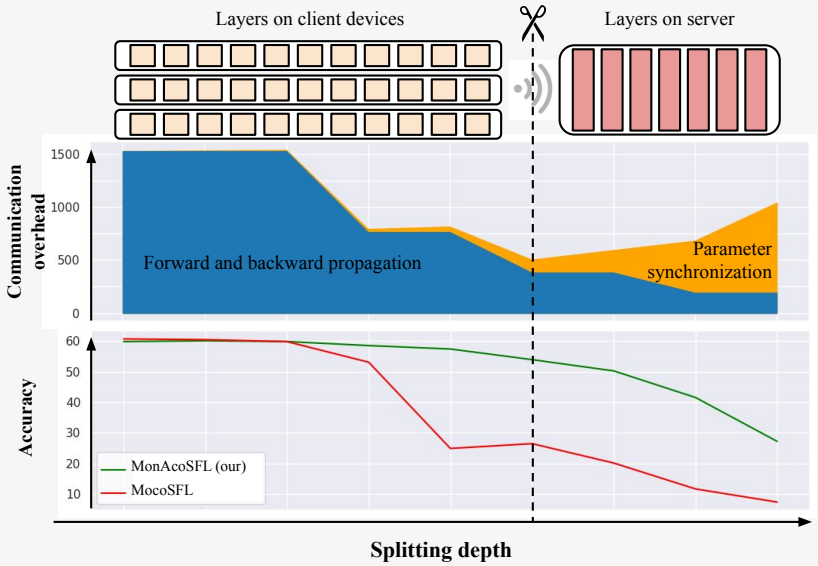

Figure 21. The relationship between communication overhead and accuracy for MocoSFL and MonAcoSFL across different splitting depths. In this scenario, the optimal communication overhead is achieved by splitting the model into 11 layers on the client side and 7 layers on the server side. Notably, MocoSFL experiences a significant drop in accuracy at this optimal split, whereas MonAcoSFL maintains high accuracy. The computational overhead is composed of forward and backward propagation (indicated in blue) and parameter synchronization (indicated in orange).

In our research paper, we delve into the relationship between communication overhead and the splitting point. We identify the optimal splitting point and highlight the poor performance of MocoSFL when aligned with it. As a remedy, we introduce Momentum-Aligned Contrastive Split Federated Learning (MonAcoSFL). In contrast to MocoSFL, which synchronizes only the online client models during the training, MonAcoSFL also synchronizes their momentum models. This change is crucial because it prevents the divergence of online and momentum models and reduces confusion during training [134].

6.5.1 Technical description of the work

SFL [46] is a practical variant of federated learning that divides the model into two parts: one part resides on the client devices and the other on the server. Formally, let N denote the number of participating clients and f_{ϕ_i} represent a copy of the model f_ϕ stored on the i-th client, parameterized by ϕ_i. The model f_ϕ is decomposed into two components, $f^s_{\phi^s} \circ f^c_{\phi^c}$, where $\phi = \phi^s \cup \phi^c$. Here, $f^c_{\phi^c}$ is distributed across the client devices, and $f^s_{\phi^s}$ is maintained on the centralized server. While there are multiple copies of $f^c_{\phi^c}$, there is only a single version of the parameters ϕ^s on the server.

Initially, all client models start with identical parameters ($\phi^c_1 = \phi^c_2 = \cdots = \phi^c_N$) and undergo training in two distinct phases:

1. Optimization of $\phi^c_1, ..., \phi^c_N, \phi^s$ with respect to the training objective $\mathcal{L}$.
2. Synchronization of $\phi^c_1, ..., \phi^c_N$ by updating each ϕ^c_i to the average parameter value $\hat{\phi}^c = \sum_{i=0}^{N} \frac{\phi^c_i}{N}$.

During the optimization phase, each client processes only its local data, which causes the parameters $\phi_1^c, ..., \phi_N^c$ to diverge over time. Therefore, the synchronization phase is essential to realign the client models.

SSL is a framework for learning data representations without the need for labeled data [110], [129]. The most common approaches in SSL today are joint-embedding architectures [72], [73], [75], [77], [92], [128], where the model f is trained by optimizing contrastive objectives. Specifically, let $\mathbf{x}'$ and $\mathbf{x}''$ be two augmented versions of a sample $\mathbf{x} \sim X$. The contrastive objectives aim to make the embeddings $f(\mathbf{x}')$ and $f(\mathbf{x}'')$ similar while preventing trivial solutions, such as producing identical embeddings for different data samples. To achieve this, most joint-embedding methods [72], [92] employ objective functions that require large batch sizes. The significant data requirements and computational overhead of contrastive objectives pose practical challenges for deploying SSL methods in highly distributed federated environments [109], [126], [134]. Momentum Contrastive Split Federated Learning (MocoSFL) [134] tackles the practical difficulties of implementing SSL methods in distributed settings by integrating SFL [121] with the Momentum Contrastive Learning (MoCo) [77] approach.

In MocoSFL, the contrastive learning objective is derived using the InfoNCE loss function [57], which leverages a memory bank of embeddings, denoted as M:

$$\mathcal{L}_{InfoNCE}(\mathbf{z}', \mathbf{z}'', M) = -\log \frac{\exp(\mathbf{z}' \cdot \mathbf{z}''/\tau)}{\exp(\mathbf{z}' \cdot \mathbf{z}''/\tau) + \sum_{\mathbf{z}_M \in M} \exp(\mathbf{z}' \cdot \mathbf{z}_M/\tau).} \tag{30}$$

Here, $\mathbf{z}' = f_\phi(\mathbf{x}')$ and $\mathbf{z}'' = f_{EMA(\phi)}(\mathbf{x}'')$ are the embeddings of two augmented versions of the same data sample $\mathbf{x}$, and $\mathbf{z}_M$ represents the embeddings stored in the memory bank M. The memory bank is updated with $\mathbf{z}''$ after each training step in a first-in-first-out manner. The parameters ϕ correspond to the *online* model, while $EMA(\phi)$, the exponential moving average of ϕ, corresponds to the *momentum* model. In the SFL setup, each client maintains its own set of parameters ϕ_i^c and $EMA(\phi_i^c)$, whereas the server holds a single set of parameters ϕ^s and $EMA(\phi^s)$. The memory bank M is also managed by the server.

MocoSFL stands out as the only SSL method capable of functioning effectively in a cross-client federated learning environment with over 100 clients, each contributing as few as 250 data samples from various distributions [134]. This capability is facilitated by a large memory bank of negative examples from all clients, which helps to mitigate the negative impact of small batch sizes on individual clients [72], [90] and reduces the likelihood of overfitting to any single client's data distribution [87]. Furthermore, by offloading the majority of the model layers and the contrastive objective to the server, the computational burden on the clients is significantly reduced [121].

MocoSFL limitations. Despite the success of MocoSFL, it has several limitations. We begin by outlining the limitations of MocoSFL, particularly focusing on the significant privacy concerns associated with sending representations from low layers. Privacy concerns caused by sending representations from low layers are illustrated in Figure 22. One can observe that representations of low ResNet18 [32] layers highly resemble the respective input data, in contrast to the activations from the higher layers. In fact, in principle, models that learn perceptive features (such as MoCo) do not retain reconstructive features in their high representations [72]. Thus, in SFL, increasing the number of layers on the client side reduces the privacy risks associated with broadcasting network representations.

Another limitation is the communication overhead. Mobile network architectures like ResNet [32] and MobileNet [58] reduce the spatial dimensions of representations while increasing their channel dimensions as layers progress. This results in smaller overall representation sizes at deeper layers, reducing bandwidth needs during the SFL optimization phase. However, having more layers on client devices means more parameters need to be exchanged during synchronization, increasing

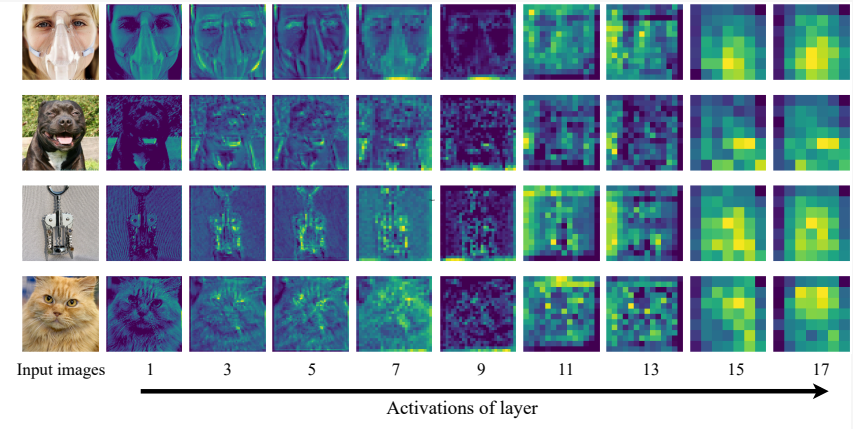

Figure 22. Activations obtained from successive layers of ResNet-18 for a sample ImageNet image. One can observe that representations of low layers highly resemble the respective input data, increasing the privacy risks associated with broadcasting network activations.

bandwidth requirements. Thus, there is a trade-off between these two types of communication overheads, with an optimal split point in the higher layers, as shown in Figure 24.

While deeper splits offer privacy and efficiency benefits, they also raise the question of how they impact MocoSFL's performance. To investigate this, we evaluated MocoSFL on the CIFAR-10 dataset [16] with 5, 20, and 200 clients, following the setup in [134]. As shown in Figure 25, we found that deeper splits lead to a significant drop in model accuracy.

MonAcoSFL. Initially, client models start with identical parameters $\phi_1^c = \phi_2^c = \cdots = \phi_N^c$. During training, these parameters diverge due to different local datasets, requiring periodic synchronization. MocoSFL synchronizes only the online client models, disrupting the MoCo assumption that online and momentum models encode similar representations, which is crucial for minimizing the contrastive objective [77]. This misalignment worsens with increased splitting depth, leading to performance degradation.

