Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

Research Article

Kolmogorov-Arnold Representation Based Informed Orchestration in Solving Partial Differential Equations for Carbon Balance Sustainability

Charles Z. Liu[1,2*], Farookh Hussain[1], Ying Zhang[1], Lu Qin[1]

[1] School of Computer Science, University of Technology Sydney, Sydney, Australia
[2] Aicloation, AlphaNest, Sydney, Australia
E-mail: charles.liu@aicloation.com

Received: 23 September 2024; **Revised:** 23 October 2024; **Accepted:** 31 October 2024

Abstract: This paper introduces the Kolmogorov-Arnold Credit Informed Network Orchestration (KACINO), a novel framework proposed to comprehensively structure and analyze the carbon dynamic system. By synthesizing dynamic processes such as carbon emission and sequestration, a credit information of carbon dynamics is accumulated. KACINO provides a holistic approach to model the complexities inherent in carbon dynamics. Through systematic analysis and scenario simulations, KACINO facilitates informed policy recommendations aimed at optimizing carbon credit utilization and achieving sustainable environmental outcomes. This paper outlines the theoretical foundation, methodology, and practical applications of KACINO, highlighting its potential to support transformative strategies in climate change mitigation and sustainable development.

Keywords: kolmogorov-arnold representation (KAR), partial differential equation, carbon informed neural networks (CINNs), carbon emission, carbon sequestration, carbon dynamic processes, sustainable development

MSC: 65L05, 34K06, 34K28

1. Introduction

Climate change poses a significant threat to global ecosystems and human societies, making the development of effective emission reduction solutions imperative [1]. In this context, carbon credits have emerged as a crucial tool [2, 3]. Each carbon credit represents the reduction of one metric ton of CO_2 or its equivalent in greenhouse gases, enabling organizations and individuals to offset their emissions by investing in emission reduction projects [4]. This market-based mechanism incentivizes greenhouse gas reduction initiatives, fostering investment in sustainable practices [5].

However, while carbon credits support the advancement of sustainable practices, they face scrutiny regarding verification accuracy and the risk of greenwashing [6, 7], a practice of misleading consumers regarding the environmental benefits of a product, service, or company practices. It often involves exaggerating or fabricating claims about sustainability to create a false impression of environmental responsibility. This can include vague language, unsubstantiated claims, or showcasing minor eco-friendly initiatives while ignoring larger, harmful practices [8]. Greenwashing poses significant risks, as it can undermine genuine sustainability efforts, mislead consumers, and lead to regulatory scrutiny. Thus, it is essential to establish rigorous standards and ensure transparency to mitigate these concerns [9].

DOI: https://doi.org/10.37256/cm.5420245775

Research into the dynamics of carbon emissions and sequestration is vital for enhancing verification accuracy and addressing greenwashing within the carbon credit system [10, 11]. By developing precise models and tools for tracking carbon fluxes, we can improve the accuracy of carbon credit assessments and minimize discrepancies between reported and actual emission reductions. Additionally, enhancing transparency through detailed reporting on the factors influencing carbon sequestration helps stakeholders verify that carbon credits represent genuine emission reductions [12–14].

The intricate relationship between carbon emissions and sequestration has complicated the creation of a comprehensive and accurate unified model. Despite considerable efforts to develop robust measurement models for these processes-covering emissions [15–17], sequestration [18–20], and integrated carbon emission-sequestration modeling [21, 22] current models remain relatively static, failing to adapt to new data or changing conditions. They often rely on linear numerical logic and assume constant emission and sequestration factors, overlooking the dynamic interactions and decay of these processes. Consequently, they fail to account for the fundamental physical laws and temporal variations inherent in carbon exchange systems, leading to inadequate predictions of long-term trends and policy effectiveness.

To address these limitations, the Physical Information Neural Network (PINN) offers significant advantages by integrating physical laws directly into the neural network training process. This integration enhances both the physical consistency and reliability of the model [23]. By incorporating physical constraints, PINNs can generate accurate predictions even when data is scarce, thus reducing the dependence on extensive datasets. They are particularly adept at solving complex scientific and engineering challenges that involve multi-physics interactions or nonlinear phenomena, naturally incorporating boundary and initial conditions [24].

Moreover, PINNs enhance computational efficiency by minimizing reliance on conventional numerical methods while simultaneously improving model interpretability. Because the predictions of PINNs align with physical laws, the outcomes remain consistent with fundamental principles [23]. By integrating physical knowledge into the training process, PINNs not only maintain statistical validity but also ensure physical relevance, making them a robust tool for addressing intricate problems [24].

Research on carbon credits, particularly regarding the dynamic interplay of various factors, remains limited. While there is a growing body of work on PINNs, most current studies focus on methodologies such as Gated Recurrent Units (GRUs) [25–28], Feedforward Neural Networks (FNNs) [29, 30], Residual Gain (REG, Residual Gain) Strategy [31–33], Deep Neural Networks (DNN) [34, 35], and Multilayer Perceptron (MLP) [36, 37]. However, a comparative analysis of these methodologies are notably lacking.

In our research, we explore PINNs and design a framework for Carbon Informed Neural Networks (CINNs) to address the challenges present in carbon physical systems. Our comparative studies reveal that these methods often necessitate more complex network structures, leading to longer training times to achieve convergence and accuracy, especially in intricate systems. To overcome these challenges, we have reviewed and contrasted these approaches and proposed a new framework that utilizes the Kolmogorov-Arnold Representation to organize data, thereby simplifying network architecture and potentially reducing training complexity.

In this paper, we present an orchestration framework with CINNs to synthesize diverse dynamics of carbon emissions and sequestration into a unified model, illustrated through a case study on Carbon Balance Dynamics. We introduce the Kolmogorov-Arnold Carbon Informed Network Orchestration (KACINO), which comprehensively considers decay rates and the interactions between carbon emissions and sequestration as an integrated system. KACINO dynamically orchestrates all necessary factors to reinforce the CINN framework, enhancing efficiency in modeling and analyzing carbon dynamics. This synthesis allows for a comprehensive understanding of how different emissions and sequestration processes interact within a single model, integrating various factors influencing carbon behavior into a cohesive framework.

2. Related works

Although classical numerical methods, such as the finite element method (FEM [38–40]) and spectral methods [41, 42], have demonstrated maturity and effectiveness in solving partial differential equations (PDEs), but they exhibit

significant limitations. One of the primary challenges is the curse of dimensionality [43, 44]. As dimensionality increases, the computational costs associated with these methods escalate exponentially, rendering them unsuitable for complex that necessitate extensive computational resources and mesh sizes that reach hundreds of thousands when addressing high-dimensional PDE applications [45, 46]. In contrast, Physics-Informed Neural Networks (PINNs) leverage the inherent flexibility of neural networks, which are particularly adept at approximating high-dimensional functions [47], thereby enabling efficient handling of complex high-dimensional problems with less increase in computational cost [48]

Additionally, classical numerical methods heavily depend on structured meshes [49, 50], complicating computations in the presence of intricate geometries. The mesh generation process can be both time-consuming and error-prone, particularly when navigating complex boundary conditions [51, 52]. PINNs, however, do not require predefined meshes; instead, they learn directly over the defined domain [53, 54]. This mesh-free approach allows them to seamlessly adapt to arbitrary geometries, offering significant advantages in computational efficiency and ease of implementation.

Another notable limitation of traditional methods pertains to their treatment of physical laws. These classical techniques often necessitate extensive preprocessing and meticulous attention to boundary conditions, which can introduce additional complexity and potential sources of error [55–57]. In contrast, PINNs incorporate physical laws directly into the training process, enabling simultaneous approximation of solutions while enforcing physical constraints [58, 59]. This end-to-end learning framework facilitates effective performance even in the absence of large labeled datasets.

In scenarios demanding rapid computation-such as real-time simulations or optimization tasks-PINNs exhibit a remarkable ability to accelerate the solution process [60, 61]. By leveraging existing physical information, PINNs can converge to accurate solutions with fewer iterations than traditional methods, which often require extensive computational resources and numerous iterations.

Moreover, the integration of physical principles into the training of PINNs not only enhances predictive performance but also improves interpretability [62–64]. In fields such as engineering, finance and healthcare, where model outputs require justifiable explanations, understanding model decisions becomes crucial. PINNs provide a clearer pathway for verification and validation, as their solutions are closely aligned with governing physical laws.

It can be seen that while classical numerical methods like FEM and spectral methods excel in accuracy for well-defined problems, they face significant limitations in high-dimensional spaces, complex geometries, and real-time applications. PINNs present a valuable alternative, effectively addressing these shortcomings with their flexibility, efficiency, and interpretability. They outperform traditional methods in scenarios that demand rapid solutions or involve complex physical systems. As computational challenges continue to evolve, integrating PINNs into numerical computing represents a promising direction for future research and practical applications.

While Physics-Informed Neural Networks (PINNs) typically demonstrate considerable advantages in computational efficiency, their performance can be uncertain or inconsistent in specific scenarios. This variability arises from the inherent characteristics of PINNs, which depend on minimizing a loss function that incorporates both the governing partial differential equations (PDEs) and boundary conditions [65]. In complex anisotropic or spatially varying settings, the optimization landscape can become intricate, posing challenges for the training process [66, 48]. Consequently, this complexity can hinder the algorithm's ability to converge to a solution that accurately reflects the underlying physics of the problem.

This study aims to identify a simple and stable computational structure that can effectively adapt to complex carbon dynamics and optimization landscapes. We propose KACINO, a structure designed to ensure consistent computational stability while accommodating various conditions and constraints. Before detailing KACINO, we will first discuss the carbon credit model and its associated computational assembly.

3. Orchestration modeling of carbon credits

The model of carbon credits involves examining the dynamics of carbon emissions, sequestration, and the accumulation of carbon credits over time. Understanding these dynamics is fundamental for managing carbon emissions,

developing administration schemes, evaluating carbon offset projects, and assessing compliance with emissions reduction targets.

3.1 *Carbon emission modeling*

In the context of carbon emissions, as economic activity, population, and energy consumption increase, emissions tend to rise at a rate proportional to their current level [67, 68]. Meanwhile, as industrialization and urbanization progress, increased carbon emissions can lead to a feedback loop where more emissions promote further economic growth (e.g., through increased energy demand). This creates a cycle that can often be approximated by exponential growth [69, 70]. In addition, empirical data on carbon emissions often shows a consistent upward trend, particularly in periods of rapid industrialization and technological advancement. This historical behavior aligns well with the characteristics of exponential growth [71, 72]. Therefore, the carbon emission rate can be modeled using an exponential growth model as

$$P(t) = P_0 \cdot e^{rt}, \tag{1}$$

where $P(t)$ refers to the rate at which carbon dioxide (CO_2) or other greenhouse gases emission into the atmosphere over a specific period of time, $P(t)$ is the are emitted rate at time t, P_0 is the initial emission rate at $t = 0$, r is the growth rate of emissions, e is the base of natural logarithm, t is time. This model describes how the emission rate changes over time assuming a constant growth rate r.

The accumulated carbon emission E represents the cumulative amount of carbon dioxide (CO_2) or its equivalent greenhouse gases released into the atmosphere over a specified period. It is typically measured in units of metric tons of CO_2 or CO_2-equivalent gases emitted within a defined timeframe, such as annually or over a specific project duration, which can be formulated as

$$E(T) = \int_0^T P(t)\, dt, \tag{2}$$

where E is the total carbon emission over time T, $P(t)$ is the emission rate at time t. This integral model sums up the emissions over time based on the emission rate function $E(t)$. to calculate the total carbon emission E over a period from $t = 0$ to $t = T$, as an integration of the accumulated emission rate $P(t)$ over time.

Substituting the exponential growth model (1) into (2), the accumulated emission can be formulated as

$$E(T) = P_0 \int_0^T e^{rt}\, dt = P_0 \frac{e^{rT} - 1}{r}, \tag{3}$$

where $E(T)$ is the total carbon emission over time T, P_0 is the initial emission rate at $t = 0$, r is the growth rate of emissions, T is the duration over which emissions are accumulated.

These mathematical models provide a framework for understanding both the rate at which carbon emissions occur over time (carbon emission rate) and the cumulative total emissions over a specified period (total carbon emission).

