Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Speakers

Ann Bostrom (UW)

Mariana Cains (NCAR)

Christopher Wirz (NCAR)

Amy McGovern (OU)

Imme Ebert-Uphoff (CSU)

Randy Chase (OU)

Antonios Mamalakis (CSU)

ARTIFICIAL INTELLIGENCE · ENVIRONMENTAL SCIENCE · RISK COMMUNICATION

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

LEAP

NOAA CENTER FOR ARTIFICIAL INTELLIGENCE

Radiant Earth Foundation
EARTH IMAGERY FOR IMPACT

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Goals

- Learn about explainability and interpretability, as well as how users think about the concepts
- Learn how to use attribution maps to gain insights into strategies a NN is using, including
  - Different types of attribution maps
  - Common pitfalls and how to interpret attribution maps

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Agenda

- 9:00 Explainability vs. Interpretability
- 9:45 *Short brain & bio break #1*
- 9:50 XAI techniques for deep learning (Part 1)
- 11:10 *Short brain & bio break #2*
- 11:15 XAI techniques for deep learning (Part 2)
- Noon: End of session

Questions?

https://app.sli.do/event/1zumy91n

Or go to sli.do and use the code TAI4ES

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Agenda

- **9:00 Explainability vs. Interpretability**
- 9:45 *Short brain & bio break #1*
- 9:50 XAI techniques for deep learning (Part 1)
- 11:10 *Short brain & bio break #2*
- 11:15 XAI techniques for deep learning (Part 2)
- Noon: End of session

Questions?

https://app.sli.do/event/1zumy91n

Or go to sli.do and use the code TAI4ES

Speakers for the first part

Ann
Bostrom
(UW)

Mariana
Cains
(NCAR)

Christopher
Wirz
(NCAR)

Warm-up and refresher from yesterday

Let’s do couple quick questions to get us back in the trustworthy AI mindset:

1. What words/phrases would you use to describe “**trustworthy** AI?”

2. What was your favorite part of yesterday’s lectures?

2.1. & 2.2. Go to sli.do and use the code TAI4ES

Explainability vs Interpretability

Now that you're warmed up, let's think about today's topic

What is "**explainable** AI"?

What is "**interpretable** AI"?

2.3. & 2.4. Go to sli.do and use the code TAI4ES

Explainability and Interpretability

TABLE 1. A non-exhaustive list of definitions of interpretability and explainability provided in the literature. Many studies not included here do not define the terms and use them interchangeably.

Source	**Interpretability**	**Explainability**
Murdoch et al. (2019)	[. . .] the use of machine-learning models for the extraction of relevant knowledge about domain relationships contained in data	N/A (no distinction made)
Rudin (2018)	An interpretable machine learning model is constrained in model form so that it is either useful to someone, or obeys structural knowledge of the domain such as [. . .] the physical constraints that come from domain knowledge.	[. . .] where a second (post-hoc) model is created to explain the black box model
Gilpin et al. (2018)	[. . .] describe the internals of a system in a way which is understandable to humans	[. . .] models that are able summarize the reasons for [black box] behavior [...] or produce insights about the causes of their decisions
Rudin et al. (2021)	An interpretable ML model obeys a domain-specific set of constraints to allow it to be more easily understood by humans. These constraints can differ dramatically depending on the domain.	Explaining a black box model with a simpler model
Doshi-Velez and Kim (2017)	[. . .] the ability to explain or to present [the model] in understandable terms to a human.	explaining the model after it is trained with post-hoc methods
Wikipedia	describes the possibility to comprehend the ML model and to present the underlying basis for decision-making in a way that is understandable to humans.	the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g., classification or regression)
Linardatos et al. (2020)	[ability] to identify cause-and-effect relationships within the system's inputs and outputs.	Explainability, [...] is associated with the internal logic and mechanics that are inside a machine learning system.
Miller (2019)	the degree to which a human can understand the cause of a decision.	N/A (no distinction is made)

Unfortunately, there is only partial consensus in the literature about the definition of these terms, and many important papers treat them as interchangeable (see Table 1).

From Flora et al. (2022, in prep.)

Explainability and Interpretability

Table 1. A non-exhaustive list of definitions of interpretability and explainability provided in the literature. Many studies not included here do not define the terms and use them interchangeably.

Source	Interpretability	Explainability
Murdoch et al. (2019)	[. . .] the use of machine-learning models for the extraction of relevant knowledge about domain relationships contained in data	N/A (no distinction made)
Rudin (2018)	An interpretable machine learning model is constrained in model form so that it is either useful to someone, or obeys structural knowledge of the domain such as [. . .] the physical constraints that come from domain knowledge.	[. . .] where a second (post-hoc) model is created to explain the black box model
Gilpin et al. (2018)	[. . .] describe the internals of a system in a way which is understandable to humans	[. . .] models that are able summarize the reasons for [black box] behavior [...] or produce insights about the causes of their decisions
Rudin et al. (2021)	An interpretable ML model obeys a domain-specific set of constraints to allow it to be more easily understood by humans. These constraints can differ dramatically depending on the domain.	Explaining a black box model with a simpler model
Doshi-Velez and Kim (2017)	[. . .] the ability to explain or to present [the model] in understandable terms to a human.	explaining the model after it is trained with post-hoc methods
Wikipedia	describes the possibility to comprehend the ML model and to present the underlying basis for decision-making in a way that is understandable to humans.	the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g., classification or regression)
Linardatos et al. (2020)	[ability] to identify cause-and-effect relationships within the system's inputs and outputs.	Explainability, [...] is associated with the internal logic and mechanics that are inside a machine learning system.
Miller (2019)	the degree to which a human can understand the cause of a decision.	N/A (no distinction is made)

From Flora et al. (2022, in prep.); McGovern et al. (2022 submitted, BAMS)

AI2ES Definitions:

Interpretability: The degree to which a human can derive meaning from the entire model and its components **without the aid of additional methods.**

Explainability: The degree to which a human can derive meaning from the entire model and its components **through the use of post-hoc methods** (e.g., verification, visualizations of important predictors, etc).

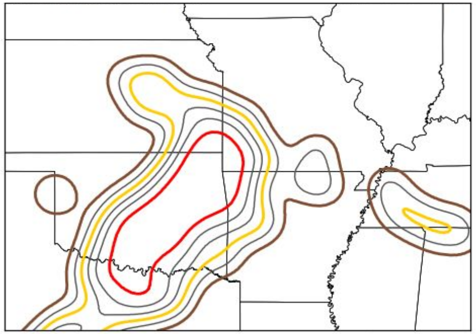
Overview of Interview Process

Interviewed National Weather Service Forecasters:

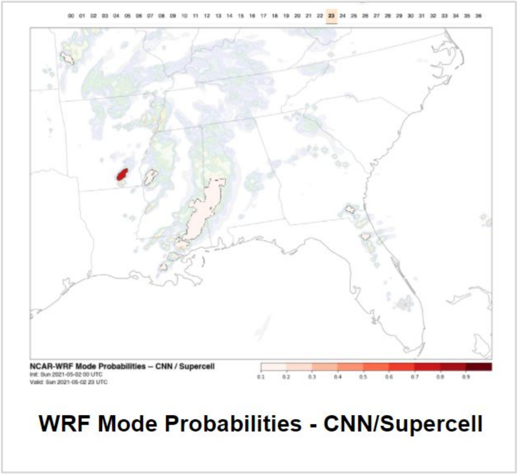
- 14 forecasters from Central, Southern, and Eastern Regions
- 7 GS 5-12 meteorologists, 4 lead meteorologist, and 5 science & operations officers

Topics covered in the interviews:

- **Perceptions of and attitudes toward AI and AI trustworthiness**
- Perceptions of and feedback about AI/ML convective forecast guidance for two products

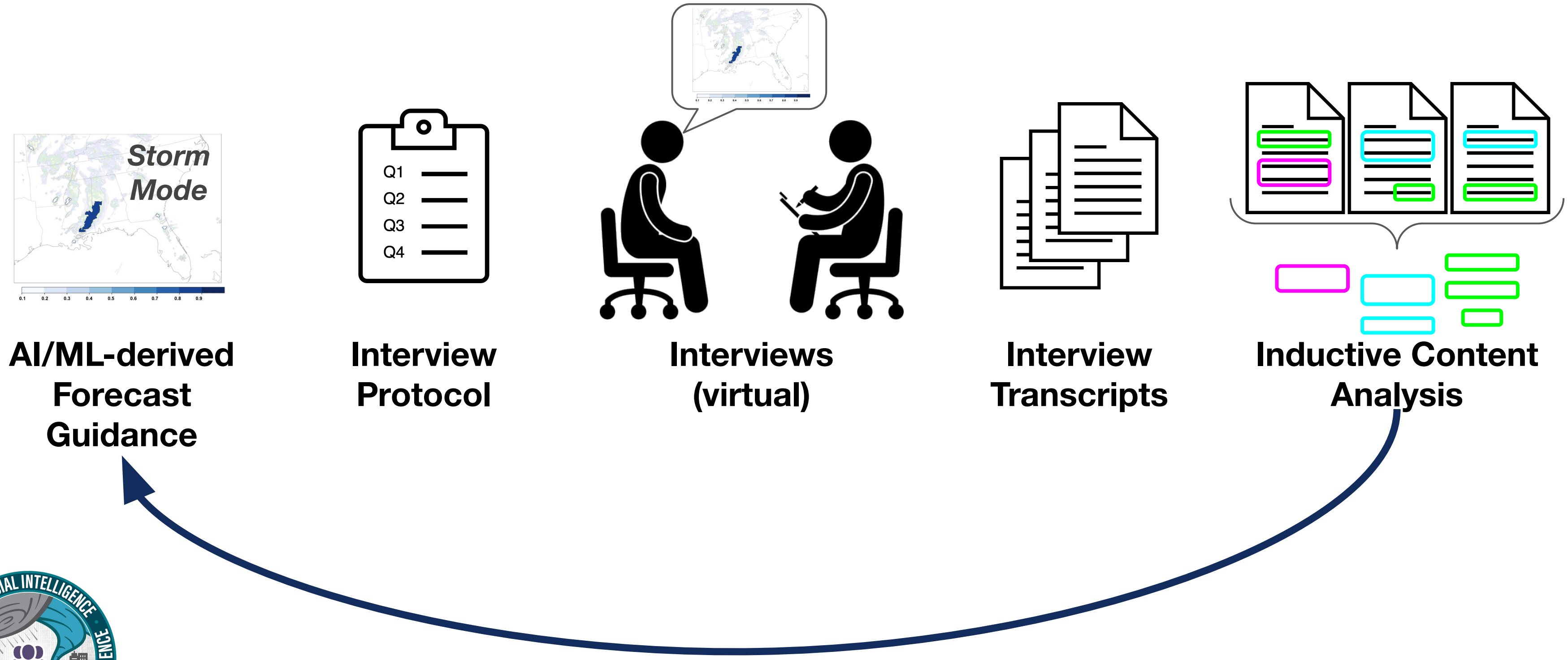


Probability of hail: Burke *et al.*, 2020



Storm mode: Sobash *et al.*, in prep.

Qualitative Data Collection and Analysis



What AI/ML explainability and interpretability mean to NWS forecasters:

Interview question prompt*:*

"These are just terms that have been used a lot in certain academic and developer circles and we're trying to get a sense of what they mean to potential users. ***Note, there is no right or wrong answer.****"*

Explainability:

What do you think about the term ***"explainability"?*** *What might it mean in this context of AI/ML guidance?*

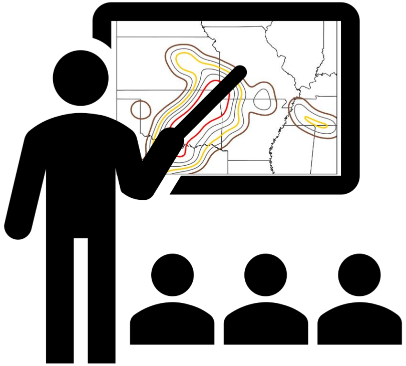
Interpretability:

What do you think about the term ***"interpretability"?*** *What might it mean in this context of AI/ML guidance?*

Results: Explainability Overview

Explainability:

1. Forecaster and guidance
2. Forecaster, guidance, others (e.g., forecasters, core partners)



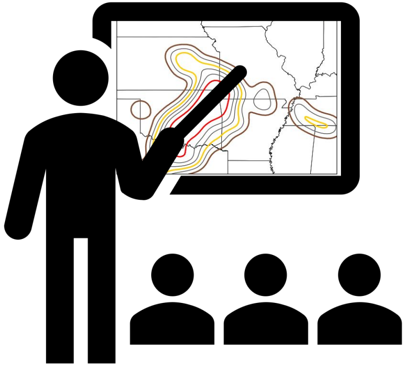
Results: Explainability Theme 1

<u>Forecaster 10:</u> “There’s explainable to other people in the meteorology community. **Can I explain what this model is showing to me to another forecaster** or can I explain it to the lead forecaster, something like that in our office and **are they going to understand it?**

If I need to explain this to a partner – say we’re really concerned about severe weather and they say, ‘why’, can I explain? You know, if I’m relying on this new model **can I explain what this model is showing in a way that they’re going to be able to, in plain language, get them to buy into what we’re trying to tell them.**”

Explainability:

1. **Forecaster and guidance**
2. Forecaster, guidance, others (e.g., forecasters, core partners)



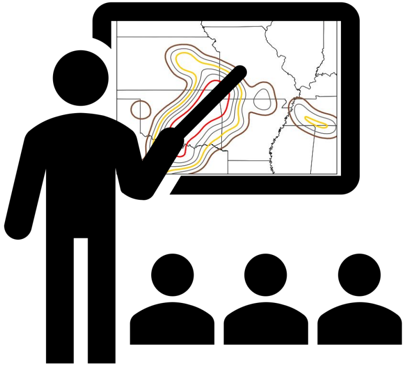
Results: Explainability Theme 2

Forecaster 5: "**Less explainable would be…if you gave us that background, but it was just like, so overly technical** or something like that. Where it was almost like explaining things like from a theoretical overview versus just kind of like, okay, **using terms that forecasters are familiar with.**"

Forecaster 7: **I don't think we need like a like 100-page manual**, kind of explaining it or anything like that. But, you know, if there's a way that you could explain it in layman's terms and just get people to kind of understand, again, kind of the strengths and limitations. And, you know, **I know a lot of forecasters really do like to know the internal workings of everything they work with, but I think probably the best way is just to keep it as simple as possible**. Simplicity is -- is always better.

Explainability:

1. Forecaster and guidance
2. **Forecaster, guidance, others (e.g., forecasters, core partners)**



Results: Explainability Theme 2

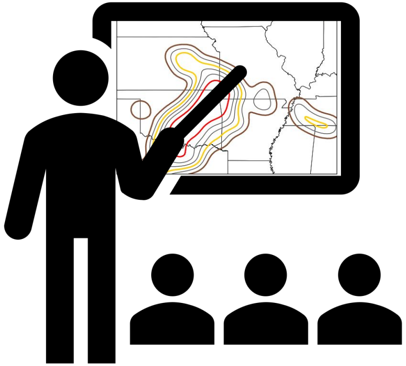
<u>Forecaster 2:</u> “I guess in an ideal world, **explainable would also mean that it's easy for my partners to understand**. And by partners, I mean anybody with either internal or external, whether it's media or EMs or in our office, SPC, that sort of thing, RFC, any of those types of things.”

<u>Forecaster 1:</u> “I think in the sense that **one meteorologist could tell another meteorologist.**”

<u>Forecaster 9:</u> So if we can **boil it down to say okay here is the basic conceptual model** [...] We gotta go that route to make sure we **don’t overcomplicate this AI to the point where it’s just not usable.**

Explainability:

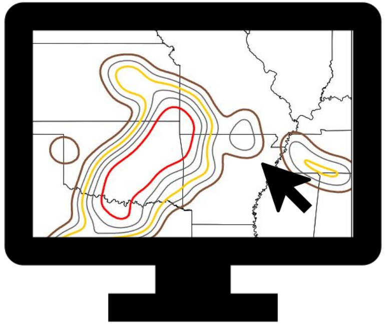
1. Forecaster and guidance
2. **Forecaster, guidance, others (e.g., forecasters, core partners)**



Results: Interpretability Theme

Interpretability:

1. **Good visualization is key to interpretation** (both for **display** and **interactivity**)

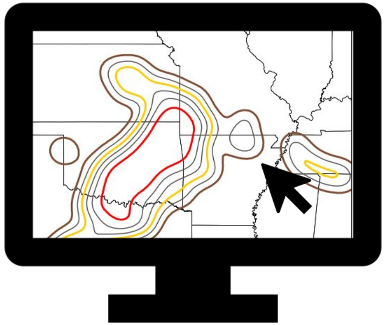


Results: Interpretability Theme

<u>**Forecaster 1:**</u> "That for me, that would be how the data is displayed which again, I think is critical. **I get frustrated when really, really good tools are hard to use for various reasons.** And so yeah, for me **"interpretable" is something that has a level of ease with it in terms of analyzing the data, the way it's displayed, the graphical interface,** you know, with **the ability to loop and go through time**, go through past runs. And see -- because I'm a big trends person, I like to see how runs trend over time. **So it needs to have an intuitive and useful way of displaying the data.**"

Interpretability:

1. **Good visualization is key to interpretation** (both for **display** and **interactivity**)

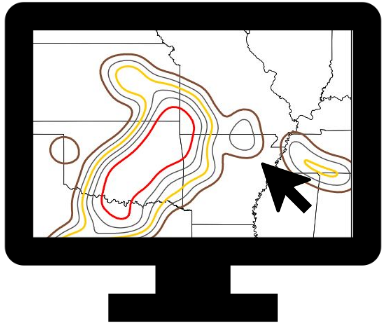


Results: Interpretability Theme

<u>Forecaster 4:</u> "[The] output product is it clear, does it – **can the forecaster interpret it quickly?** In other words if it's – **does it clearly meet the needs of what it's intended to help forecast, right?** If it's a graphical product, does the graphic explain it? If it's a statistical product, there's some sort of a bar graph or numbers, yeah, how is it – how is the interpretability of that product for the forecasters? **Can they get a quick assessment of what it's meant to try to predict?**"

Interpretability:

1. **Good visualization is key to interpretation** (both for **display** and interactivity)

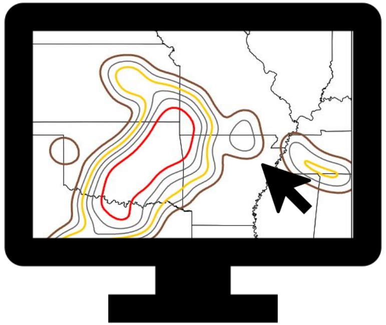


Results: Interpretability Theme

<u>Forecaster 2:</u> "I think that how [guidance] shows what's going to happen and then the realm of possibilities from that. If there's a deterministic, I mean, that's a logical output to have one output. But then **you have the ensembles, and how that's displayed, I think would be important too, so that I can look at the entire realm of possibilities.**"

Interpretability:

1. **Good visualization is key to interpretation** (both for display and **interactivity**)



Big picture take away points from the interview data

- **Developers**: Focus more on understanding how and why the model works
- **Forecasters**: Focus on utility of the model and output for forecasting needs, as well as inter-personal explanations

- Forecasters discuss AI/ML weather product explainability and interpretability within the context of being able to perform core functions their job, e.g.:
    - Display of model **output is intuitive** and meets forecasting needs
    - Able to **explain and discuss model output** amongst forecasters
    - Effectively communicate model output in **understandable language to core partners**

Rudin's principles for 'creating a predictive model that is not a black box'

"In cases where the underlying distribution of data changes (called domain shift, which occurs often in practice), problems arise if users cannot troubleshoot the model in real-time, which is much harder with black box models than interpretable models."

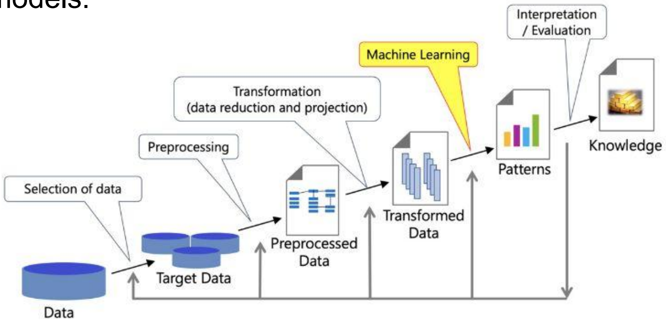


FIG 1. *Knowledge Discovery Process. Figure adapted from [95].*

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. Statistics Surveys, 16, 1-85.

Principle 1 An interpretable machine learning model **obeys a domain-specific set of constraints to allow** it (or its predictions, or the data) to be more easily understood by humans. These constraints can differ dramatically depending on the domain.

Principle 2 Despite common rhetoric, interpretable models do not necessarily create or enable trust – they could also enable distrust. They **simply allow users to decide whether to trust them**. In other words, they permit a decision of trust, rather than trust itself.

Principle 3 It is important not to assume that one needs to make a sacrifice in accuracy in order to gain interpretability. In fact, interpretability often begets accuracy, and not the reverse. **Interpretability versus accuracy is, in general, a false dichotomy in machine learning**.

Principle 4 As part of the full data science process, one should **expect both the performance metric and interpretability metric to be iteratively refined**.

Principle 5 For high stakes decisions, interpretable models should be used if possible, rather than "explained" black box models.

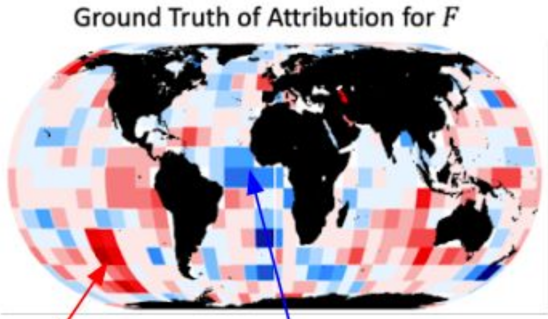
2.5. & 2.6. Go to sli.do and use the code TAI4ES

Activity break!

How do you **interpret** this?
How would you **explain** this?

From *Mamalakis et al. (2021)*

y_n : 0.0660
NN prediction: 0.0802



Positive contribution
Negative contribution

Red color highlights features that contributed positively to y

Blue color highlights features that contributed negatively to y

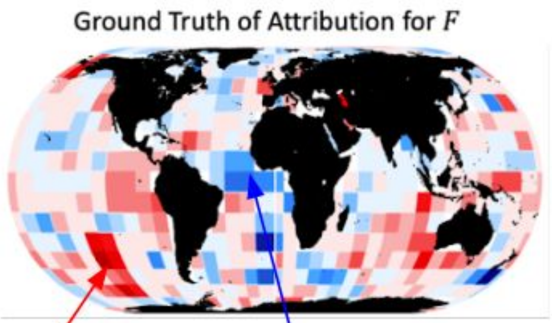
2.7. Go to sli.do and use the code TAI4ES

Activity break!