Algorithm. We introduce **Mo**men**t**um-**A**ligned **Co**ntrastive **SFL** as shown in Figure 23, which aligns online and momentum client models by synchronizing the momentum models whenever the online models are synchronized. Specifically, the momentum model of each client is updated as follows:

$$\widehat{EMA(\phi^c)} = \frac{\sum_{i=0}^{N} EMA(\phi_i^c)}{N}.$$

Since $EMA(\phi_1^c), \ldots, EMA(\phi_N^c)$ are the EMAs of the individual $\phi_1^c, \ldots, \phi_N^c$, their average corresponds to the EMA of the average online parameters $(\hat{\phi^c})$, i.e., $\widehat{EMA(\phi^c)} = EMA(\hat{\phi^c})$.

Hardware. We emulate the distributed setup on a single machine, utilizing an NVidia A100 GPU to host and execute both client and server models.

Architecture. Our experiments employ the mobile-optimized ResNet18 [32] and MobileNetV2 [58] architectures.

Data. We perform our experiments using the CIFAR-10 and CIFAR-100 datasets [16]. To simulate a realistic scenario, we distribute the data equally among clients, ensuring each client has access to only a small subset. We adopt a challenging non-IID setting, where data is not independently and identically distributed across clients. Specifically, each client receives images from a random selection of 2 classes for CIFAR-10 or 20 classes for CIFAR-100.

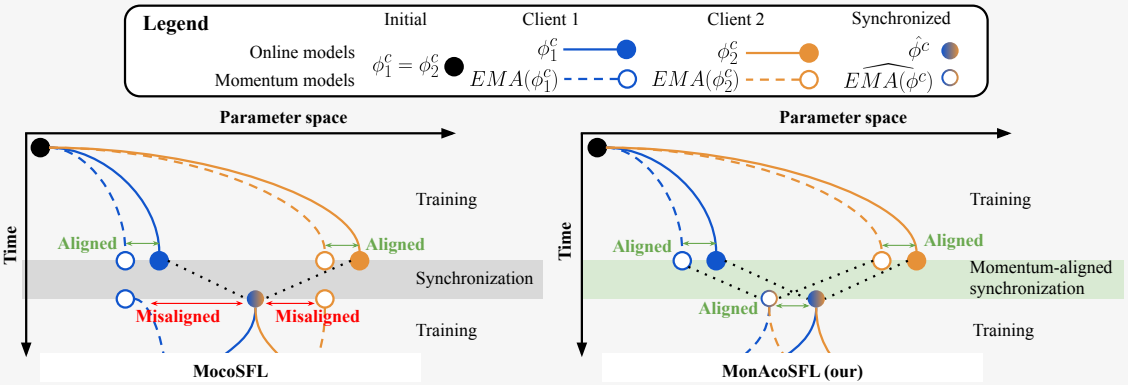

Figure 23. Visualization of parameters changing in MocoSFL (left) and MonAcoSFL (right) for two clients. The solid and dashed lines represent the progression of online and momentum parameters, respectively. The dotted lines symbolize the synchronization of parameters. The difference between MocoSFL and MonAcoSFL lies in the synchronization procedure that, in the case of MonAcoSFL, ensures that both online and the momentum models remain aligned, preserving their ability to optimize the contrastive objective.

Number of clients. We compare MocoSFL and MonAcoSFL in cross-silo (5 or 20 clients) and cross-client (200 clients) settings. Models are synchronized 10 times per epoch, equating to every 1000, 250, and 25 images for 5, 20, and 200 clients, respectively. We adjust batch sizes to keep the server-side batch size around 100. For example, in the 5-client setting, the batch size is 20, while in the 200-client setting, 100 clients are randomly selected per epoch with a batch size of 1.

SSL model. We employ MoCo-v2 [73], enhanced with a 2-layer MLP projector network with a hidden size of 1024 [72], [130], which is removed post-SSL pretraining. The server manages a FIFO queue memory of 6000 negative embeddings, updated with momentum model embeddings from the latest mini-batch after each training step. MocoSFL is trained for 200 epochs using the SGD optimizer, starting with a learning rate of 0.06, momentum of 0.9, and weight decay of 0.0005. The learning rate follows a cosine decay schedule throughout the training process.

Evaluation. After each epoch, we validate the model using k-NN on 20% of the validation set. This method is commonly used to assess self-supervised representations during training [98], [134]. We select the model with the highest k-NN performance for final evaluation. The final performance is determined using the linear evaluation protocol [72], [75], [98], [126], [134]. Specifically, we freeze the pretrained backbone, add a random linear layer, and train this layer on the labeled dataset for 100 epochs with a batch size of 128, using the Adam optimizer [41] with an initial learning rate of 0.001 and a cosine learning rate schedule.

Figure 26 illustrates the performance of MonAcoSFL and MocoSFL across different client configurations using the ResNet architecture. While both methods show a decline in accuracy with deeper splits, MonAcoSFL's accuracy drop is significantly less pronounced compared to MocoSFL. At the optimal communication-efficient split (layers 11-13), MonAcoSFL outperforms MocoSFL by over 30 percentage points. This advantage is even more evident with the MobileNet architecture, as shown in Figure 27. Specifically, with 20 clients, MonAcoSFL achieves over 40 percentage points higher accuracy than MocoSFL from the 3rd to the 15th cut layer on the CIFAR-10 dataset.

We evaluate the privacy-preserving features of MonAcoSFL and MoCoSFL using a Model Inversion Attack (MIA)[24]. Assuming the attacker has access to 1% of the training data, we train a decoder to reconstruct images from client model embeddings. The decoder is trained using the MSE loss and Adam optimizer with a learning rate of 0.001 over 50 epochs and a batch size of 32. The attack targets ResNet-18 models pretrained on the CIFAR-100 dataset with 20 clients

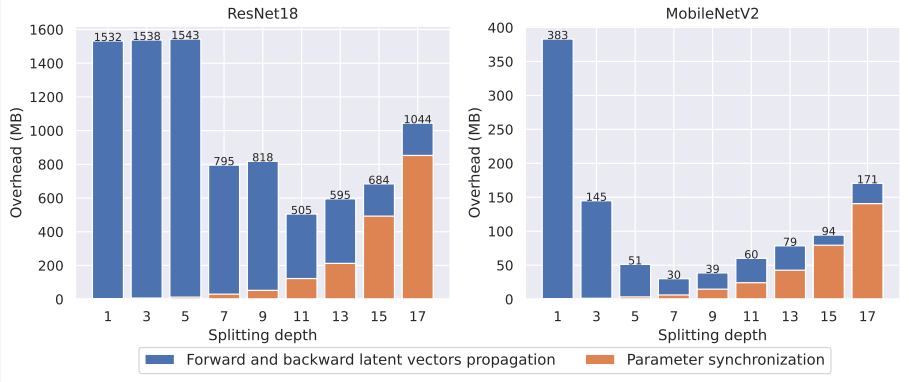

Figure 24. Communication overhead of a single client device for one training epoch of ResNet-18 [32] and MobileNet-V2 [58] for different splitting depths. The 11-th and 7-th layers are the most communication-efficient for ResNet-18 and MobileNetV2, respectively. Note that the training epoch corresponds to 250 images of resolution 224×224 are processed, and 10 synchronizations of parameters. Moreover, the blue bars correspond to communication in the optimization phase, and the orange bars correspond to parameter synchronization.

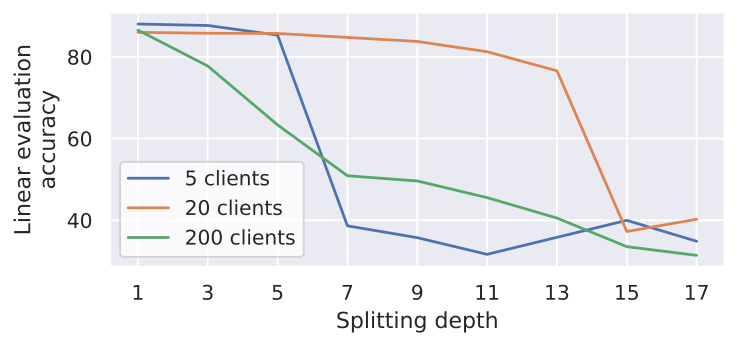

Figure 25. Accuracy of MocoSFL drops significantly with increased splitting depth regardless of the number of clients. Here, presented for CIFAR-10 [16].

and various cut-layers. No pre-training or additional privacy techniques like TAResSFL [134] are employed.

Figure 28 shows the comparison of original and reconstructed images in terms of MSE, where higher MSE values indicate better privacy. For cut-layers 1-13, both MocoSFL and MonAcoSFL exhibit similar reconstruction errors, with MonAcoSFL showing slightly higher errors. The MSE remains stable for layers 1-7 and increases for layers 9-17, suggesting better privacy for these deeper layers. In conclusion, deeper cut-layers (9-13) not only enhance computational efficiency but also improve client data protection.

To empirically confirm that maintains the alignment between online and momentum model parameters, we track this alignment during training. We quantify the average misalignment as the mean absolute difference between the online and momentum parameters, defined as:

$$\sum_{i=0}^{N} \frac{|\phi_i^c - EMA(\phi_i^c)|}{N \cdot dim(\phi^c)}, \tag{31}$$

where N represents the number of clients and $dim(\phi^c)$ denotes the dimensionality of the client

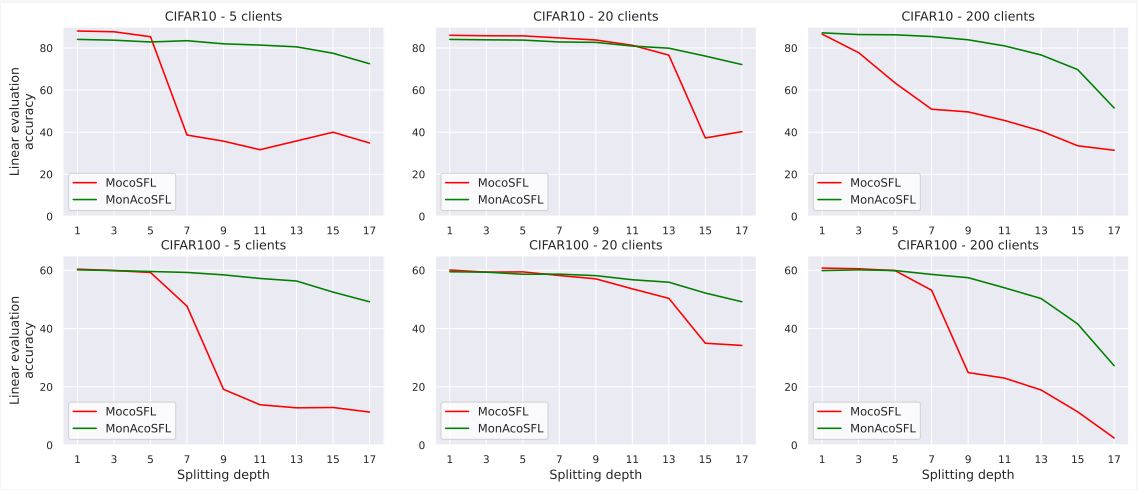

Figure 26. Linear evaluation of MocoSFL and MonAcoSFL on ResNet18 architecture. MonAcoSFL maintains the accuracy with increasing cut-layers, whereas the performance of MocoSFL rapidly deteriorates.

