EXTENDING TWO EXPLAINABLE ARTIFICIAL INTELLIGENCE METHODS FOR DEEP CLIMATE EMULATORS

Wei Xu, Xihaier Luo, Yihui Ren, Shinjae Yoo
Brookhaven National Laboratory, USA
{xuw, xluo, yren, sjyoo}@bnl.gov

Luke Van Roekel, Balu T. Nadiga
Los Alamos National Laboratory, USA
{lvanroekel, balu}@lanl.gov

Rui Qiu, Han-Wei Shen
The Ohio State University, USA
qiu.580@buckeyemail.osu.edu, shen.94@osu.edu

ABSTRACT

Climate change presents a complex and critical challenge that spans forecasting across various temporal horizons to reconstruction using data from sparsely distributed sensors. Recent advancements in deep learning have shown promising results in emulating complex climate dynamics and reconstructing physical fields using real-time measurements. Given their data-driven nature, it is crucial to investigate how these deep emulators learn and represent the underlying physics. This paper aims to address these concerns by employing Explainable Artificial Intelligence (XAI) techniques, focusing specifically on two methods–feature attribution and influence functions and demonstrating their explainability of a cutting-edge implicit neural network that learns a continuous and reliable representation from sparse sampling climate data.

1 INTRODUCTION

Climate change presents a multifaceted challenge, requiring precise prediction and reconstruction of physical fields over varying temporal and spatial scales. As global climate models continue to evolve, the ability to leverage data from sparsely distributed sensors and integrate them with deep learning algorithms is essential for tackling both short-term forecasting and long-term climate projections Nguyen et al. (2023a); Rolnick et al. (2022); Verma et al. (2024); Nguyen et al. (2023b); Bi et al. (2023). Implicit Neural Representations (INRs) offer a powerful framework for mapping input coordinates of a continuous field (e.g., spatial and temporal) to output values, which can represent physical quantities like temperature, pressure or humidity at those coordinates. After being trained, INR models are capable of generalizing to unseen coordinates, allowing them to predict field values in regions where no data has been directly observed. Such a framework is particularly well-suited to the domain of climate prediction where data from sparsely distributed sensors must be leveraged to reconstruct and forecast complex environmental dynamics across spatial and time. While deep learning methods have demonstrated promising capabilities in capturing complex climate dynamics, their "*black-box*" nature raises concerns about the trustworthiness of predictions Castelvecchi (2016); Alvarez Melis & Jaakkola (2018); Alaa & van der Schaar (2019); Huang et al. (2021); Madsen et al. (2022); Kaur et al. (2022). For climate scientists and policymakers, understanding the underlying mechanisms by which these models process and interpret input data becomes a necessity.

A critical step toward this understanding is the development of methods that provide *explainability* and *transparency* in neural climate models. *Feature attribution* methods, as a branch of research in Explainable Artificial Intelligence (XAI), offer a promising pathway to explore how neural networks allocate weights to various input features—such as temperature, pressure, or sensor locations—during prediction and reconstruction tasks. Gradient-based feature attribution methods calculate the gradients of the model's output with respect to its input features, thereby indicating how sensitive the predictions are to small changes in each input Higa et al. (2021); Reulen & Mehrkanoon (2024); Toms et al. (2020). While these methods reveal how a trained model's output responds to perturbations in input features or latent variables, they do not directly address "which

training examples most influence a particular prediction". *Influence functions* Koh & Liang (2020); Guo et al. (2020) bridge this gap by quantifying how a model's parameters—and thereby its predictions—change if individual training instances are removed or re-weighted. This provides a new perspective on model behaviors, facilitating insights into data quality, labeling errors, and regions of high or low impact in the training set.

This paper focuses on extending these two types of XAI methods for an INR-type deep climate emulator and making it more understandable and able to answer domain-driven questions.