Which regional oceanic basins had the biggest positive contribution to y in this example?

From *Mamalakis et al. (2021)*

y_n : 0.0660
NN prediction: 0.0802



Positive contribution
Negative contribution

Red color highlights features that contributed positively to y

Blue color highlights features that contributed negatively to y

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Agenda

- 9:00 Explainability vs. Interpretability
- ***9:45 Short brain & bio break #1***
- 9:50 XAI techniques for deep learning (Part 1)
- 11:10 *Short brain & bio break #2*
- 11:15 XAI techniques for deep learning (Part 2)
- Noon: End of session

Questions?

https://app.sli.do/event/1zumy91n

Or go to sli.do and use the code TAI4ES

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

LEAP

NOAA CENTER FOR ARTIFICIAL INTELLIGENCE

Radiant Earth Foundation
EARTH IMAGERY FOR IMPACT

Overview

XAI techniques for deep learning - Part 1:

1) Introduction to XAI for neural networks:

i) Motivation for XAI - the general idea
ii) Opportunities that XAI brings
iii) Representative methods and categories of XAI

2) Popular XAI methods and Examples:

i) Gradient (sensitivity)
ii) Input*Gradient (attribution)
iii) Layer-wise Relevance Propagation (attribution)
ii) SHAP – SHapley Additive exPlanations (attribution)

XAI techniques for deep learning - Part 2:

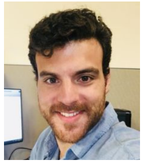
3) Benchmarking XAI:

i) Motivation - General idea
ii) Regression Example
iii) Classification Example

4) Final comments / big picture

Speakers for XAI techniques for deep learning

Randy
Chase
(OU)



Antonios
Mamalakis
(CSU)

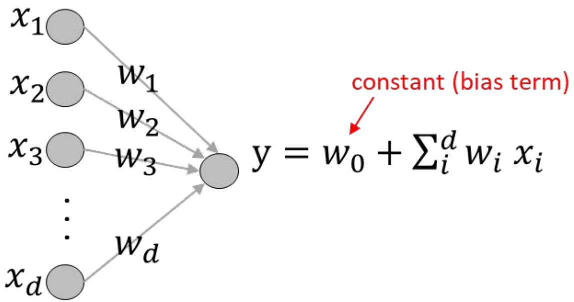
Imme
Ebert-Uphoff
(CSU)

Introduction to XAI methods for deep learning (neural networks)

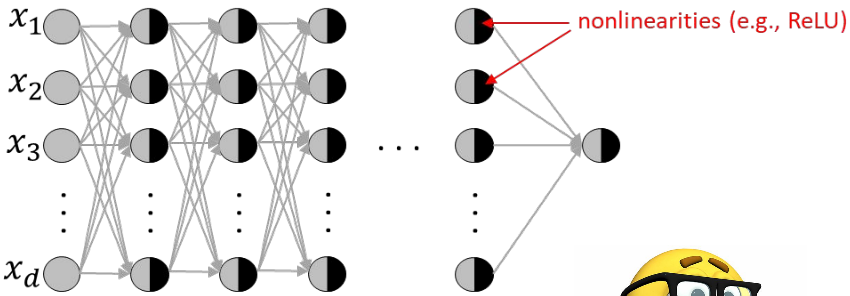
Why is XAI necessary?

➡ **Scientists need to understand what the AI model is doing; what the decision-making process is.**

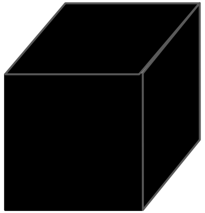
Linear model: inherently interpretable



Neural Network: not inherently interpretable



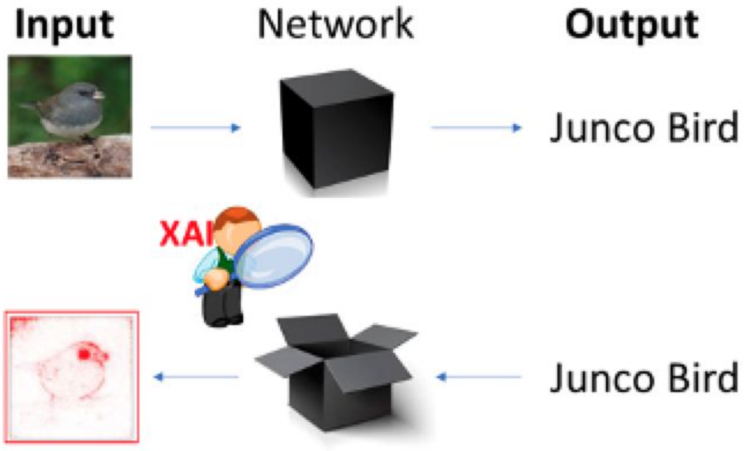
Neural networks have for long been perceived as black boxes



Why is XAI necessary?

Methods of eXplainable Artificial Intelligence (XAI) aim to explain how a Neural Network makes predictions, i.e., what the *decision strategy* is.

XAI methods highlight which features in the input space are important for the prediction: They produce the so-called *explanation/relevance heatmaps*.



Explanation Heatmap

From *Adebayo et al. (2020)*

Why is XAI necessary?

site: https://lrpserver.hhi.fraunhofer.de/image-classification

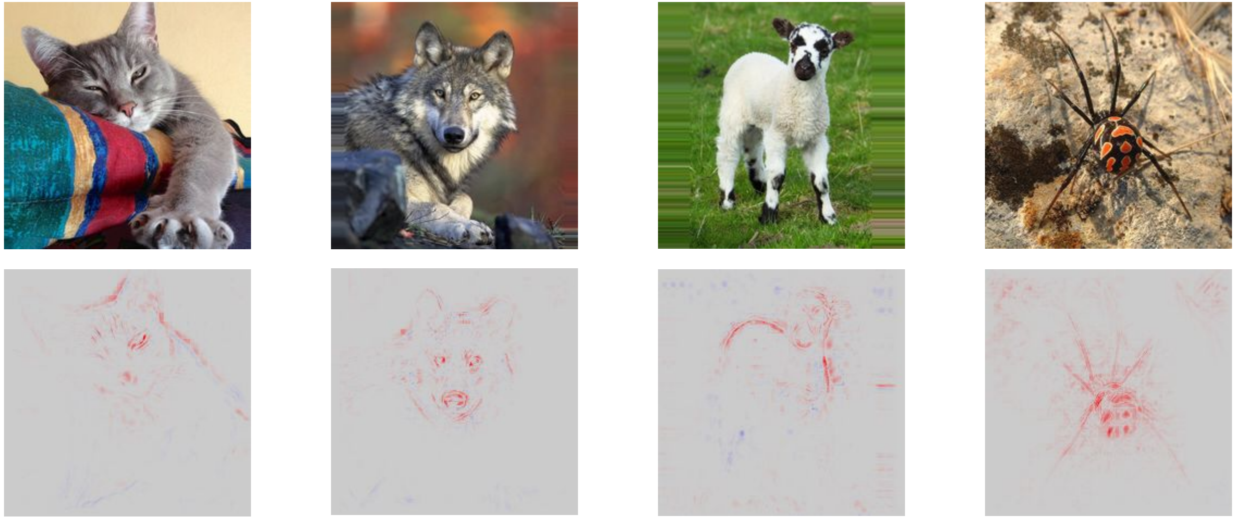
Methods of eXplainable Artificial Intelligence (XAI) aim to explain how a Neural Network makes predictions, i.e., what the *decision strategy* is.

XAI methods highlight which features in the input space are important for the prediction: They produce the so-called *explanation/relevance heatmaps*.

tabby cat | white wolf | ram | black widow

Network Input

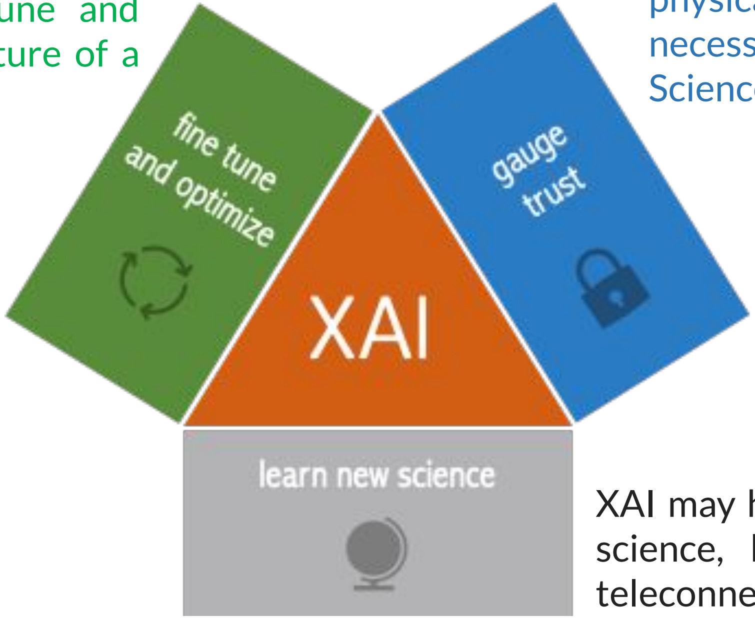
Explanation Heatmap



XAI: A potential *game changer* for prediction in Earth Sciences

XAI may help fine-tune and optimize the architecture of a flawed model

XAI helps calibrate model trust and physically interpret the network, which is a necessity in many applications in Earth Sciences.



XAI may help accelerate establishing new science, like investigating new climate teleconnections and gaining new insights.

From *Mamalakis et al. (2022)*

XAI methods and categories

AI models

Interpretable models
(e.g., linear models, decision trees)

Post-hoc Explainable models

Today's topic

Covered on Monday

Global XAI methods
(e.g., optimal input, permutation importance)

Local XAI methods

sensitivity
(e.g., Gradient)

attribution
(e.g., LRP, SHAP)

. . .

Table 3 Glossary of Interpretability Methods With Abbreviations Referenced Throughout Our Review

Method		Abbrv.	Method		Abbrv.
Anchors	[148]	ANCH	Layer-wise Relevance Propagation (full)	[13]	LRP
ApproShapley (Shapley Value Sampling)	[29]	AS	LRP (composite strategy)	[103], [126]	LRP-CMP
Class Activation Mapping	[206]	CAM	LRP (specific variants)	[13], [126]	LRP-*
Contextual Prediction Difference Analysis	[56]	CPDA	Local Interpretable Model-agnostic Explanations	[147]	LIME
DeconvNet	[201]	DCN	Meaningful Perturbation	[47]	MP
DeepLIFT	[170]	DL	NeuronConductance	[36]	NC
DeepLIFT (Rescale)	[170]	DLR	NeuronGuidedBackprop	[178]	NGB
DeepLIFT SHAP	[116]	DLSHAP	NeuronIntegratedGradients	[172]	NIG
Deep Taylor Decomposition	[127]	DTD	Occlusion Analysis	[201]	OCC
ExcitationBackprop	[202]	EB	PatternAttribution	[90]	PA
ExtremalPerturbation	[46]	EP	PatternNet	[90]	PN
GNNExplainer	[198]	GNNEXP	Prediction Difference Analysis	[208]	PDA
GNN-LRP	[162]	GLRP	Randomized Input Sampling for Explanation	[142]	RISE
GradCAM	[167]	GC	Saliency Analysis / Gradient	[14], [174]	SA
Gradient SHAP	[116]	GSHAP	SHapley Additive exPlanations	[116]	SHAP
Gradient × Input	[170]	GI	SHAP Interaction Index	[115]	SHAPIDX
GuidedBackprop	[178]	GB	SmoothGrad	[176]	SG
Guided GradCam	[167]	GGC	SmoothGrad^2	[76]	SG-SQ
Integrated Gradients	[183]	IG	Spectral Relevance Analysis	[104]	SpRAy
Internal Influence	[110]	II	TreeExplainer	[115]	TEXP
Kernel SHAP	[116]	KSHAP	VarGrad	[1]	VG
LayerConductance	[172]	LC	Testing with Concept Activation Vectors	[89]	TCAV
Local Rule-based Explanations	[58]	LORE	TotalConductance	[36]	TC

From *Samek et al. (2021)*

The Big Picture / Setting Expectations

- Applying local XAI methods to identify the strategies a NN has learned … is a **Detective Game**.
- Expect to only get clues - rather than complete answers.
- It's usually a lengthy process, where you try one method after the other to find clues, then generate hypotheses.

- Why lengthy?
  - Because there are many different methods that tell you different things.
  - Because local XAI methods look at one sample at a time.

- Questions you will have to tackle:
  - Which methods should I use?
  - Which samples should I look at?
  - How do I ensure results are consistent across other samples without looking at *all* samples?
  - How should I interpret the results?
  - If I use visual inspection of results: how objective is that?

First - A Guiding Application: SEVIR

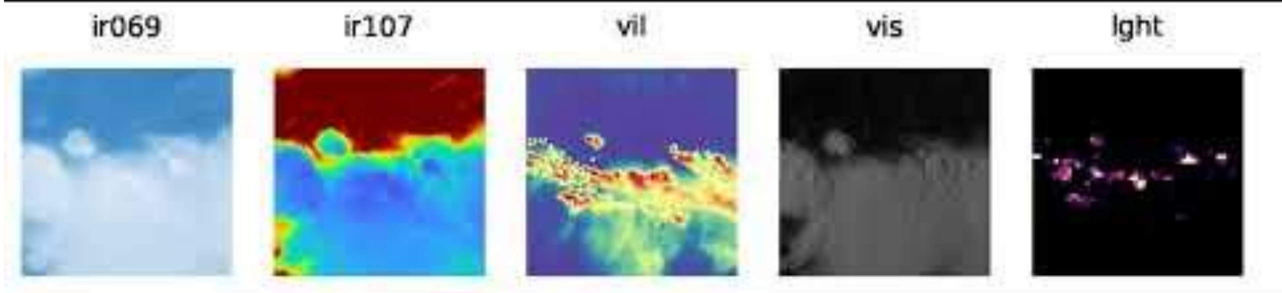
We will use this application to demonstrate what XAI methods might (and might not) give you.

A Guiding application: SEVIR

The Storm EVent ImagRy (**SEVIR**) dataset (Veillette et al. 2020):

Over 10,000 events

1 TB in size...

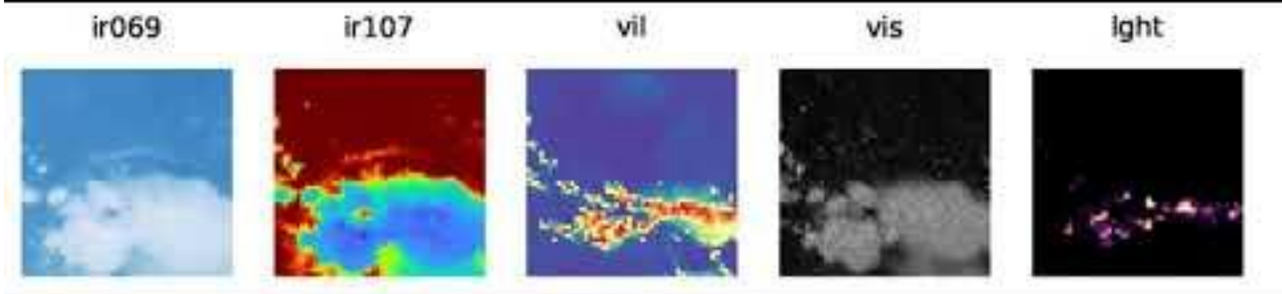


A Guiding application: SEVIR

sub-SEVIR

Resampled to only have 48x48 pixels

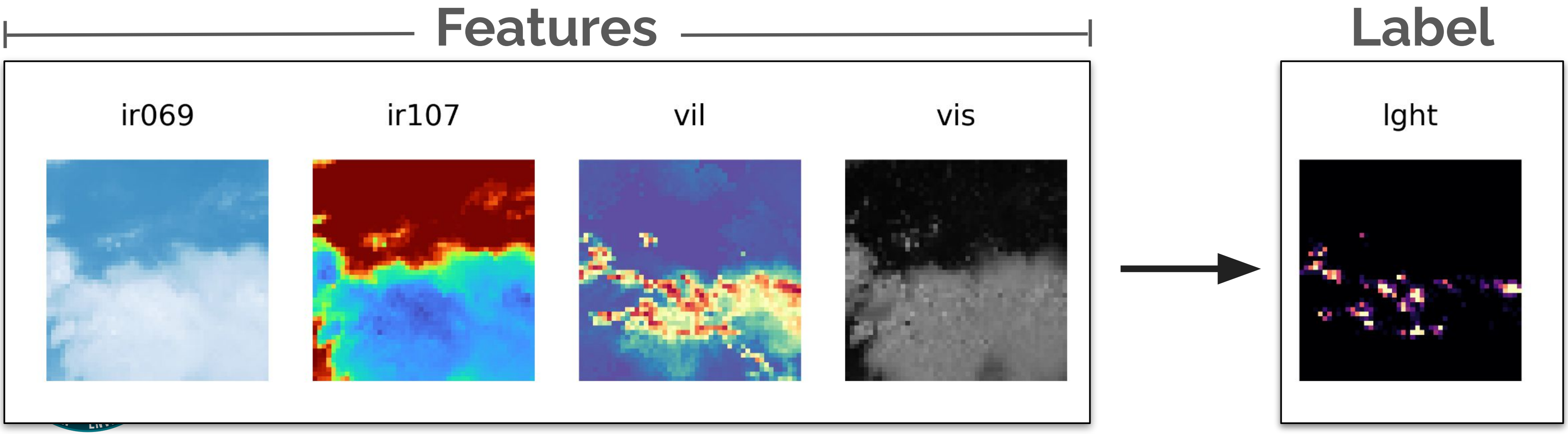
2 GB in size



A Guiding application: SEVIR

The Machine Learning Tasks: [Chase et al. 2022 & Chase et al. *in prep.*]

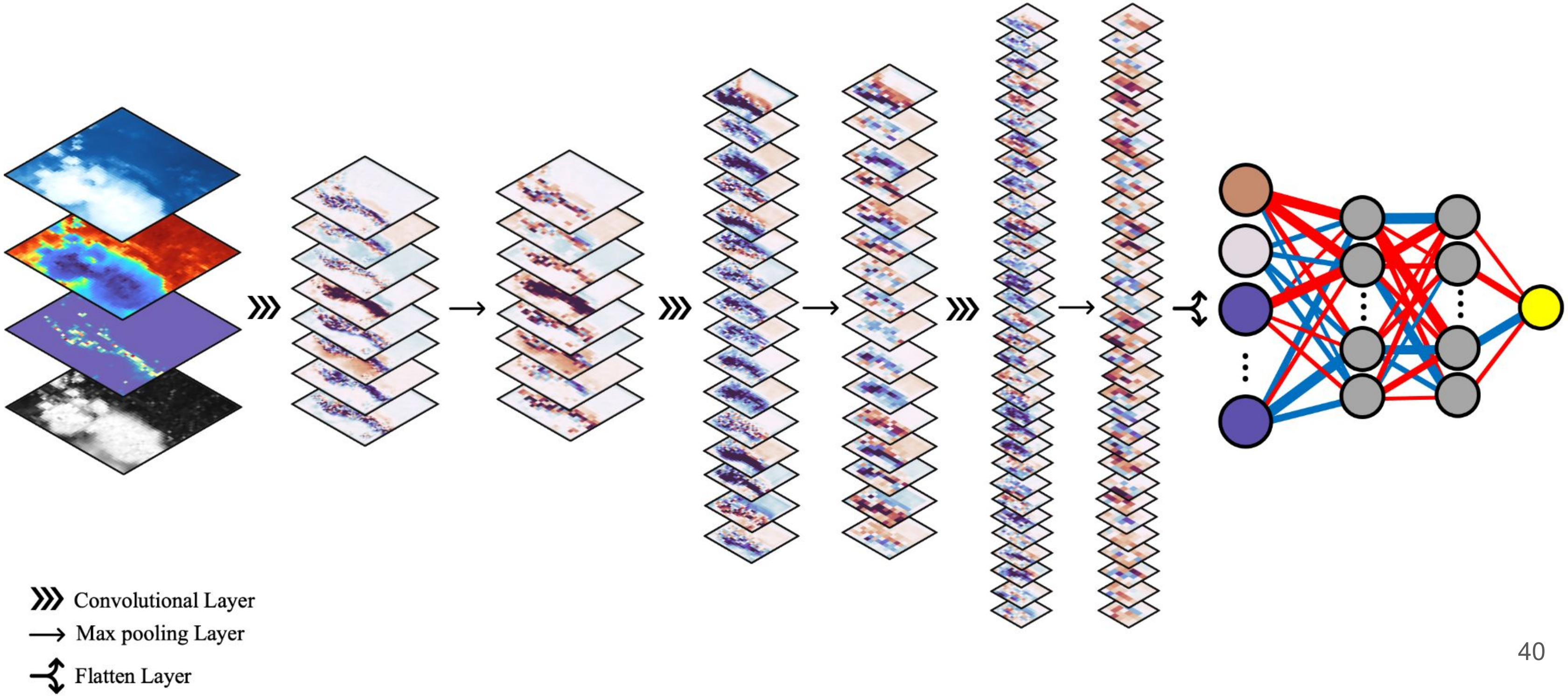
(1) Does this image contain a thunderstorm? (**classification**)
(2) How many lightning flashes are in this image? (**regression**)



A Guiding application: SEVIR

The Machine Learning Models: [Chase et al. *in prep.*]

Example CNN Architecture



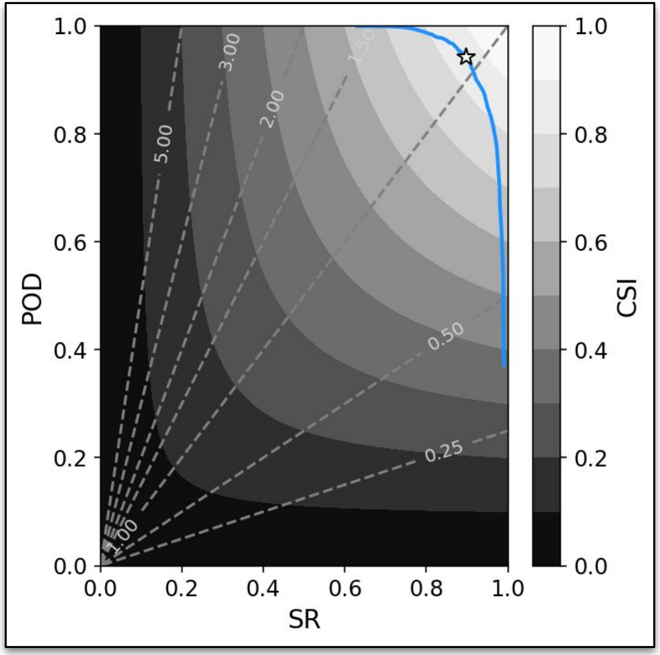
A Guiding application: SEVIR

CNN Skill [Chase et al. *in prep.*]

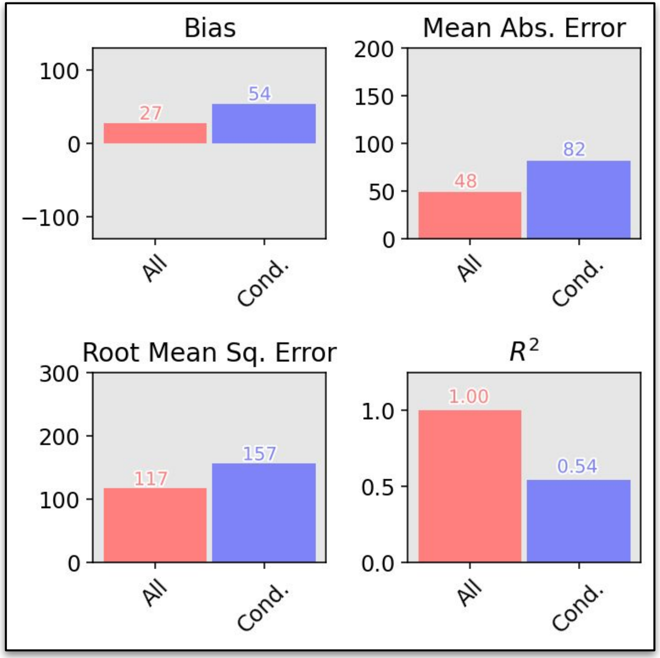
Step 1:
Before we try to explain the model, analyze its overall performance.
Is it working well?