They are essential tools for analyzing emission trends, projecting future impacts, and informing policy decisions aimed at mitigating climate change and promoting sustainability.

Carbon emissions are the release of carbon dioxide (CO_2) or its equivalent greenhouse gases into the atmosphere, primarily from human activities such as burning fossil fuels, deforestation, and industrial processes. The decay rate of carbon emission can be conceptualized as the rate at which the emission intensity or emission rate changes over time due to factors such as technological advancements, policy interventions, and shifts in energy sources.

In a natural emission environment unaffected by external factors, its dynamics directly correlate with $P(t)$. However, when measures or environmental changes are implemented, these emissions are influenced by a factor related to the carbon emission level on the basis of $P(t)$. Then the rate of change of carbon emissions $E(t)$ over time t is given by

$$\frac{dE}{dt} = P(t) - \gamma E(t), \tag{4}$$

where $E(t)$ represents the amount of carbon emissions at time t, $P(t)$ is the carbon emission rate at time t, and γ is the decay rate or removal rate of carbon emissions, accounting for processes such as carbon capture or natural absorption.

3.2 *Carbon sequestration*

The carbon sequestration rate $R(t)$ represents the rate at which carbon dioxide (CO_2) or its equivalent greenhouse gases are removed from the atmosphere or stored in carbon sinks. In saturation, carbon sequestration processes (e.g., plant growth, soil absorption) have a maximum capacity [73, 74], as carbon is sequestered over time, the rate of sequestration approaches this maximum but never exceeds it [75]. Many studies of ecological systems and carbon cycling have shown that sequestration rates often follow a logistic pattern [76, 77], and the logistic growth model is well-suited for representing carbon sequestration because it captures the saturation effects, and realistic ecological behavior associated with carbon uptake in natural systems [78–80].

Considering the upper limit of the rate of sequestration, the rate can be formulated with a logistic growth model base on an initial increase as

$$R(t) = \frac{R_{\max}}{1 + e^{-k(t-t_0)}}, \tag{5}$$

where $R_{\max}$ is the maximum sequestration rate, k is the growth rate constant, and t_0 is the time of inflection where the rate starts to level off.

This model provides frameworks for understanding how the carbon sequestration rate changes over time in various environmental and policy contexts.

The Accumulated Carbon Sequestration $S(t)$ represents the total amount of carbon sequestered up to time T. It is obtained by integrating the sequestration rate $R(t)$ over time, which can be formulated as

$$S(T) = \int_0^T R(t)\, dt = \int_0^T \frac{R_{\max}}{1 + e^{-k(t-t_0)}}\, dt, \tag{6}$$

where $R_{\max}$ is the maximum sequestration rate, k is the growth rate constant, and t_0 is the time when sequestration starts to level off. The integral (6) typically lacks a straightforward analytical solution For practical purposes, it is often evaluated using numerical methods or suitable analytical techniques, depending on the specific values of $R_{\max}$, k, and t_0.

Carbon sequestration refers to the process of capturing and storing carbon dioxide from the atmosphere, primarily through natural processes (e.g., photosynthesis in plants and trees) or engineered solutions (e.g., carbon capture and storage technologies). The decay rate of carbon sequestration refers to the rate at which carbon is removed or stored from the atmosphere and how this rate may vary over time due to environmental factors, land use changes, and technological developments.

Similar to carbon emission, when considering the decay rate of the sequestration, the rate of change of carbon sequestration $S(t)$ over time t is described by

$$\frac{dS}{dt} = R(t) - \lambda S(t), \tag{7}$$

where $S(t)$ denotes the amount of carbon sequestered (removed from the atmosphere) at time t, $R(t)$ is the carbon sequestration rate at time t, and λ is the decay rate or loss rate of carbon sequestration due to processes like carbon release from storage.

3.3 *Modeling of carbon credits with carbon balance*

Modeling carbon credits involves quantifying the reduction or removal of greenhouse gases (GHGs) to mitigate climate change. Carbon credits represent the unit of measurement for these reductions or removals, typically measured in metric tons of carbon dioxide equivalent (tCO_2e) which is a measure of the effect of different greenhouse gases on the climate. In essence, the net carbon balance determines whether an entity has a surplus (carbon credits) or deficit (carbon debits) in greenhouse gas emissions compared to its emissions reduction and sequestration efforts, in which carbon credits provide a means to incentivize and quantify activities that mitigate climate change by reducing net emissions or increasing carbon removal from the atmosphere.

By introducing the carbon emission $E(t)$ and sequestration $S(t)$ into the balance, the net carbon balance at time t, which determines the carbon credits, can be expressed as

$$B(t) = E(t) - S(t), \tag{8}$$

The carbon credit $C(t)$ corresponding to the balance $B(t)$ can be formulated as

$$C(t) = -\frac{1}{K}B(t) = \frac{S(t) - E(t)}{K}, \tag{9}$$

where K is a scaling factor influenced by policy and market, representing the equivalence of carbon credits to metric tons of CO_2e. If the process of carbon dynamic is autonomous, then the balance accumulation can be formulated as

$$\Gamma(t) = \int_0^t (P(t) - R(t))\, dt, \tag{10}$$

where $\Gamma(t)$ represents the net balance of emissions and sequestration up to time t. When considering the decay of the emission and sequestration, the dynamics (4) and (7) need to be applied in carbon credit.

3.4 *Carbon credit modeling orchestration*

Carbon credit modeling orchestration involves integrating and analyzing various dynamic processes related to carbon emission, sequestration, and credit accumulation. These models provide a comprehensive framework for modeling the flow of carbon emissions, carbon sequestration, and the accumulation of carbon credits in environmental systems. They are crucial tools for assessing carbon management strategies, guiding policy decisions, and ensuring environmental sustainability.

To orchestrate carbon credit modeling effectively, we first define variables and dynamics by identifying key variables such as carbon emission rates, sequestration rates, and credit accumulation rates, and construct a Coherent framework based on the modeling of carbon emission, sequestration and credit as

$$\begin{bmatrix} dE(t) \\ dS(t) \end{bmatrix} = \begin{bmatrix} 1 & -\gamma & & \\ & & 1 & -\lambda \end{bmatrix} \begin{bmatrix} P(t) \\ E(t) \\ R(t) \\ S(t) \end{bmatrix} dt, \tag{11}$$

$$C(t) = \frac{S(t) - E(t)}{K}. \tag{12}$$

This framework formulates the system as a representation of the interactions between emission, sequestration, and credits with the consideration of decay rates, external influences, and policy impacts. This orchestration models the interactions between carbon emissions, sequestration, and credit accumulation to assess their interdependencies. It also can be used to analyze how policy interventions, technological advancements, and natural variability impact carbon dynamics over time. With the dynamic interaction, this framework can be used to forecast future carbon credit availability based on different scenarios and policy implementations. By using insights gained from orchestration, we also can optimize carbon credit trading mechanisms and enhance climate change mitigation efforts. Carbon credits are crucial in climate change mitigation strategies as they incentivize reductions in greenhouse gas emissions and promote carbon sequestration activities.

4. Kolmogorov anorld credit informed network ochestration

4.1 *Kolmogorov-arnold representation*

The Kolmogorov-Arnold representation (KAR) theorem states that any continuous multivariate function can be expressed as a superposition of univariate functions. For a continuous function $f(x_1, x_2, \ldots, x_n)$, KAR asserts the following representation:

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{N} g_i(h_i(x_1), h_i(x_2), \ldots, h_i(x_n)), \tag{13}$$

in which g_i represents mappings that can take multiple inputs h_i, reflecting their dependence on the mappings, and N refers to the number of mapping in the combination of functions, which can be finite or infinite.

With the superposition principle, KAR simplifies the representation of complex multivariate functions into a sum of simpler univariate components of KAR. It simplifies the solving of complex differential equations by breaking them down into simpler parts. In computational contexts, KAR reduces the computational complexity of high-dimensional problems by decomposing them into lower-dimensional operations. It facilitates effective nonlinear approximations, where complex relationships between variables can be approximated by simpler functions, aiding in interpretability and efficiency.

4.2 *Minimal viable KACINO*

Carbon Informed Neural Networks (CINNs) integrate carbon theories and constraints directly into a neural network architectures to enhance predictive accuracy and model reliability in carbon applications. KACINO serves as a type of Finance Informed Neural Networks (CINNs) with Kolmogorov-Arnold Representation (KAR), integrating carbon domain knowledge and sophisticated mathematical frameworks to enhance the predictive capabilities and interpretability of carbon models.

Consider a carbon system characterized by a set of differential equations representing carbon dynamics, such as option pricing models or portfolio optimization:

$$\mathscr{F}[\mathbf{u}(\mathbf{x},\, t);\kappa] = 0, \quad \mathbf{x} \in \Omega, \quad t \in [0,\, T], \tag{14}$$

where $\mathscr{F}$ denotes the carbon differential operator, $\mathbf{u}(\mathbf{x},\, t)$ represents the carbon variable of interest (e.g., option price, asset value), κ includes parameters of the carbon model, Ω represents the domain of carbon variables, $[0,\, T]$ denotes the time period of interest.

In KACINO, the neural network $\mathbf{u}_\theta(\mathbf{x},\, t)$ is constrcuted as a Kolmogrov Anorld Network (KAN) with parameters θ to approximate the solution $\mathbf{u}(\mathbf{x},\, t)$. Utilizing KAR, equation (13) can be represented as a network to reflect the decomposition as

$$\mathbf{u}_\theta(\mathbf{x},\, t) = \sum_{i=1}^{L_2} g_{i,\,\theta} \left(\sum_{j=1}^{L_1} h_{ij,\,\theta}(x_j,\, t) \right), \tag{15}$$

where $g_{i,\,\theta}(\cdot)$ and $h_{ij,\,\theta}(\cdot)$ are learned by the network, $L_1,\, L_2$ are the number of the function units used in layers 1 and 2. By learning the parameters, KACINO that any continuous multivariate function can be decomposed into the sum of univariate functions of linear combinations of the input variables.

In light of the absence of analytical solutions, the evaluation of the reliability of the derived approximate solution necessitates the introduction of the concept of credit. This measure is quantified by establishing explicit criteria for the understanding of the underlying mechanism and assessing the extent to which the approximate solution adheres to these criteria. A straightforward approach involves defining credit through a loss function. The modeling credit is formulated as the minimal loss, and the parameters $\theta,\, g_{i,\,\theta}(\cdot)$ and $h_{ij,\,\theta}(\cdot)$ are optimized by minimizing the total loss function, which can be formulated as

$$\theta^* = \arg\min_\theta \mathscr{L}(\theta), \tag{16}$$

The loss function in the context of carbon constraints and system comprises several key components, including initial value loss, boundary loss, and carbon governing loss.

$$\mathscr{L}(\theta) = \mathscr{L}_{\text{initial}}(\theta) + \mathscr{L}_{\text{boundary}}(\theta) + \mathscr{L}_{\text{governing}}(\theta), \tag{17}$$

in which the initial value loss $\mathscr{L}_{\text{initial}}(\theta)$ ensures that the neural network's prediction at the initial time $t = 0$ matches the given initial condition; the boundary loss $\mathscr{L}_{\text{boundary}}(\theta)$ ensures that the neural network solution satisfies the boundary conditions of the carbon domain; the carbon governing loss $\mathscr{L}_{\text{governing}}$ enforces adherence to the carbon models and constraints.

The formulation of the loss function is crucial in developing a robust minimally viable KACINO. Integration of convex learning mechanisms [81] ensure that all loss functions satisfy convex constraints related to Initial, Boundary, and carbon considerations. The mean squared error (MSE) metric is particularly effective in this regard, as it precisely measures the average disparities between estimated outputs and their true references, offering a straightforward and robust framework for defining the loss function.

When confronted with a situation where the mechanism is partially understood yet lacks an analytical solution, we infer through logical contradiction that any solution diverging from the known mechanism is inherently unreliable. This approach gives rise to a modeling paradigm termed Credit Informed Modeling. In the context where the model takes the form of a neural network, it specifically refers to a Credit Informed Neural Network (CINN).