2 The Implicit Neural Network

An INR model is defined as a function $f_\theta(\mathbf{x}, t) : \mathbb{R}^d \to \mathbb{R}^k$, where f_θ is a neural network parameterized with θ to map the input space of d dimension (e.g., spatio-temporal coordinates) to the output of k dimension (e.g., the value of the physical field at a given coordinate). The training objective of an INR model is to approximate the physical field $u(\mathbf{x}, t)$ for a given set of coordinates $(\mathbf{x}, t)$. One of the defining characteristics of INR models is the use of latent representations to capture underlying semantics and context of the environmental dynamics. These latent representations z are trained to encapsulate contextual information such as data distribution, geographical features, and temporal trends, allowing the model to generalize and make predictions even when data is sparse or incomplete. In an INR with the encoder-decoder structure, a typical latent variables z_t are learned to capture contextual information from the observed data at time step t, guiding the model's predictions at unobserved locations and times:

$$z_t = E_\phi(U_t), \quad u(\mathbf{x}, t) = D_\theta(z_t, \mathbf{x}), \tag{1}$$

where, E_ϕ represents the encoder that extracts the latent representation from the observed data U_t, and D_θ is the decoder that predicts the field values based on the latent code and spatial coordinates x. In contrast, the decoder-only approaches treat latent variables z_t as a learnable code-book to assist the decoder in the reconstruction or prediction of field values from input coordinates directly. This flexibility allows INR models to effectively capture complex spatiotemporal dynamics, enhancing their utility for precise environmental predictions in climate forecasting.

In this work, we aim at explaining a specific INR model, i.e., Multiplicative and Modulated Gabor Network (MMGN) Luo et al. (2024). On top of a typical encoder-decoder INR, MMGN introduces a context-aware indexing for time and factorizes target signals into a set of multiplicative basis functions. The detailed model description is included in Appendix A.1.

3 Methodology

3.1 Feature Attribution for INR

Gradient-based feature attribution methods provide a quantitative way to assess how variations in inputs affect the model's predictions. Specifically, for MMGN, we aim to compute gradients with respect to input spatial coordinates $\mathbf{x}$ and latent representations $\mathbf{z}_t$ to quantify their impact on the predicted field (i.e., the surface temperature) $u(\mathbf{x}, t)$.

The first gradient quantifies the sensitivity of the predicted field to each spatial dimension at location $\mathbf{x}$ and time t, while the magnitude and direction of the gradient provide insights into how small perturbations in spatial coordinates affect the model's output, effectively highlighting regions of high spatial influence. Respectively, the second gradient measures the influence of the latent variables on the model's predictions, revealing how changes in this higher-level representation impact the predicted physical fields. Such analysis is particularly valuable in climate forecasting, where it is essential to comprehend the model's response to spatial variations and underlying physical processes encoded in the latent space. A detailed mathematical description can be found in Appendix A.2

3.2 Influence Function for INR

For the aforementioned INR model f_θ, let $\mathcal{D} = \{ (\mathbf{x}_i, t_i, u_i) \}_{i=1}^N$ denote the training dataset consisting of coordinate-time-value triplets. The model parameters θ are estimated by minimizing a loss

function $\mathcal{L}(\theta)$ that sums the individual losses over the entire training set:

$$\hat{\theta} \;=\; \underset{\theta}{\operatorname{argmin}} \;\frac{1}{N}\sum_{i=1}^{N} L\big(f_\theta,\,(\mathbf{x}_i,t_i,u_i)\big), \tag{2}$$

where $L(\cdot)$ is a suitable loss (e.g., mean squared error or ℓ_1-norm) for climate forecasting or reconstruction tasks. Influence functions quantify how up-weighting (or down-weighting) a single training example $\big(\mathbf{x}_j,t_j,u_j\big)$ by a small scalar ϵ would alter the model parameters, and subsequently, how this parameter shift affects predictions for a *target* input $\big(\mathbf{x}_{\text{test}},t_{\text{test}}\big)$. Formally, define a perturbed objective:

$$\mathcal{L}_\epsilon(\theta) \;=\; \frac{1}{N}\sum_{i=1}^{N} L\big(f_\theta,(\mathbf{x}_i,t_i,u_i)\big) \;+\; \epsilon\cdot L\big(f_\theta,(\mathbf{x}_j,t_j,u_j)\big). \tag{3}$$

Denote the resulting perturbed minimizer by $\hat{\theta}_\epsilon$, the *influence* of the j-th training example on the model's prediction at $\big(\mathbf{x}_{\text{test}},t_{\text{test}}\big)$ can be conceptualized via:

$$\mathcal{I}\big(\mathbf{x}_{\text{test}},t_{\text{test}};\,j\big) \;=\; \frac{\partial}{\partial\epsilon}\,f_{\hat{\theta}_\epsilon}\big(\mathbf{x}_{\text{test}},t_{\text{test}}\big)\,\Big|_{\epsilon=0}. \tag{4}$$