Classification

AUC: 0.97
CSI: 0.87
Acc: 90%

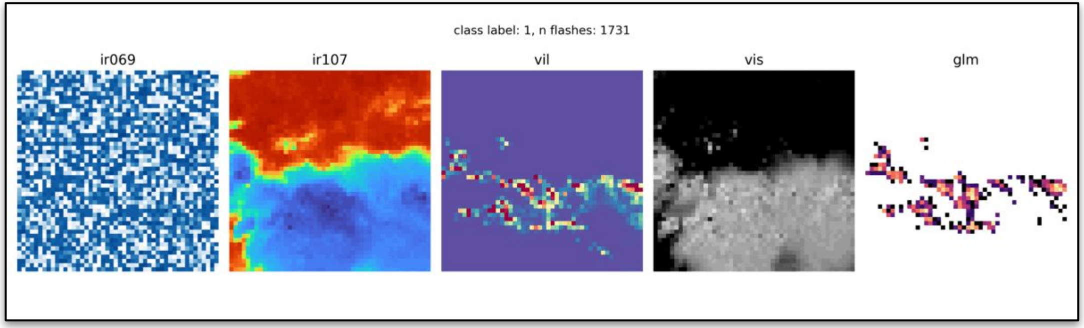


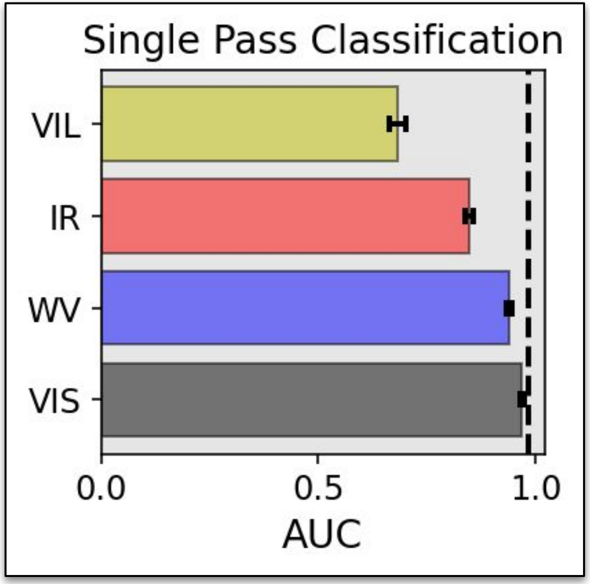
Regression

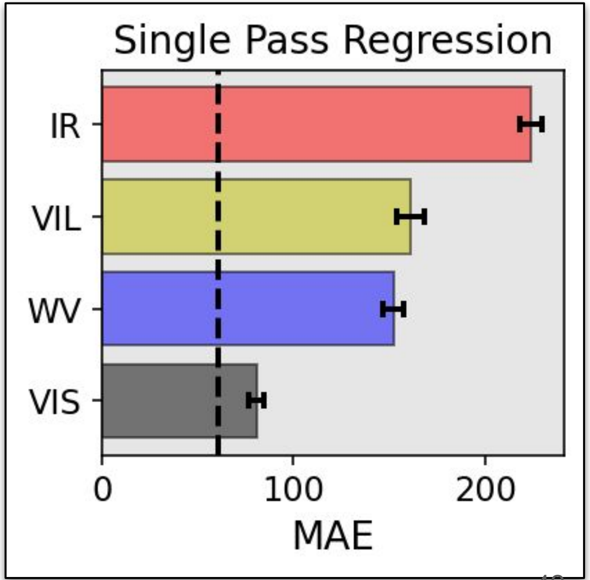


A Guiding application: SEVIR

CNN Permutation Importance







Step 2:
Start with global explanation methods (covered yesterday), before looking at local methods (covered today).

Here: permutation importance.

A Guiding application: SEVIR Notebooks

All of the following XAI examples were made on Google Colab using the following notebooks:

Saliency:
https://colab.research.google.com/drive/1nkhmeyYEZeXYFtTkd1GfGWA8o-nHuKvC?usp=sharing

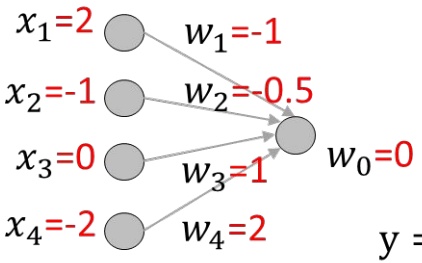
Shap:
https://colab.research.google.com/drive/1HbpR37bmPxyMPhqWXne4Pr2KuasWEXtk?usp=sharing

Popular Local XAI Methods For NNs

The difference between “sensitivity” and “attribution”

- ***Sensitivity*** refers to how sensitive the value of the output is to a specific input feature. It is essentially the gradient (i.e., the first derivative if we think the network as a function) of the output with respect to the input. **[units output/units input]**
- ***Attribution*** refers to the relative contribution of a specific input feature to the output. **[units output]**

Simple case: Linear model



$y = w_0 + \sum_i^4 w_i\, x_i$

= 0+(-2)+0.5+0+(-4)= -5.5

Explanation of -5.5:

- The ***sensitivity*** of -5.5 to the feature x_i is $\frac{\partial y}{\partial x_i} = \boldsymbol{w_i}$.

(sensitivity is NOT dependent on x point; not true in nonlinear models)

- The ***attribution*** of -5.5 to the feature x_i is $\boldsymbol{w_i x_i}$.

sensitivity heatmap
-1
-0.5
1
2

attribution heatmap
-2
0.5
0
-4

The difference between "sensitivity" and "attribution"

Another way to think of this difference: Warren Buffet example (famous investor, super rich now).

Let's say you want to **learn from Warren Buffet's investment strategies.**

Which question would you like to ask?

A) **Sensitivity question:**
Changes from current situation: Given his current financial situation, which financial actions would change his net worth the most (up or down)?

B) **Attribution question:**
How did he get here: Given his current financial situation, how did he get here, say from the situation he was in 10/20/30 years ago?

There's no right and wrong question - but each question will give you different insights - so use and interpret them accordingly. Often attribution is what you want.

The difference between “sensitivity” and “attribution”

Another example: **Identifying a thunderstorm from inputs (SEVIR)**

Which question would you like to ask?

A) **Sensitivity Question:**
Modifications of current situation: Given you just identified a storm based on the inputs with certain confidence - **what modifications to the inputs would change this assessment the most** (up or down)?

B) **Attribution Question:**
How did we get here: Given you just identified a storm based on the inputs with certain confidence - what were the **most important reasons in the input to yield that confidence**?

There’s no right and wrong question - but each question will give you different insights - so use and interpret them accordingly. Often **attribution** is what you want.

Gradient (sensitivity)

- ***Sensitivity*** refers to how much sensitive the value of the output is to a specific input feature. It is essentially the gradient (i.e., the first derivative if we think the network as a function) of the output with respect to the input. **[units output/units input]**

Relevance of feature i for prediction n

Network

$$R_{i,n} = \left.\frac{\partial \hat{F}}{\partial \mathrm{X}_i}\right|_{\mathrm{X}_i = x_{i,n}}$$

Value of feature i in sample n

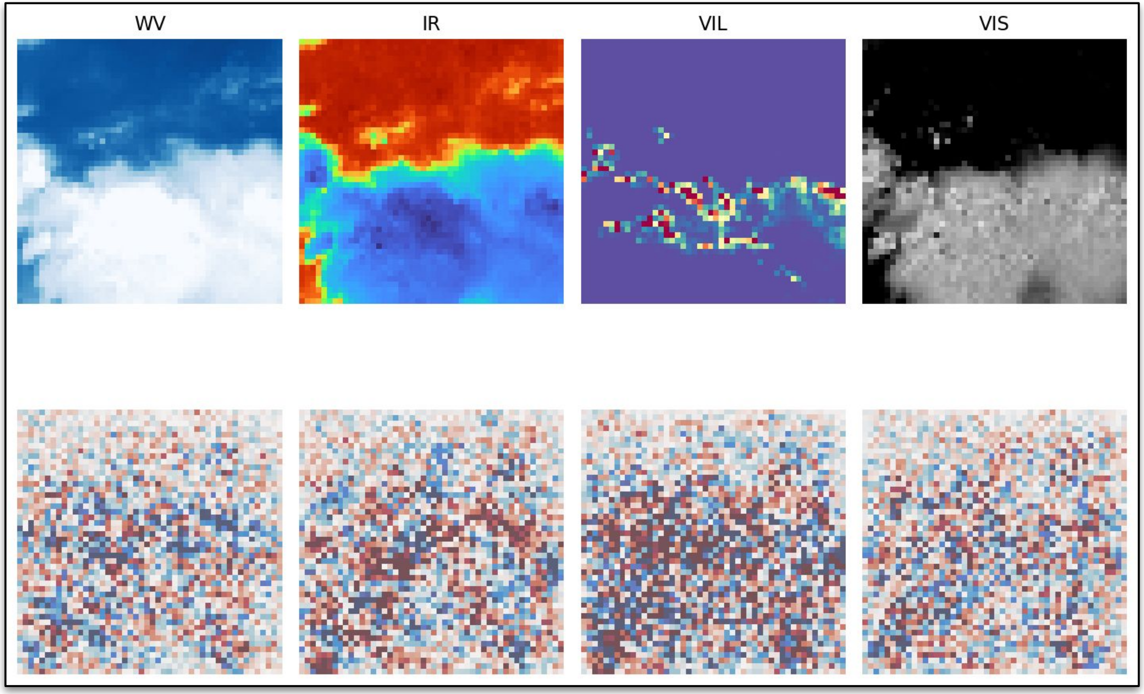
Partial derivative

Gradient (sensitivity): Classification Example

p(lightning | input) = 0.998

Inputs

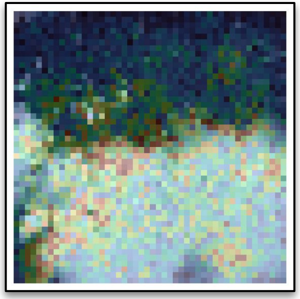
Gradient



Positive Sensitivity

Negative Sensitivity

“Heatmap”

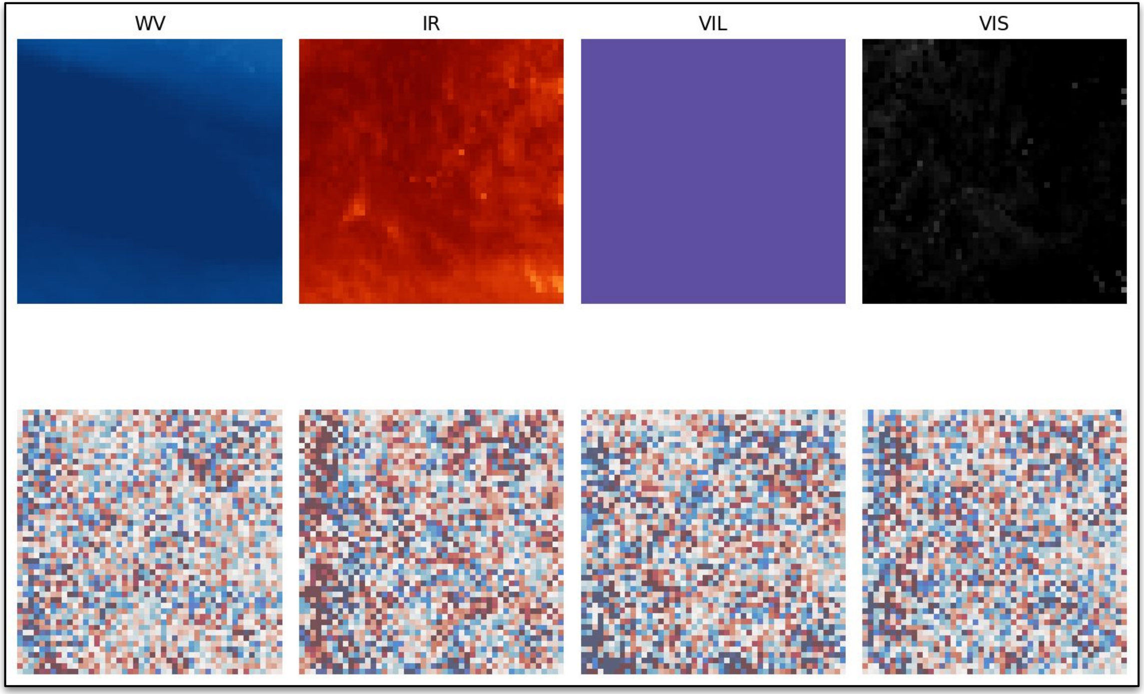


Gradient (sensitivity): Classification Example 2

p(lightning | input) = 0.006

Inputs

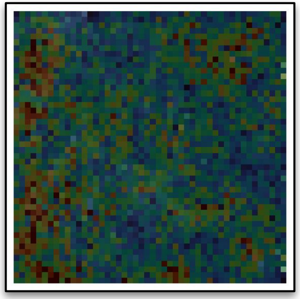
Gradient



Positive Sensitivity

Negative Sensitivity

"Heatmap"

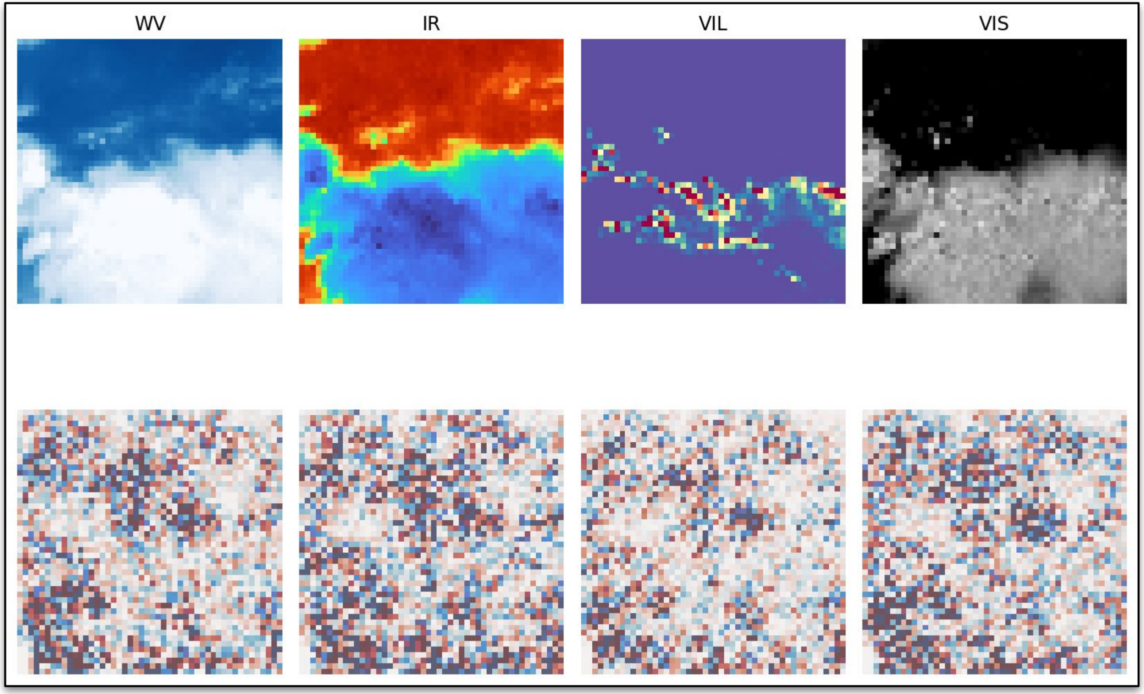


Gradient (sensitivity): Regression Example

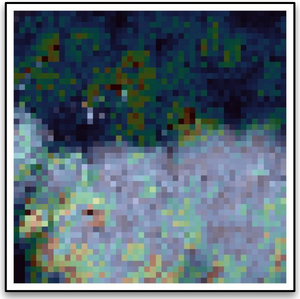
Predicted flash number = **761 flashes**

Inputs

Gradient



"Heatmap"

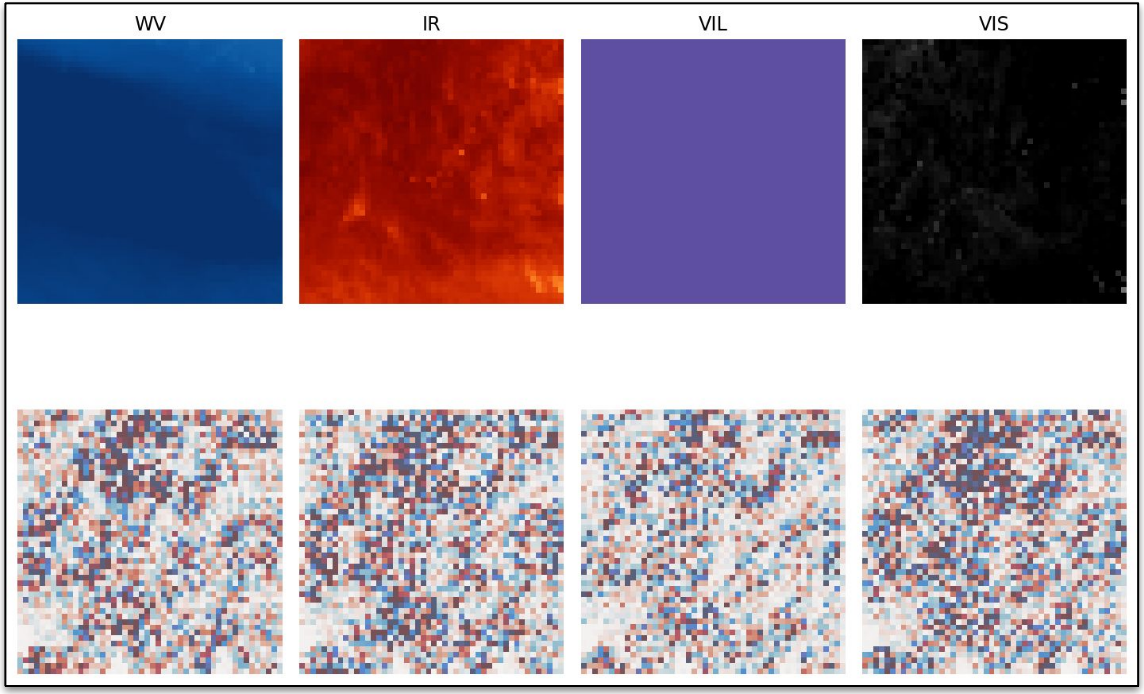


Gradient (sensitivity): Regression Example 2

Predicted flash number = **4 flashes**

Inputs

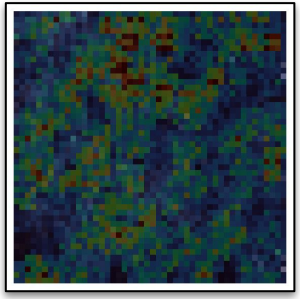
Gradient



Positive Sensitivity

Negative Sensitivity

"Heatmap"



Smooth Gradient

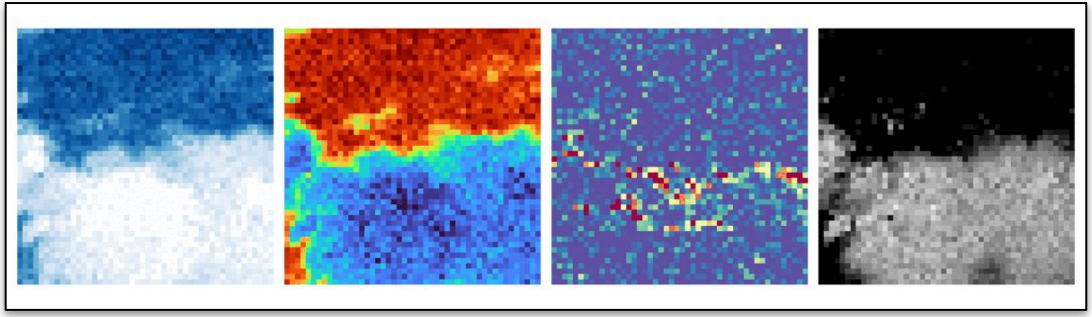
Motivation:

- Gradient tends to amplify noise.
- Result: generates maps that have lots of blue and red pixels right next to each other - due to amplified noise, which are hard to interpret …

Simple trick to remove much of that noise:

- Use “Smooth Gradient” method
- Smooth gradient is where we run the gradient method many times (e.g., 100) with the same image with a slight bit of noise added in. Then we take the average gradient of all 100 runs as the new ‘smooth gradient’