4.3 *Generalization of deep KACINO*

We can further generalise the KACINO neural network representation (15) to deeper cases. Let the first layer be $\Phi_1(\cdot) = \sum h(\cdot)$ and the second layer $\Phi_2(\cdot) = \sum g(\cdot)$. The equation (15) can be represented as

$$\mathbf{u}_\theta(\mathbf{x},\, t) = \sum_{i=1}^{L_2} g_{i,\,\theta} \left(\sum_{j=1}^{L_1} h_{ij,\,\theta}(x_j,\, t) \right) = \Phi_2 \circ \Phi_1 z \tag{18}$$

$$\Phi_2 = [g_1(\cdot),\, g_2(\cdot),\, \cdots,\, g_{L_2}(\cdot)] \tag{19}$$

$$\Phi_1 = \begin{bmatrix} h_{1,\,1}(\cdot) & h_{1,\,2}(\cdot) & \cdots & h_{1,\,L_1}(\cdot) \\ h_{2,\,1}(\cdot) & h_{2,\,2}(\cdot) & \cdots & h_{2,\,L_1}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ h_{L_2,\,1}(\cdot) & h_{L_2,\,2}(\cdot) & \cdots & h_{L_2,\,L_1}(\cdot) \end{bmatrix} \tag{20}$$

$$z = \Big[(x_1,\, t)(x_2,\, t) \ldots (x_{L_1},\, t) \Big]^T. \tag{21}$$

Similarly, Φ_k can be defined to perform the deep KACINO that can be formulated as

$$\mathbf{u}_\theta(\mathbf{x},\, t) = \Phi_L \circ \Phi_{L-1} \cdots \circ \Phi_k \circ \cdots \circ \Phi_1 z, \tag{22}$$

$$\Phi_k = \begin{bmatrix} \Phi_{1,\,1}(\cdot) & \Phi_{1,\,2}(\cdot) & \cdots & \Phi_{1,\,L_k}(\cdot) \\ \Phi_{2,\,1}(\cdot) & \Phi_{2,\,2}(\cdot) & \cdots & \Phi_{2,\,L_k}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{L_{k+1},\,1}(\cdot) & \Phi_{L_2,\,2}(\cdot) & \cdots & \Phi_{L_{k+1},\,L_k}(\cdot) \end{bmatrix}. \tag{23}$$

The general loss function can be formulated as a weighted measure of a series of sub loss functions $\mathscr{L}_l(\theta)$, which can be formulated as

$$\mathscr{L}(\theta) = \sum \lambda_l \mathscr{L}_l(\theta), \tag{24}$$

Table 1 provides a comprehensive list of symbols utilized in the context of carbon modeling, as a reference of various components related to carbon dynamics and neural network representations. The Kolmogorov Arnold Representation employs univariate functions g_i to decompose complex carbon systems into simpler, interpretable components, allowing for a modular approach. Mappings h_i define interactions among carbon physics, capturing nonlinear relationships essential for carbon ecological modeling. The adjustable superposition of terms N enables the model to scale with data intricacies, accommodating both simple and intricate relationships.

The carbon differential operator $\mathscr{F}$ formulates differential equations governing the dynamics of carbon variables over time, while $\mathbf{u}(\mathbf{x},\, t)$ represents the core carbon variable of interest. The neural network model $\mathbf{u}_\theta(\mathbf{x},\, t)$ leverages machine learning to capture complex patterns and enhance predictive accuracy.

Model parameters σ are crucial for tuning and ensuring alignment with observed data, and the domain Ω encompasses all relevant variables. Finally, the time period $[0,\ T]$ facilitates temporal analysis, enabling researchers to evaluate trends and impacts, particularly important for understanding climate change dynamics.

Table 1. List of symbols used in the carbon modeling context

Symbol	Descriptio
g_i	Univariate functions in Kolmogrov Anorld Representation
h_i	Mappings that define how each univariate function interacts with each variable
N	Number of terms in the superposition, which can be finite or infinite
$\mathscr{F}$	Carbon differential operator
$\mathbf{u}(\mathbf{x},\ t)$	Carbon variable of interest
$\mathbf{u}_\theta(\mathbf{x},\ t)$	Neurual Network Representation of the carbon variable of interest
σ	Parameters of the carbon model
Ω	Domain of carbon variables
$[0,\ T]$	Time period of interest

By employing a combination of traditional mathematical representations and modern machine-learning techniques, the models can effectively capture the nuances of interactions within carbon systems for quantitative analysis and predictive simulations, contributing to more informed decision-making in environmental management and policy.

5. Case study: Carbon balance dynamics

5.1 *KACINO for carbon balance*

For balance, KACINO leverages the principles of carbon emission (4) and sequestration (7) while incorporating data-driven neural network techniques to enhance prediction accuracy and model robustness. This case study illustrates how KACINO can be employed to carbon balance effectively as a case in point for greenhouse gas (GHG) crediting program application.

In the context of GHG credit, the KACINO model defines the carbon orchestration (CO) model through the differential operator derived from the carbon balance differential equation. The CO equation can be expressed as:

$$f_{CO} = dD(t) - AG(t)dt = 0, \tag{25}$$

where

$$A = \begin{bmatrix} 1 & -\gamma & 0 & 0 \\ 0 & 0 & 1 & -\lambda \end{bmatrix},\ D(t) = \begin{bmatrix} E(t) \\ S(t) \end{bmatrix},\ G(t) = \begin{bmatrix} P(t) \\ E(t) \\ R(t) \\ S(t) \end{bmatrix}.$$

Matrix $D(t)$ represents the state of carbon dynamics, specifically the emissions $E(t)$ and sequestration $S(t)$ at time t. It captures the net effect of carbon emissions and storage. In contrast, $G(t)$ encapsulates the total carbon dynamics involving emission rate $P(t)$, emissions $E(t)$, sequestration rate $R(t)$, and sequestration $S(t)$. This vector serves as a comprehensive representation of all contributing factors in the carbon cycle.

The coefficient A acts as a transformation matrix that relates the dynamics of the emissions and sequestration represented in $D(t)$ to the broader dynamics captured in $G(t)$. The coefficients γ and λ are parameters that represent the rates of interactions between the components of the carbon balance, defining how changes in production and respiration influence emissions and storage

Thus, the equation (25) serves as the governing constraint for carbon balance in partial differential equation (PDE) form, capturing the essential dynamics of carbon flows within the KACINO model.

By applying (22)-(24), we build the KACINO to learn the KAR to decompose the carbon orchestration of physics $D(G,\, t)$ using KAR into a series of simpler functions:

$$\mu_\theta(G,\, t) = \sum_{i=1}^{N} g_i(h_i(G),\, t) \approx D(G,\, t), \tag{26}$$

where g_i are univariate functions, and $h_i(x)$ denote the mappings of these functions with respect to S and t, N refers to the number of layers of the deep learning and θ the parameters of the learning system. This decomposition allows for a more granular and interpretable representation of the carbon balance dynamics. The neural network component of KACINO is designed to learn the parameters of these univariate functions and their interactions. The network is trained using a composite loss function that includes Initial Value Loss, Boundary Loss and Carbon Governing Constraint Loss. The carbon governing constraint loss $\mathscr{L}_{\text{carbon}}(\theta)$ is built based on the (25) as a constraint to enforce the CO PDE constraints within the domain.

Based on (17), the corresponding initial value loss $\mathscr{L}_{\text{initial}}(\theta)$, boundary loss $\mathscr{L}_{\text{boundary}}(\theta)$, and carbon governing loss $\mathscr{L}_{\text{carbon}}(\theta)$ can be formulated as follows

$$\mathscr{L}_{\text{initial}}(\theta) = \frac{1}{N_{\text{init}}} \sum_{i=1}^{N_{\text{init}}} \left|\mathbf{u}_\theta(\mathbf{x}_i,\, 0) - \mathbf{u}_0(\mathbf{x}_i)\right|^2, \tag{27}$$

$$\mathscr{L}_{\text{boundary}}(\theta) = \frac{1}{N_b} \sum_{j=1}^{N_b} \left|\mathbf{u}_\theta(\mathbf{x}_j,\, t_j) - \mathbf{b}(\mathbf{x}_j,\, t_j)\right|^2, \tag{28}$$

$$\mathscr{L}_{\text{carbon}}(\theta) = \frac{1}{N_f} \sum_{k=1}^{N_f} \left|\mathscr{F}[\mathbf{u}_\theta(\mathbf{x}_k,\, t_k); \lambda]\right|^2. \tag{29}$$

In this context, $\mathscr{L}_{\text{initial}}(\theta)$ quantifies the difference between the neural network's prediction at the initial time $t = 0$, $\mathbf{u}_\theta(\mathbf{x}_i,\, 0)$, and the known initial condition $\mathbf{u}_0(\mathbf{x}_i)$ across N_{init} points. The boundary loss $\mathscr{L}_{\text{boundary}}(\theta)$ measures the discrepancy between the predictions at boundary points $\mathbf{u}_\theta(\mathbf{x}_j,\, t_j)$ and the known boundary conditions $\mathbf{b}(\mathbf{x}_j,\, t_j)$, evaluated over N_b boundary points. Finally, the carbon governing loss $\mathscr{L}_{\text{carbon}}(\theta)$ assesses how well the predictions satisfy the carbon differential operator $\mathscr{F}$, with λ representing the parameters of the carbon model, across N_f points used to enforce the carbon constraints.

5.2 *Governing loss for carbon orchestration*

To mathematically model the governing loss for carbon dynamics using the CO model within a neural network framework, we need to ensure that the network's output adheres to the CO differential equation. The loss function should penalize deviations from the CO model, thereby enforcing the carbon constraints.

Let the neural network output for the call carbon balance be $u_\theta(G,\ t)$, where θ denotes the network parameters. The carbon governing loss can be constructed by computing the residual of the CO when applied to the network's output. This residual should ideally be zero across the domain of interest. The residual $\mathscr{R}(G,\ t)$ is given by:

$$\mathscr{R}(G,\ t) = \frac{du_\theta}{dt} - AG(t), \tag{30}$$

The carbon governing loss $\mathscr{L}_{\text{carbon}}$ can then be defined as the mean squared error of the residual over the sampled points $(G_i,\ t_i)$ in the domain:

$$\mathscr{L}_{\text{carbon}} = \frac{1}{N}\sum_{i=1}^{N} \mathscr{R}^2(G_i,\ t_i), \tag{31}$$

where N is the number of sample points. By integrating this carbon governing loss into the overall loss function of the neural network, we ensure that the network's output respects the carbon dynamics dictated by the CO model.

5.3 *Initial condition for carbon balance*

The initial value loss for carbon dynamics using a neural network ensures that the network's output satisfies the initial conditions of the option pricing model. For a carbon balance, the initial condition is determined by the payoff function at the time of issuance (time $t = 0$).

If the carbon content in the land is saturated, then the system will no longer be able to absorb carbon, so it will take its upper limit value S_K then the maximum will be S_K. The initial condition for a carbon sequestration balance at $t = 0$ is given by the:

$$S(G,\ 0) = \min(S,\ S_K), \tag{32}$$

where $S(G,\ t)$ is the carbon sequestration, G is the underlying asset carbon dynamics, and S_K is the upper bound of the sequestration dynamics. If considering the upper bound of carbon emission in a region, then the carbon dynamics presents as a limited emission system, which means there is an upper limit E_K for the emission. Therefore, the initial condition for a carbon emission balance at $t = 0$ is given by

$$E(G,\ 0) = \min(E,\ E_K), \tag{33}$$

The initial value loss $\mathscr{L}_{\text{initial}}$ ensures that the neural network output at $t = 0$ matches the initial condition $\min(E,\ E_K)$. Mathematically, this loss can be defined as the mean squared error between the neural network's predicted carbon balances and the actual initial condition over a set of sampled carbon sequestration dynamics G_i.

Let $D_K = [E_K,\ S_K]^T$, $D_i(t) = [E_i(t),\ S_i(t)]^T$ and $u_\theta(G,\ t)$ be the output of the neural network with parameters θ, representing the carbon balance, and the initial loss can be formulated as

$$\mathscr{L}_{\text{initial}} = \frac{1}{M}\sum_{i=1}^{M} \left(u_\theta(G_i,\ 0) - \min(D_i(0),\ D_K)\right)^2, \tag{34}$$

where M is the number of sampled carbon points G_i.