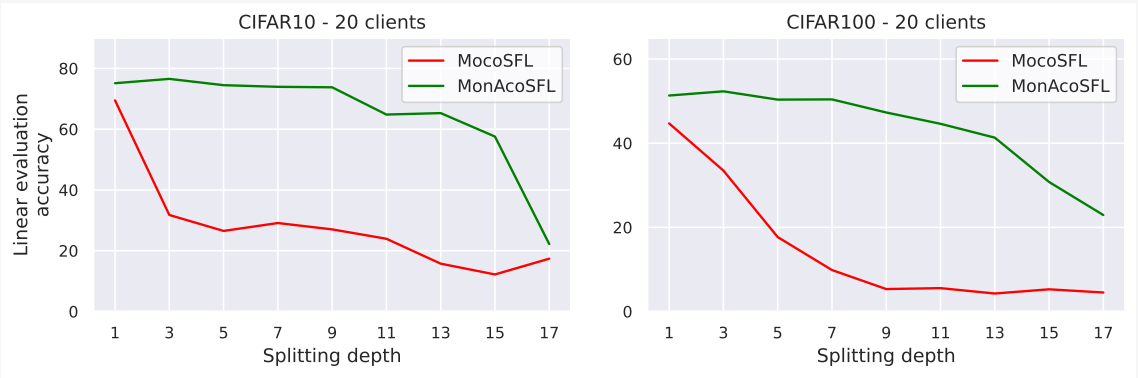

Figure 27. Linear evaluation accuracy on MobileNetV2 backbone trained with MocoSFL and MonAcoSFL. There is significant discrepancy between MonAcoSFL and MoCoSFL at every splitting depth.

model parameters.

We illustrate the misalignment values for the initial 1500 training steps (out of approximately 50000) of ResNet18 trained on CIFAR-100 by 20 clients, using MocoSFL and MonAcoSFL with a split at the 11th layer, in Figure 29. During the first 125 steps, both methods show similar misalignments. However, when parameters are synchronized, the misalignment between the online and momentum models in MocoSFL increases significantly. In contrast, MonAcoSFL maintains a consistent alignment between the momentum and online models throughout the training.

6.5.2 Relevant Publications

1. *A deep cut into Split Federated Self-Supervised Learning*; (Przewięźlikowski, M.; Osial, M.; Zieliński, B.; and Śmieja, M.; European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD),2024).

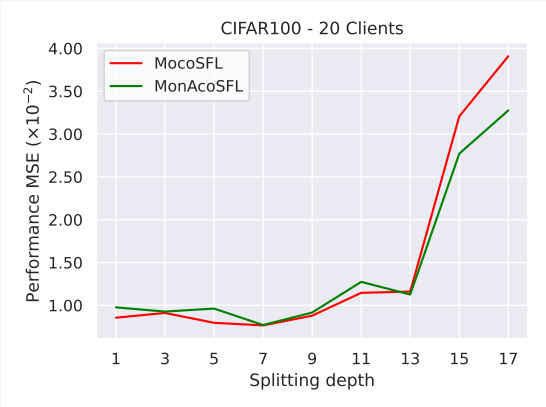

Figure 28. MSE of reconstructing the original images by the attacker. Higher MSE values for deeper cut-layers indicates a better resistance to attack (better privacy).

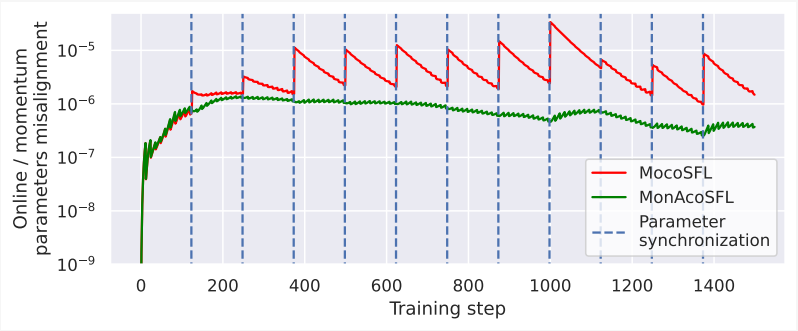

Figure 29. Average misalignment between online and momentum client models during the early training stages for MonAcoSFL and MocoSFL (lower values are preferable). Blue lines mark the parameter synchronization intervals (every 125 steps). In MocoSFL, misalignment spikes significantly during these intervals, while in MonAcoSFL, it stays relatively stable, leading to more consistent training.

6.5.3 Relevant Resources

- **Code of MonAcoSFL method**
  - Available on GitHub: https://github.com/gmum/MonAcoSFL

6.5.4 Relevant Use Cases

MonAcoSFL is particularly advantageous in scenarios that demand data privacy and efficient distributed communication. This framework is ideal for applications involving sensitive, unlabeled data that must remain decentralized, such as in the healthcare and finance sectors. The split model architecture of MonAcoSFL further enhances its utility by enabling deployment on resource-constrained devices, making it suitable for autonomous systems. For instance, drones used in wildlife protection, fire detection, and safety services can leverage MonAcoSFL to collaboratively learn from shared experiences without compromising the data privacy of of individuals, critical infrastructure, or private properties.

6.6 Selecting Images with Entropy for Frugal Knowledge Distillation

Contribution partner: CERTH

Knowledge Distillation (KD) is a mechanism aimed at reducing the size and complexity of these models without significantly sacrificing performance [95], i.e. aiming at frugal learning. In our work, we are interested in the frugality of KD in two ways. Our first objective is to produce an accurate and compressed student model. Additionally, our second objective is to ensure that the KD process itself is lightweight, meaning it should require minimal time, resources, and energy to complete. This is particularly important when the student model needs to be distilled in an edge computing environment rather than in a data center, which typically follows a centralized approach [65].

6.6.1 Technical Description

The main proposed idea involves utilizing entropy on image representations to identify image samples that make a greater contribution to the KD process, resulting in a student model with high predictive performance. The utilization of entropy in the image representations is the most important step in frugal KD workflow as depicted in Figure 30 and Figure 31.

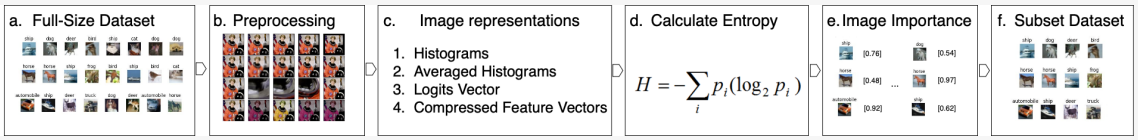

Figure 30. Subset of Dataset: Selecting Images with Entropy

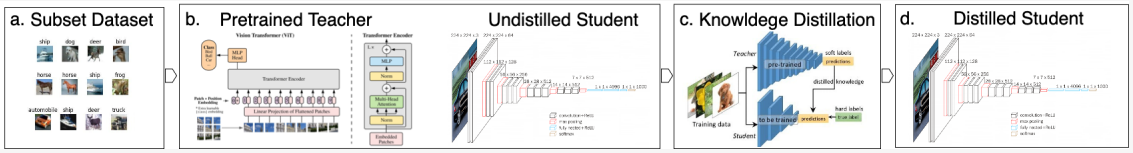

Figure 31. Knowledge Distillation with Image Selection

Image Representations. In the Image Representation step, we aim to generate a vector representation that encapsulates the unique insights and characteristics of each image. As we can see in Figure 30, step c, in these image vector representations we apply the cross entropy formula to provide a quantitative measure of the information content and complexity within the images. We have explored various image representations, each offering distinct characteristics for encoding visual content complexity. These include: 1) histograms, 2) adjusted histograms, 3) logits vectors, and 4) compressed feature vectors.

Entropy Formula. We use the entropy formula on the image representations as we can see in Figure 30 step d. Entropy in the context of image analysis, measures the uncertainty or disorder within the distribution of pixel intensities, latent representations, or other image features. A higher entropy value indicates greater complexity or information content within the image, while lower entropy suggests more predictability or uniformity. By computing entropy for each image representation in an image dataset, we can gauge its relative importance based on the diversity and wealth of information it contains.

The entropy function $H(X)$ is defined as:

$$H(X) = -\sum_{i=1}^{n} P(i) \log P(i),$$

where $P(i)$ denotes the probability of occurrence of each image feature i. The natural logarithm log amplifies the importance of rare features, while the summation process aggregates contributions from all possible features in the image.

Image Importance. Image importance is a measure of the significance or relevance of individual images in a dataset. As we can see in Figure 30 step e, we use an image importance criterion to prioritize images to be included in the subset. In our approach we quantify image importance through the calculation of entropy. Specifically, we use the entropy formula to assess the importance of images within a dataset, which comprises images categorized into a number of classes. It is important to ensure balanced representation across categories. Thus we select an equal number of images per class.

Output: Subset Dataset. After quantifying the importance of images within the dataset, our focus shifts to constructing a curated subset, as we can see in Figure 30 step f, optimized for efficient knowledge transfer in model distillation. Leveraging entropy-based image importance metrics, we employ a selection strategy aimed at capturing diverse visual patterns, while maintaining class balance. By prioritizing images with higher entropy scores indicative of greater complexity and information content, we curate an image subset that encapsulates the essential characteristics of the original dataset. This curated subset facilitates efficient knowledge extraction and transfer from the teacher to the student model. It is also the input to the KD process and the first step in the pipeline that is depicted in Figure 31.

Image Representations. We assess four distinct methodologies for representing images, each offering unique insights into their visual characteristics and content. These methodologies encompass histogram analysis, averaged adjusted histograms, logits vector extraction, and compressed feature vectors obtained through autoencoder-based encoding as depicted in Figure 32. By representing the images as vectors and applying the entropy formula to them, we assess the significance of each image for inclusion in the image subset.