That is, how sensitively the model's output at a specific test coordinate/time depends on changes in the weight of the j-th training point. Influence functions Koh & Liang (2020) estimate this effect as follows:

$$\mathcal{I}\big(\mathbf{x}_{\text{test}},t_{\text{test}};\,j\big) \;\approx\; -\,\nabla_\theta f_{\hat{\theta}}\big(\mathbf{x}_{\text{test}},t_{\text{test}}\big)\;\mathbf{H}_{\hat{\theta}}^{-1}\;\nabla_\theta\,L\big(f_{\hat{\theta}},(\mathbf{x}_j,t_j,u_j)\big), \tag{5}$$

where $\mathbf{H}_{\hat{\theta}}=\nabla_\theta^2\,\mathcal{L}(\theta)\big|_{\theta=\hat{\theta}}$ is the Hessian of the training loss evaluated at $\hat{\theta}$. In practice, inverting $\mathbf{H}_{\hat{\theta}}$ directly can be computationally infeasible for large models. Instead, we use iterative approximations such as LiSSA Guo et al. (2020) to estimate $\mathbf{H}_{\hat{\theta}}^{-1}$ efficiently.

When used in conjunction with climate spatiotemporal data, influence functions can: identify key training regions, reveal data dependencies, and guide data augmentation or quality control. These properties make influence functions an appealing complement to gradient-based feature attribution for achieving a deeper and more holistic understanding of neural climate models, including INRs.

4 Results

4.1 Explaining INRs via Gradient-Based Analysis

Gradient with Respect to Spatial Coordinates. To evaluate how changes in $\mathbf{x}$ affect the predicted value $D_\phi(\mathbf{z}_t,\mathbf{x})$, we compute the gradients w.r.t. $\mathbf{x}$ across all forecasted time steps (e.g., $N=1024$) and aggregate the results by computing the standard deviation at each spatial location.

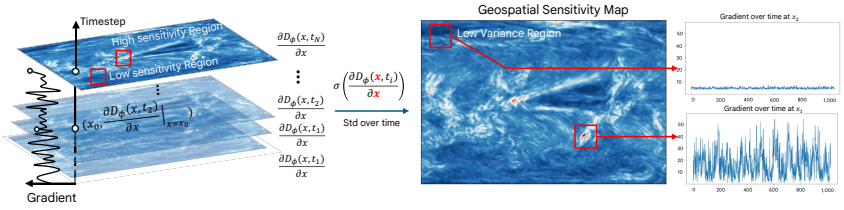

Figure 1: Standard deviation map of spatial gradients $\nabla_{\mathbf{x}} D_\phi(\mathbf{z}_t,\mathbf{x})$ over $N=1024$ time steps. Higher values (red regions) suggest areas where small changes in $\mathbf{x}$ lead to large temporal fluctuations in the predicted field.

The standard deviation map $\sigma_{\mathbf{x}}$ in Figure 1 shows that regions with larger $\sigma_{\mathbf{x}}$ exhibit stronger temporal variability in their gradients, indicating the model is more sensitive to spatial perturbations in these areas. Such high-variance zones often correspond to physically dynamic regions (e.g., ocean currents, turbulent hotspots). These findings provide insights for domain scientists: 1) suggesting additional data collection locations for uncertainty reduction, 2) revealing boundary or transition

zones where the model must capture strong spatial gradients (e.g., land-ocean boundaries), and 3) validating known physical phenomena—if the model's sensitivity map aligns with domain expertise.

Gradient with Respect to Latent Variables. We next compute the gradient of the decoder output w.r.t. the latent representation $\mathbf{z}$. Empirically, we observe that, for MMGN, $\nabla_{\mathbf{z}} D_\phi$ often appears nearly constant across spatial locations and time steps. This means that for any $\mathbf{x}$ and t, the gradient remains the same, indicating a uniform sensitivity of the model's output to changes in the latent variables. We performed a mathematical derivation based on the MMGN's architecture to understand this phenomenon. The detailed description and an exemplar result are included in Appendix A.3.

We conclude the following interpretation: 1) *Global Influence.* A near-constant gradient suggests that $\mathbf{z}_t$ encapsulates global temporal features rather than location-specific details. Perturbations in $\mathbf{z}_t$ shift the prediction uniformly across $\mathbf{x}$. 2) *High- vs. Low-Value Regions.* Even if the gradient is spatially uniform, its magnitude can vary for different latent dimensions or time steps. High-gradient dimensions imply the model is more responsive to the corresponding latent factors, whereas low-gradient dimensions have less impact. 3) *Domain Significance.* Such uniform dependence indicates that $\mathbf{z}_t$ might capture large-scale climatic cycles (e.g., seasonal trends) consistently applied over the spatial domain. This knowledge can assist domain experts in fine-tuning the physical models.