5.4 *Boundary loss for carbon dynamics*

To mathematically model the boundary loss for carbon dynamics using a neural network, we ensure that the network's output satisfies the boundary conditions of the option pricing model. For a carbon balance, the boundary conditions are determined by the behavior of the carbon balance as the underlying asset carbon dynamics approaches zero and infinity, and at the expiration time.

The boundary conditions for a carbon balance include the zero boundary and the infinity boundary. For zero boundary, as the underlying asset carbon G approaches zero, the carbon balance should also approach zero:

$$D(0,\, t) = 0 \quad \forall t. \tag{35}$$

For infinity boundary, as the underlying asset carbon dynamics G approaches infinity, the carbon balance should asymptotically approach based on the dynamic processing during the emission and sequestration balance as $\min(D,\, D_K)$ that

$$\lim_{G\to\infty} D(G,\, t) = \min(D(t),\, D_K) \quad \forall t, \tag{36}$$

The boundary loss $\mathscr{L}_{\text{boundary}}$ ensures that the neural network output satisfies these boundary conditions. Mathematically, this loss can be defined as the sum of the mean squared errors at the boundary points.

The zero boundary loss can be formulated as

$$\mathscr{L}_{\text{boundary},\,0} = \frac{1}{N_0}\sum_{i=1}^{N_0} \left(u_\theta(0,\, t_i)\right)^2, \tag{37}$$

where N_0 is the number of sampled points t_i. The infinity boundary can be formulated as

$$\mathscr{L}_{\text{boundary},\,\infty} = \frac{1}{N_\infty}\sum_{i=1}^{N_\infty} \left(u_\theta(G_i,\, t) - \min(D_i - D_K)\right)^2, \tag{38}$$

where N_∞ is the number of sampled points G_i.

The total boundary loss $\mathscr{L}_{\text{boundary}}$ is the sum of these individual losses:

$$\mathscr{L}_{\text{boundary}} = \mathscr{L}_{\text{boundary},\,0} + \mathscr{L}_{\text{boundary},\,\infty}. \tag{39}$$

5.5 *Implementation in the loss function*

We collect the loss function results as a tensor $\mathscr{L}$ that

$$\mathscr{L} = [\mathscr{L}_{\text{initial}} \quad \mathscr{L}_{\text{boundary},\,0} \quad \mathscr{L}_{\text{boundary},\,\infty} \quad \mathscr{L}_{\text{carbon}}], \tag{40}$$

The corresponding weight for each loss is enclosed in a tensor σ that

$$\sigma = [\sigma_{\text{initial}} \quad \sigma_{\text{boundary, 0}} \quad \sigma_{\text{boundary, }\infty} \quad \sigma_{\text{carbon}}], \tag{41}$$

The total loss is obtained by

$$\mathscr{L}_{\text{total}} = \sigma \mathscr{L}^T, \tag{42}$$

The loss function (42) is convex because it is a quadratic function (sum of squares), and quadratic functions are convex. Therefore, there exists at least one minimizer θ^* such that $\mathscr{L}(\theta^*) \leq \mathscr{L}(\theta)$ for all θ in the parameter space according to convex optimization theory.

6. Experiment and results

6.1 *Experiment setting and implementation*

In this section, we detail the experimental setup for evaluating the performance of our proposed KACINO framework in the context of European call carbon. The key parameters and settings used in the experiments are outlined below in Table 2.

Table 2. Parameter settings for carbon balance test

Parameter	Value
Emission Upper Bound (E_K)	200
Emission Growth Rate (r)	0.05
Emission Decay (γ)	0.05
Initial emission rate (P_0)	10
Sequestration Upper Bound (S_K)	700
Maximum sequestration rate (R_{max})	90
Sequestration growth rate (k)	0.1
Sequestration Decay (λ)	0.05
Expiration (T)	10
Emission Range (E)	$E \in [0,\ 100]$
Sequestration Range (S)	$S \in [0,\ 100]$
Time Range (t)	$t \in [0,\ T]$
Start time (t_0)	0
Number of Samples (N_{sample})	10,000
Carbon Info Weight (σ_{carbon})	0.1
Initial Value Weight ($\sigma_{initial}$)	1
Zero Boundary Weight ($\sigma_{boundary,\ 0}$)	1
Infinity Boundary Weight ($\sigma_{boundary,\ \infty}$)	1

The study generates sample data within specified asset prices and time ranges. Each data point includes the asset's carbon emission E, sequestration S, and the corresponding time t. The CO model serves as the benchmark for the KACINO training framework. The framework is trained using these data points, incorporating initial value, boundary, and carbon governing losses to accurately model carbon dynamics.

The evaluation compares the models' loss, using metrics like mean squared error (MSE) to assess accuracy and robustness according to the given governing rules. This setup aims to validate KACINO's effectiveness in carbon balance computation, showcasing its potential in carbon modeling and risk management. To further validate the effectiveness of the KACINO model, we compare its performance with CINNO REG [31–33], CINNO FNN [29, 30, 82, 83], CINNO DNN [34, 35, 84], CINNO MLP [36, 37, 85], and CINNO GRU [25–28] as baselines under the same experiment setting. The following results show the performance after training with 5,000 epochs.

6.2 *Results*

In this section, we present the results of our KACINO-based approach to carbon dynamics.

Figure 1 illustrate the learning results as the outcomes of the KACINO system following 5,000 training epochs, conducted under the conditions specified in Table 2. The results represent the solutions to the underlying partial differential equations (PDEs) that govern the processes of carbon emissions and sequestration, incorporating relevant initial and boundary conditions.

In this context, the "Time of Cycle" corresponds to the variable t, reflecting the temporal evolution of the system. The designated point values, referred to as "Spot E" and "Spot S", represent the specific values of emission and sequestration at designated spatial locations and times. The calculations of emissions and sequestration are informed by the extensive knowledge encapsulated within the KACINO model, which synthesizes various factors that influence carbon dynamics.

These calculations adhere to the constraints outlined in equation (17), thereby ensuring that the resulting solutions are consistent with the physical principles governing the carbon cycle.

Table 3 provides an evaluation of six different methods-CINNO DNN, CINNO FFN, CINNO GRU, CINNO MLP, CINNO REG, and KACINO-over 5,000 epochs, focusing on Total Loss and Sequestration. The metrics provided include the mean, standard deviation (std), minimum (min), and maximum (max) values. This analysis offers a detailed comparative assessment of these methods based on the performance metrics.

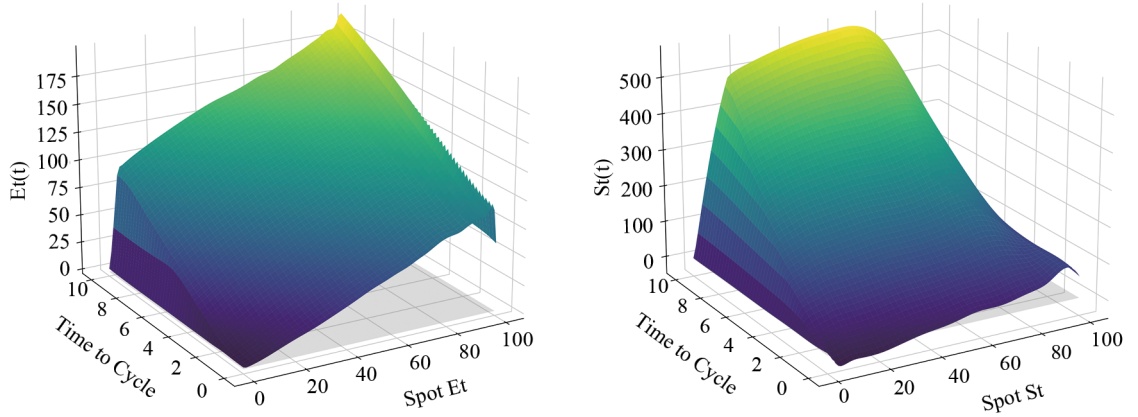
Figures 2 and 3 present the solutions derived from various methods after 5,000 training epochs. The results indicate that KACINO has achieved more comprehensive solutions for carbon emission and sequestration when compared to the other methods under the same training conditions. This superiority suggests that KACINO is more effective in capturing the underlying dynamics of the partial differential equations (PDEs) governing these processes.

The enhanced performance of KACINO may be attributed to its structural design, which enables better adaptation to the complexities inherent in carbon dynamics. In contrast, the other methods have not yet reached a comparable level of solution completeness, highlighting potential limitations in their ability to model the intricate relationships within the carbon emission and sequestration frameworks. These findings underscore the importance of selecting appropriate computational structures when addressing complex PDEs in environmental applications.

We also analyze the convergence of CINNO models during the training process. All the loss function values are monitored over successive training iterations. Figures 4 and 5 show the trajectories loss functions during training. KACINO shows rapid convergence, with the stabilizing cross-all loss functions, reflecting the efficiency of the KACINO approach in training neural networks for carbon applications.

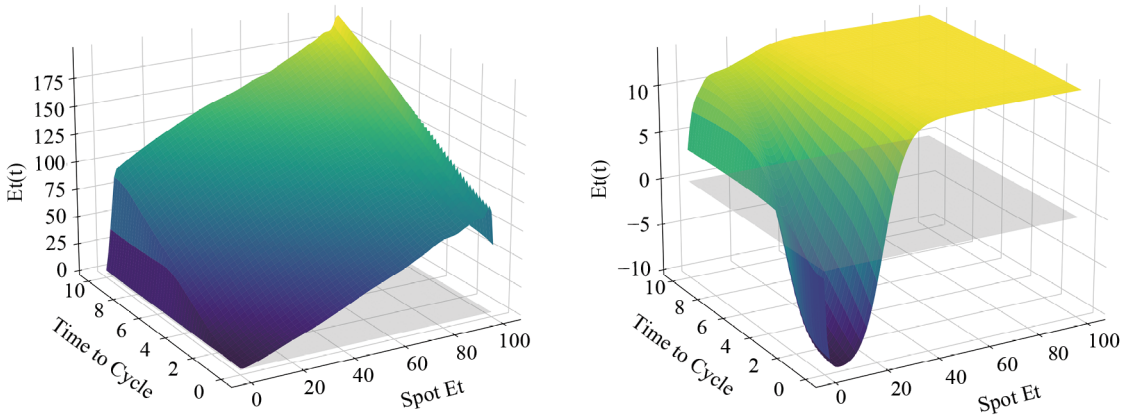
Table 3 shows the evaluation of six different methods-CINNO DNN, CINNO FFN, CINNO GRU, CINNO MLP, CINNO REG, and KACINO-over 5,000 epochs, focusing on Total Loss and Sequestration. Metrics provided include mean, standard deviation (std), minimum (min), and maximum (max) values. This analysis offers a detailed comparative assessment of these methods based on these performance metrics.

Table 4 presents performance metrics of different methods over 5,000 epochs in governing Loss. It can be seen in Table 4 that the KACINO model outperforms existing deep CINN approaches in terms of accuracy within the limited training epochs.