100 noised examples

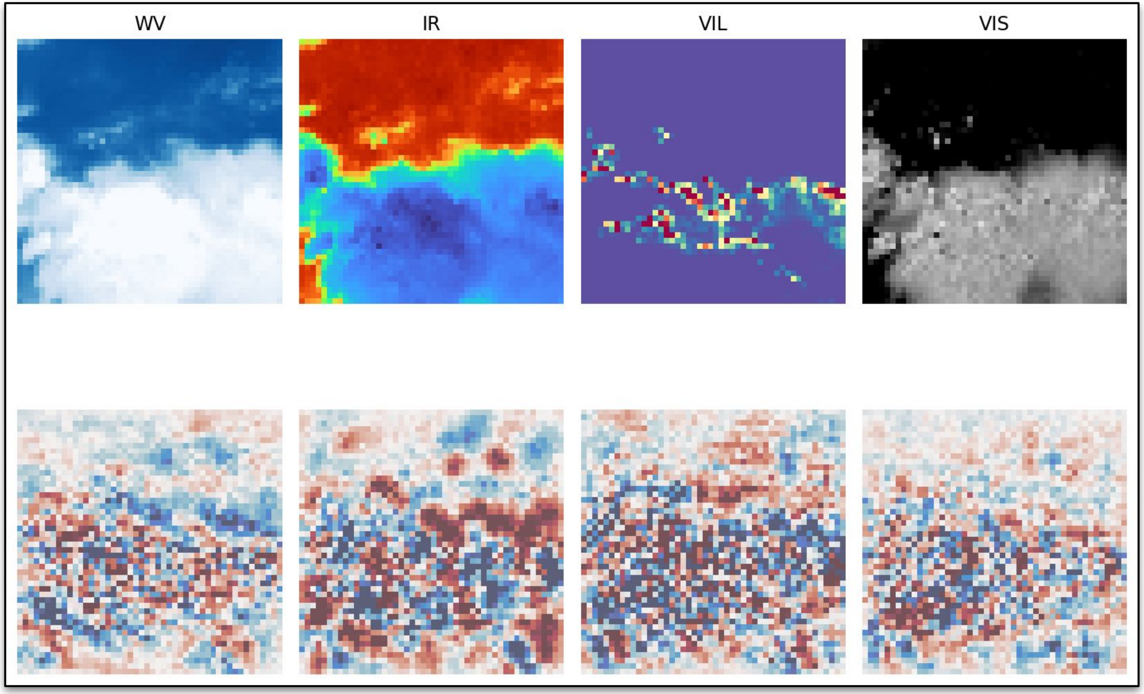


Smooth Gradient: Classification Example

p(lightning | input) = 0.998

Inputs

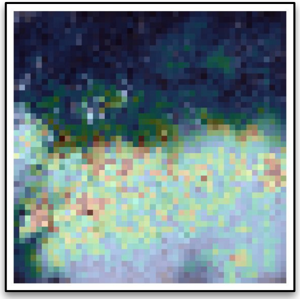
Smooth Gradient



Positive Sensitivity

Negative Sensitivity

"Heatmap"

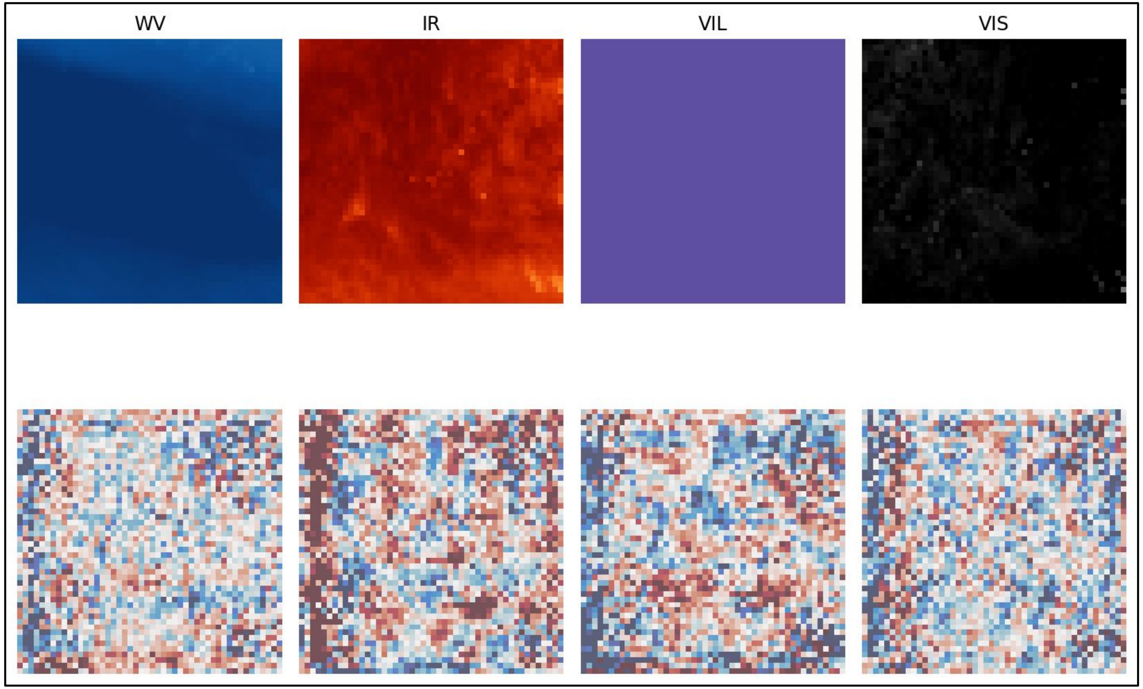


Smooth Gradient: Classification Example 2

p(lightning | input) = 0.006

Inputs

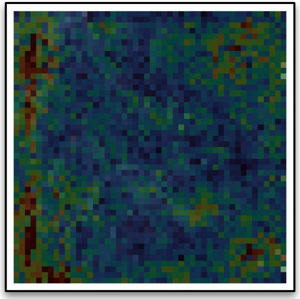
Smooth Gradient



Positive Sensitivity

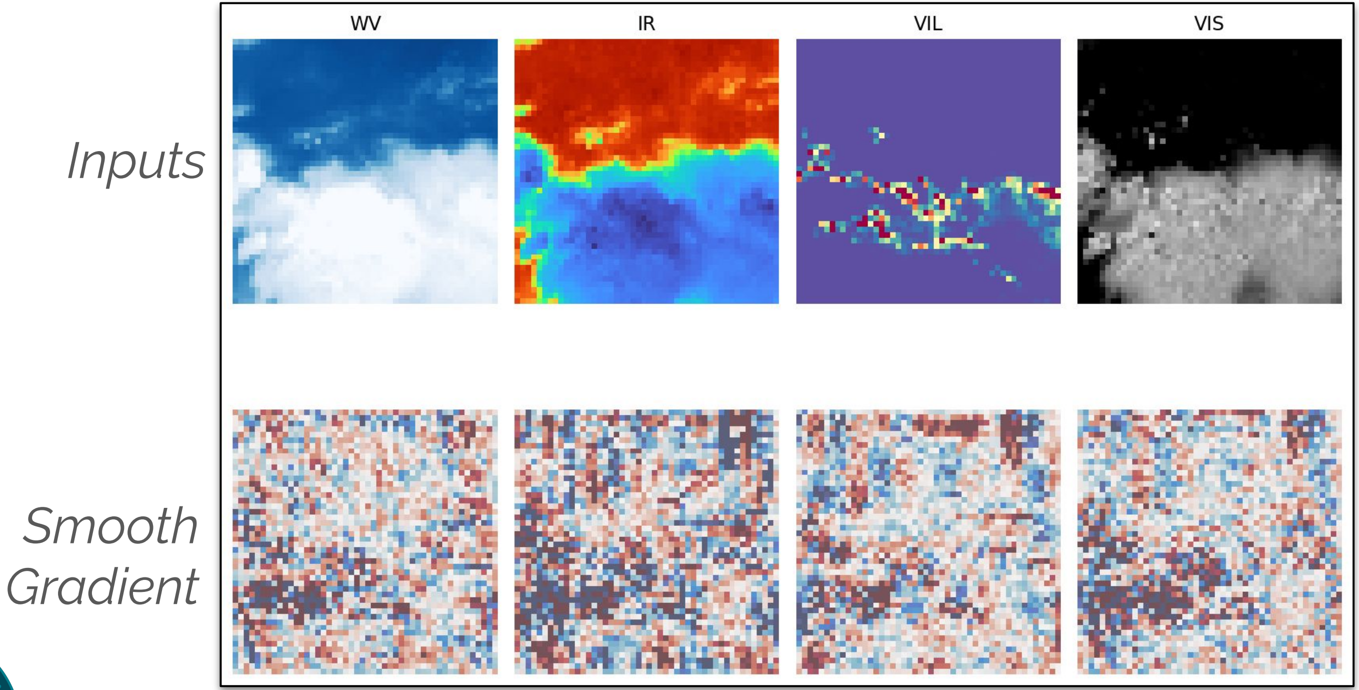
Negative Sensitivity

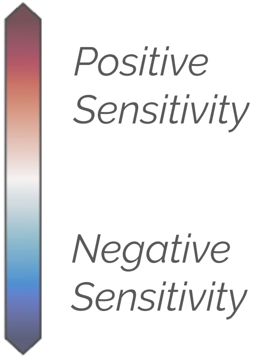
"Heatmap"



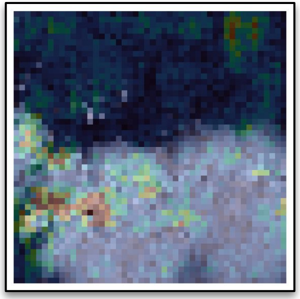
Smooth Gradient: Regression Example

Predicted flash number = **761 flashes**



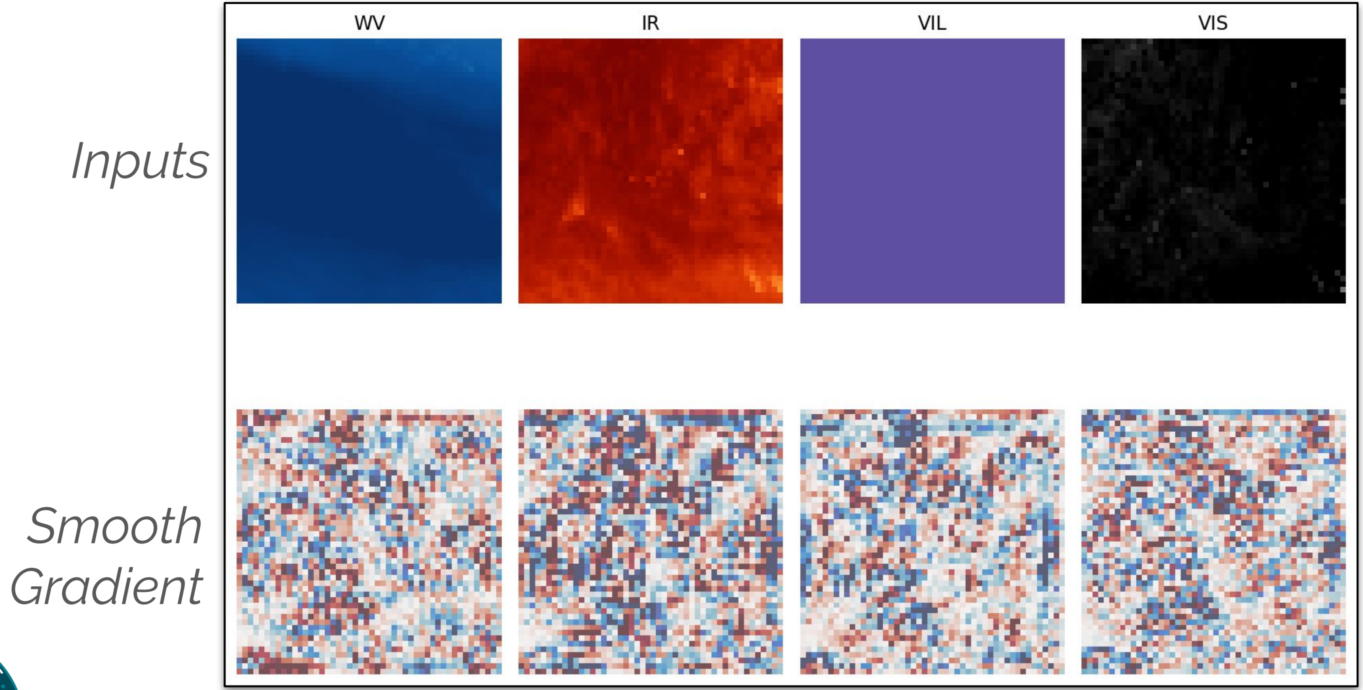


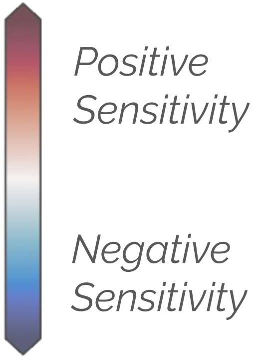
"Heatmap"



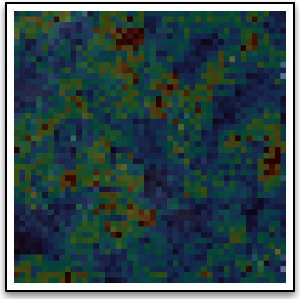
Smooth Gradient: Regression Example 2

Predicted flash number = **4 flashes**





"Heatmap"



This was all sensitivity…

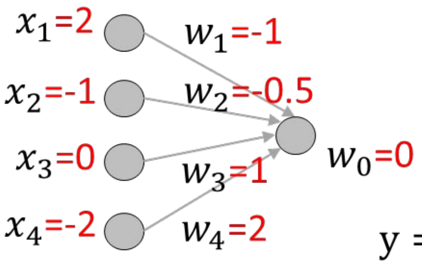
(gradient/saliency, smooth gradient)

… now moving on to attribution.

The difference between "sensitivity" and "attribution"

- *Sensitivity* refers to how much sensitive the value of the output is to a specific input feature. It is essentially the gradient (i.e., the first derivative if we think the network as a function) of the output with respect to the input. **[units output/units input]**
- ***Attribution*** refers to the relative contribution of a specific input feature to the output. **[units output]**

Simple case: Linear model



$$y = w_0 + \sum_i^4 w_i\, x_i$$

= 0+(-2)+0.5+0+(-4)= -5.5

Explanation of -5.5:

- The ***sensitivity*** of -5.5 to the feature x_i is $\frac{\partial y}{\partial x_i}$= $\boldsymbol{w_i}$.

(sensitivity is NOT dependent on x point; not true in nonlinear models)

- The ***attribution*** of -5.5 to the feature x_i is $\boldsymbol{w_i x_i}$.

sensitivity heatmap	***attribution heatmap***
-1	-2
-0.5	0.5
1	0
2	-4

Input*Gradient (attribution)

- ***Attribution*** refers to the relative contribution of a specific input feature to the output. **[units output]**

Relevance of feature i for prediction n

Input

Gradient

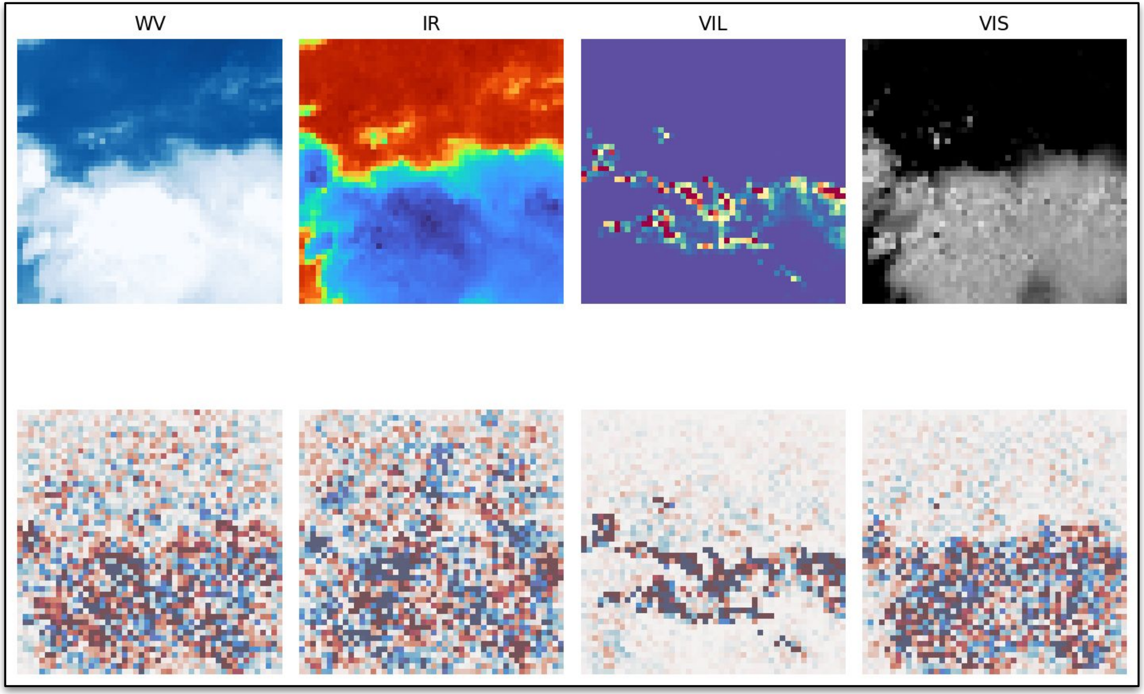
$$R_{i,n} = x_{i,n} * \left.\frac{\partial \hat{F}}{\partial \mathrm{X}_i}\right|_{\mathrm{X}_i = x_{i,n}}$$

input*gradient: Classification Example

p(lightning | input) = **0.998**

Inputs

*input*gradient*

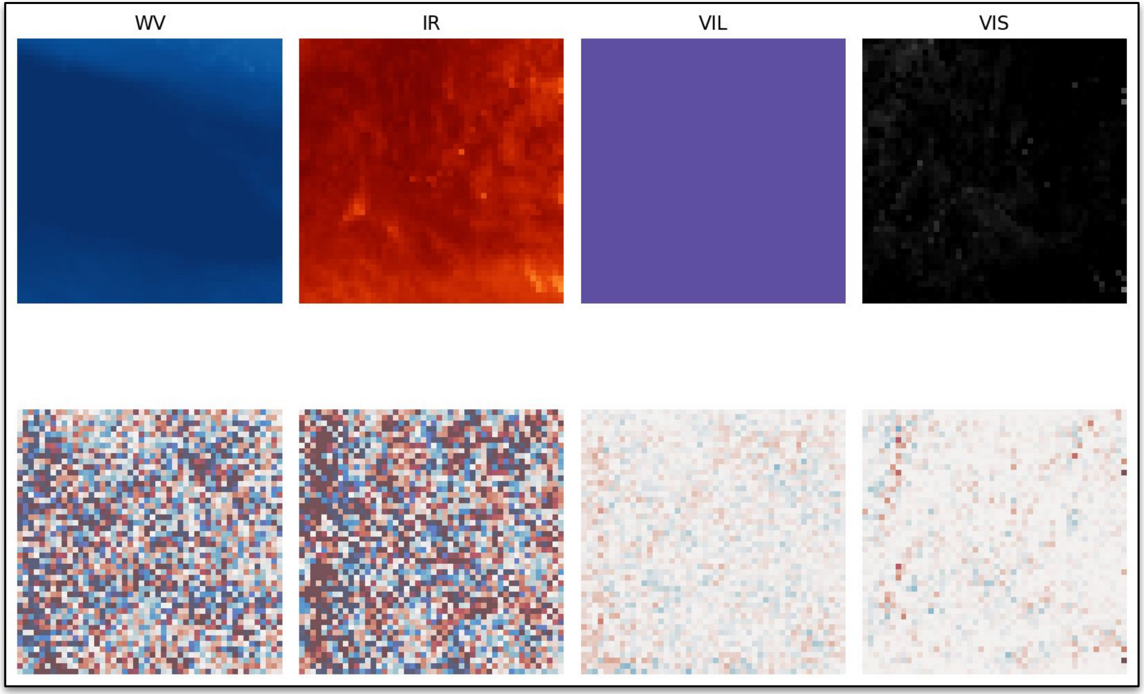


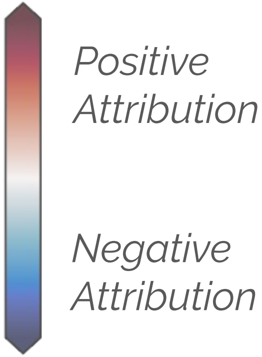
input*gradient: Classification Example 2