Experimental Evaluation. The proposed model has been implemented and experimentally evaluated using Python 3 and the modules, OpenCV, argparse, NumPy, pandas, Scikit-learn, and PyTorch. The environment used for the evaluation was an Ubuntu Linux computer with an Intel i5-4670K CPU, 16 GBs of RAM at 2400 MHz and an ASUS RTX 2060 GPU. In our experimental evaluation, we also made experiments with Random Image Subset Selection [78], Highest Variance Criterion, and Manifold Learning-based Data Sampling [52]. Our experimental outcomes confirm that images with higher entropy contain more information and make a more substantial contribution to the KD process compared to those with lower entropy, random selection and other criteria such as highest variance and manifold learning-based sampling. Furthermore, experiments with different image representation methods shed light on their impact on model performance and efficiency with the average adjusted histograms method exhibited 3% better accuracy than any other method.

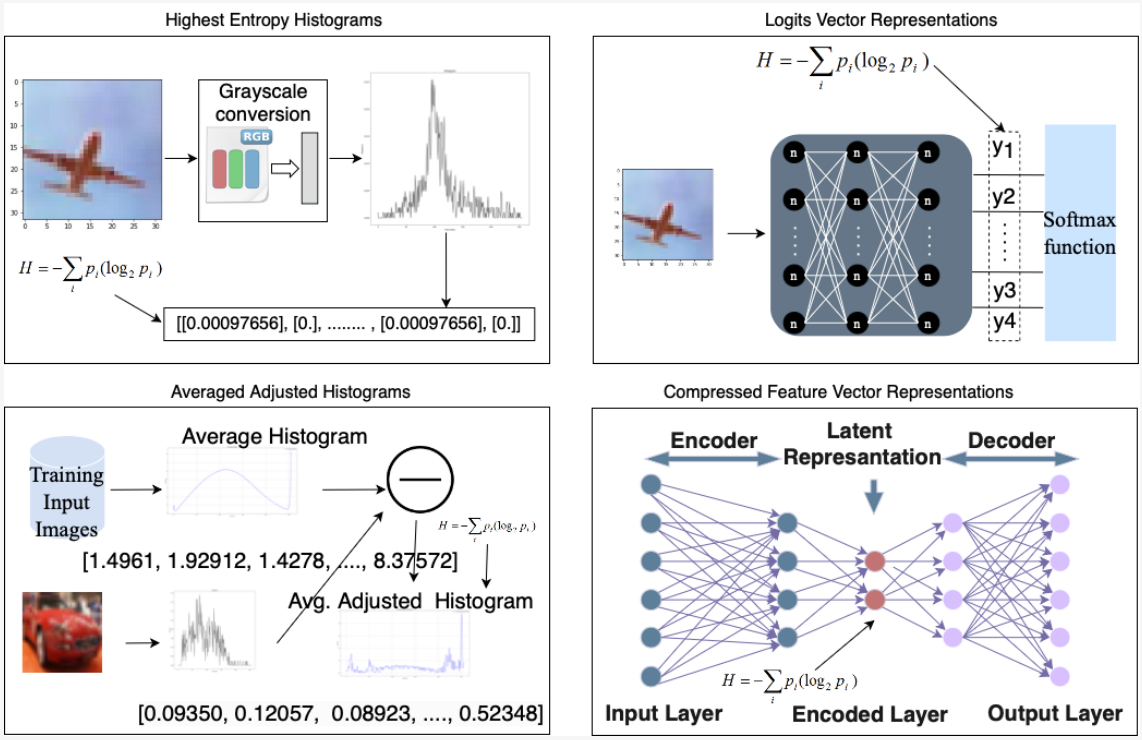

Figure 32. Image Representations

6.6.2 Relevant Publications

- An article submission, titled "Selecting Images with Entropy for Frugal Knowledge Distillation" by N. K. Karapiperis, J. Violos, M. Kinnas, S. Papadopoulos, and I. Kompatsiaris, is under preparation at the time of writing this deliverable.

6.6.3 Relevant Software Releases / Datasets

- The code will be made publicly available on GitHub once the paper is accepted at https://github.com/NikosKarapiperis/Entropy-Image-Selection-KD

6.6.4 Relevant Use Cases

The proposed methodology can potentially be applied to any use case within the ELIAS project where there is a need to compress a large deep learning model with minimal computational resources and KD time.

6.7 Reducing the Energy Requirements of Inference using two Heterogeneous CNNs

Contribution partner: CERTH

The goal of this research work is to propose a methodology based on the synergy of two small CNNs to achieve the performance of a large CNN while maintaining the energy efficiency of a small,

more compact model [27].

6.7.1 Technical Description

Our methodology encompasses two heterogeneous CNNs paired with a memory component as seen in Figure 33. Upon receiving an input for classification, the memory component first verifies whether a classification has been previously assigned to the particular input. If such a classification exists, the stored result is immediately retrieved without engaging any of the CNNs. In the event that no prior classification exists, the input undergoes processing by the initial CNN and a prediction confidence score is computed. This score is then compared against a predefined threshold. If the score surpasses this threshold, the classification is considered accurate. Conversely, the input is forwarded to the second CNN for classification. Following this, a new prediction confidence score is calculated for the second CNN. A final decision regarding the more accurate classification is determined by comparing these scores and the outcome is saved in the memory component.

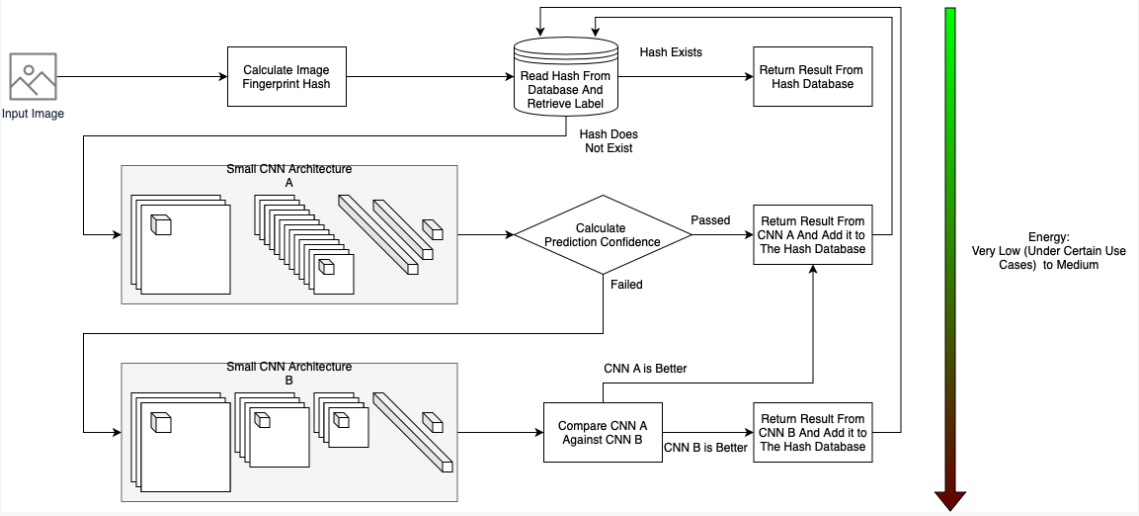

Figure 33. Two heterogeneous small CNN's with a memory component.

Two Heterogeneous CNNs. Our methodology incorporates two compact CNN architectures, allowing them to complement each other's deficiencies. In instances of classification ambiguity, the utilization of an alternative CNN mitigates such limitations, given that each CNN captures unique facets of the dataset's information. Opting for two small models, as opposed to a mixture of large and small ones, further minimizes power consumption. In order to quantify the heterogeneity of a selected pair of CNNs models A and B of similar size and complexity, we define the heterogeneity factor as given in equation 32 where A and B are the number of true predictions of models A and B and N the number of inputs.

$$heterogeneity(A, B) = \frac{A \cup B - 2 \cdot A \cap B}{N} \tag{32}$$

Score Functions. The CNNs in a classification problem produce a logits vector $\vec{z}$, which, upon passing each value of the vector through a softmax function (equation 33), is converted into a probability distribution vector $\vec{p}$ where i and j are the i-th and j-th element of the logits vector.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{33}$$

Each dimension of this vector corresponds to a class, with the value indicating the probability that the input belongs to that class. We then operate on this probability vector applying Max Probability, Difference score, or Entropy score function as presented in [55]:

Score Comparison. The calculated score value of the selected scoring function is utilized in two distinct manners. First, it is employed to compare the score value of the initial neural network CNN against a predetermined threshold. This comparison determines whether to trigger the subsequent neural network. Secondly, subsequent to the invocation of the second CNN, a second score is computed. The two scores are juxtaposed against each other, and the prediction of the CNN with the highest (or lowest, if the entropy score function is applied) score is employed.

Threshold hyper-parameter. The threshold hyperparameter is a fixed value that determines the extent of usage of the second CNN. We select the value that maximizes the accuracy while minimizing the use of the second CNN.

Memory Component. We incorporate a Memory component designed to reduce energy consumption during predictions under specific conditions. This component aims to recall whether a previous classification has been made for a given input, thereby bypassing the need to invoke the CNN when possible. To implement this we explored two Perceptual Hashing methods, the Difference Hash and the Invariants of Complex Moments [15].