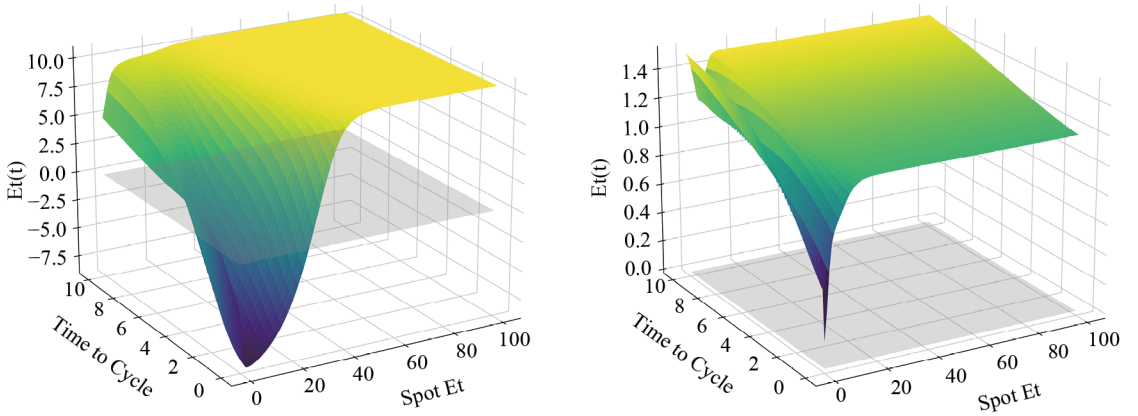


(a) KACINO Emission Solution (b) KACINO Sequestration Solution

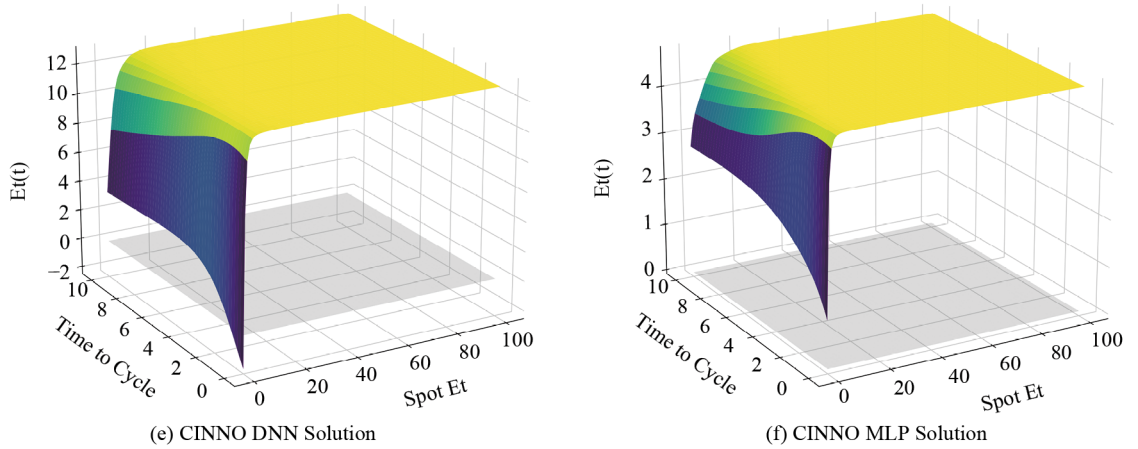
Figure 1. KACINO Solution with 5,000 epochs



(a) KACINO Solution (b) CINNO REG Solution

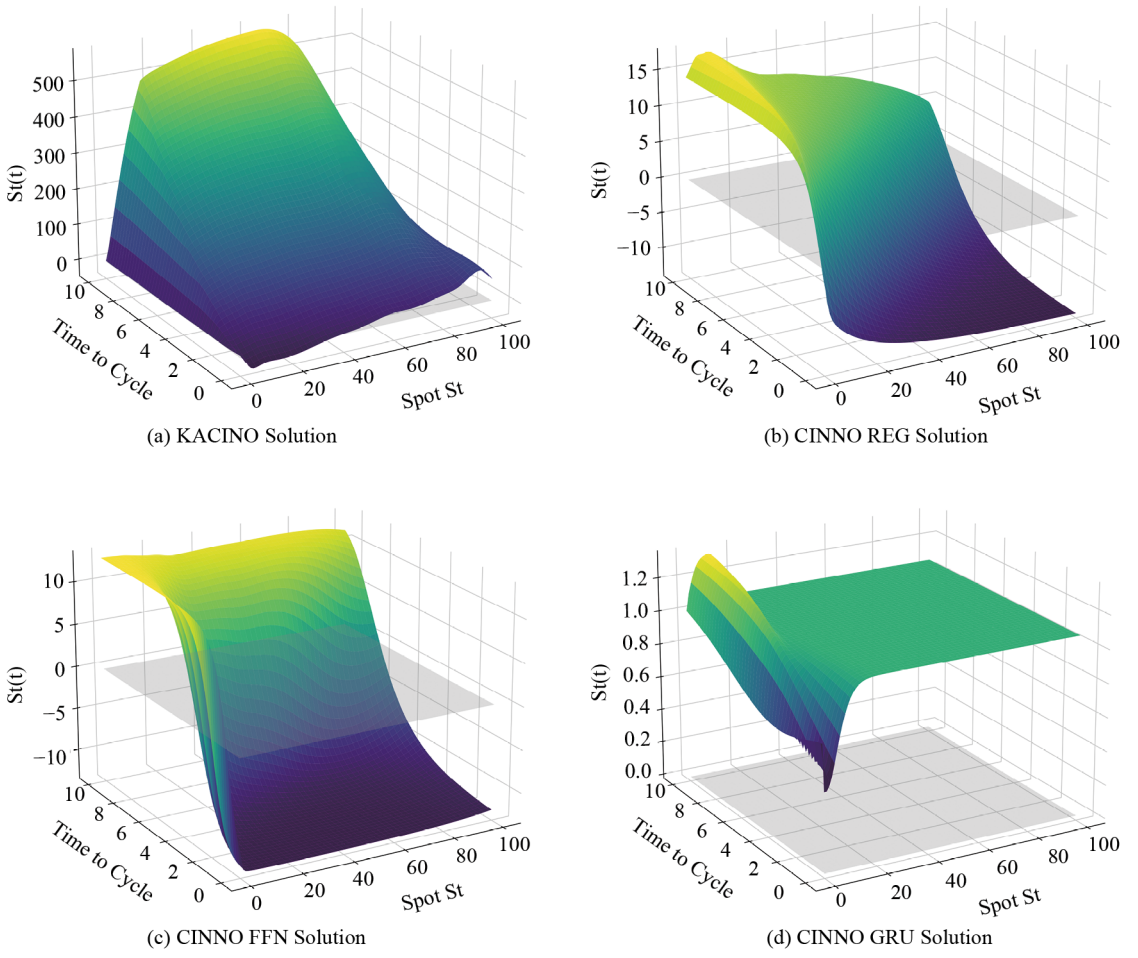


(c) CINNO FFN Solution (d) CINNO GRU Solution



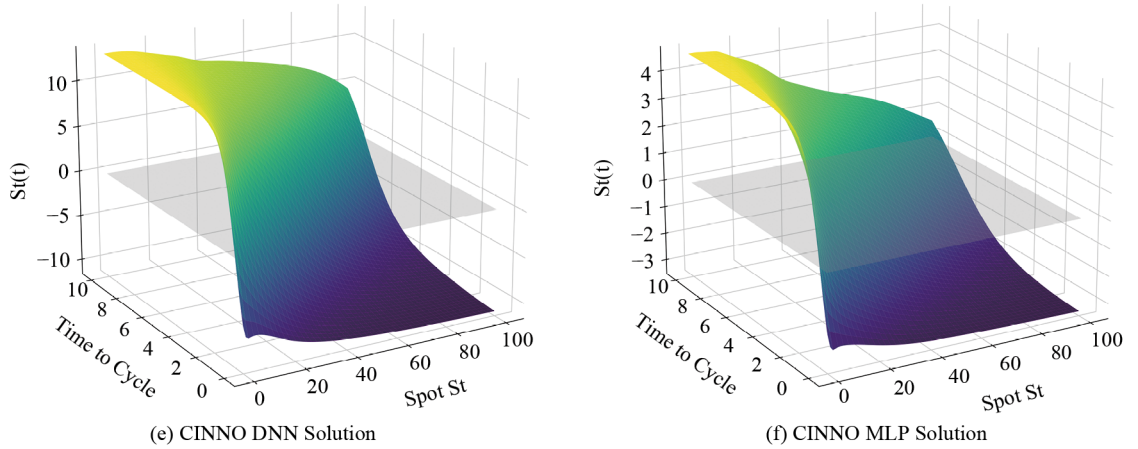
(e) CINNO DNN Solution (f) CINNO MLP Solution

Figure 2. Comparison among the methods with 5,000 epochs in emission



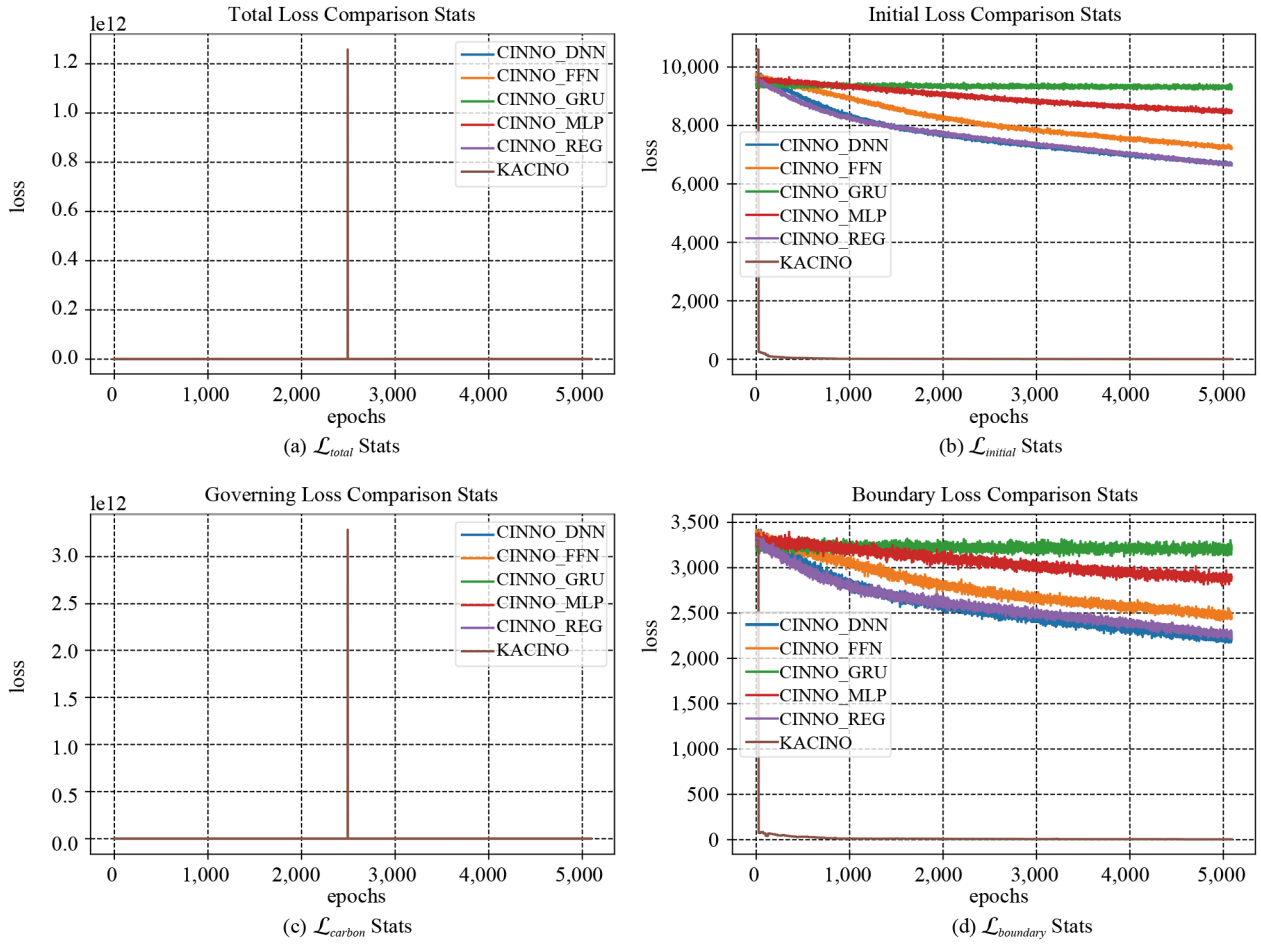
(a) KACINO Solution (b) CINNO REG Solution