p(lightning | input) = 0.006

Inputs

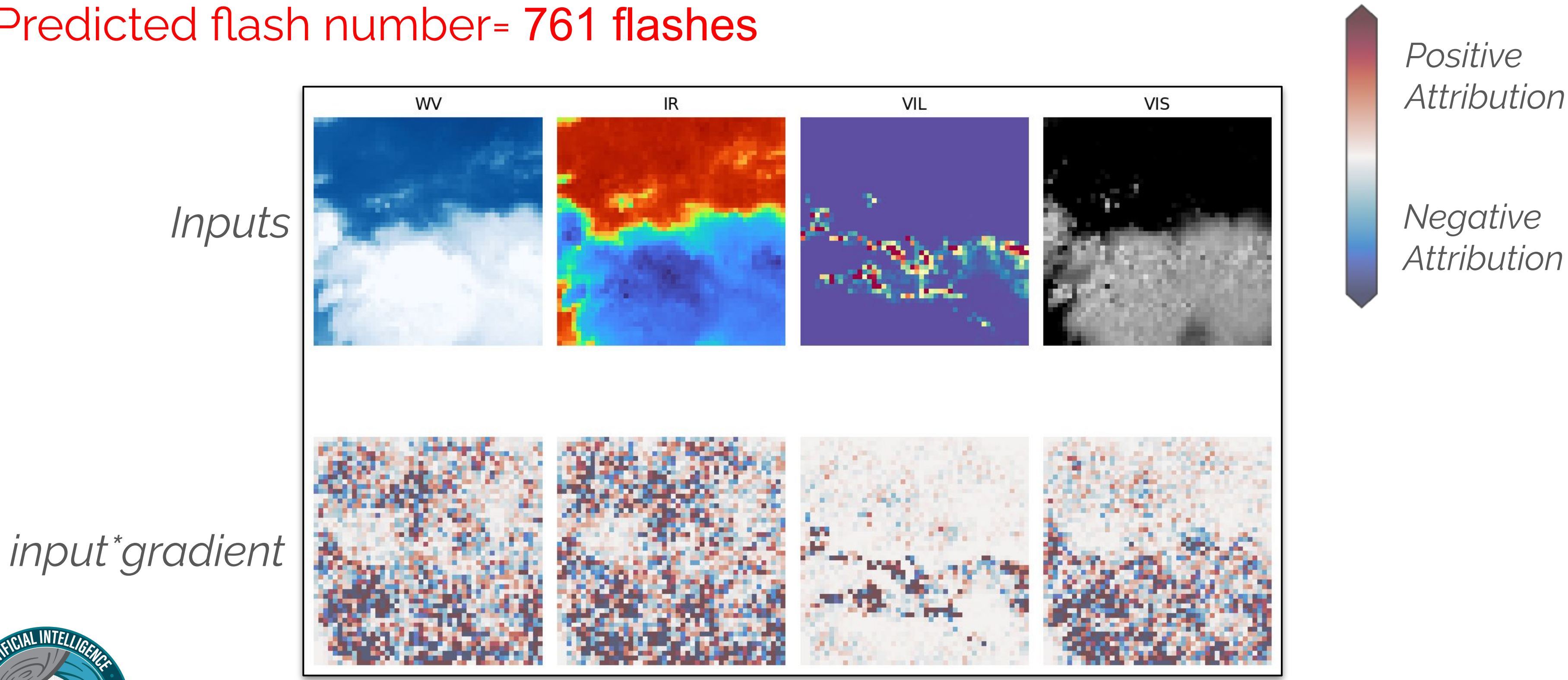
input*gradient





input*gradient: Regression Example

Predicted flash number= **761 flashes**

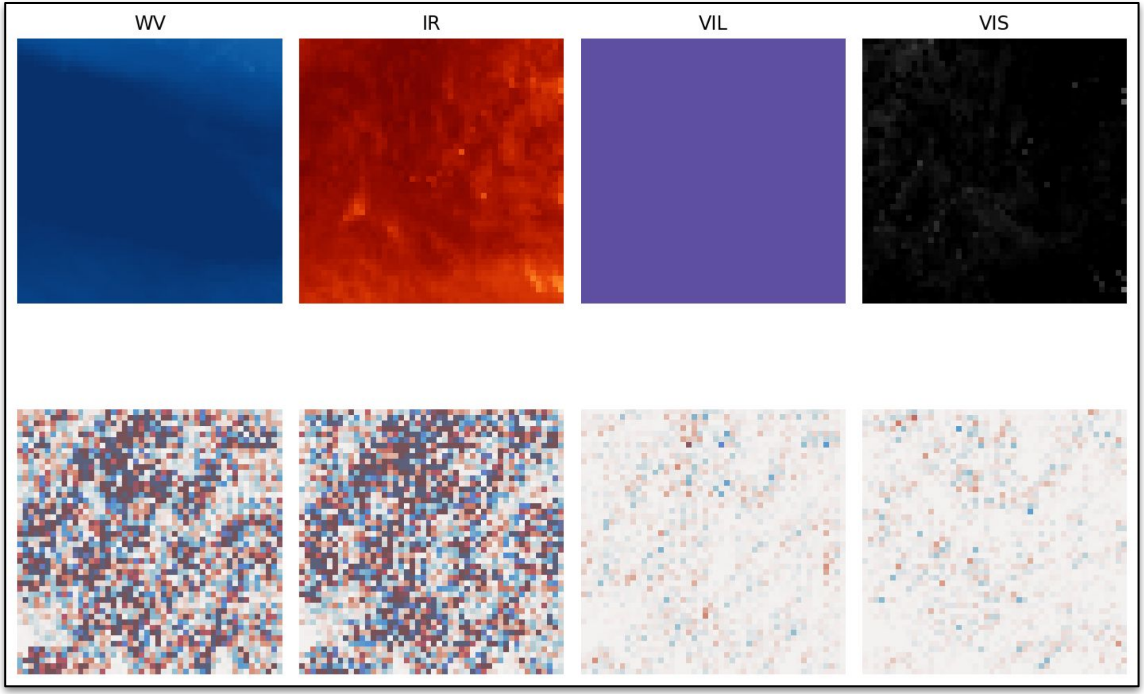


input*gradient: Regression Example 2

Predicted flash number = **4 flashes**

Inputs

*input*gradient*



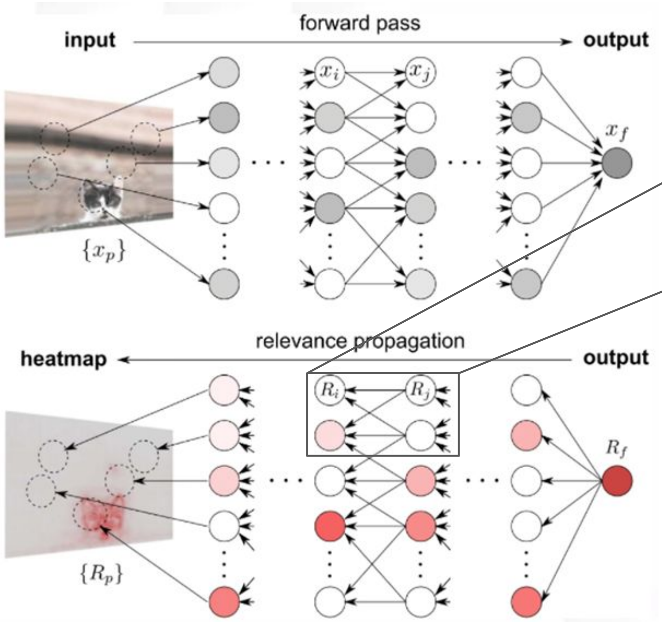
Positive Attribution

Negative Attribution

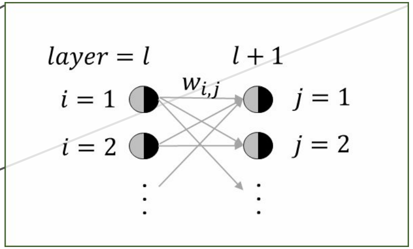
These were **simple attribution methods**…

… now moving on to **more complex attribution methods**.

LRP: Layerwise Relevance Propagation (attribution)



From *Bach et al., (2015)*



Relevance of neuron i in layer l

Preactivation from i to j

$$\textbf{LRP}_\textbf{z}: \quad R_i^l = \sum_j \frac{w_{i,j}x_i}{\sum_i w_{i,j}x_i + w_0} R_j^{l+1}$$

Other popular LRP rules :

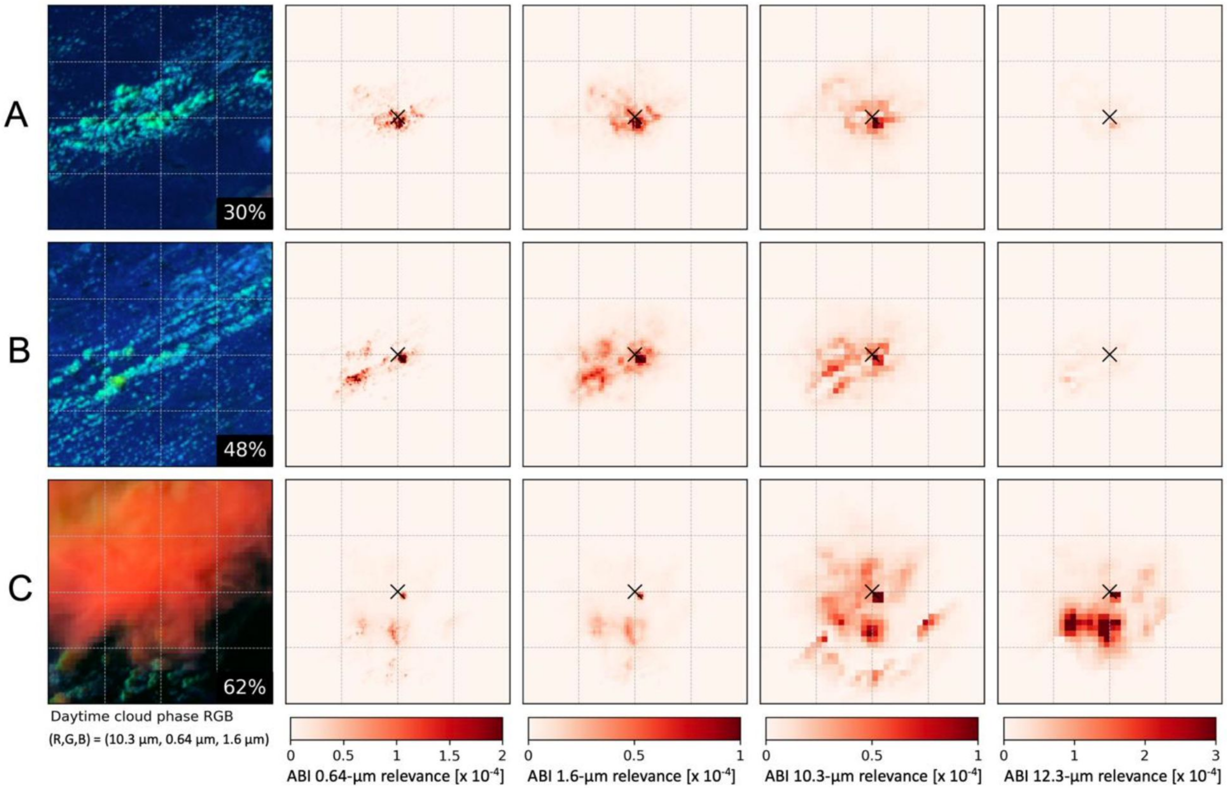
$$\text{LRP}_{\alpha\beta}: \quad R_i^l = \sum_j \left(\alpha \frac{(w_{i,j}x_i)^+}{\sum_i (w_{i,j}x_i)^+ + w_0{}^+} + \beta \frac{(w_{i,j}x_i)^-}{\sum_i (w_{i,j}x_i)^- + w_0{}^-} \right) R_j^{l+1}$$

LRP_{comp} : a combination of LRP_z and $\text{LRP}_{\alpha\beta}$

LRP: Example 1

Adapted from Cintineo et al. (2022; WAF)

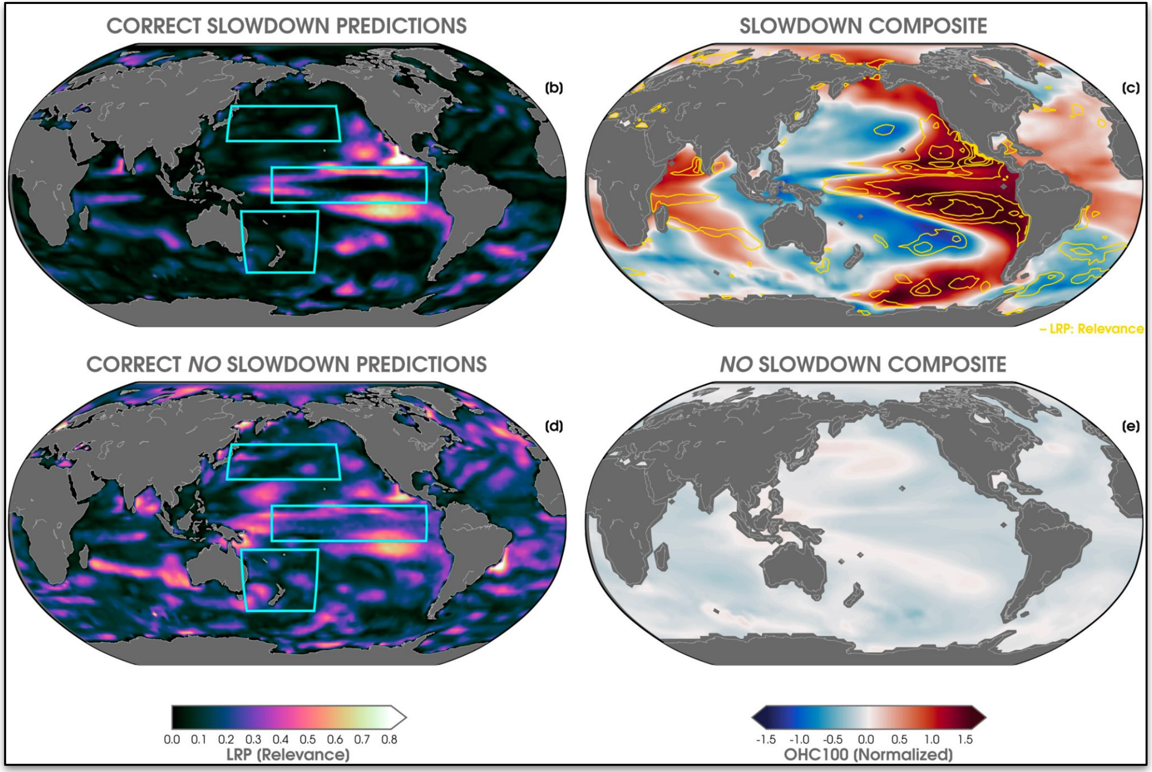
For a UNET forecasting lightning flash location (Cintineo et al. 2022)
LRP highlights relevant meteorological information used in its prediction



LRP Example 2

Adapted from Labe and Barnes (2022; GRL)

LRP of an ANN to detect 'Climate Slowdowns' (Labe and Barnes et al. 2022) suggests precursors to the Interdecadal Pacific Oscillation are important to detecting a slowdown



Big picture comments

Personal notes (Imme):

- **First wave:** The simple methods were used in our field first.
  First "gradient" (aka "saliency"), then "input*gradient", "integrated gradient".

- **Second wave:** Then came LRP and many other methods.
  Our research group used LRP as primary method for quite some time (other groups may have preferred other methods), because it was very suitable for our applications.

  - But now LRP is hard to run: common implementations not compatible with TensorFlow 2.x. Tedious to have to go back to earlier TF versions.
    (That's why we didn't include examples for SEVIR for LRP here.)

  - There are other drawbacks for LRP, too (see Part 2 later). But still very useful for some tasks. So don't discard it, but maybe not first go-to tool.

- **Third wave:** Recently, Shapley / DeepShap is the newest tool. Lots of advantages. Becoming very popular.

  See next slides…

SHAP: SHapley Additive exPlanations (attribution)

Consider the general class of explanation models:

$$f(\mathbf{x}) = R_0 + \sum_i R_i \qquad (1)$$

network

input

attribution to feature i

Any XAI method that can be represented as in Eq. (1), we will say it is an ***additive feature attribution method***.

LRP and other popular XAI methods (e.g., LIME, DeepLIFT) are essentially different solutions to Eq. (1).

Theorem:

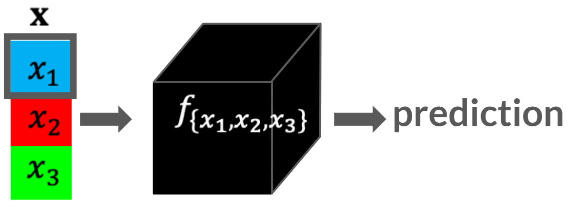
The only solution to Eq. (1) that satisfies the desirable properties of local accuracy, missingness, consistency emerges when R_i are equal to the Shapley values.

$$R_i = \sum_{S \subseteq M \setminus \{i\}} \frac{|S|!\,(|M| - |S| - 1)!}{|M|!} \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S(x_S) \right]$$

From *Lundberg and Lee (2017)*

SHAP (attribution)

Step 1: Consider all the subsets of the input that contain x_1.
Step 2: For all sets in step 1, calculate the importance of x_1, as the difference between the model output when x_1 is present and wher it is missing.
Step 3: The Shapley value of x_1 is the weighted average of the quantities calculated in step 2.



Let's say I want to calculate the Shapley value of x_1

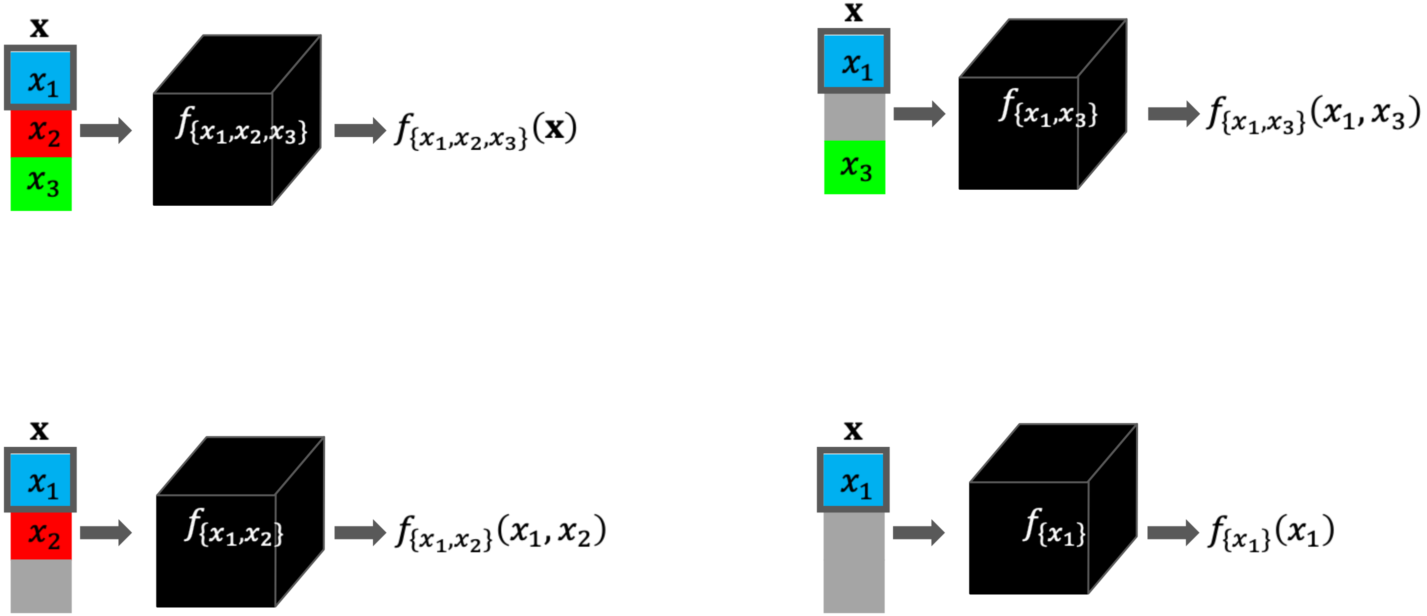
SHAP (attribution)

Let's say I want to calculate the Shapley value of x_1

Step 1: Consider all the subsets of the input that contain x_1.

Step 2: For all sets in step 1, calculate the importance of x_1, as the difference between the model output when x_1 is present and when it is missing.

Step 3: The Shapley value of x_1 is the weighted average of the quantities calculated in step 2.



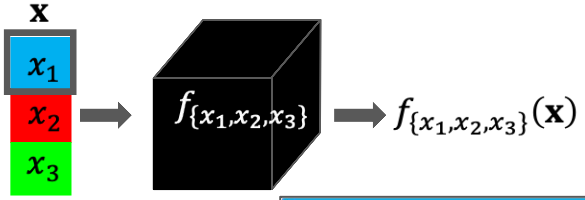
SHAP (attribution)

Let's say I want to calculate the Shapley value of x_1

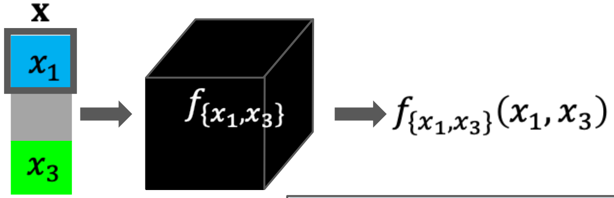
Step 1: Consider all the subsets of the input that contain x_1.

Step 2: For all sets in step 1, calculate the importance of x_1, as the difference between the model output when x_1 is present and when it is missing.

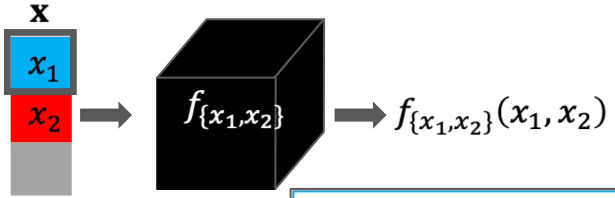
Step 3: The Shapley value of x_1 is the weighted average of the quantities calculated in step 2.