4.2 Explaining INRs via Influence Functions

A Case Study: Influence of a Single Training Example The MMGN model was trained using the simulation data generated by the Community Earth System Model version 2 (CESM2) (Danabasoglu et al., 2020), a fully coupled global climate model to simulate Earth's climate states. The data is preprocessed by taking the monthly averaged global surface temperature and removing the seasonal cycles to represent an atmospheric field.

To illustrate the practical utility of influence functions in an INR context, we focus on a scenario where our model predicts a temperature field at high resolution, given sparse observations. We select a specific target coordinate $(\mathbf{x}_{\text{test}}, t_{\text{test}})$ of interest (highlighted in yellow on the left image of Figure 2) and compute the influence $\mathcal{I}(\mathbf{x}_{\text{test}}, t_{\text{test}}; j)$ of every training point $(\mathbf{x}_j, t_j)$ on that target prediction. Our reconstruction objective is a combination of ℓ_1 loss (to measure prediction error) plus an ℓ_2 regularization on the model parameters. This objective is minimized over multiple time-stamped data loaders, each containing spatial coordinates and temperature values for a particular time step.

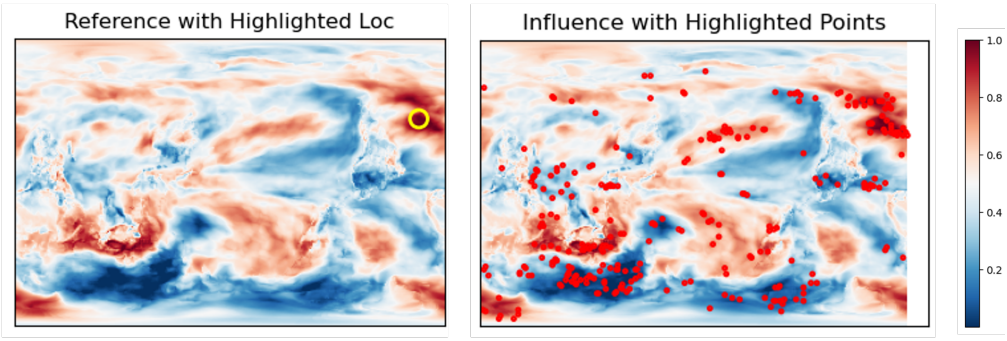

Figure 2: **Left:** The reference temperature field at a target time step, with a chosen location of interest (yellow circle). **Right:** Influence map (red dots) indicating which training coordinates exert the greatest positive or negative influence on the chosen prediction.

Figure 2 (right) visualizes the spatial distribution of influential training points, overlaid on the same reference field used for context. We observe: 1) **Strong Regional Clusters.** Red circles tend to cluster in specific areas, suggesting that certain geographic or atmospheric regions are especially correlated with the target location's temperature. 2) **Physical Plausibility.** Many of these high-influence training points can potentially relate to regions of strong atmospheric currents or ocean boundaries, aligning with known teleconnections in climate systems. 3) **Potential Data Issues.** If spurious or noisy measurements in a high-influence region are detected, re-checking or re-sampling the data points can substantially improve the prediction quality at the target location.

5 CONCLUSION

In this paper, we introduced the adoption of two XAI approaches–feature attribution and influence functions–enhancing the understanding of an INR model for continuous representation learning built with sparsely sampled climate data. We demonstrated the interpretation capability from two distinct perspectives by answering, *which coordinates or latent features matter most* as well as *which training examples matter most*, for the prediction of interest. As a work in progress, we are working on fusing the two approaches together in order to provide a holistic understanding of the INR model's behavior and data dependencies. The result part will also be improved toward more quantitative evaluation and case studies with domain scientists. Finally, we plan to test the generalizability to other INR models.

6 ACKNOWLEDGEMENT

This research is supported by the U.S. Department of Energy (DOE) Office of Science, Office of Advanced Scientific Computing Research, and Office of Biological and Environmental Research, through the Scientific Discovery through Advanced Computing (SciDAC) program, under Contract No. DE-SC0012704, Award No. DE-AC02-06CH11357, and FWP No. CC122. There was also support from Brookhaven National Laboratory's LDRD #24-063. Additional support was provided by the DOE through the Los Alamos National Laboratory, which is operated by Triad National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy (Contract No. 89233218CNA000001).