(c) CINNO FFN Solution (d) CINNO GRU Solution



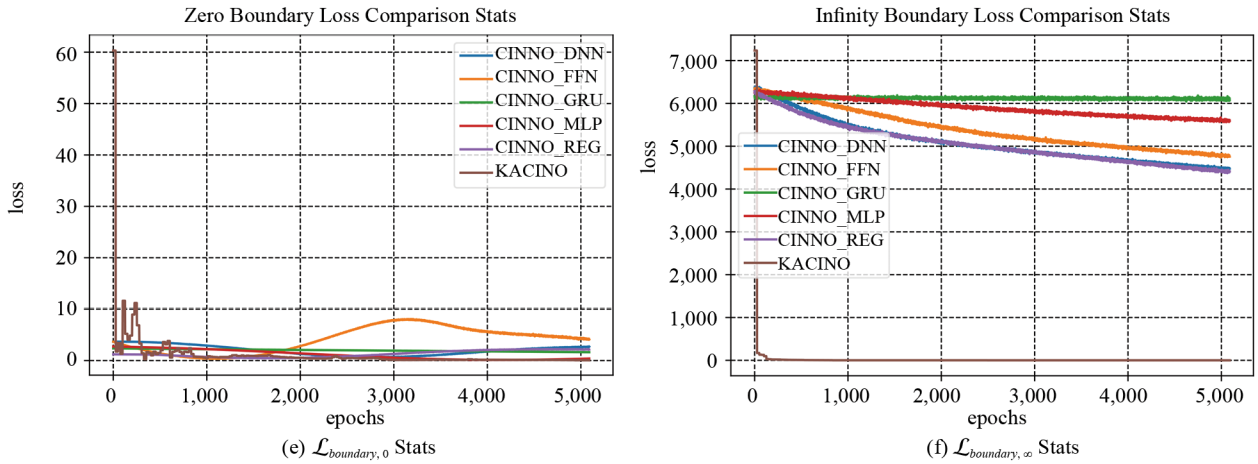
(e) CINNO DNN Solution (f) CINNO MLP Solution

Figure 3. Comparison among the methods with 5,000 epochs in sequestration



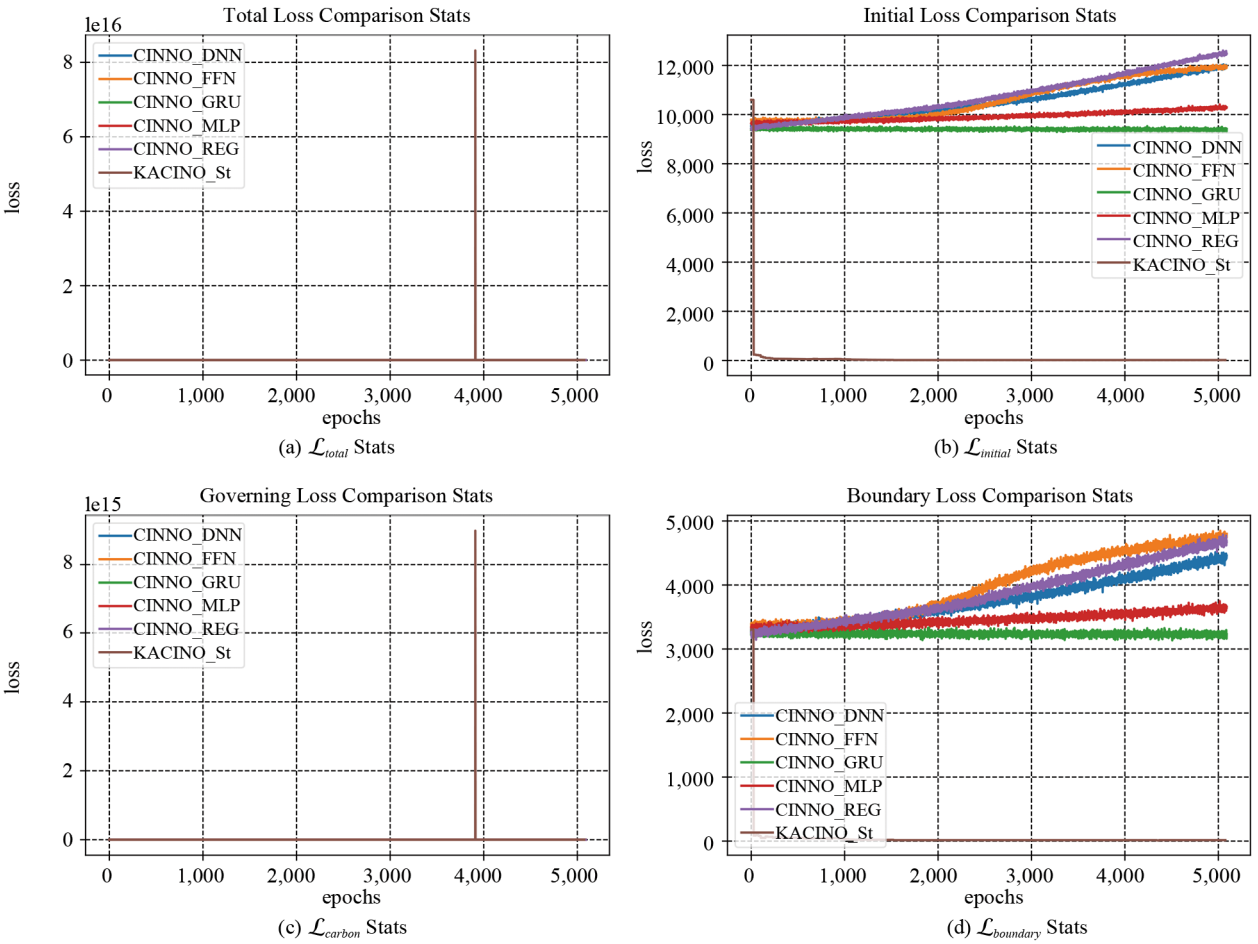
(a) $\mathcal{L}_{total}$ Stats (b) $\mathcal{L}_{initial}$ Stats

(c) $\mathcal{L}_{carbon}$ Stats (d) $\mathcal{L}_{boundary}$ Stats



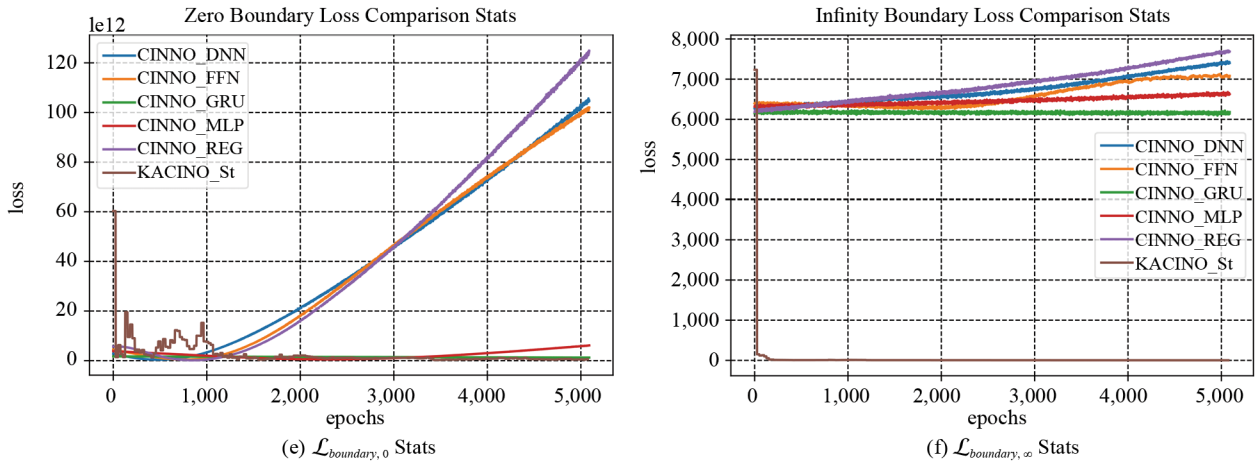
(e) $\mathcal{L}_{boundary,\,0}$ Stats (f) $\mathcal{L}_{boundary,\,\infty}$ Stats

Figure 4. Overall comparison of loss among CINNOs on emission



(a) $\mathcal{L}_{total}$ Stats (b) $\mathcal{L}_{initial}$ Stats

(c) $\mathcal{L}_{carbon}$ Stats (d) $\mathcal{L}_{boundary}$ Stats



(e) $\mathcal{L}_{boundary,\,0}$ Stats (f) $\mathcal{L}_{boundary,\,\infty}$ Stats

Figure 5. Overall comparison of loss among CINNOs on emission

6.3 *Emission analysis*

CINNO REG achieves the lowest mean total loss of 3.82×10^3, demonstrating superior average performance in minimizing total loss compared to other methods. This value is significantly lower than those of CINNO DNN and CINNO FFN, indicating mean total losses of 9.01×10^3 and 9.41×10^3, respectively. Although CINNO DNN and CINNO FFN are competitive, they are less efficient in minimizing total loss compared to CINNO REG. On the other hand, KACINO exhibits a considerably higher mean total loss of 2.68×10^8, indicating a significant performance gap compared to the CINNO methods

Table 3. Total loss stats with 5,000 epochs

Emission	CINNO DNN	CINNO FFN	CINNO GRU	CINNO MLP	CINNO REG	KACINO
mean	9.01×10^3	9.41×10^3	1.07×10^4	1.03×10^4	3.82×10^3	2.68×10^8
std	7.84×10^2	7.66×10^2	4.08×10^1	3.30×10^2	3.26×10^2	1.77×10^{10}
min	7.96×10^3	8.30×10^3	1.05×10^4	9.74×10^3	3.31×10^3	4.82×10^0
max	1.11×10^4	1.11×10^4	1.08×10^4	1.10×10^4	4.67×10^3	1.26×10^{12}
Sequestration	CINNO DNN	CINNO FFN	CINNO GRU	CINNO MLP	CINNO REG	KACINO
mean	7.93×10^5	7.67×10^5	8.39×10^5	8.26×10^5	7.83×10^5	1.67×10^{13}
std	3.40×10^4	4.97×10^4	3.79×10^3	1.08×10^4	4.00×10^4	1.17×10^{15}
min	7.21×10^5	6.88×10^5	8.22×10^5	7.95×10^5	7.03×10^5	5.09×10^1
max	8.51×10^5	8.52×10^5	8.53×10^5	8.51×10^5	8.48×10^5	8.32×10^{16}

However, despite KACINO's higher mean total loss, it demonstrates remarkable convergence capabilities. The high standard deviation (1.77×10^{10}) observed in KACINO, while indicating variability, also reflects its robustness in rapidly reducing loss values over time. KACINO's ability to quickly bring the loss back towards zero, regardless of initial system initialization, highlights its effectiveness in reaching optimal solutions faster and more stably. This characteristic allows KACINO to recover from suboptimal initial conditions and converge to minimal loss levels efficiently.

In comparison, CINNO GRU shows the lowest standard deviation (4.08×10^1), indicating its stable performance in total loss across epochs as an advantageous application inconsistent outcomes. CINNO REG and CINNO MLP achieve lower minimum total losses (3.31×10^3 and 9.74×10^3, respectively) compared to other methods, but their maximum total

losses are relatively higher, with CINNO REG reaching 4.67×10^3 and CINNO MLP reaching 1.10×10^4. KACINO's extensive range in total loss values, with a minimum of 4.82×10^0 and a maximum of 1.26×10^{12}, underscores its variability. Nonetheless, the KACINO framework's capacity to achieve rapid and effective convergence demonstrates its superior ability to quickly stabilize performance and reach optimal solutions, regardless of initial performance challenges.

In terms of Governing Loss, CINNO REG exhibits the lowest mean emission Governing Loss at 1.22×10^3 among the methods evaluated, indicating superior average performance in minimizing this metric. This is closely followed by CINNO FFN with a mean loss of 1.23×10^3 and CINNO DNN with 1.35×10^3. The other CINNO variant, CINNO GRU and CINNO MLP, show mean losses of 1.32×10^3 and 1.33×10^3, respectively, which are slightly higher but still competitive within the evaluated methods.

CINNO DNN shows the lowest standard deviation (1.35×10^1), which reflects its relatively stable performance across epochs. CINNO GRU and CINNO MLP also demonstrate low standard deviations (1.19×10^1 and 1.17×10^1, respectively), indicating consistency in their loss performance. Conversely, CINNO FFN and CINNO REG have higher standard deviations (6.95×10^1 and 7.84×10^1), suggesting greater variability in their performance. The minimum governing Loss values are relatively close across the CINNO methods, with CINNO FFN and CINNO REG achieving the lowest minimum values of 1.08×10^3. CINNO DNN and CINNO MLP have slightly higher minimum losses (1.30×10^3).