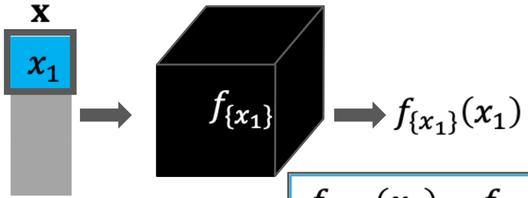
$$f_{\{x_1,x_2,x_3\}}(\mathbf{x}) - f_{\{x_2,x_3\}}(x_2, x_3)$$



$$f_{\{x_1,x_3\}}(x_1, x_3) - f_{\{x_3\}}(x_3)$$



$$f_{\{x_1,x_2\}}(x_1, x_2) - f_{\{x_2\}}(x_2)$$



$$f_{\{x_1\}}(x_1) - f_{\{\emptyset\}}(\emptyset)$$

SHAP (attribution)

Step 1: Consider all the subsets of the input that contain x_1.
Step 2: For all sets in step 1, calculate the importance of x_1, as the difference between the model output when x_1 is present and when it is missing.
Step 3: The Shapley value of x_1 is the weighted average of the quantities calculated in step 2.

$$R_1 = \frac{1}{3}\Big(f_{\{x_1,x_2,x_3\}}(\mathbf{x}) - f_{\{x_2,x_3\}}(x_2,x_3)\Big) + \frac{1}{6}\Big(f_{\{x_1,x_3\}}(x_1,x_3) - f_{\{x_3\}}(x_3)\Big)$$
$$+ \frac{1}{6}\Big(f_{\{x_1,x_2\}}(x_1,x_2) - f_{\{x_2\}}(x_2)\Big) + \frac{1}{3}\Big(f_{\{x_1\}}(x_1) - f_{\{\varnothing\}}(\varnothing)\Big)$$

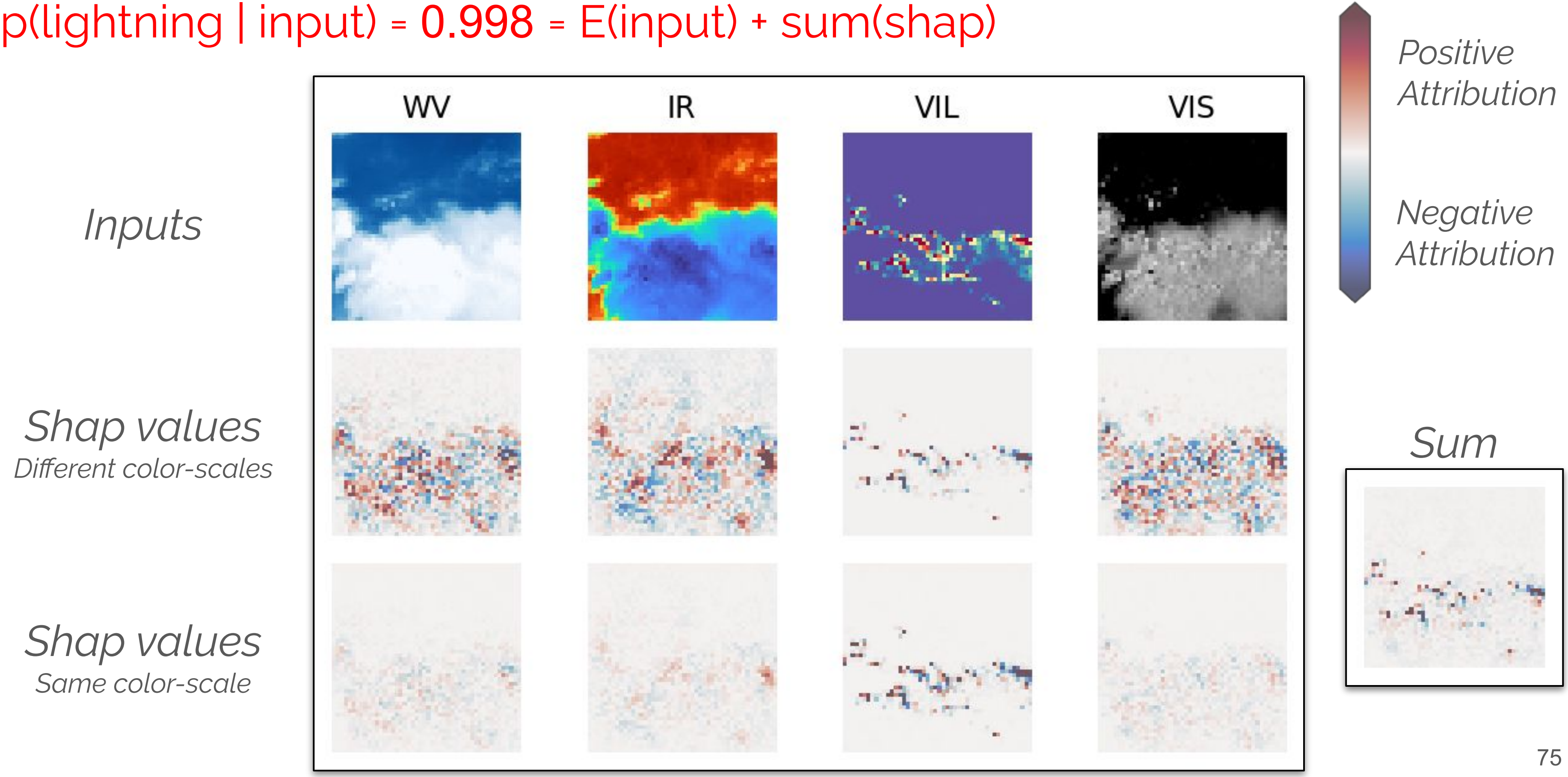
This was only for x_1. The same needs to be done for x_2 and x_3 to get a “heatmap”.

Computationally, estimating the Shapley values for the entire network is very expensive, so the SHAP method uses an approximate algorithm (Deep SHAP), specifically designed for deep neural networks.

Deep SHAP is similar to LRP, except that instead of propagating the relevance, it propagates the Shapley values.

DeepShap: Classification Example

p(lightning | input) = 0.998 = E(input) + sum(shap)



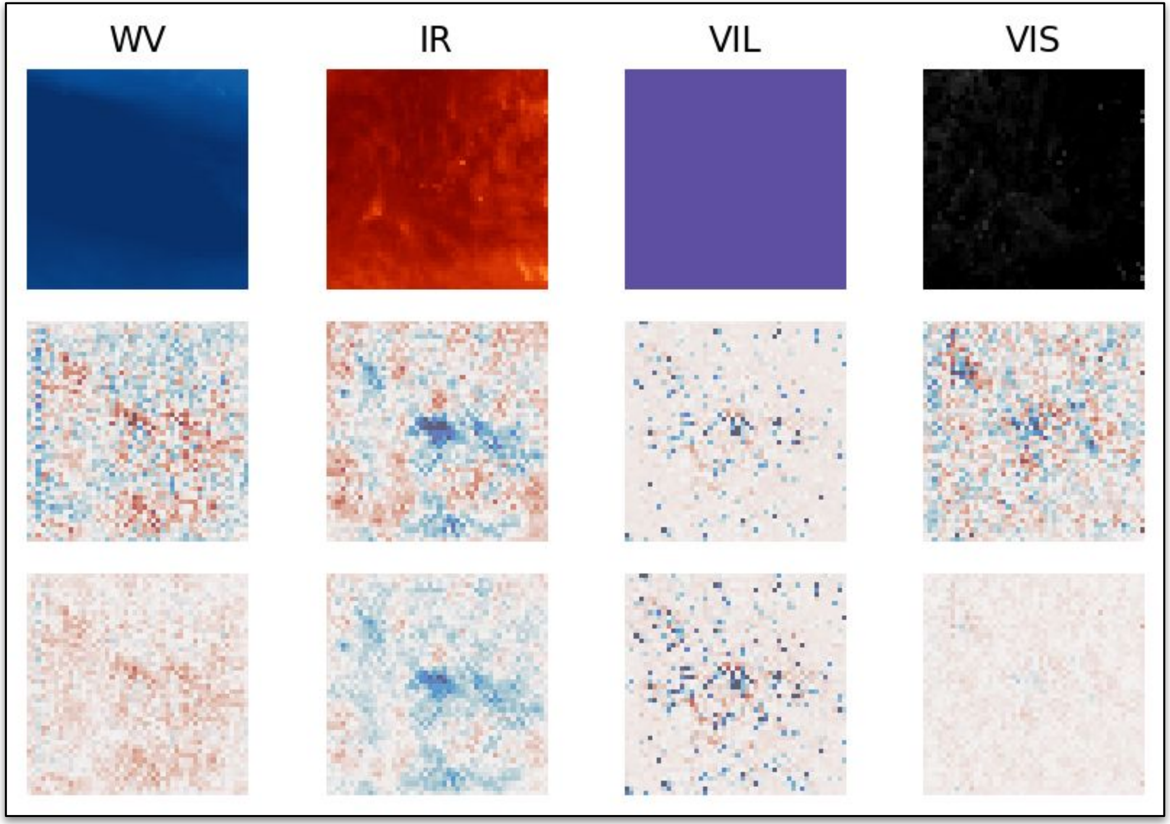
DeepShap: Classification Example 2

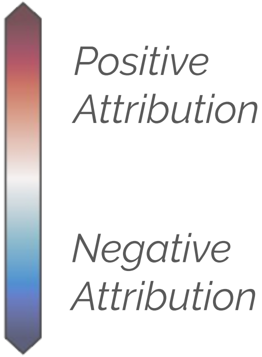
p(lightning | input) = **0.006** = E(input) + sum(shap)

Inputs

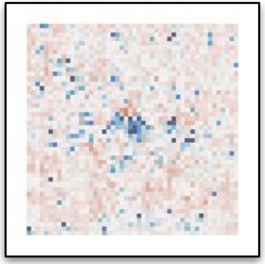
Shap values
Different color-scales

Shap values
Same color-scale





Sum



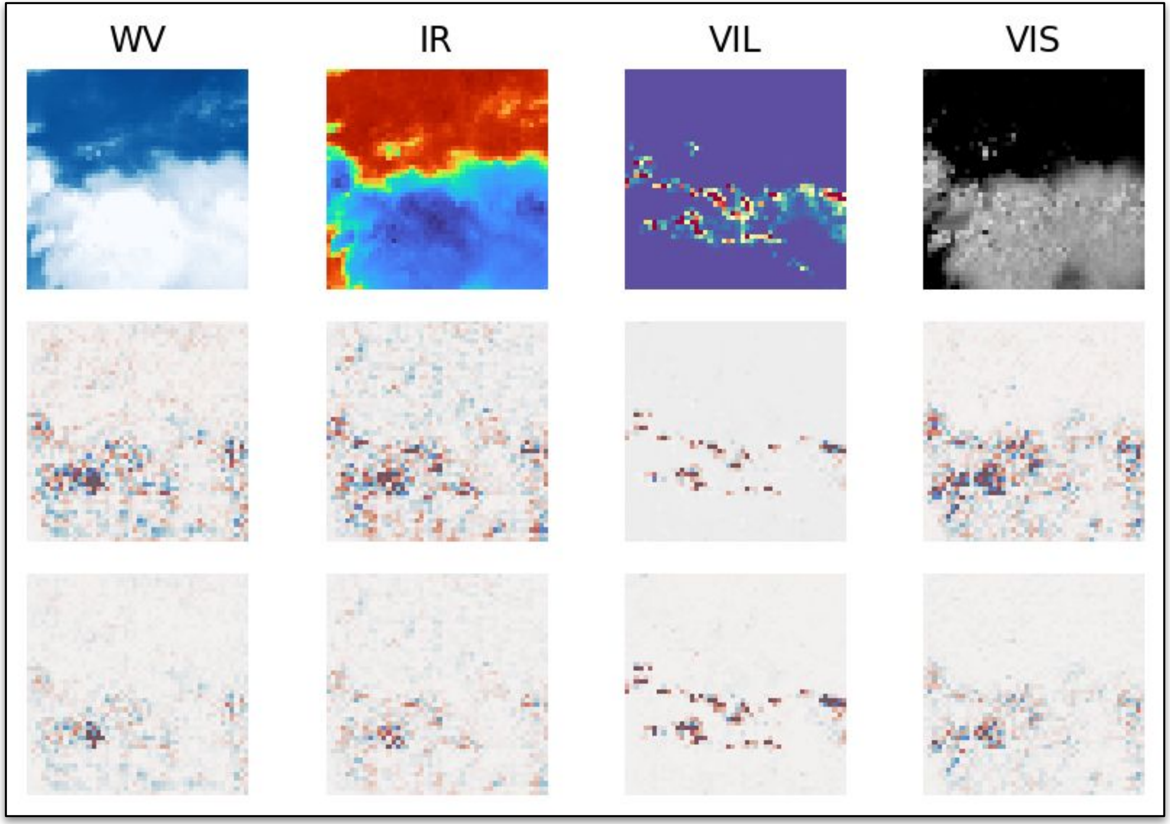
DeepShap: Regression Example

Predicted flash number = **761 flashes** = E(input) + sum(shap)

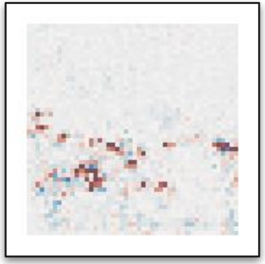
Inputs

Shap values
Different color-scales

Shap values
Same color-scale



Sum



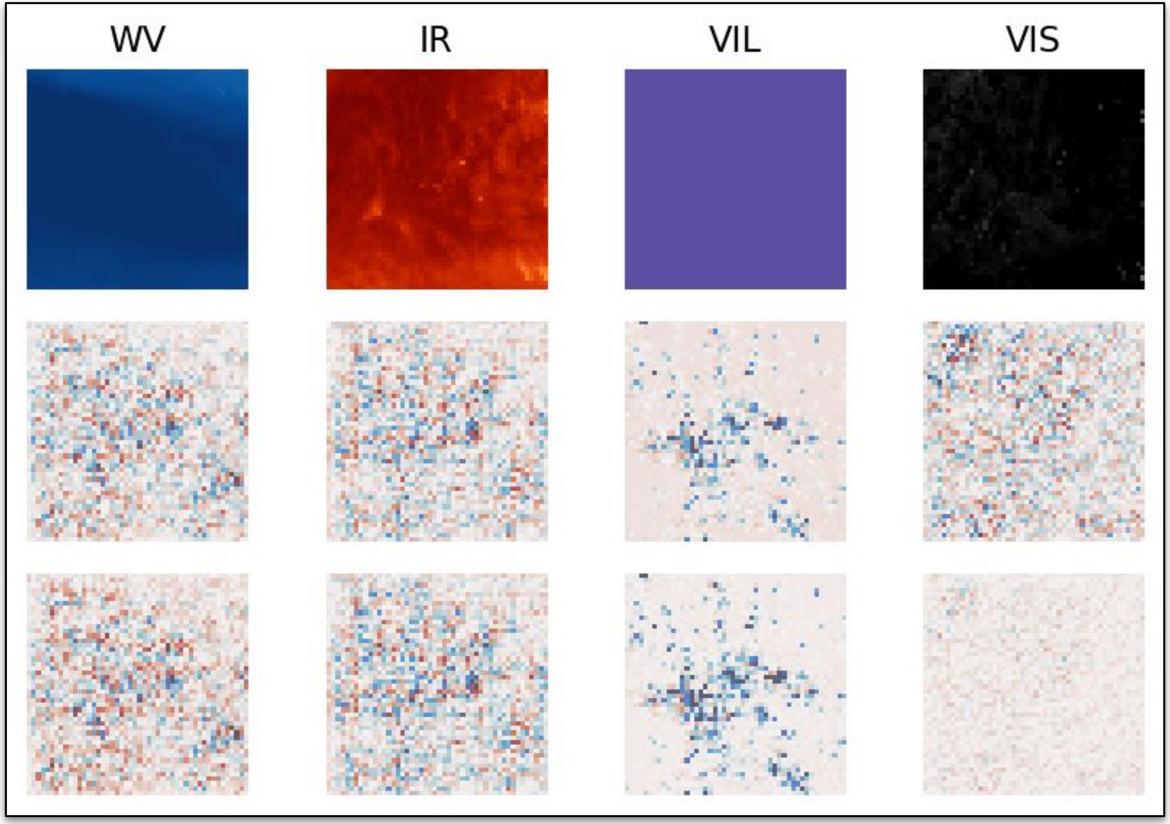
DeepShap: Regression Example 2

Predicted flash number = **4 flashes** = E(input) + sum(shap)

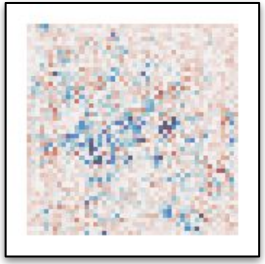
Inputs

Shap values
Different color-scales

Shap values
Same color-scale



Sum



Reminder

All of the XAI examples above were made on Google Colab using the following notebooks:

Saliency:
https://colab.research.google.com/drive/1nkhmeyYEZeXYFtTkd1GfGWA8o-nHuKvC?usp=sharing

Shap:
https://colab.research.google.com/drive/1HbpR37bmPxyMPhqWXne4Pr2KuasWEXtk?usp=sharing

BIG THANKS to **Randy Chase** (at OU) for creating these notebooks!

DeepShap Comments

Personal notes (Imme):

- **Third wave:** Shapley / DeepShap. Becoming very popular.

- DeepShap has many advantages:
  - Better mathematical basis than many other methods.
  - As you just saw for the SEVIR example: often delivers nice strong signal.

- But DeepShap also comes with its own challenges:
  - **DeepShap is very slow:**
    It can be so slow that it might limit the number of samples you can look at.

  - **Memory needs for baseline calculations can be a problem:**
    The first step of DeepShap is to calculate a baseline based on a subset of your training data. Problems:
    - If each sample is very big (e.g., high resolution and many channels), then this first step easily runs out of memory for a decent number of samples.
    - If you use too few samples, then the baseline - and thus results - are not robust.
  - Nevertheless - rapidly increasing in popularity. Might soon be most popular tool.

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Agenda

- 9:00 Explainability vs. Interpretability
- 9:45 *Short brain & bio break #1*
- 9:50 XAI techniques for deep learning (Part 1)
- **11:10 *Short brain & bio break #2***
- 11:15 XAI techniques for deep learning (Part 2)
- Noon: End of session

Questions?

https://app.sli.do/event/1zumy91n

Or go to sli.do and use the code TAI4ES

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

LEAP

NOAA CENTER FOR ARTIFICIAL INTELLIGENCE

Radiant Earth Foundation
EARTH IMAGERY FOR IMPACT

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Day 2: Agenda

- 9:00 Explainability vs. Interpretability
- 9:45 *Short brain & bio break #1*
- 9:50 XAI techniques for deep learning (Part 1)
- 11:10 *Short brain & bio break #2*
- **11:15 XAI techniques for deep learning (Part 2)**
- Noon: End of session

Questions?

https://app.sli.do/event/1zumy91n

Or go to sli.do and use the code TAI4ES

NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

LEAP

NOAA CENTER FOR ARTIFICIAL INTELLIGENCE

Radiant Earth Foundation EARTH IMAGERY FOR IMPACT

Benchmarking XAI

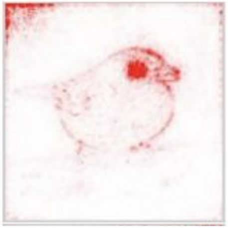
The need for *objectivity* in assessing XAI

Which input features were important for this classification?

Original Image

Guided BackProp

XAI method



XAI heatmap

From *Adebayo et al. (2020)*

Issues : 1) No ground truth to assess the estimated explanations.

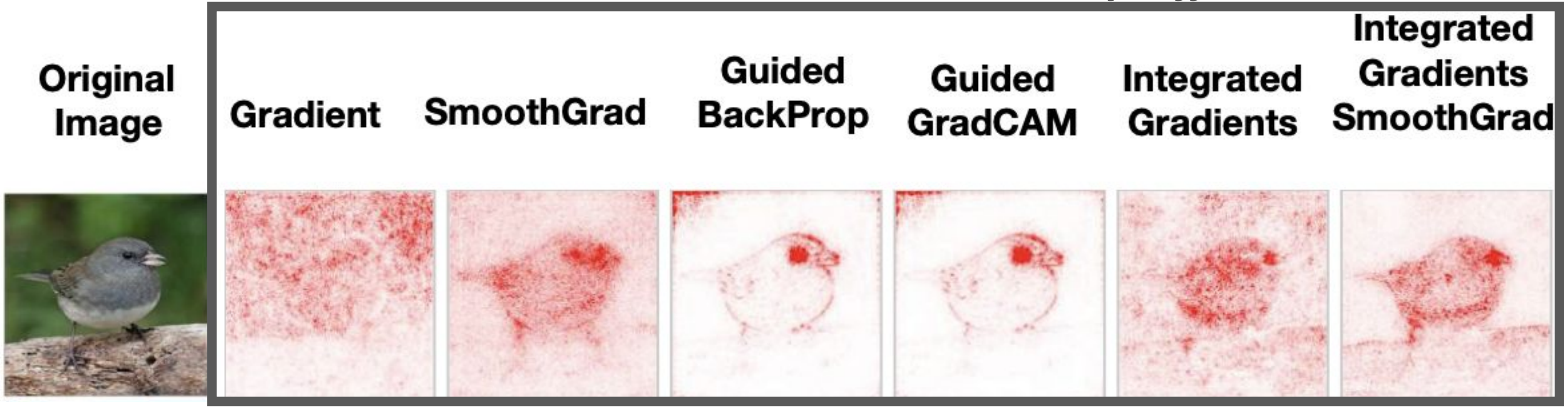
➡ **Pushing back the phrase: "*The explanation looks reasonable*"**

➡ **Remember: The human perception of the explanation alone is NOT a solid criterion for its trustworthiness.**

The need for *objectivity* in assessing XAI

Which input features were important for this classification?

Many Different XAI methods



From *Adebayo et al. (2020)*

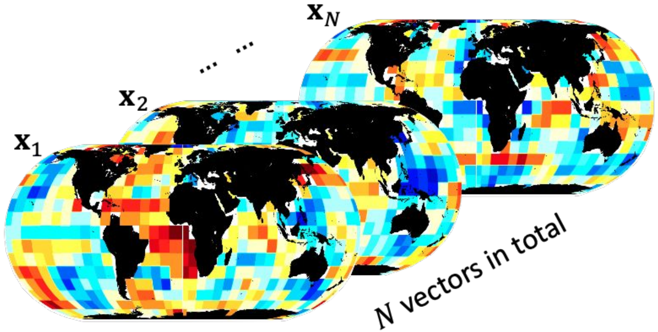
Issues : 1) No ground truth to assess the estimated explanations.
2) Different methods provide different answers.

➡ This is problematic: The uncertainty on how the network decides, leads to limited trust when using neural networks in environmental problems.

➡ We need objective frameworks to rigorously assess XAI methods and gain insights about relative strengths and weaknesses.