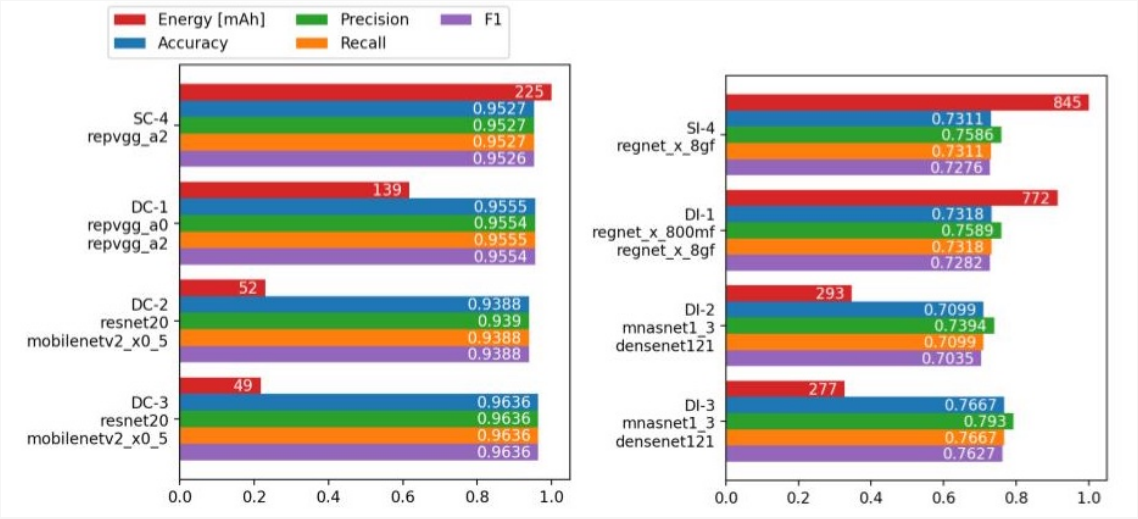

Figure 34. Experiment results for CIFAR10 (left) and ImageNet (right)

Experimental Evaluation. We conducted our experiments on a Jetson Nano computer powered through the USB at approximately 5.15V, and we used a USB power meter capable of measuring milliamps (mAh) and watt-hours (Wh) to the second decimal digit. The datasets used in our experiments are CIFAR-10 and ImageNet. Figure 34 illustrates results. From top to bottom: Single

large CNN (SC-4 and SI-4), state-of-the-art big-little [27] (DC-1 and DI-1), our implementation of two heterogeneous small CNNs (DC-2 and DI-2), our implementation using oracle score function (DC-3 and DI-3).

Our proposed model with two heterogeneous DL models (DC-2) for the CIFAR-10 dataset used 76.9% less energy compared to the single large CNN (SC-4), with a 1.4% decrease in Top-1% prediction accuracy. Additionally, when compared to the big/little configuration (DC-1), our implementation consumed 62.6% less energy, with a 1.7% reduction in Top-1% prediction accuracy. For the ImageNet dataset, our proposed model with two heterogeneous DL models (DI-2) consumed 65.3% less energy than the single large CNN (SI-4), with a 2.1% decrease in Top-1% prediction accuracy. Furthermore, compared to the big/little configuration (DI-1), our implementation achieved a 62% reduction in energy consumption, with a 2.2% reduction in Top-1% prediction accuracy.

6.7.2 Relevant Publications

- An article submission, titled "Reducing the Energy Requirements of Inference using two Heterogeneous CNNs" by M. Kinnas, J. Violos, S. Papadopoulos, and I. Kompatsiaris, is under preparation at the time of writing this deliverable.

6.7.3 Relevant Software Releases / Datasets

- The code will be made publicly available on GitHub once the paper is accepted at https://github.com/michaelkinnas/Reducing-the-energy-requirements-of-inference-using-two-heterogeneous-CNNs

6.7.4 Relevant Use Cases

The proposed methodology involving two small heterogeneous CNNs can potentially be applied to any use case within the ELIAS project where image classification needs to occur on resource-constrained devices with minimal response times.

6.8 Energy-Efficient Gaussian Processes Using Low-Precision Arithmetic

Contributing partner: HPI

The widespread use of artificial intelligence requires finding energy-efficient paradigms for the field. We propose to reduce the energy consumption of Gaussian process regression using low-precision floating-point representations. Gaussian process regression is a probabilistic and data-efficient model. We explored how low-precision (similar core idea as pruning or quantization) representations impact the results of Gaussian process regression and how data set properties, implementation approach, model performance, and energy consumption interact.

6.8.1 Technical Description

We suggest using low-precision Gaussian process regression (GPR) as a means of decreasing the power consumption of this AI method. GPR is typically used for small data sets where the prediction of uncertainty is of key importance. Enhancing the efficiency of these routine tasks has the potential to generate significant overall power savings. We investigate the connection between Gaussian process regression, arbitrary low-precision utilization, and power consumption.

Gaussian process regression is a mature model and a powerful tool for regression. The model [9] has been widely adopted, as evidenced by its reception in the academic community and its inclusion in many libraries for industry use.

We propose low-precision Gaussian processes to reduce the power consumption of Gaussian process regression. Our low-precision approach can be directly utilized by many processors and GPUs that have inherent abilities to use smaller floating-point representations through SIMD or other native hardware implementations without the need for special hardware. However, low-precision floating-point representations can accumulate large round-off errors. Determining the appropriate low-precision representation is a non-trivial task, as the algorithm implementation, the chosen kernel, the specific data set, and the desired model performance interact with a specific numerical representation. It is not known what precision to use for reasonable model performance. Existing work from numerical linear algebra provides theory-guided upper error bounds of specific arithmetic operations and even subtasks in Gaussian process regression. However, the complexity of what precision delivers a reasonable model performance in an end-to-end perspective for Gaussian process regression with real data is only feasible through an empirical evaluation.

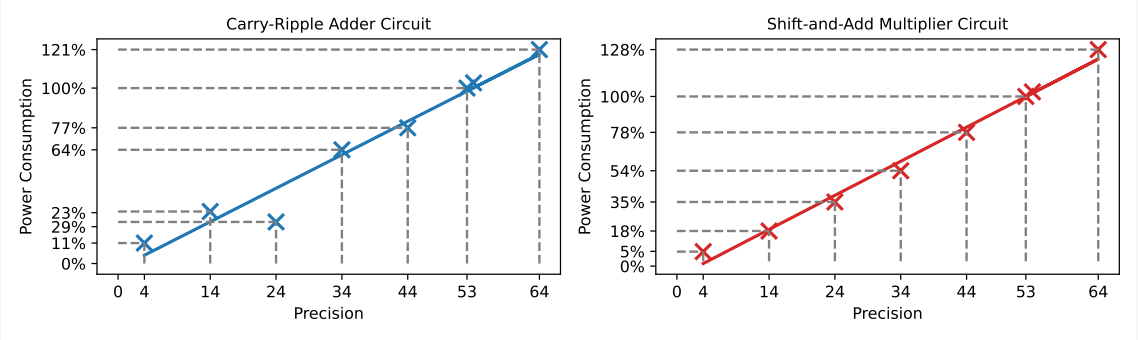

Figure 35. Results of the power consumption benchmark for addition and multiplication circuits relative to 53-bit double-precision, showing a linear power pattern in line with their circuit-size complexities. The power measurements were conducted on an FPGA-core.

To determine power consumption and potential savings in a generic and platform-independent manner, we focused on comparing relative reductions rather than absolute values. During our experiments, we counted the arithmetic operations that were performed while fitting the training data and during the inference phase. We then established a relationship between these operations and the power consumption of the corresponding arithmetic circuits on an FPGA, taking into account the precision used. To determine the extent of power savings, we performed a benchmark of the energy usage of simplified arithmetic circuits at different precision levels at the circuit level. Simplified refers to the implementation of integer-based circuits as they exhibit the same scaling properties as their complex IEEE floating-point counterparts. Given that the majority of operations in Gaussian process regression involve addition and multiplication or can be traced back to it, we focused on implementing these two operations on circuits and measuring their power consumption.

We implement a Carry-Ripple Adder circuit (by using a 2s complement, subtraction is equivalent to using an addition circuit). The state machines used in (non-)restoring division and add-and-shift multiplication have very similar designs and exhibit the same scaling properties. For simplicity, we assume that the power consumption of all other operations is independent of a numeric representation. Thus, we can assign any numerical operation in GPR to addition, multiplication, division, or being constant (not optimized). The energy reduction is the weighted mean of the power measurements for a representation, depending on the fraction of additions/subtractions and multiplications/divisions. Figure 35 displays the benchmarking results for power consumption for addition and multiplication. Both circuits show a linear power consumption pattern, which

is consistent with the expected behavior for circuits with linear size complexity. Anomalies in power consumption are observed in the 24-bit addition circuit, but these may be due to an efficient compiler mapping to our FPGA core, rather than a general rule. All other measurements for both circuits conform closely to the regression line.

	Data	Precision	Conditioning	Δ Train	Δ Test	Δ UC	% Operations	Δ Energy
Cholesky	fr	3 (4/5)	10 ± 2	0.12	0.04	−0.09		−98.30%
		8	$\mathbf{9 \pm 1}$	**0.01**	**−0.02**	**0.01**	**87.02%**	**−88.94%**
		53	9 ± 2	-	-	-		-
	pl	**14**	$\mathbf{25,093 \pm 39,926}$	**0.00**	**0.00**	**−0.01**	**83.27%**	**−75.16%**
		24 (4/5)	$66,100 \pm 107,969$	0.00	0.00	0.00		−71.09%
		53 (4/5)	$12,815 \pm 10,215$	-	-	-		-
	ch	24	$4,574,294 \pm 2,490,886$	0.00	0.05	−0.01		−70.30%
		34	$\mathbf{2,856,070 \pm 1,929,682}$	**0.00**	**0.01**	**−0.03**	**87.43%**	**−40.02%**
		53	$3,222,575 \pm 1,872,118$	-	-	-		-

	Data	Precision	Conditioning	Δ Train	Δ Test	Δ UC	% Operations	Δ Energy
Conjugate Gradient	fr	3	10 ± 3	0.50	0.03	−0.14		−98.38%
		8	$\mathbf{9 \pm 1}$	**0.01**	**0.00**	**0.00**	**97.99%**	**−89.01%**
		53	9 ± 1	-	-	-		-
	pl	4 (4/5)	$6,758 \pm 2,368$	4.17	3.34	−0.07		−91.20%
		8	$\mathbf{8,274 \pm 5,364}$	**0.04**	**0.00**	**−0.03**	**96.96%**	**−88.97%**
		53	$10,842 \pm 8,206$	-	-	-		-
	ch	4 (1/5)	$1,156,326 \pm 0$	144.37	144.82	0.59		−91.25%
		5 (4/5)	$3,483,705 \pm 1,927,678$	2.82	2.77	0.33		−94.64%
		14	$\mathbf{3,394,682 \pm 2,659,590}$	**−0.03**	**−0.02**	**0.01**	**98.08%**	**−75.99%**
		24 (4/5)	$3,647,790 \pm 2,357,889$	−0.03	0.00	−0.01		−70.19%
		34 (4/5)	$2,878,770 \pm 1,136,307$	−0.02	−0.01	0.00		−40.11%
		44	$2,526,487 \pm 1,514,300$	−0.03	0.01	−0.01		−22.22%
		53	$4,818,647 \pm 3,902,685$	-	-	-		-

Figure 36. The impact of low-precision numeric representations on energy consumption and model performance in Gaussian Process Regression with Cholesky decomposition and conjugate gradients implementation. The table includes selected results: lowest precision with stable computations, lowest precision with competitive (bold) performances (Δ UC, Train and Test RMSE) and double precision. Brackets in Precision column indicate the number of stable experiments to total experiments.