KACINO's minimum governing Loss (9.43×10^0) is substantially lower but comes with a significantly higher maximum value (3.29×10^{12}). It also exhibits an exceptionally high standard deviation (4.62×10^{10}) Despite the significantly higher mean Governing Loss of 6.97×10^8 observed for KACINO, it is noteworthy for its rapid convergence to minimal loss values. KACINO demonstrates strong convergence properties, achieving near-minimal losses relatively quickly after 200 epochs, following its initially high loss values. This indicates that while KACINO starts with a higher loss, its ability to converge efficiently over time is a key strength, highlighting its potential effectiveness in scenarios where rapid improvement and convergence are critical.

.

6.4 *Sequestration analysis*

KACINO demonstrates an exceptionally high mean sequestration of 1.67×10^{13}, significantly surpassing the other methods. This high mean value indicates KACINO's substantial capacity in sequestration, highlighting its superior performance in this regard. The CINNO methods, including CINNO DNN and CINNO FFN, have comparable but lower mean sequestration values, making them competitive yet unable to match KACINO's capacity.

Although KACINO's mean sequestration is notably high, it also exhibits the highest standard deviation in sequestration (1.17×10^{15}), reflecting considerable variability in its performance. This variability is contrasted by CINNO GRU, which shows the lowest standard deviation (3.79×10^3), indicative of more consistent performance across epochs. Such consistency is advantageous in applications demanding reliable and steady results.

Moreover, KACINO displays a broad range of sequestration values, with minimum and maximum values spanning from 5.09×10^1 to 8.32×10^{16}, revealing the greatest variability among the methods. Despite this variability, it is noteworthy that KACINO achieves rapid convergence to minimal loss values, demonstrating its effective convergence capabilities. In contrast, CINNO DNN, CINNO FFN, and other CINNO methods show narrower ranges and more consistent sequestration values, with CINNO GRU reaching the highest maximum sequestration of 8.53×10^5. Thus, while KACINO's metrics are higher, its robust convergence to minimal loss illustrates its strong performance improvement over time.

In terms of the governing loss in sequestration, KACINO stands out with a remarkably high mean sequestration value of 1.80×10^{12}. This performance is orders of magnitude higher than that of the CINNO methods, which have mean sequestration values ranging from 7.56×10^5 to 8.30×10^5. Among the CINNO methods, CINNO GRU and CINNO MLP show slightly higher mean sequestration values (8.30×10^5 and 8.16×10^5, respectively) compared to CINNO DNN, CINNO FFN, and CINNO REG.

KACINO also displays the highest standard deviation in sequestration governing loss (1.26×10^{14}), reflecting considerable variability in its performance. In contrast, CINNO GRU shows the lowest standard deviation (3.79×10^3),

indicating greater stability in its sequestration performance. CINNO DNN and CINNO MLP have moderate standard deviations (3.47×10^4 and 1.10×10^4, respectively), while CINNO FFN and CINNO REG exhibit higher variability.

KACINO demonstrates the widest range in sequestration governing loss values, with a minimum of 3.44×10^2 and a maximum of 8.98×10^{15}. This broad range underscores its highly variable performance. Conversely, the CINNO methods have narrower ranges, with their maximum sequestration values reaching up to 8.43×10^5 for CINNO GRU. The minimum values for the CINNO methods are more consistent, ranging from 6.76×10^5 to 8.12×10^5.

6.5 *Comparative discussion*

KACINO shows impressive performance in sequestration but with substantial variability. This high performance in magnitude comes with significant unpredictability, This suggests that KACINO may be effective in specific contexts, but its overall reliability is questionable.

CINNO REG stands out due to its low average total loss and stable performance with a relatively low standard deviation. However, its maximum total loss is higher than other CINNO methods, suggesting that while it performs well on average, it may encounter higher losses in some cases. CINNO GRU and CINNO FFN offer more consistent performance in both total loss and sequestration. This stability makes them suitable for applications where dependable and predictable results are essential. CINNO DNN and CINNO MLP exhibit moderate performance, being outperformed by CINNO REG in total loss and lagging behind KACINO in sequestration.

7. Conclusion

In this paper, we propose a KACINO framework for carbon dynamic system orchestration. Insights from the orchestration framework enable policymakers, businesses, and stakeholders to derive actionable recommendations. These recommendations are crucial for optimizing carbon credit utilization and advancing climate goals. Strategies may include implementing policies that incentivize emission reductions, promoting investments in carbon sequestration projects, and fostering innovative technologies to enhance efficiency in carbon trading mechanisms.

The orchestration framework enables stakeholders to assess the effectiveness of different interventions. This analytical rigor supports evidence-based policy formulation and strategic planning aimed at mitigating climate change impacts and fostering sustainable development pathways. Thus, carbon credit modeling orchestration enhances our understanding of carbon markets and empowers decision-makers with insights to navigate toward a more sustainable and resilient future.

Table 4. Governing loss stats with 5,000 epochs

Emission	CINNO DNN	CINNO FFN	CINNO GRU	CINNO MLP	CINNO REG	KACINO
mean	1.35×10^3	1.23×10^3	1.32×10^3	1.33×10^3	1.22×10^3	6.97×10^8
std	1.35×10^1	6.95×10^1	1.19×10^1	1.17×10^1	7.84×10^1	4.62×10^{10}
min	1.30×10^3	1.08×10^3	1.27×10^3	1.30×10^3	1.08×10^3	9.43×10^0
max	1.39×10^3	1.36×10^3	1.36×10^3	1.39×10^3	1.37×10^3	3.29×10^{12}
Sequestration	CINNO DNN	CINNO FFN	CINNO GRU	CINNO MLP	CINNO REG	KACINO
mean	7.82×10^5	7.56×10^5	8.30×10^5	8.16×10^5	7.72×10^5	1.80×10^{12}
std	3.47×10^4	5.05×10^4	3.79×10^3	1.10×10^4	4.09×10^4	1.26×10^{14}
min	7.09×10^5	6.76×10^5	8.12×10^5	7.84×10^5	6.91×10^5	3.44×10^2
max	8.41×10^5	8.42×10^5	8.43×10^5	8.42×10^5	8.39×10^5	8.98×10^{15}

Conflict of interest

The authors declare no competing financial interest.

References

[1] Masson-Delmotte V, Zhai P, Pirani A, Connors SL, Péan C, Berger S, et al. Climate change 2021: The physical science basis. *Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. 2021; 2(1): 2391.

[2] Balmford A, Keshav S, Venmans F, Coomes D, Groom B, Madhavapeddy A, et al. Realizing the social value of impermanent carbon credits. *Nature Climate Change*. 2023; 13(11): 1172-1178.

[3] Mathews JA. How carbon credits could drive the emergence of renewable energies. *Energy Policy*. 2008; 36(10): 3633-3639.

[4] Ramstein C, Dominioni G, Ettehad S, Lam L, Quant M, Zhang J, et al. *State and Trends of Carbon Pricing 2019*. USA: The World Bank; 2019.

[5] Asadnabizadeh M, Moe E. A review of global carbon markets from kyoto to paris and beyond: The persistent failure of implementation. *Frontiers in Environmental Science*. 2024; 12: 1368105. Available from: https://doi.org/10.3389/fenvs.2024.1368105.

[6] Santos C, Coelho A, Marques A. A systematic literature review on greenwashing and its relationship to stakeholders: State of art and future research agenda. *Management Review Quarterly*. 2024; 74(3): 1397-1421.

[7] Yang Z, Nguyen TTH, Nguyen HN, Nguyen TTN, Cao TT. Greenwashing behaviours: Causes, taxonomy and consequences based on a systematic literature review. *Journal of Business Economics and Management*. 2020; 21(5): 1486-1507.

[8] Adamkiewicz J, Kochańska E, Adamkiewicz I, Łukasik RM. Greenwashing and sustainable fashion industry. *Current Opinion in Green and Sustainable Chemistry*. 2022; 38: 100710. Available from: https://doi.org/10.1016/j.cogsc.2022.100710.

[9] Kreibich N, Hermwille L. Caught in between: Credibility and feasibility of the voluntary carbon market post-2020. *Climate Policy*. 2021; 21(7): 939-957.

[10] Song J, Wu D. Modeling forest carbon sink trading with carbon credit using stochastic differential game. *Environmental Science and Pollution Research*. 2023; 30(26): 68934-68950.

[11] Yadav V, Malanson GP, Bekele E, Lant C. Modeling watershed-scale sequestration of soil organic carbon for carbon credit programs. *Applied Geography*. 2009; 29(4): 488-500.

[12] Li Y, Ukkusuri SV, Fan J. Managing congestion and emissions in transportation networks with dynamic carbon credit charge scheme. *Computers and Operations Research*. 2018; 99: 90-108. Available from: https://doi.org/10.1016/j.cor.2018.06.014.

[13] Fedi L. The monitoring, reporting and verification of ships' carbon dioxide emissions: A European substantial policy measure towards accurate and transparent carbon dioxide quantification. *Ocean Yearbook Online*. 2017; 31(1): 381-417.

[14] Wood RG. *Carbon Finance and Pro-Poor Co-Benefits: The Gold Standard and Climate, Community and Biodiversity Standards*. London: International Institute for Environment and Development; 2011.

[15] Huisingh D, Zhang Z, Moore JC, Qiao Q, Li Q. Recent advances in carbon emissions reduction: Policies, technologies, monitoring, assessment and modeling. *Journal of Cleaner Production*. 2015; 103: 1-12. Available from: https://doi.org/10.1016/j.jclepro.2015.04.098.

[16] Cheng Y, Zhang N, Wang Y, Yang J, Kang C, Xia Q. Modeling carbon emission flow in multiple energy systems. *IEEE Transactions on Smart Grid*. 2018; 10(4): 3562-3574.

[17] Gao H, Wang X, Wu K, Zheng Y, Wang Q, Shi W, et al. A review of building carbon emission accounting and prediction models. *Buildings*. 2023; 13(7): 1617.

[18] Masera OR, Garza-Caligaris J, Kanninen M, Karjalainen T, Liski J, Nabuurs G, et al. Modeling carbon sequestration in afforestation, agroforestry and forest management projects: The CO_2FIX V. 2 approach. *Ecological Modelling*. 2003; 164(2-3): 177-199.

[19] Zirkle G, Lal R, Augustin B. Modeling carbon sequestration in home lawns. *Hort Science*. 2011; 46(5): 808-814.