Attribution benchmarks for XAI

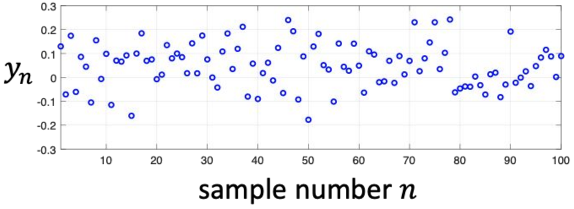
Step 1: Generate N samples of $\mathbf{X} \in \mathbb{R}^d$ from a MVN



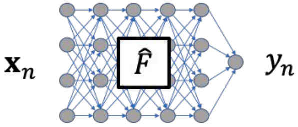
$y_n = F(\mathbf{x}_n)$

Known $F: \mathbb{R}^d \rightarrow \mathbb{R}$

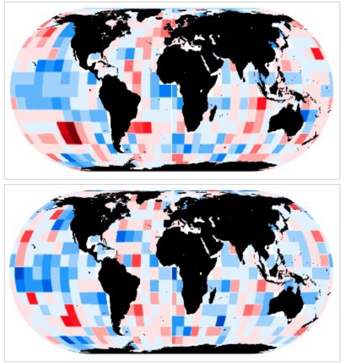
Step 2: Use a known function F that maps each vector $\mathbf{x}_n$ into a scalar y_n



Step 3: Pretend function F is not known and train a NN using inputs $\mathbf{x}_n$ and outputs y_n



Step 4: Use XAI methods to explain the NN and compare with the ground truth from F



F: ground truth

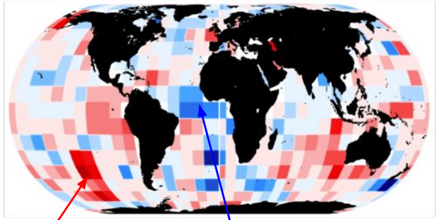
$\hat{F}$: from XAI method

From *Mamalakis et al. (2021)*

Regression Benchmark - Fully Connected Network

y_n : 0.0660
NN prediction: 0.0802

Ground Truth of Attribution for F



Positive contribution
Negative contribution

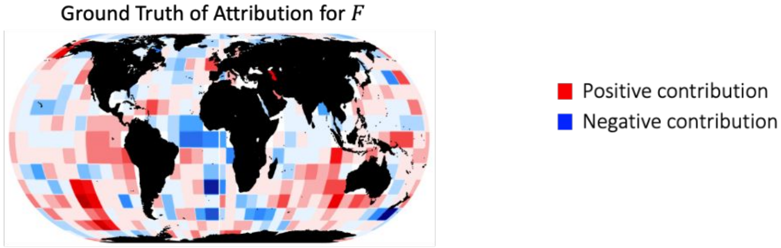
Red color highlights features that contributed positively to y

Blue color highlights features that contributed negatively to y

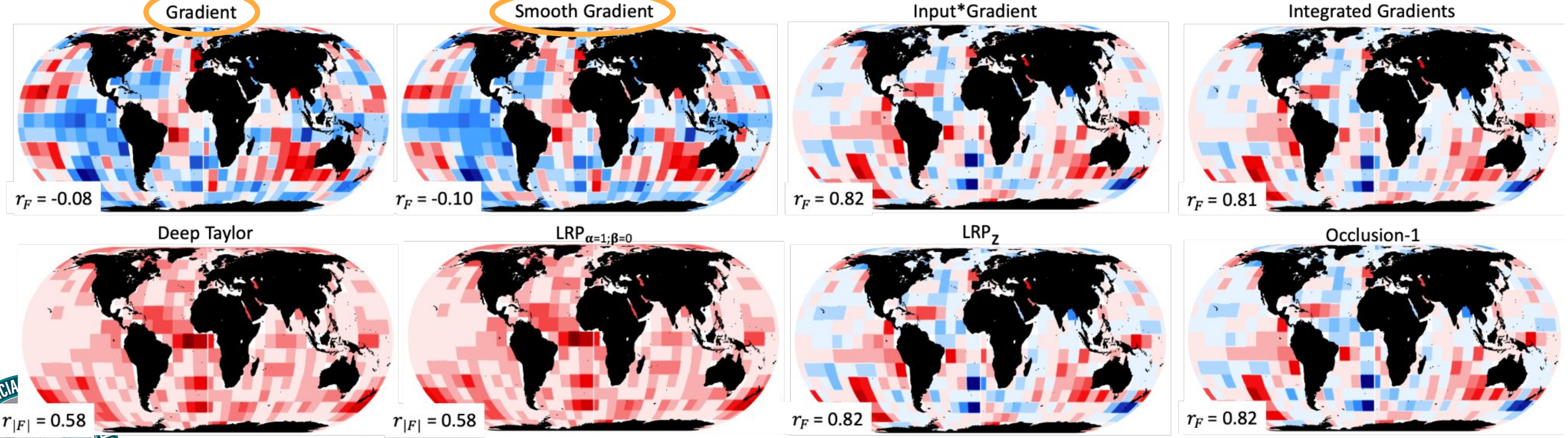
From *Mamalakis et al. (2021)*

Regression Benchmark - Fully Connected Network

y_n : 0.0660
NN prediction: 0.0802

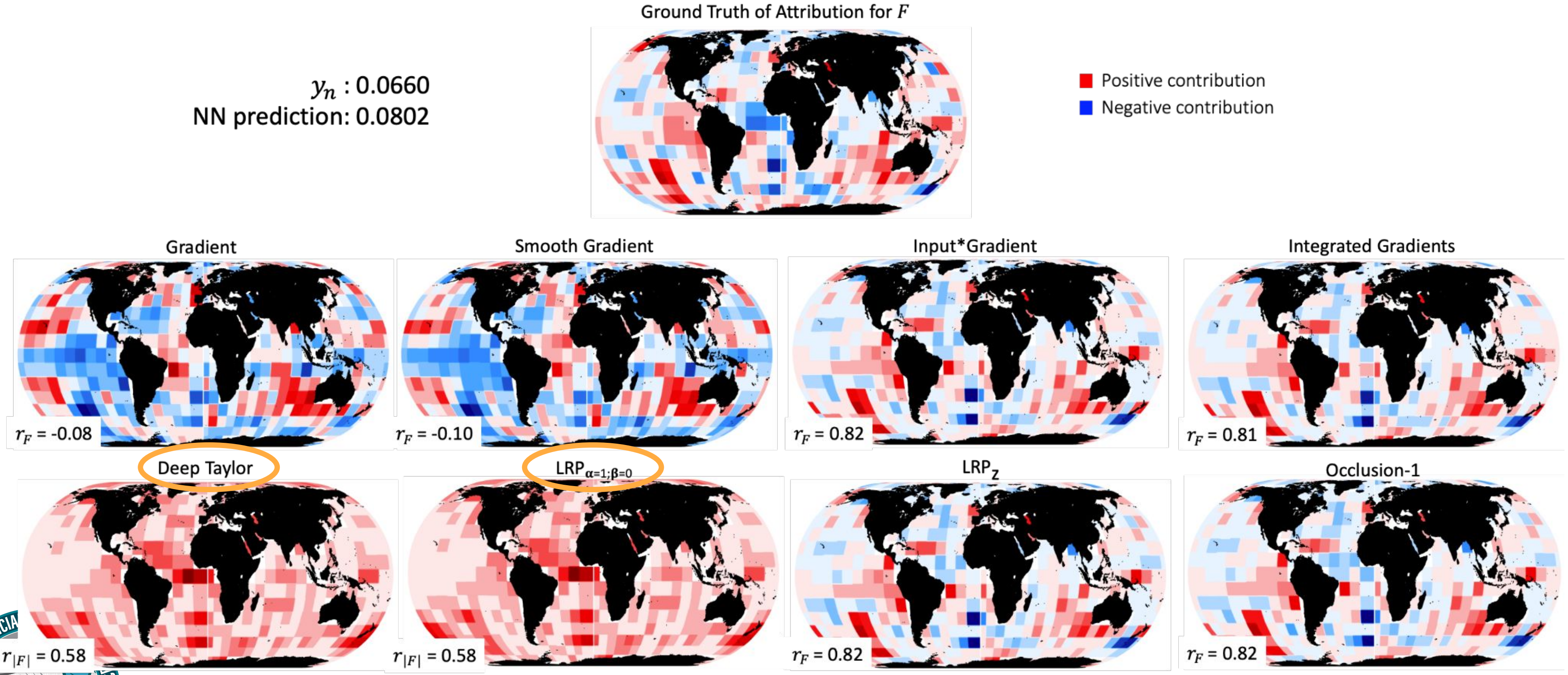


Do not correlate with the ground truth of attribution



From *Mamalakis et al. (2021)*

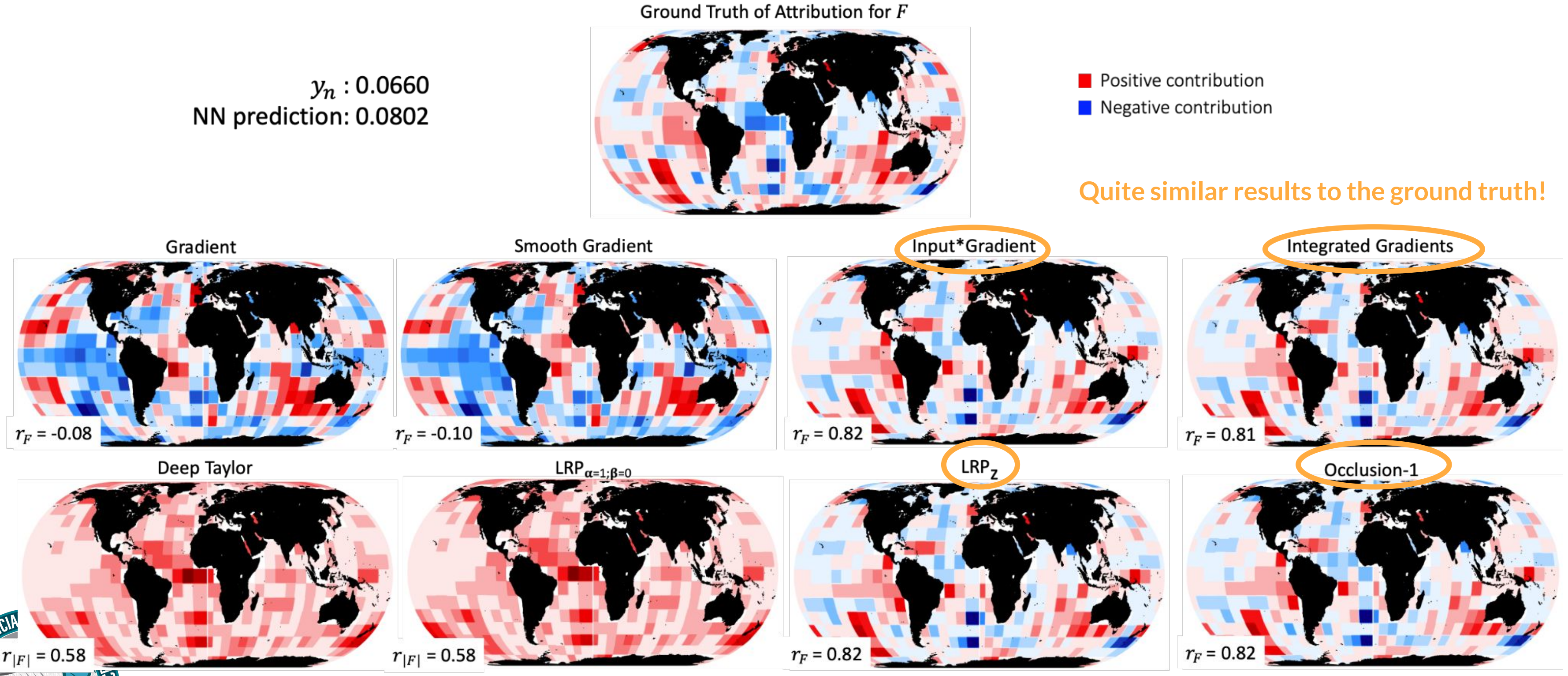
Regression Benchmark - Fully Connected Network



Provide only positive attributions. Cannot distinguish the true sign of attribution.

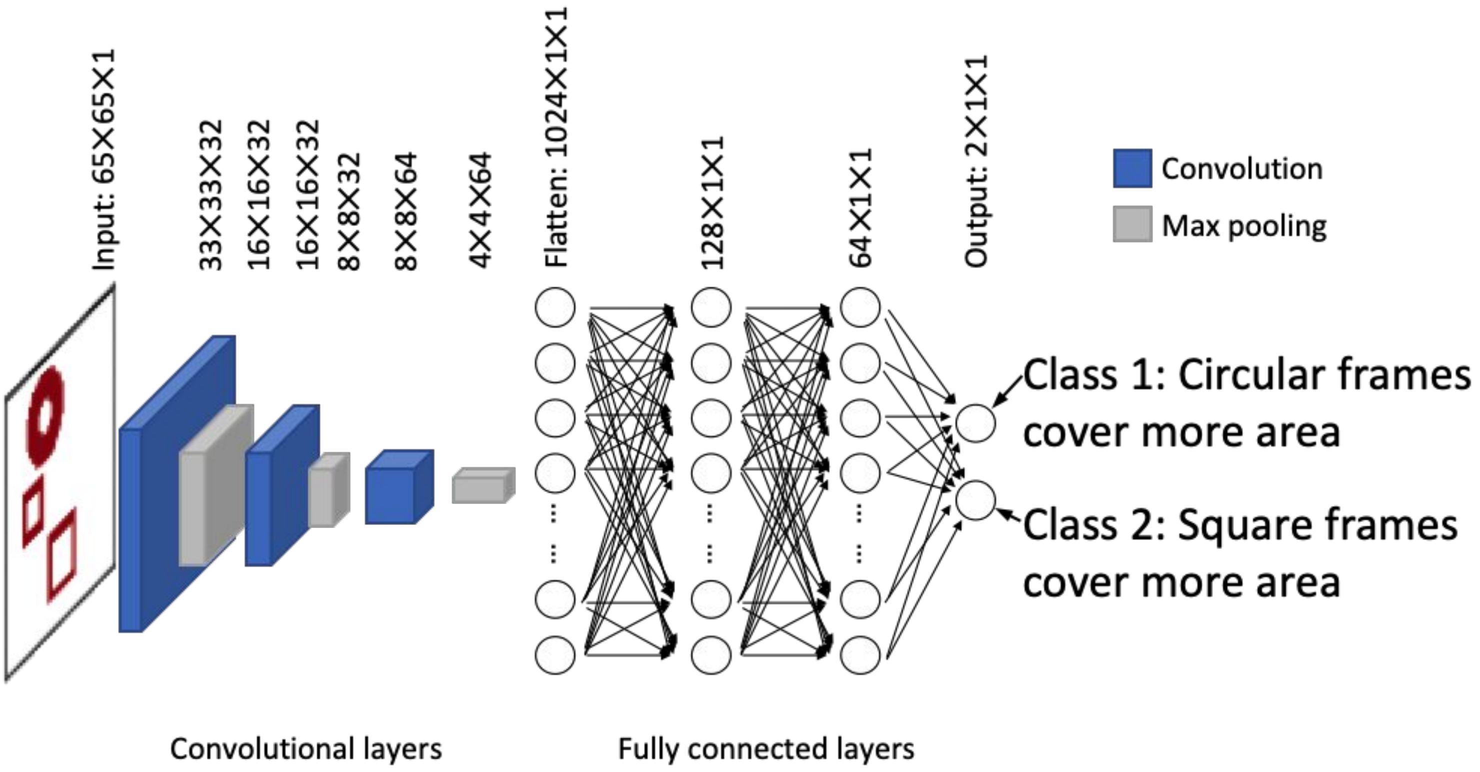
From *Mamalakis et al. (2021)*

Regression Benchmark - Fully Connected Network



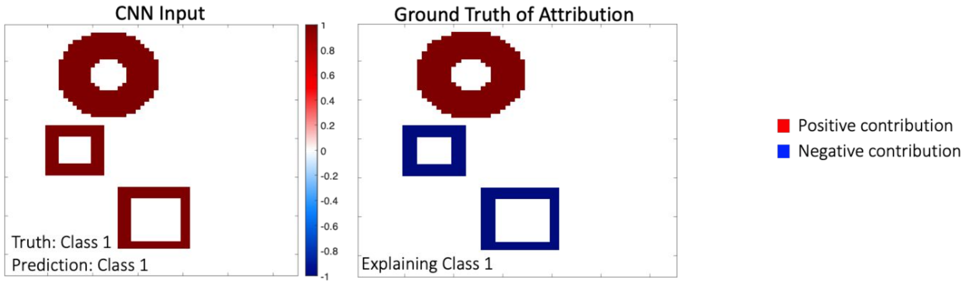
From *Mamalakis et al. (2021)*

Classification Benchmark - Convolutional Network



From *Mamalakis et al. (2022)*

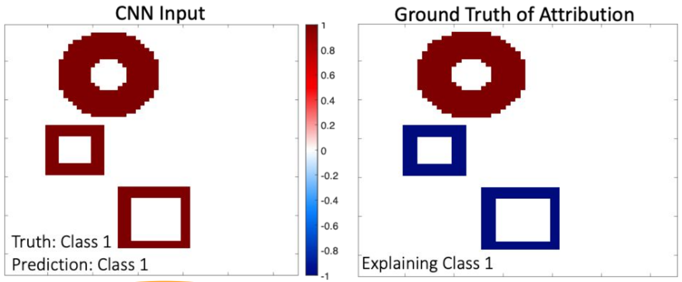
Classification Benchmark - Convolutional Network



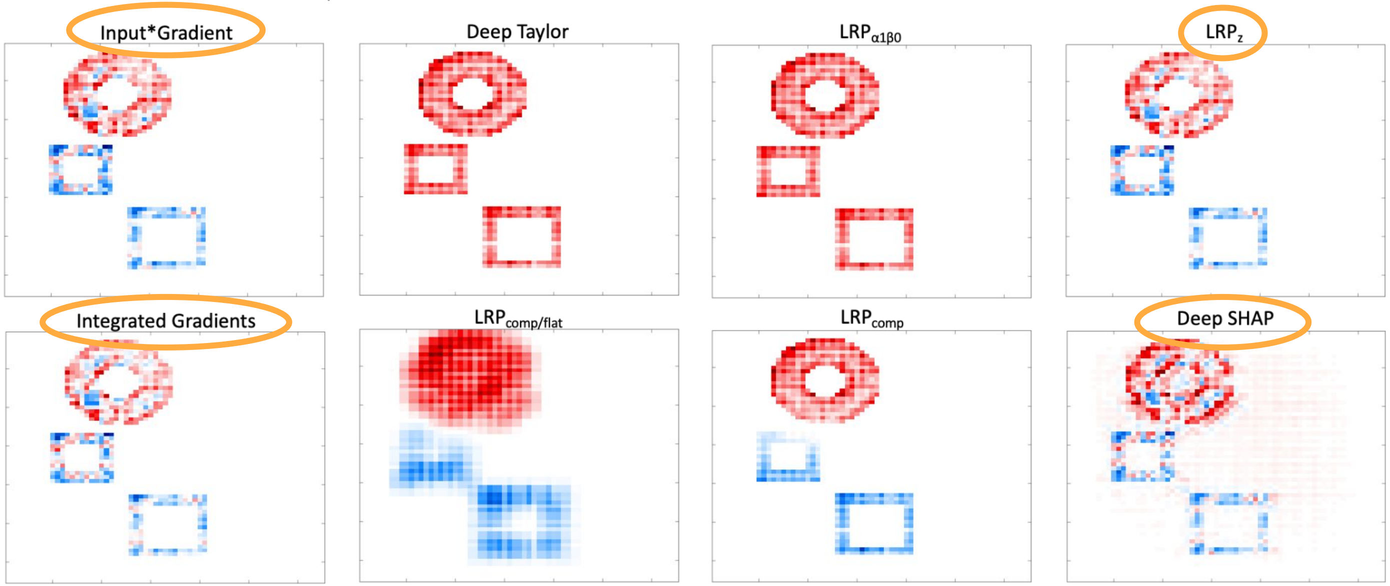
- **In this image, the truth is that the circular frame covers more area than the square frames. The CNN has correctly classified this input.**
- **Regarding to the ground truth of the attribution, we expect that the presence of the circular frame contributed positively to the CNN's decision, while the presence of the two square frames decreased its certainty.**

From *Mamalakis et al. (2022)*

Classification Benchmark - Convolutional Network

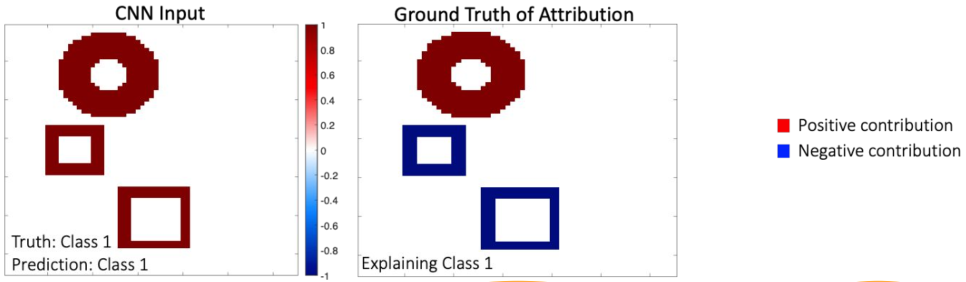


These methods exhibit noisy results. This is related to a phenomenon called *gradient shattering* that typically occurs in deep networks.

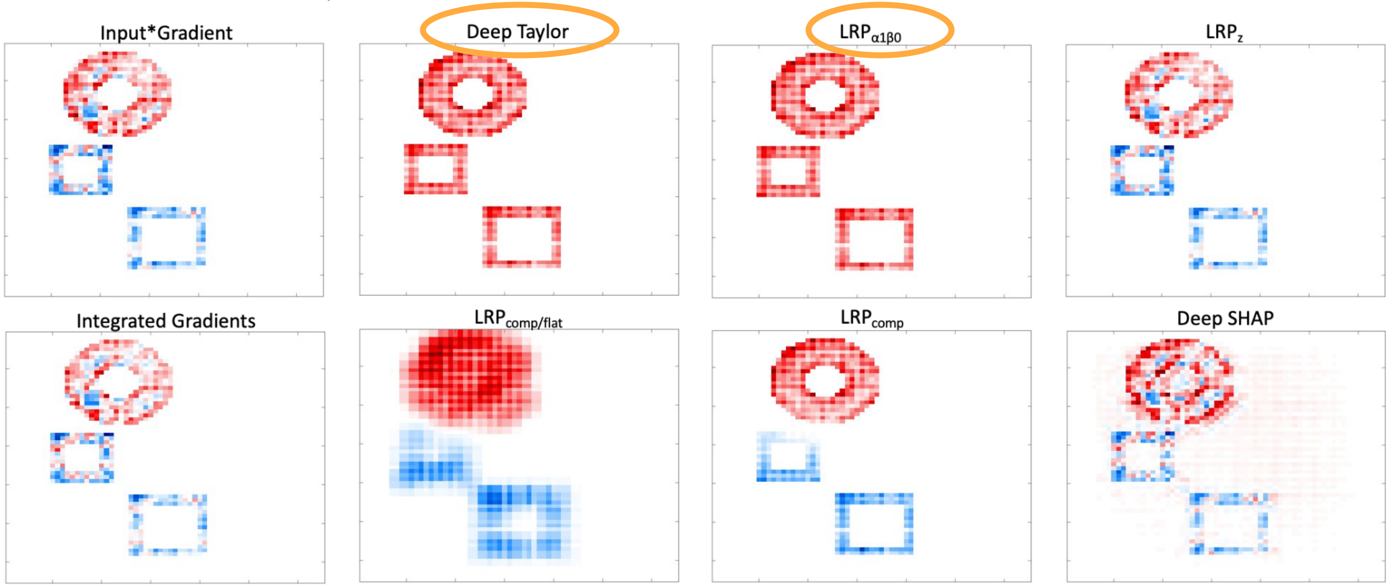


From *Mamalakis et al. (2022)*

Classification Benchmark - Convolutional Network

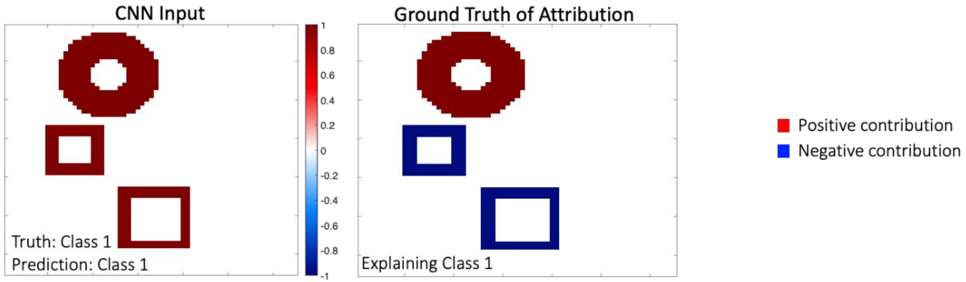


As shown before, these methods cannot distinguish between the sign of the attribution.