Figure 36 displays the combined results. For Gaussian process regression using Cholesky decomposition, we achieved a power reduction of up to 88.94% for 83.27% to 87.43% of all operations compared to using double-precision floating-point representations. When using conjugate gradients, we achieved a decrease of up to 89.01% in energy consumption for 96.96% to 98.08% of all operations. Using low-precision representations leads to less than ±0.02 deviation in root mean squared error of the test set and less than ±0.04 deviation on the train set. The uncertainty calibration changes by less than ±0.03.

6.8.2 Relevant Publications

- Alder, N., Herbrich, R. Energy-Efficient Gaussian Processes Using Low-Precision Arithmetic. In Forty-first International Conference on Machine Learning (2024).

6.8.3 Relevant Software Releases / Datasets

- The source code for "Energy-Efficient Gaussian Processes Using Low-Precision Arithmetic" is publicly available at https://github.com/nicolas-alder/energy-efficient-gps.

6.8.4 Relevant Use Cases

The proposed methodology can be applied to any use case within the ELIAS project where Gaussian process regression is applicable or algorithms perform matrix inversions. Furthermore, our research suggests that low-precision arithmetic is a reasonable overall approach for reducing power consumption as long as the quality of outputs meets the individual use case requirements.

References

[1] R. Evans, "The nature of the liquid-vapour interface and other topics in the statistical mechanics of non-uniform, classical fluids," *Adv. Phys.*, vol. 28, no. 2, pp. 143–200, 1979, ISSN: 14606976. DOI: 10.1080/00018737900101365.

[2] R. M. Neal, *Bayesian Learning for Neural Networks* (Lecture Notes in Statistics), 1st ed. New York, NY: Springer, 1996, vol. 118.

[3] D. Schläpfer, C. C. Borel, J. Keller, and K. I. Itten, "Atmospheric precorrected differential absorption technique to retrieve columnar water vapor," *Remote Sensing of Environment*, vol. 65, no. 3, pp. 353–366, 1998, ISSN: 0034-4257. DOI: https://doi.org/10.1016/S0034-4257(98)00044-3.

[4] M. Kennedy and A. O'Hagan, "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] H. Hansen-Goos and R. Roth, "Density functional theory for hard-sphere mixtures: The White Bear version mark II," *J. Phys. Condens. Matter*, vol. 18, no. 37, pp. 8413–8425, Sep. 2006, ISSN: 0953-8984, 1361-648X. DOI: 10.1088/0953-8984/18/37/002. (visited on 10/22/2022).

[7] A. O'Hagan, "Bayesian analysis of computer code outputs: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 10-11, pp. 1290–1300, 2006.

[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York: The MIT Press, 2006.

[9] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* (Adaptive Computation and Machine Learning). The MIT Press, 2006.

[10] R. Roth, "Introduction to Density Functional Theory of Classical Systems: Theory and Applications," *Lecture Notes*, 2006.

[11] A. T. Cohen, G. Agnelli, F. A. Anderson, J. I. Arcelus, D. Bergqvist, J. G. Brecht, I. A. Greer, J. A. Heit, J. L. Hutchinson, A. K. Kakkar, *et al.*, "Venous thromboembolism (vte) in europe," *Thrombosis and haemostasis*, vol. 98, no. 10, pp. 756–764, 2007.

[12] L. Gomez-Chova, G. Camps-Valls, J. Calpe-Maravilla, L. Guanter, and J. Moreno, "Cloud-screening algorithm for envisat/meris multispectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4105–4118, 2007.

[13] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer New York, 2008.

[14] M. B. Ashcroft, L. A. Chisholm, and K. O. French, "Climate change at the landscape scale: Predicting fine-grained spatial heterogeneity in warming and potential refugia for vegetation," *Global Change Biology*, vol. 15, no. 3, pp. 656–667, 2009.

[15] J. Flusser, B. Zitova, and T. Suk, *Moments and moment invariants in pattern recognition*. John Wiley & Sons, 2009.

[16] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *J. Fluid Mech.*, vol. 641, pp. 115–127, 2009.

[18] G. Camps-Valls, J. Mooij, and B. Schölkopf, "Remote sensing feature selection by kernel dependence measures," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 3, pp. 587–591, 2010. DOI: http://dx.doi.org/10.1109/LGRS.2010.2041896.

[19] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index," *Computer Physics Communications*, vol. 181, no. 2, pp. 259–270, 2010.

[20] K. E. Kocher, W. J. Meurer, R. Fazel, P. A. Scott, H. M. Krumholz, and B. K. Nallamothu, "National trends in use of computed tomography in the emergency department," *Annals of emergency medicine*, vol. 58, no. 5, pp. 452–462, 2011.

[21] S. A. Oldham *et al.*, "Ctpa as the gold standard for the diagnosis of pulmonary embolism," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, no. 4, pp. 557–563, 2011.

[22] J. Verrelst, L. Alonso, G. Camps-Valls, J. Delegido, and J. Moreno, "Retrieval of vegetation biophysical parameters using Gaussian process techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 5 PART 2, pp. 1832–1843, 2012.

[23] G. E. Raskob, P. Angchaisuksiri, A. N. Blanco, H. Buller, A. Gallus, B. J. Hunt, E. M. Hylek, A. Kakkar, S. V. Konstantinides, M. McCumber, *et al.*, "Thrombosis: A major contributor to global disease burden," *Arteriosclerosis, thrombosis, and vascular biology*, vol. 34, no. 11, pp. 2363–2371, 2014.

[24] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICML)*, 2015.

[26] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *NeurIPS*, Curran Associates, Inc., 2015.

[27] E. Park, D. Kim, S. Kim, Y.-D. Kim, G. Kim, S. Yoon, and S. Yoo, "Big/little deep neural network for ultra low power inference," in *2015 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Oct. 2015, pp. 124–132. DOI: 10.1109/CODESISSS.2015.7331375. [Online]. Available: https://ieeexplore.ieee.org/document/7331375 (visited on 04/19/2024).

[28] G. Tramontana, K. Ichii, G. Camps-Valls, E. Tomelleri, and D. Papale, "Uncertainty analysis of gross primary production upscaling using Random Forests, remote sensing and eddy covariance data," *Remote Sensing of Environment*, vol. 168, pp. 360–373, 2015. DOI: http://dx.doi.org/10.1016/j.rse.2015.07.015.

[29] G. Camps-Valls, J. Verrelst, J. Muñoz-Marí, V. Laparra, F. Mateo-Jiménez, and J. Gómez-Dans, "A survey on Gaussian processes for Earth Observation Data Analysis," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, 2016.

[30] M. Edelmann and R. Roth, "A numerical efficient way to minimize classical density functional theory," *J. Chem. Phys.*, vol. 144, no. 7, p. 074105, Feb. 2016, ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.4942020. (visited on 03/09/2024).

[31] J. L. Gómez-Dans, P. E. Lewis, and M. Disney, "Efficient emulation of radiative transfer codes using gaussian processes and application to land surface parameter inferences," *Remote Sensing*, vol. 8, no. 2, p. 119, 2016.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[33] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations (ICLR)*, 2016.

[34] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition*. Society for Industrial and Applied Mathematics, 2016.

[35] Z. Li and D. Hoiem, "Learning without forgetting," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, ser. Lecture Notes in Computer Science, vol. 9908, 2016, pp. 614–629.

[36] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *ECCV*, Springer, 2016.

[37] J. Verrelst, J. Rivera, A. Gitelson, J. Delegido, J. Moreno, and G. Camps-Valls, "Spectral Band Selection for Vegetation Properties Retrieval using Gaussian Processes Regression," *International Journal of Applied Earth Observation and Geoinformation*, vol. 52, pp. 554–567, 2016. DOI: http://dx.doi.org/10.1016/j.jag.2016.07.016.

[38] A. M. Wendelboe and G. E. Raskob, "Global burden of thrombosis: Epidemiologic aspects," *Circulation research*, vol. 118, no. 9, pp. 1340–1347, 2016.

[39] A. Berk and F. Hawes, "Validation of MODTRAN6 and its line-by-line algorithm," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 203, pp. 542–556, 2017.

[40] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," *arXiv preprint arXiv:1712.00409*, 2017.

[41] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[42] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[43] Z. Li and D. Hoiem, "Learning without forgetting," *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2017.

[44] M. Main-Knorn, B. Pflug, J. Louis, V. Debaecker, U. Müller-Wilm, and F. Gascon, "Sen2Cor for Sentinel-2," in *Image and Signal Processing for Remote Sensing XXIII*, vol. 10427, SPIE, Oct. 2017, p. 3.

[45] J. Mairhofer and J. Gross, "Numerical aspects of classical density functional theory for one-dimensional vapor-liquid interfaces," *Fluid Phase Equilibria*, vol. 444, pp. 1–12, Jul. 2017, ISSN: 03783812. DOI: 10.1016/j.fluid.2017.03.023. (visited on 03/09/2024).

[46] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., ser. Proceedings of Machine Learning Research, vol. 54, PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html.

[47] L. M. Prevedello, B. S. Erdal, J. L. Ryu, K. J. Little, M. Demirer, S. Qian, and R. D. White, "Automated critical test findings identification and online notification system using artificial intelligence in imaging," *Radiology*, vol. 285, no. 3, pp. 923–931, 2017.

[48] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "Icarl: Incremental classifier and representation learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[49] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "Icarl: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 5533–5542.

[50] E. Belouadah and A. Popescu, "Deesil: Deep-shallow incremental learning," *TaskCV Workshop ECCV 2018.*, 2018.