[20] Doraiswamy PC, McCarty GW, Hunt ERJ, Yost R, Doumbia M, Franzluebbers AJ. Modeling soil carbon sequestration in agricultural lands of Mali. *Agricultural Systems*. 2007; 94(1): 63-74.
[21] Qiu JJ, Li CS, Wang LG, Tang HJ, Li H, Ranst EV. Modeling impacts of carbon sequestration on net greenhouse gas emissions from agricultural soils in China. *Global Biogeochemical Cycles*. 2009; 23(1): 1-16.
[22] Yan Y. Integrate carbon dynamic models in analyzing carbon sequestration impact of forest biomass harvest. *Science of the Total Environment*. 2018; 615: 581-587. Available from: https://doi.org/10.1016/j.scitotenv.2017.09.326.
[23] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*. 2019; 378: 686-707. Available from: https://doi.org/10.1016/j.jcp.2018.10.045.
[24] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nature Reviews Physics*. 2021; 3(6): 422-440.
[25] Li XZ, Lin FF, Wang H, Zhang X, Ma H, Wen CY, et al. Temporal modeling for power converters with physics-in-architecture recurrent neural network. *IEEE Transactions on Industrial Electronics*. 2024; 71(11): 14111-14123.
[26] Liu H, Shen L. Forecasting carbon price using empirical wavelet transform and gated recurrent unit neural network. *Carbon Management*. 2020; 11(1): 25-37.
[27] Lv ZH, Wang NN, Lou RR, Tian YJ, Guizani M. Towards carbon neutrality: Prediction of wave energy based on improved GRU in maritime transportation. *Applied Energy*. 2023; 331(C): 120394.
[28] Chen H, Wu H, Kan T, Zhang J, Li H. Low-carbon economic dispatch of integrated energy system containing electric hydrogen production based on VMD-GRU short-term wind power prediction. *International Journal of Electrical Power and Energy Systems*. 2023; 154: 109420. Available from: https://doi.org/10.1016/j.ijepes.2023.109420.
[29] Hainaut D, Casas A. Option pricing in the heston model with physics inspired neural networks. *Annals of Finance*. 2024; 20(3): 353-376.
[30] Hainaut D. Valuation of guaranteed minimum accumulation benefits (GMABs) with physics-inspired neural networks. *Annals of Actuarial Science*. 2024; 1: 1-32. Available from: https://doi.org/10.1017/S1748499524000095.
[31] Du HH, Zhao Z, Cheng HJ, Yan JH, He QZ. Modeling density-driven flow in porous media by physics-informed neural networks for CO_2 sequestration. *Computers and Geotechnics*. 2023; 159: 105433. Available from: https://doi.org/10.1016/j.compgeo.2023.105433.
[32] Hou M, Fu H, Hu Z, Wang J, Chen Y, Yang Y. Numerical solving of generalized Black-Scholes differential equation using deep learning based on blocked residual connection. *Digital Signal Processing*. 2022; 126: 103498. Available from: https://doi.org/10.1016/j.dsp.2022.103498.
[33] Dhiman A, Hu Y. Physics informed neural network for option pricing. *arXiv:231206711*. 2023.
[34] Ibrahim AQ, Götschel S, Ruprecht D. Parareal with a physics-informed neural network as coarse propagator. In: *European Conference on Parallel Processing*. Heidelberg: Springer; 2023. p.649-663.
[35] Villarino JP, Leitao Á, Rodríguez JG. Boundary-safe PINNs extension: Application to non-linear parabolic PDEs in counterparty credit risk. *Journal of Computational and Applied Mathematics*. 2023; 425: 115041. Available from: https://doi.org/10.1016/j.cam.2022.115041.
[36] Santos DDS, Ferreira TAE. Neural network learning of black-scholes equation for option pricing. *arXiv:240505780*. 2024.
[37] Kumar H, Yadav N. Deep learning algorithms for solving differential equations: A survey. *Journal of Experimental and Theoretical Artificial Intelligence*. 2023; 1-40. Available from: https://doi.org/10.1080/0952813X.2023.2242356.
[38] Gockenbach MS. *Understanding and Implementing the Finite Element Method*. USA: Society for Industrial and Applied; 2006.
[39] Dziuk G, Elliott CM. Finite element methods for surface PDEs. *Acta Numerica*. 2013; 22: 289-396. Available from: https://doi.org/10.1017/S0962492913000056.
[40] Schneider T, Hu Y, Gao X, Dumas J, Zorin D, Panozzo D. A large-scale comparison of tetrahedral and hexahedral elements for solving elliptic pdes with the finite element method. *ACM Transactions on Graphics (TOG)*. 2022; 41(3): 1-14.
[41] Hafez R, Youssri Y. Review on Jacobi-Galerkin spectral method for linear PDEs in applied mathematics. *Contemporary Mathematics*. 2024; 5(2): 2503-2540.
[42] Zayernouri M, Ainsworth M, Karniadakis GE. A unified Petrov-Galerkin spectral method for fractional PDEs. *Computer Methods in Applied Mechanics and Engineering*. 2015; 283(1): 1545-1569.

[43] Köppen M. The curse of dimensionality. In: *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*. Berlin: Springer; 2000. p.1-22.
[44] Hutzenthaler M, Jentzen A, Kruse T, Anh Nguyen T, von Wurstemberger P. Overcoming the curse of dimensionality in the numerical approximation of semilinear parabolic partial differential equations. *Proceedings of the Royal Society A*. 2020; 476(2244): 20190630.
[45] Kunisch K, Walter D. Semiglobal optimal feedback stabilization of autonomous systems via deep neural network approximation. *ESAIM: Control, Optimisation and Calculus of Variations*. 2021; 27(16): 59.
[46] Hu Z, Shukla K, Karniadakis GE, Kawaguchi K. Tackling the curse of dimensionality with physics-informed neural networks. *Neural Networks*. 2024; 176: 106369. Available from: https://doi.org/10.2139/ssrn.4641406.
[47] Grohs P, Hornung F, Jentzen A, von Wurstemberger P. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv:180902362*. 2023.
[48] Grossmann TG, Komorowska UJ, Latz J, Schönlieb CB. Can physics-informed neural networks beat the finite element method? *IMA Journal of Applied Mathematics*. 2024; 89(1): 143-174.
[49] Bao W, Zhao Q. A structure-preserving parametric finite element method for surface diffusion. *SIAM Journal on Numerical Analysis*. 2021; 59(5): 2775-2799.
[50] Upadhyay BD, Sonigra SS, Daxini SD. Numerical analysis perspective in structural shape optimization: A review post 2000. *Advances in Engineering Software*. 2021; 155(6): 102992.
[51] Lu F, Qi L, Jiang X, Liu G, Liu Y, Chen B, et al. NNW-GridStar: interactive structured mesh generation software for aircrafts. *Advances in Engineering Software*. 2020; 145: 102803. Available from: https://doi.org/10.1016/j.advengsoft.2020.102803.
[52] Lo S. Finite element mesh generation and adaptive meshing. *Progress in Structural Engineering and Materials*. 2002; 4(4): 381-399.
[53] Huang Y, Zhang Z, Zhang X. A direct-forcing immersed boundary method for incompressible flows based on physics-informed neural network. *Fluids*. 2022; 7(2): 56.
[54] Sharma R, Shankar V. Accelerated training of physics-informed neural networks (PINNs) using meshless discretizations. *arXiv:220509332*. 2022.
[55] Zhang Z, Sun C. Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. *Structural Health Monitoring*. 2021; 20(4): 1675-1688.
[56] Perrella M, Gerbino S, Citarella R. BEM in biomechanics: Modeling advances and limitations. In: *Numerical Methods and Advanced Simulation in Biomechanics and Biological Processes*. Netherlands: Elsevier; 2018. p.145-167.
[57] Neto MA, Amaro A, Roseiro L, Cirne J, Leal R. *Engineering Computation of Structures: The Finite Element Method*. Heidelberg: Springer; 2015.
[58] Sukumar N, Srivastava A. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*. 2022; 389: 114333. Available from: https://doi.org/10.1016/j.cma.2021.114333.
[59] Xie Y, Ma Y, Wang Y. Automatic boundary fitting framework of boundary dependent physics-informed neural network solving partial differential equation with complex boundary conditions. *Computer Methods in Applied Mechanics and Engineering*. 2023; 414(6): 116139.
[60] Wandel N, Weinmann M, Neidlin M, Klein R. Spline-pinn: Approaching pdes without data using fast, physics-informed hermite-spline cnns. *arXiv:210907143*. 2019.
[61] Nabian MA, Gladstone RJ, Meidani H. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*. 2021; 36(8): 962-977.
[62] Xu Z, Zhao K, Wang J, Bashir M. Physics-informed probabilistic deep network with interpretable mechanism for trustworthy mechanical fault diagnosis. *Advanced Engineering Informatics*. 2024; 62(C): 102806.
[63] Bi H, Abhayapala TD. Point neuron learning: A new physics-informed neural network architecture. *arXiv:240816969*. 2024.
[64] Ramabathiran AA, Ramachandran P. SPINN: Sparse, physics-based, and partially interpretable neural networks for PDEs. *Journal of Computational Physics*. 2021; 445: 110600. Available from: https://doi.org/10.1016/j.jcp.2021.110600.

[65] Salvatore C, Cola VSD, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*. 2022; 92(3): 88.

[66] Cuadro CAD, Vanzulli MC, Galione PA. Exploring the capability of PINNs for solving material identification problems. *Computational Mechanics*. 2023; 40(32): 1183-1194.

[67] Sibly RM, Hone J. Population growth rate and its determinants: An overview. *Philosophical Transactions of the Royal Society of London Series B: Biological Sciences*. 2002; 357(1425): 1153-1170.

[68] Khodabin M, Maleknejad K, Rostami M, Nouri M. Interpolation solution in generalized stochastic exponential population growth model. *Applied Mathematical Modelling*. 2012; 36(3): 1023-1033.

[69] Laurmann JA, Rotty RM. Exponential growth and atmospheric carbon dioxide. *Journal of Geophysical Research: Oceans*. 1983; 88(C2): 1295-1299.

[70] Guo J, Liu W, Tu L, Chen Y. Forecasting carbon dioxide emissions in BRICS countries by exponential cumulative grey model. *Energy Reports*. 2021; 7(9): 7238-7250.

[71] Fatima S, Ali SS, Zia SS, Hussain E, Fraz TR, Khan MS. Forecasting carbon dioxide emission of Asian countries using ARIMA and simple exponential smoothing models. *International Journal of Economic and Environmental Geology*. 2019; 10(1): 64-69.

[72] Nahleh AY, Alhindawi R, Arun K, Shiwakoti N. Projection of greenhouse gas emissions for the road transport sector based on multivariate regression and the double exponential smoothing model. *Sustainability*. 2020; 12(21): 9152.

[73] Chandel AK, Jiang L, Luo Y. Microbial models for simulating soil carbon dynamics: A review. *Journal of Geophysical Research: Biogeosciences*. 2023; 128(8): 1-27.

[74] Barré P, Angers DA, Basile-Doelsch I, Bispo A, Cécillon L, Chenu C, et al. Ideas and perspectives: Can we use the soil carbon saturation deficit to quantitatively assess the soil carbon storage potential, or should we explore other strategies? *Biogeosciences Discussions*. 2017; 2017: 1-12. Available from: https://doi.org/10.5194/bg-2017-395.

[75] Stewart CE, Paustian K, Conant RT, Plante AF, Six J. Soil carbon saturation: Concept, evidence and evaluation. *Biogeochemistry*. 2007; 86(1): 19-31.

[76] Cai W, He N, Li M, Xu L, Wang L, Zhu J, et al. Carbon sequestration of Chinese forests from 2010 to 2060: Spatiotemporal dynamics and its regulatory strategies. *Science Bulletin*. 2022; 67(8): 836-843.

[77] He N, Wen D, Zhu J, Tang X, Xu L, Zhang L, et al. Vegetation carbon sequestration in Chinese forests from 2010 to 2050. *Global Change Biology*. 2017; 23(4): 1575-1584.

[78] Kridiborworn P, Chidthaisong A, Yuttitham M, Tripetchkul S. Carbon sequestration by mangrove forest planted specifically for charcoal production in Yeesarn, Samut Songkram. *Journal of Sustainable Energy and Environment*. 2012; 3(2): 87-92.

[79] Qi L, Jian-Hua Z, Yuan F, Wen-Fa X. Carbon storage and carbon sequestration potential of the forest in China. *Advances in Climate Change Research*. 2018; 14(3): 287.

[80] He G, Zhang Z, Zhu Q, Wang W, Peng W, Cai Y. Estimating carbon sequestration potential of forest and its influencing factors at fine spatial-scales: A case study of Lushan city in southern China. *International Journal of Environmental Research and Public Health*. 2022; 19(15): 9184.

[81] Argyriou A, Evgeniou T, Pontil M. Convex multi-task feature learning. *Machine Learning*. 2008; 73: 243-272. Available from: https://doi.org/10.1007/s10994-007-5040-8.

[82] Bolderman M, Fan D, Lazar M, Butler H. Generalized feedforward control using physics-informed neural networks. *IFAC-PapersOnLine*. 2022; 55(16): 148-153.

[83] Yuan L, Ni YQ, Deng XY, Hao S. A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *Journal of Computational Physics*. 2022; 462: 111260. Available from: https://doi.org/10.1016/j.jcp.2022.111260.

[84] Rao C, Sun H, Liu Y. Physics-informed deep learning for incompressible laminar flows. *Theoretical and Applied Mechanics Letters*. 2020; 10(3): 207-212.

[85] Markidis S. From complexity to simplicity: Brain-inspired modularization of pinn solvers. *arXiv:240115661*. 2024.