REFERENCES

Ahmed M Alaa and Mihaela van der Schaar. Demystifying black-box models with symbolic metamodels. *Advances in neural information processing systems*, 32, 2019.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

Gokhan Danabasoglu, J-F Lamarque, J Bacmeister, DA Bailey, AK DuVivier, Jim Edwards, LK Emmons, John Fasullo, R Garcia, Andrew Gettelman, et al. The community earth system model version 2 (cesm2). *Journal of Advances in Modeling Earth Systems*, 12(2):e2019MS001916, 2020.

Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*, 2020.

Maiki Higa, Shinya Tanahara, Yoshitaka Adachi, Natsumi Ishiki, Shin Nakama, Hiroyuki Yamada, Kosuke Ito, Asanobu Kitamoto, and Ryota Miyata. Domain knowledge integration into deep learning for typhoon intensity classification. *Scientific reports*, 11(1):12972, 2021.

Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. *Advances in Neural Information Processing Systems*, 34:22745–22757, 2021.

Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durresi. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)*, 55(2):1–38, 2022.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions, 2020. URL `https://arxiv.org/abs/1703.04730`.

Xihaier Luo, Wei Xu, Yihui Ren, Shinjae Yoo, and Balu Nadiga. Continuous field reconstruction from sparse observations with implicit neural networks. *arXiv preprint arXiv:2401.11611*, 2024.

Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42, 2022.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. In *International Conference on Machine Learning*, pp. 25904–25938. PMLR, 2023a.

Tung Nguyen, Jason Kyle Jewik, Hritik Bansal, Prakhar Sharma, and Aditya Grover. Climatelearn: Benchmarking machine learning for weather and climate modeling. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023b. URL https://openreview.net/forum?id=RZJEkLFlPx.

Eloy Reulen and Siamak Mehrkanoon. Ga-smaat-gnet: Generative adversarial small attention gnet for extreme precipitation nowcasting. *arXiv preprint arXiv:2401.09881*, 2024.

David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

Benjamin A Toms, Elizabeth A Barnes, and Imme Ebert-Uphoff. Physically interpretable neural networks for the geosciences: Applications to earth system variability. *Journal of Advances in Modeling Earth Systems*, 12(9):e2019MS002002, 2020.

Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xuY33XhEGR.

A Appendix

This appendix presents supplementary materials

A.1 MMGN

The MMGN model is defined recursively as:

$$\begin{aligned} h^{(1)} &= g_1(\mathbf{x}), \\ h^{(i+1)} &= g_i(\mathbf{x}) \odot \left(W_h^{(i)} h^{(i)} + W_z^{(i)} \mathbf{z} + \mathbf{b}_h^{(i)}\right), \quad i = 1, \ldots, L-1, \\ D_\phi(\mathbf{z}, \mathbf{x}) &= W^{(L)} h^{(L)} + \mathbf{b}^{(L)}, \end{aligned} \tag{6}$$

where

- $\mathbf{x} \in \mathbb{R}^d$ is the spatial coordinate,
- $\mathbf{z} \in \mathbb{R}^m$ is the latent representation at time t,
- $h^{(i)}$ is the hidden feature at layer i,
- $g_i(\mathbf{x})$ is a modulation function (e.g., a Gabor function) that may incorporate sinusoidal components,
- $W_h^{(i)}, W_z^{(i)}$ are weight matrices, $\mathbf{b}_h^{(i)}$ is a bias vector,
- $\odot$ denotes element-wise multiplication.

A.2 Gradient-based Feature Attribution

First, the gradient of the output with respect to the spatial coordinates $\mathbf{x}$ is given by:

$$\nabla_{\mathbf{x}} u(\mathbf{x}, t) = \frac{\partial D_\theta(\mathbf{z}_t, \mathbf{x})}{\partial \mathbf{x}} = \left[\frac{\partial D_\theta(\mathbf{z}_t, \mathbf{x})}{\partial x_1} \vdots \frac{\partial D_\theta(\mathbf{z}_t, \mathbf{x})}{\partial x_d} \right] \in \mathbb{R}^d. \tag{7}$$

This gradient vector quantifies the sensitivity of the predicted field to each spatial dimension at location $\mathbf{x}$ and time t. The magnitude and direction of $\nabla_{\mathbf{x}} u(\mathbf{x}, t)$ provide insights into how small perturbations in spatial coordinates affect the model's output, effectively highlighting regions of high spatial influence.