[51] G. Camps-Valls, D. Svendsen, L. Martino, J. Muñoz-Marí, V. Laparra, M. Campos-Taberner, and D. Luengo, "Physics-aware Gaussian processes in remote sensing," *Applied Soft Computing*, vol. 68, pp. 69–82, Jul. 2018. DOI: https://doi.org/10.1016/j.asoc.2018.03.021.

[52] S. Chen, S. Dorn, M. Lell, M. Kachelrieß, and A. Maier, "Manifold Learning-based Data Sampling for Model Training," de, in *Bildverarbeitung für die Medizin 2018*, Springer Vieweg, Berlin, Heidelberg, 2018, pp. 269–274, ISBN: 978-3-662-56537-7. DOI: $10.1007/978-3-662-56537-7_70$. [Online]. Available: $https://link.springer.com/chapter/10.1007/978-3-662-56537-7_70$ (visited on 05/13/2024).

[53] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] G. Heinze, C. Wallisch, and D. Dunkler, "Variable selection - a review and recommendations for the practicing statistician," *Biometrical journal*, vol. 60, no. 3, pp. 431–449, 2018.

[55] N. K. Jayakodi, A. Chatterjee, W. Choi, J. R. Doppa, and P. P. Pande, "Trading-off accuracy and energy of deep inference on embedded systems: A co-design approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2881–2893, Nov. 2018, ISSN: 1937-4151. DOI: 10.1109/tcad.2018.2857338. [Online]. Available: http://dx.doi.org/10.1109/TCAD.2018.2857338.

[56] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICML)*, 2018.

[57] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[58] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474.

[59] D. R. Thompson, V. Natraj, R. O. Green, M. C. Helmlinger, B.-C. Gao, and M. L. Eastwood, "Optimal estimation for imaging spectrometer atmospheric correction," *Remote Sensing of Environment*, vol. 216, pp. 355–373, 2018.

[60] H. M. Ali, M. S. Kaiser, and M. Mahmud, "Application of convolutional neural network in segmenting brain regions from mri data," in *International conference on brain informatics*, Springer, 2019, pp. 136–146.

[61] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González, *Deep Gaussian Processes for Multi-fidelity Modeling*, 2019. DOI: 10.48550/ARXIV.1903.07320.

[62] S. Horiguchi, D. Ikami, and K. Aizawa, "Significance of softmax-based features in comparison to distance metric learning-based features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 5, pp. 1279–1285, 2019.

[63] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 831–839.

[64] S. C. Lin and M. Oettel, "A classical density functional from machine learning and a convolutional neural network," *SciPost Phys.*, vol. 6, no. 2, pp. 1–13, 2019, ISSN: 25424653. DOI: 10.21468/SciPostPhys.6.2.025. arXiv: 1811.05728.

[65] J. Park, S. Wang, A. Elgabli, S. Oh, E. Jeong, H. Cha, H. Kim, S.-L. Kim, and M. Bennis, *Distilling On-Device Intelligence at the Network Edge*, arXiv:1908.05895 [cs, eess, math], Aug. 2019. DOI: 10.48550/arXiv.1908.05895. [Online]. Available: http://arxiv.org/abs/1908.05895 (visited on 05/24/2024).

[66] M. Reichstein, G. Camps-Valls, B. Stevens, J. Denzler, N. Carvalhais, M. Jung, and Prabhat, "Deep learning and process understanding for data-driven Earth System Science," *Nature*, vol. 566, pp. 195–204, Feb. 2019. DOI: https://doi.org/10.1038/s41586-019-0912-1.

[67] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[68] S. Soffer, A. Ben-Cohen, O. Shimon, M. M. Amitai, H. Greenspan, and E. Klang, "Convolutional neural networks for radiologic images: A radiologist's guide," *Radiology*, vol. 290, no. 3, pp. 590–606, 2019.

[69] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.

[70] J. Verrelst, Z. Malenovsky, C. Van der Tol, G. Camps-Valls, J.-P. Gastellu-Etchegorry, P. Lewis, P. North, and J. Moreno, "Quantifying vegetation biophysical variables from imaging spectroscopy data: A review on retrieval methods," *Surveys in Geophysics*, vol. 40, no. 3, pp. 589–629, 2019.

[71] H. West, N. Quinn, and M. Horswell, "Remote sensing for drought monitoring & impact assessment: Progress, past challenges and future opportunities," *Remote Sensing of Environment*, vol. 232, p. 111 291, 2019.

[72] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 13–18 Jul 2020, pp. 1597–1607. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html.

[73] X. Chen, H. Fan, R. B. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *CoRR*, vol. abs/2003.04297, 2020. arXiv: 2003.04297. [Online]. Available: https://arxiv.org/abs/2003.04297.

[74] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *Computer vision-ECCV 2020-16th European conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, Springer, vol. 12365, 2020, pp. 86–102.

[75] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu koray, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 21 271–21 284. [Online]. Available: https://proceedings.neurips.cc/paper%5C_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[76] T. L. Hayes and C. Kanan, "Lifelong machine learning with deep streaming linear discriminant analysis," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2020.

[77] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[78] J. M. Johnson and T. M. Khoshgoftaar, "The Effects of Data Sampling with Deep Learning and Highly Imbalanced Big Data," en, *Information Systems Frontiers*, vol. 22, no. 5, pp. 1113–1131, Oct. 2020, ISSN: 1572-9419. DOI: 10.1007/s10796-020-10022-7. [Online]. Available: https://doi.org/10.1007/s10796-020-10022-7 (visited on 05/29/2024).

[79] K. Keller, L. Hobohm, M. Ebner, K.-P. Kresoja, T. Münzel, S. V. Konstantinides, and M. Lankeit, "Trends in thrombolytic treatment and outcomes of acute pulmonary embolism in germany," *European heart journal*, vol. 41, no. 4, pp. 522–529, 2020.

[80] Y. Liu, S. Parisot, G. Slabaugh, X. Jia, A. Leonardis, and T. Tuytelaars, "More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning," in *European Conference on Computer Vision*, Springer, 2020, pp. 699–716.

[81] M. Polsinelli, L. Cinque, and G. Placidi, "A light cnn for detecting covid-19 from ct scans of the chest," *Pattern recognition letters*, vol. 140, pp. 95–100, 2020.

[82] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[83] J. Vicent, J. Verrelst, N. Sabater, L. Alonso, J. P. Rivera-Caicedo, L. Martino, J. Muñoz-Marí, and J. Moreno, "Comparative analysis of atmospheric radiative transfer models using the Atmospheric Look-up table Generator (ALG) toolbox (version 2.0).," *Geoscientific Model Development*, vol. 13, no. 4, 2020.

[84] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[85] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. van de Weijer, "Semantic drift compensation for class-incremental learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, IEEE, 2020, pp. 6980–6989.

[86] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, "Semantic drift compensation for class-incremental learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[87] F. Zhang, K. Kuang, Z. You, T. Shen, J. Xiao, Y. Zhang, C. Wu, Y. Zhuang, and X. Li, *Federated unsupervised representation learning*, 2020. arXiv: 2010.08982 [cs.LG].

[88] A. del Águila and D. S. Efremenko, "Fast hyper-spectral radiative transfer model based on the double cluster low-streams regression method," *Remote Sensing*, vol. 13, no. 3, 2021. DOI: 10.3390/rs13030434.

[89] A. Bastos, R. Orth, M. Reichstein, P. Ciais, N. Viovy, S. Zaehle, P. Anthoni, A. Arneth, P. Gentine, E. Joetzjer, *et al.*, "Vulnerability of european ecosystems to two compound dry and hot summers in 2018 and 2019," *Earth system dynamics*, vol. 12, no. 4, pp. 1015–1035, 2021.

[90] A. Bulat, E. Sánchez-Lozano, and G. Tzimiropoulos, "Improving memory banks for unsupervised learning with large mini-batch, consistency and hard negative mining," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1695–1699. DOI: 10.1109/ICASSP39728.2021.9414389.

[91] G. Camps-Valls, D. Tuia, X. Zhu, and M. Reichstein, *Deep learning for the Earth Sciences: A comprehensive approach to remote sensing, climate science and geosciences*. Wiley & Sons, 2021, ISBN: 9781119646143.

[92] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[93] P. Cats, S. Kuipers, S. De Wind, R. Van Damme, G. M. Coli, M. Dijkstra, and R. Van Roij, "Machine-learning free-energy functionals using density profiles from simulations," *APL Mater.*, vol. 9, no. 3, 2021, ISSN: 2166532X. DOI: 10.1063/5.0042558. arXiv: 2101.01942.

[94] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 15 750–15 758.

[95] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," en, *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021, ISSN: 1573-1405. DOI: 10.1007/s11263-021-01453-z. [Online]. Available: https://doi.org/10.1007/s11263-021-01453-z (visited on 05/24/2024).

[96] T. T. Ho, T. Kim, W. J. Kim, C. H. Lee, K. J. Chae, S. H. Bak, S. O. Kwon, G. Y. Jin, E.-K. Park, and S. Choi, "A 3d-cnn model with ct-based parametric response mapping for classifying copd subjects," *Scientific Reports*, vol. 11, no. 1, pp. 1–12, 2021.

[97] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, and et al., "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[98] H. Lee, K. Lee, K. Lee, H. Lee, and J. Shin, "Improving transferability of representations via augmentation-aware self-supervision," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 17 710–17 722. [Online]. Available: https://proceedings.neurips.cc/paper%5C_files/paper/2021/file/94130ea17023c4837f0dcdda95034b65-Paper.pdf.

[99] A. Li, R. B. Perez, S. Wiggin, S. C. Ward, P. A. Wood, and D. Fairen-Jimenez, "The launch of a freely accessible MOF CIF collection from the CSD," *Matter*, vol. 4, no. 4, pp. 1105–1106, Apr. 2021, ISSN: 2590-2393, 2590-2385. DOI: 10.1016/j.matt.2021.03.006. (visited on 06/26/2024).

[100] L. Martino and J. Read, "A joint introduction to Gaussian Processes and Relevance Vector Machines with connections to Kalman filtering and other kernel smoothers," *Information Fusion*, vol. 74, pp. 17–38, 2021.

[101] H. Meyer and E. Pebesma, "Predicting into unknown space? estimating the area of applicability of spatial prediction models," *Methods in Ecology and Evolution*, vol. 12, no. 9, pp. 1620–1633, 2021.