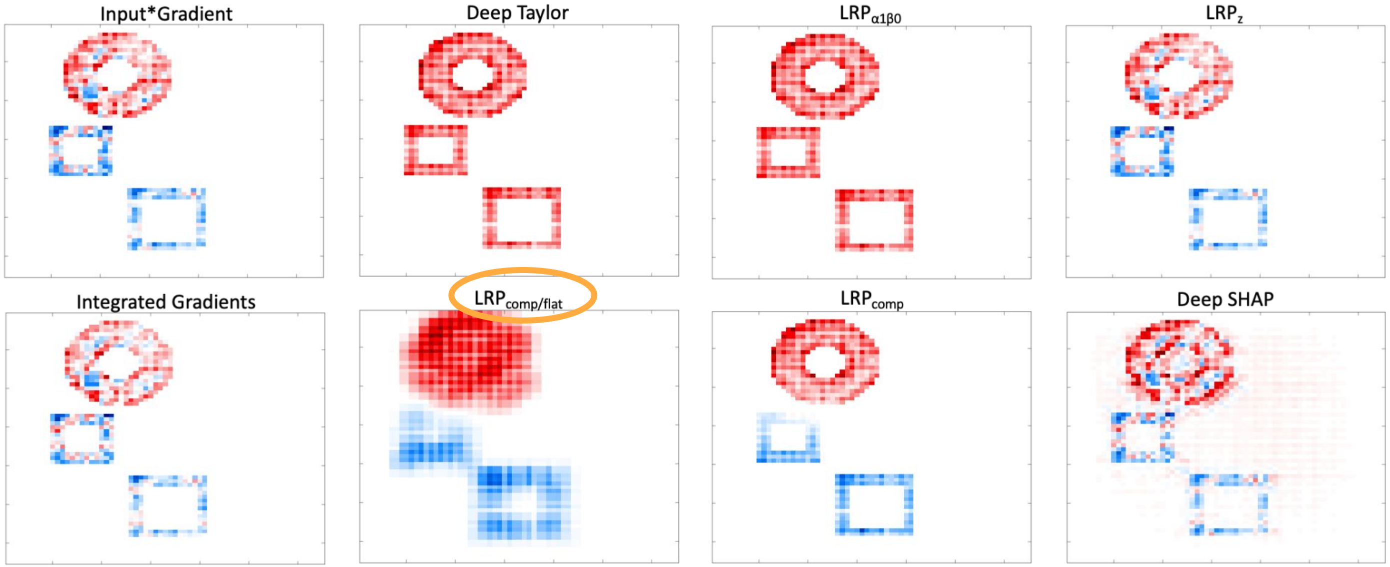


From *Mamalakis et al. (2022)*

Classification Benchmark - Convolutional Network

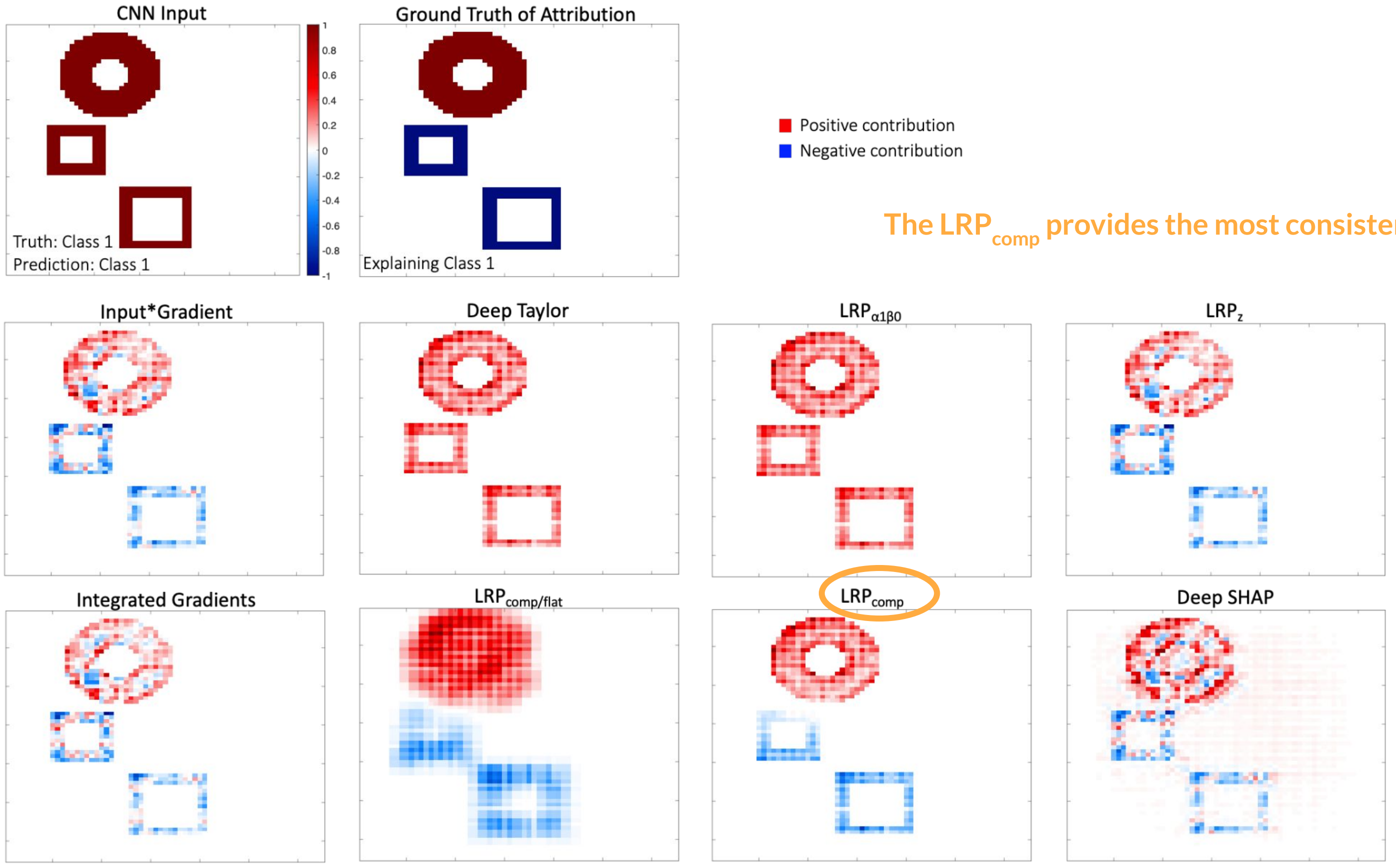


The LRP$_{comp/flat}$ provides a coarser picture of the true attribution; not ideal if local accuracy is of interest.



From *Mamalakis et al. (2022)*

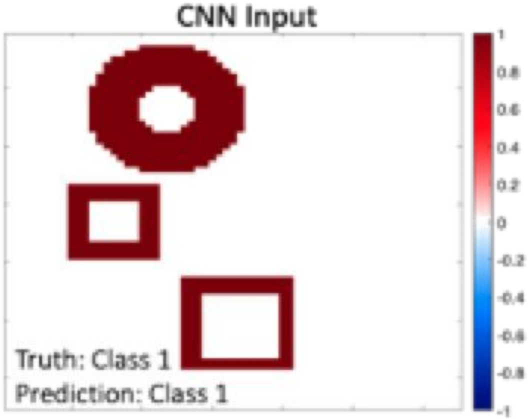
Classification Benchmark - Convolutional Network



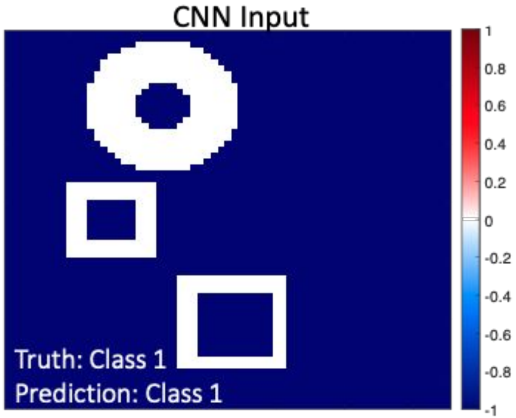
The LRP_{comp} provides the most consistent attribution.

From *Mamalakis et al. (2022)*

Classification Benchmark - Shifting the Input



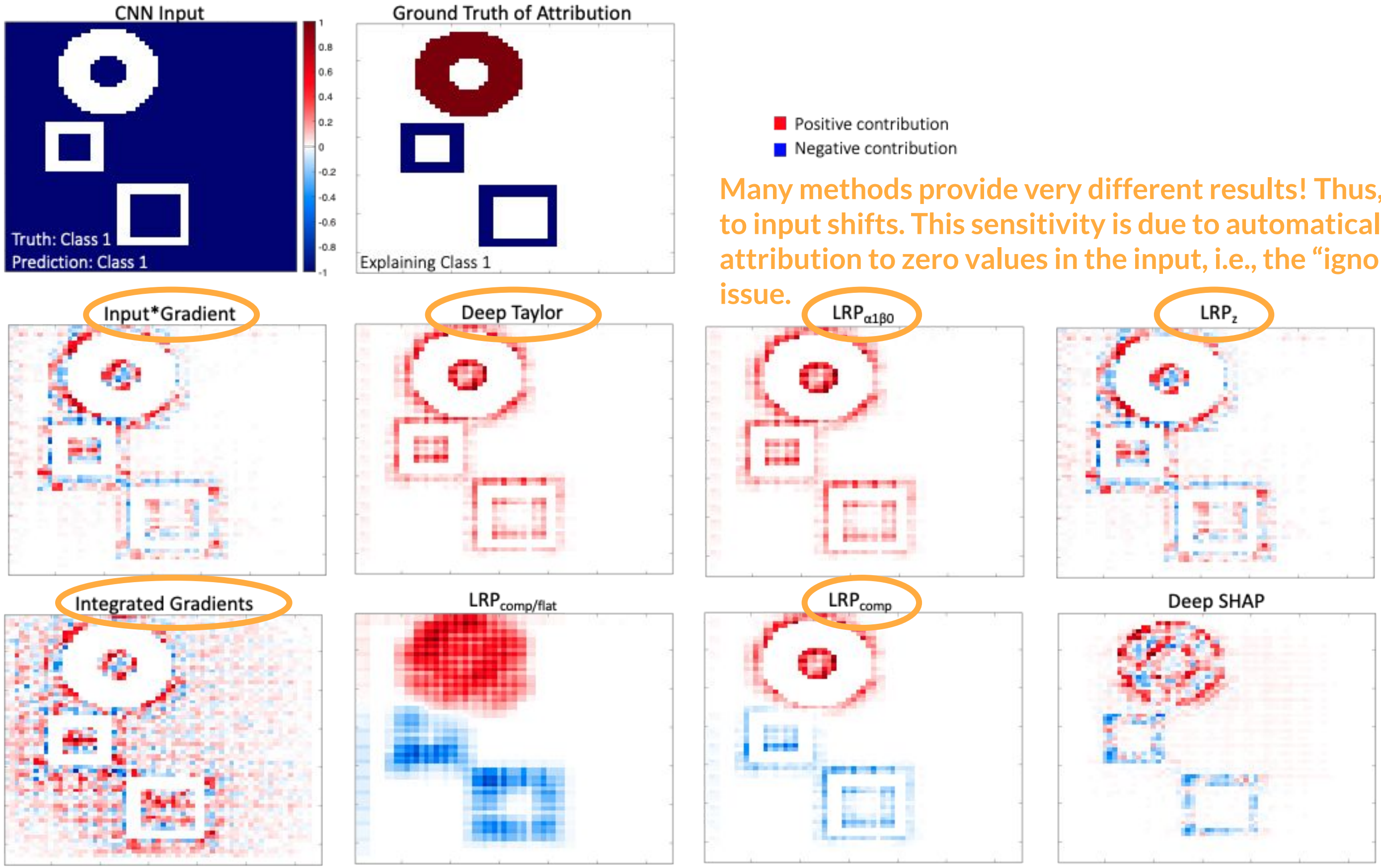
$x_{shifted} = x - 1$



We modify the first layer of the network so that the final predictions are exactly the same.

From *Mamalakis et al. (2022)*

Classification Benchmark - Shifting the Input



Many methods provide very different results! Thus, they are sensitive to input shifts. This sensitivity is due to automatically assigning a zero attribution to zero values in the input, i.e., the "ignorant to zero input" issue.

From *Mamalakis et al. (2022)*

Best practices of XAI

Our investigation revealed aspects that need to be considered when applying XAI methods:

i) Gradient shattering is the phenomenon of noisy patterns in the gradient, the level of which is a function of the depth of the network. Gradient shattering might lead to overwhelmingly noisy patterns that make the explanation of any gradient-based method incomprehensible.

ii) Unable to disentangle positive and negative contributions. This may lead to a very distorted picture of what the network's strategy is and possibly limit trust in the predictive model itself.

iii) Ignorant to zero input: Some methods automatically assign a zero attribution to zero values in the input, despite the fact that in specific settings a zero input value could be important for the prediction.

From *Mamalakis et al. (2022)*

Best practices of XAI

Table 1. Summary of XAI methods considered in this study. Practical strengths (✓) and weaknesses (✗) of each method are also reported.

XAI method		Brief summary of the method	Desired property for CNN applications as explored in this study: disentangles the sign of relevance	insensitive to gradient shattering	not ignorant to zero input	Extra comments/insights
Gradient (Simonyan et al., 2014)		Calculates the first partial derivative of the model output with respect to the input. (sensitivity)	✓	✗	✓	Estimates the sensitivity of the output to the input, which is not the same as the attribution; see Appendix B
Smooth Gradient (Smilkov et al., 2017)		Calculates the average gradient across many perturbed inputs. (sensitivity)	✓	✗	✓	
Input*Gradient (Shrikumar et al., 2017)		Multiplies the input with the gradient. (attribution)	✓	✗	✗	
Integrated Gradients (Sundararajan et al., 2017)		Multiplies the average gradient along the straight line between the input point and a reference point with the corresponding distance between the two points. (attribution)	✓	✗	✓	
LRP	**α1β0** (Bach et al., 2015)	Layer-wise back propagation of each neuron's relevance based on the α1β0-rule. (attribution)	✗	✓	✗	Considers only positive preactivations
	z (Bach et al., 2015)	Layer-wise back propagation of each neuron's relevance based on the z-rule (attribution)	✓	✗	✗	Equivalent to Input*Gradient for networks using ReLU activations
	comp (Kohlbrenner et al., 2020)	Layer-wise back propagation of each neuron's relevance by combining the α1β0-rule and the z-rule. (attribution)	✓	✓	✗	Combines the strengths of LRPz and $LRP_{\alpha1\beta0}$
	comp/flat (Kohlbrenner et al., 2020)	Layer-wise back propagation of each neuron's relevance by combining the α1β0-rule, the z-rule and the flat rule. (attribution)	✓	✓	✓	Provides a coarser picture of attribution; not suitable if local accuracy necessary
Deep Taylor (Montavon et al., 2017)		Applies Taylor decomposition of the relevance function for each neuron recursively. (attribution)	✗	✓	✗	Equivalent to $LRP_{\alpha1\beta0}$ for networks using ReLU activations; not defined for negative predictions
PatternNet (Kindermans et al., 2017a)		Calculates the signal in the input for each neuron recursively. (signal)	✗	✓	✓	Estimates the signal (not the same as the attribution)
PatternAttribution (Kindermans et al., 2017a)		Calculates the attribution in the direction of the signal for each neuron recursively. (attribution)	✗	✓	✓	
Deep SHAP (Lundberg and Lee, 2017)		Approximates Shapley values for each neuron recursively (attribution)	✓	✗	✓	Based on well-founded theory; computationally expensive

From *Mamalakis et al. (2022)*

Key take home messages from benchmarks

- **XAI methods show potential to be a game-changer in how we predict/detect patterns in Earth Sciences. We can use these tools to calibrate model trust, fine-tune models and learn new science.**

- **Given the plethora and the diversity of methods out there, the lack of a ground truth to assess their fidelity has the risk of allowing subjective assessment, and cherry-picking certain methods. It is important to introduce *objectivity* in XAI assessment and shed light to relative strengths and weaknesses.**

- ***Engagement* of attribution benchmarks may lead to a more *cautious* and *successful* implementation of XAI methods.**

Final comments / big picture

Recall - The Big Picture

- Applying local XAI methods to identify the strategies a NN has learned … is a **Detective Game.**
- Expect to only get clues - rather than complete answers.
- It's usually a <u>lengthy</u> process, where you try one method after the other to find clues, then generate hypotheses.
  - Different methods tell you different things.
  - Local XAI methods look at one sample at a time.

What you can hope for:

- Finding clues that tell you about *potential* strategies.
- Then you can design tests to verify these strategies.
- You will never find *all* strategies - instead hope to find most important ones (but no guarantee).

Questions to tackle

1. **Which methods should I use?**
   a. Carefully consider which question you want to ask.
   b. Look at benchmark to see (and table above) to help you decide what is best for your application.
   c. Try more than one method, and interpret results accordingly.

2. **Which samples should I look at?**
   Often we select samples to show extremes:
   Which samples produced highest (lowest) output?
   Which samples did the model perform best/worst for?
   Which samples represent extreme conditions (strong/no lightning)?

3. **How do I ensure results are consistent across other samples without looking at *all* samples?**
   First of all, you can't guarantee that - ever - with local tools.
   Once you have a hypothesis you can devise test, e.g., with synthetic data, by modifying existing samples, etc.

Recall - The Big Picture

Questions you will have to tackle:

4. **How should I interpret the results?**
    - Interpret results as clues / as hypotheses to be tested.
    - Environmental scientist needs to be core person interpreting results (with help of data scientist).

5. **If I use visual inspection of results: how objective is that?**
    - **Output of these XAI method is an image** (heatmap).
    - **Image needs to be interpreted by a human.**
    - Potential for confirmation bias:
      we may "see" those patterns in the images that we *want* to see, i.e. that match our hypothesis.
    - Also - we all love to cherry-pick, and we all do it!
      (Cherry-picking = showing only "good" results)
    - Again, if we treat our results as hypotheses, rather than firm statements, we have the right mindset to interpret them accordingly.
    - Try to confirm with other means → design experiments to test hypothesis. Not always possible.

ARTIFICIAL INTELLIGENCE · ENVIRONMENTAL SCIENCE · RISK COMMUNICATION

Online Resources

INNVESTIGATE	https://github.com/albermax/innvestigate https://innvestigate.readthedocs.io/en/latest/modules/analyzer.html
SHAP	https://github.com/slundberg/shap
Saliency	https://github.com/PAIR-code/saliency https://pair-code.github.io/saliency/#home
Example SEVIR Notebooks (by Randy Chase)	Saliency: https://colab.research.google.com/drive/1nkhmeyYEZeXYFtTkd1GfGWA8o-nHuKvC?usp=sharing Shap: https://colab.research.google.com/drive/1HbpR37bmPxyMPhqWXne4Pr2KuasWEXtk?usp=sharing

References

J. Adebayo et al. (2020) "Sanity checks for saliency maps," arXiv preprint, https://arxiv.org/abs/1810.03292.

Bach, et al. (2015) "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation", *PLOS ONE*, https://doi.org/10.1371/journal.pone.0130140

Chase, et al. (2022) "A Machine Learning Tutorial for Operational Meteorology, Part I: Traditional Machine Learning", *WAF*, *accepted*, https://arxiv.org/abs/2204.07492

Chase, et al. (*in prep.*) "A Machine Learning Tutorial for Operational Meteorology, Part II: Neural Networks", *WAF.*

Cintineo et al. (2022) "ProbSevere LightningCast: A deep-learning model for satellite-based lightning nowcasting", *WAF*, pp https://doi.org/10.1175/WAF-D-22-0019.1

Labe and Barnes (2022) "Predicting Slowdowns in Decadal Climate Warming Trends With Explainable Neural Networks", *GRL* https://doi.org/10.1029/2022GL098173

Lundberg, S. M. and S. I. Lee (2017) "A unified approach to interpreting model predictions," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 4768-4777.

Mamalakis, A., I. Ebert-Uphoff and E.A. Barnes (2021) "Neural Network Attribution Methods for Problems in Geoscience: A Novel Synthetic Benchmark Dataset", arXiv preprint, https://arxiv.org/abs/2103.10005, accepted in *Environmental Data Science*, doi: 10.1017/eds.2022.7.

Mamalakis, A., I. Ebert-Uphoff, E.A. Barnes "Explainable Artificial Intelligence in Meteorology and Climate Science: Model fine-tuning, calibrating trust and learning new science," in *Beyond explainable Artificial Intelligence* by Holzinger et al. (Editors), Springer Lecture Notes on Artificial Intelligence, open access at: *https://link.springer.com/chapter/10.1007/978-3-031-04083-2_16*

Mamalakis, A., E.A. Barnes, I. Ebert-Uphoff (2022) "Investigating the fidelity of explainable artificial intelligence methods for applications of convolutional neural networks in geoscience", arXiv preprint, https://arxiv.org/abs/2202.03407, accepted in *Artificial Intelligence for the Earth Systems.*

Samek, et al. (2021), "Explaining Deep Neural Networks and Beyond: A review of Methods and Applications", in *Proceedings of the IEEE*, vol. 109, no. 3, pp. 247-278, March 2021, doi: 10.1109/JPROC.2021.3060483

Trustworthy Artificial Intelligence for Environmental Science (TAI4ES) Summer School

Time for any open questions!

Questions?

https://app.sli.do/event/1zumy91n

Or go to sli.do
and use the
code TAI4ES

Thank you!

- This material is based upon work supported by the National Science Foundation under Grant No. ICER-2019758.
- This summer school is being supported by NCAR/UCAR
- Thank you to:
  - Taysia Peterson and the multi-media team @ NCAR
  - Susan Dubbs @ OU
  - Our sponsors! NCAR/UCAR, Google cloud, LEAP, Radiant Earth, NCAI.
  - All of our speakers
  - All of you for coming and participating!