Second, the gradient of the output with respect to the latent representation $\mathbf{z}_t$ is given by:

$$\nabla_{\mathbf{z}_t} u(\mathbf{x}, t) = \frac{\partial D_\theta(\mathbf{z}_t, \mathbf{x})}{\partial \mathbf{z}_t} \in \mathbb{R}^m. \tag{8}$$

This gradient measures the influence of the latent variables on the model's predictions. Since $\mathbf{z}_t$ encapsulates contextual information from the observed data U_t, $\nabla \mathbf{z}_t u(\mathbf{x}, t)$ reveals how changes in this higher-level representation impact the predicted physical fields. By computing these gradients, we can identify which input features and latent variables most significantly affect the model's output. Such analysis is particularly valuable in climate forecasting, where it is essential to comprehend the model's response to spatial variations and underlying physical processes encoded in the latent space.

A.3 Mathematical Derivation of the Attribution of Latent Variables

Figure 1 illustrates an example of these latent gradients at multiple time steps.

Starting from the output layer, the derivative of the decoder output with respect to $\mathbf{z}$ is given by:

$$\frac{\partial D_\phi}{\partial \mathbf{z}} = W^{(L)} \frac{\partial h^{(L)}}{\partial \mathbf{z}}. \tag{9}$$

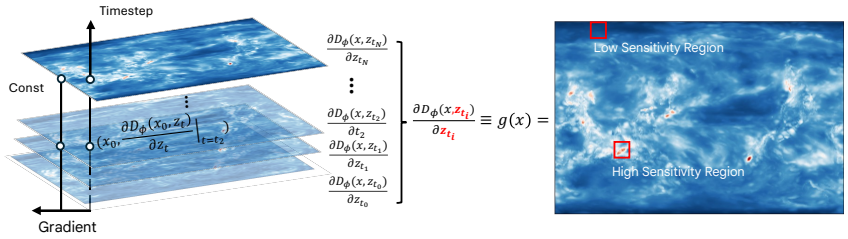

Figure 1: Latent gradient maps across multiple time steps. The near-constant patterns indicate a uniform effect of perturbations in the latent variable $\mathbf{z}_t$ on the model's output, independent of $\mathbf{x}$.

We recursively expanded $\frac{\partial h^{(i)}}{\partial \mathbf{z}}$ through each layer of the network:

$$\begin{aligned}
\frac{\partial h^{(L)}}{\partial \mathbf{z}} &= g_{L-1}(\mathbf{x}) \odot \left(W_z^{(L-1)} + W_h^{(L-1)} \frac{\partial h^{(L-1)}}{\partial \mathbf{z}} \right), \\
\frac{\partial h^{(L-1)}}{\partial \mathbf{z}} &= g_{L-2}(\mathbf{x}) \odot \left(W_z^{(L-2)} + W_h^{(L-2)} \frac{\partial h^{(L-2)}}{\partial \mathbf{z}} \right), \\
&\vdots \\
\frac{\partial h^{(2)}}{\partial \mathbf{z}} &= g_1(\mathbf{x}) \odot W_z^{(1)}.
\end{aligned} \tag{10}$$

By unrolling these recursive equations, we noticed that each term $\frac{\partial h^{(i)}}{\partial \mathbf{z}}$ depends on the product of the modulation functions $g_j(\mathbf{x})$ and weight matrices $W_z^{(j)}$ and $W_h^{(j)}$, but crucially, the modulation functions are applied element-wise and are independent of $\mathbf{z}$. This structure leads to the gradient $\frac{\partial D_\phi}{\partial \mathbf{z}}$ being a composition of constant terms and modulation functions that, due to their multiplicative nature, result in the gradient being constant with respect to $\mathbf{x}$:

$$\frac{\partial D_\phi}{\partial \mathbf{z}} = \left(W^{(L)} \prod_{i=1}^{L-1} g_i(\mathbf{x}) \odot W_z^{(i)} \right). \tag{11}$$

Since the modulation functions $g_i(\mathbf{x})$ are deterministic and depend only on $\mathbf{x}$, and the weight matrices $W_z^{(i)}$ are fixed after training, the entire expression evaluates to a constant with respect to $\mathbf{x}$ and t.