[102] S. Mishra and R. Molinaro, "Physics informed neural networks for simulating radiative transfer," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 270, p. 107 705, 2021, ISSN: 0022-4073.

[103] C. Requena-Mesa, V. Benson, M. Reichstein, J. Runge, and J. Denzler, "Earthnet2021: A large-scale dataset and challenge for earth surface forecasting as a guided video prediction task.," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1132–1142.

[104] J. S. Smith, Y. Hsu, J. Balloch, Y. Shen, H. Jin, and Z. Kira, "Always be dreaming: A new approach for data-free class-incremental learning," in *IEEE/CVF International Conference on Computer Vision, ICCV 2021*, IEEE, 2021, pp. 9354–9364.

[105] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan, *Pycil: A python toolbox for class-incremental learning*, 2021. arXiv: 2112.12533 [cs.LG].

[106] F. Zhu, Z. Cheng, X.-y. Zhang, and C.-l. Liu, "Class-incremental learning via dual augmentation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[107] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5871–5880.

[108] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[109] W. Zhuang, X. Gan, Y. Wen, S. Zhang, and S. Yi, "Collaborative unsupervised visual representation learning from decentralized data," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021, pp. 4912–4921.

[110] S. Albelwi, "Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging," *Entropy*, vol. 24, no. 4, 2022, ISSN: 1099-4300. DOI: 10.3390/e24040551. [Online]. Available: https://www.mdpi.com/1099-4300/24/4/551.

[111] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman Theory for Dynamical Systems," *SIAM Review*, vol. 64, no. 2, pp. 229–340, 2022.

[112] J. Cortés-Andrés, G. Camps-Valls, S. Sippel, E. Székely, D. Sejdinovic, E. Diaz, A. Pérez-Suay, Z. Li, M. Mahecha, and M. Reichstein, "Physics-aware Nonparametric Regression Models for Earth Data Analysis," *Environmental Research Letters*, vol. 17, no. 5, 2022. DOI: https://doi.org/10.1088/1748-9326/ac6762.

[113] T. Doan, S. I. Mirzadeh, and M. Farajtabar, "Continual learning beyond a single model," *arXiv preprint arXiv:2202.09826*, 2022.

[114] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *CVPR*, 2022.

[115] "How artificial intelligence improves radiological interpretation in suspected pulmonary embolism," *European Radiology*, pp. 1–12, Mar. 2022, ISSN: 14321084. DOI: 10.1007/S00330-022-08645-2/FIGURES/5. [Online]. Available: https://link.springer.com/article/10.1007/s00330-022-08645-2.

[116] I. Katsoularis, O. Fonseca-Rodríguez, P. Farrington, H. Jerndal, E. H. Lundevaller, M. Sund, K. Lindmark, and A. M. F. Connolly, "Risks of deep vein thrombosis, pulmonary embolism, and bleeding after covid-19: Nationwide self-controlled cases series and matched cohort study," *BMJ*, vol. 377, e069590, Apr. 2022, ISSN: 1756-1833. DOI: 10.1136/BMJ-2021-069590. [Online]. Available: https://www.bmj.com/content/377/bmj-2021-069590%20https://www.bmj.com/content/377/bmj-2021-069590.abstract.

[117] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022.

[118] R. S. Millan-Castillo, L. Martino, E. Morgado, and F. Llorente, "An exhaustive variable selection study for linear models of soundscape emotions: Rankings and Gibbs analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2460–2474, 2022.

[119] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *International Conference on Machine Learning*, PMLR, 2022, pp. 20 827–20 840.

[120] E. Tartaglione, A. Bragagnolo, A. Fiandrotti, and M. Grangetto, "Loss-based sensitivity regularization: Towards deep sparse neural networks," *Neural Networks*, 2022.

[121] C. Thapa, P. C. Mahawaga Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, pp. 8485–8493, Jun. 2022. DOI: 10.1609/aaai.v36i8.20825. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20825.

[122] J. Vicent Servera, J. P. Rivera-Caicedo, J. Verrelst, J. Muñoz-Marí, N. Sabater, B. Berthelot, G. Camps-Valls, and J. Moreno, "Systematic Assessment of MODTRAN Emulators for Atmospheric Correction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–17, 2022. DOI: 10.1109/TGRS.2021.3071376.

[123] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[124] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai, *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 795–813, 2022.

[125] K. Zhu, W. Zhai, Y. Cao, J. Luo, and Z.-J. Zha, "Self-sustaining representation expansion for non-exemplar class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9296–9305.

[126] W. Zhuang, Y. Wen, and S. Zhang, "Divergence-aware federated self-supervised learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=oVE1z8NlNe.

[127] C. H. Ali Mehmeti-Göpel and J. Disselhoff, "Nonlinear advantage: Trained networks might not be as complex as you think," in *ICML*, PMLR, 2023.

[128] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 15 619–15 629.

[129] R. Balestriero, M. Ibrahim, V. Sobal, A. S. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," *ArXiv*, vol. abs/2304.12210, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:258298825.

[130] F. Bordes, R. Balestriero, Q. Garrido, A. Bardes, and P. Vincent, "Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning," *Transactions on Machine Learning Research*, 2023, ISSN: 2835-8856. [Online]. Available: https://openreview.net/forum?id=ZgXfXSz51n.

[131] G. Camps-Valls, A. Gerhardus, U. Ninad, G. Varando, G. Martius, E. Balaguer-Ballester, R. Vinuesa, E. Diaz, L. Zanna, and J. Runge, "Discovering causal relations and equations from data," *Physics Reports*, vol. 1044, pp. 1–68, 2023, Discovering causal relations and equations from data.

[132] D. Goswami, Y. Liu, B. Twardowski, and J. van de Weijer, "FeCAM: Exploiting the heterogeneity of class distributions in exemplar-free continual learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[133] L.Martino, R. S. Millan-Castillo, and E. Morgado, "Spectral information criterion for automatic elbow detection," *Expert Systems with Applications*, vol. 231, p. 120 705, 2023.

[134] J. Li, L. Lyu, D. Iso, C. Chakrabarti, and M. Spranger, "MocoSFL: Enabling cross-client collaborative self-supervised learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=2QGJXyMNoPz.

[135] J. Li, P. Chen, Z. He, S. Yu, S. Liu, and J. Jia, "Rethinking out-of-distribution (ood) detection: Masked image modeling is all you need," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11 578–11 589.

[136] L. Li, J.-F. Wang, M. Franklin, Q. Yin, J. Wu, G. Camps-Valls, Z. Zhu, C. Wang, Y. Ge, and M. Reichstein, "Improving air quality assessment using physics-inspired deep graph learning," *npj Climate and Atmospheric Science*, 2023.

[137] Z. Liao, V. Quétu, V.-T. Nguyen, and E. Tartaglione, "Can unstructured pruning reduce the depth in deep neural networks?" In *ICCV*, 2023.

[138] T. Malepathirana, D. Senanayake, and S. Halgamuge, "Napa-vq: Neighborhood-aware prototype augmentation with vector quantization for continual learning," in *International Conference on Computer Vision (ICCV)*, 2023.

[139] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide, "Fetril: Feature translation for exemplar-free class-incremental learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 3911–3920.

[140] G. Rypeść, S. Cygert, V. Khan, T. Trzcinski, B. M. Zieliński, and B. Twardowski, "Divide and not forget: Ensemble of selectively trained experts in continual learning," in *The Twelfth International Conference on Learning Representations*, 2023.

[141] F. Sammüller, S. Hermann, D. De Las Heras, and M. Schmidt, "Neural functional theory for inhomogeneous fluids: Fundamentals and applications," *Proc. Natl. Acad. Sci. USA*, vol. 120, no. 50, e2312484120, Dec. 2023, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2312484120. (visited on 12/20/2023).

[142] W. Shi and M. Ye, "Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning," in *International Conference on Computer Vision (ICCV)*, 2023.

[143] E. d. A. Soares, A. G. Barreto Jr., and F. W. Tavares, *Classical Density Functional Theory Reveals Structural Information of H2 and CH4 Fluids Adsorbed in MOF-5*, Mar. 2023. arXiv: 2303.11384 [cond-mat, physics:physics]. (visited on 06/15/2023).

[144] D. Svendsen, D. Hernandez-Lobato, V. Laparra, L. Martino, A. Moreno-Martínez, and G. Camps-Valls, "Inference over Radiative Transfer Models using Variational and Expectation Maximization Methods," *Machine Learning*, vol. 112, pp. 921–937, Jun. 2023. DOI: https://doi.org/10.1007/s10994-021-05999-4.

[145] J. Vicent Servera, L. Martino, J. Verrelst, and G. Camps-Valls, "Multifidelity Gaussian Process Emulation for Atmospheric Radiative Transfer Models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–10, 2023.

[146] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan, "Pycil: A python toolbox for class-incremental learning," *SCIENCE CHINA Information Sciences*, 2023.

[147] J. Dijkman, M. Dijkstra, R. van Roij, M. Welling, J.-W. van de Meent, and B. Ensing, *Learning Neural Free-Energy Functionals with Pair-Correlation Matching*, May 2024. DOI: 10.48550/arXiv.2403.15007. arXiv: 2403.15007 [cond-mat].

[148] D. Goswami, A. Soutif-Cormerais, Y. Liu, S. Kamath, B. Twardowski, and J. van de Weijer, "Resurrecting old classes with new data for exemplar-free continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[149] Z. Liao, V. Quétu, V.-T. Nguyen, and E. Tartaglione, "Nepenthe: Entropy-based pruning as a neural network depth's reducer," *arXiv preprint arXiv:2404.16890*, 2024.

[150] V. Quétu and E. Tartaglione, "Dsd^2: Can we dodge sparse double descent and compress the neural network worry-free?" In *AAAI*, 2024.

[151] A. Simon, J. Weimar, G. Martius, and M. Oettel, "Machine Learning of a Density Functional for Anisotropic Patchy Particles," *J. Chem. Theory Comput.*, vol. 20, no. 3, pp. 1062–1077, Feb. 2024, ISSN: 1549-9618. DOI: 10.1021/acs.jctc.3c01238.

[152] J. Vicent Servera, L. Martino, J. Verrelst, J. P. Rivera-Caicedo, and G. Camps-Valls, "Multioutput feature selection for emulation and sensitivity analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–11, 2024. DOI: 10.1109/TGRS.2024.3358231.