berlin

ATB
Leibniz-Institut für
Agrartechnik und Bioökonomie

Quantus x Climate - Applying Explainable AI Evaluation in Climate Science

Philine Lou Bommer*

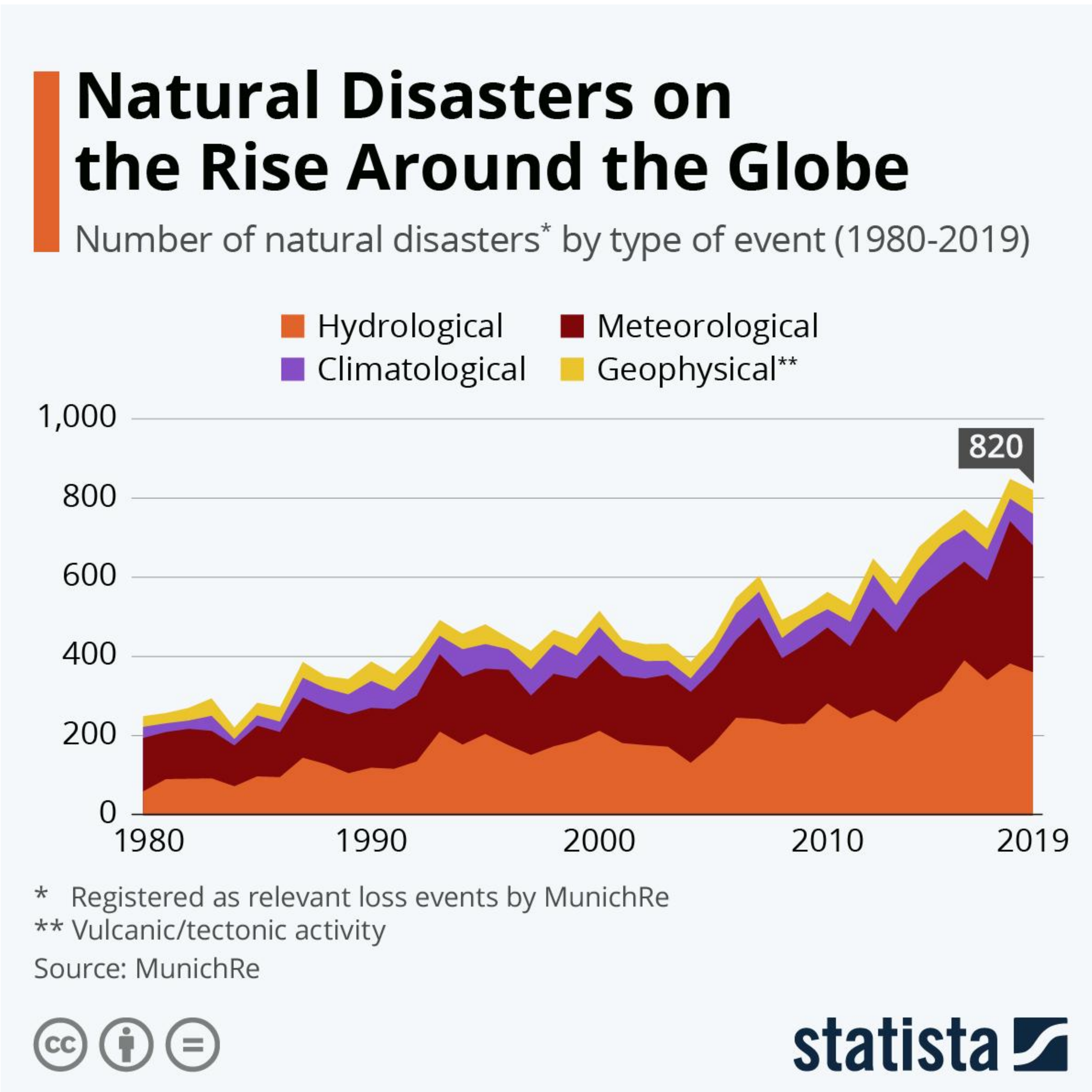
Anna Hedström*

Marlene Kretschmer

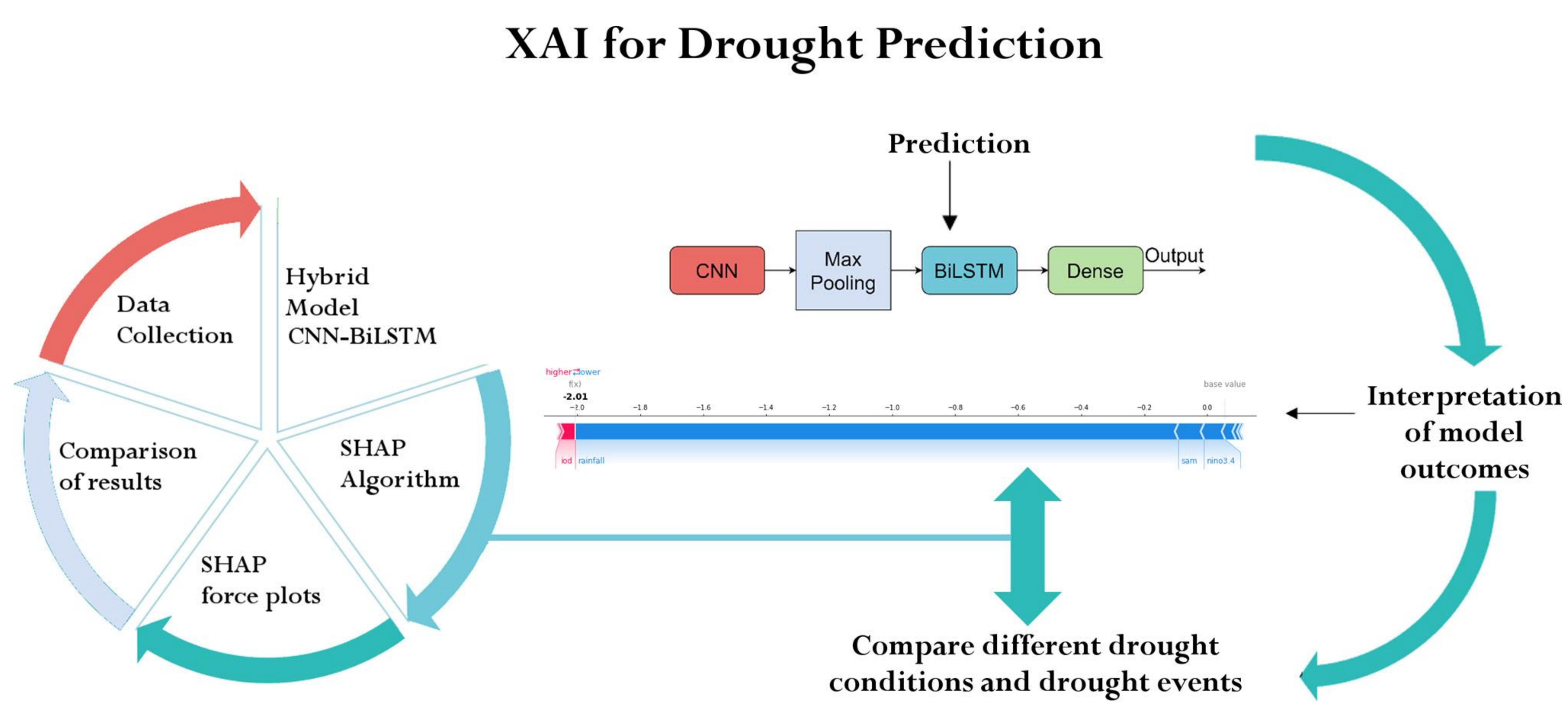
Marina M.-C. Höhne

ICLR
International Conference On
Learning Representations

1. Introduction



- ❖ **Climate change** causes economical and humanitarian losses
- ❖ **Climate research** to tackle monitoring and prediction
- ❖ **AI** more important in climate research but **black-box**
- ❖ **Explainable AI (XAI)**: deeper understanding of network decision
  - ➢ assessment of the model skill (trustworthiness and reliability)



Dikshit et. al 2021

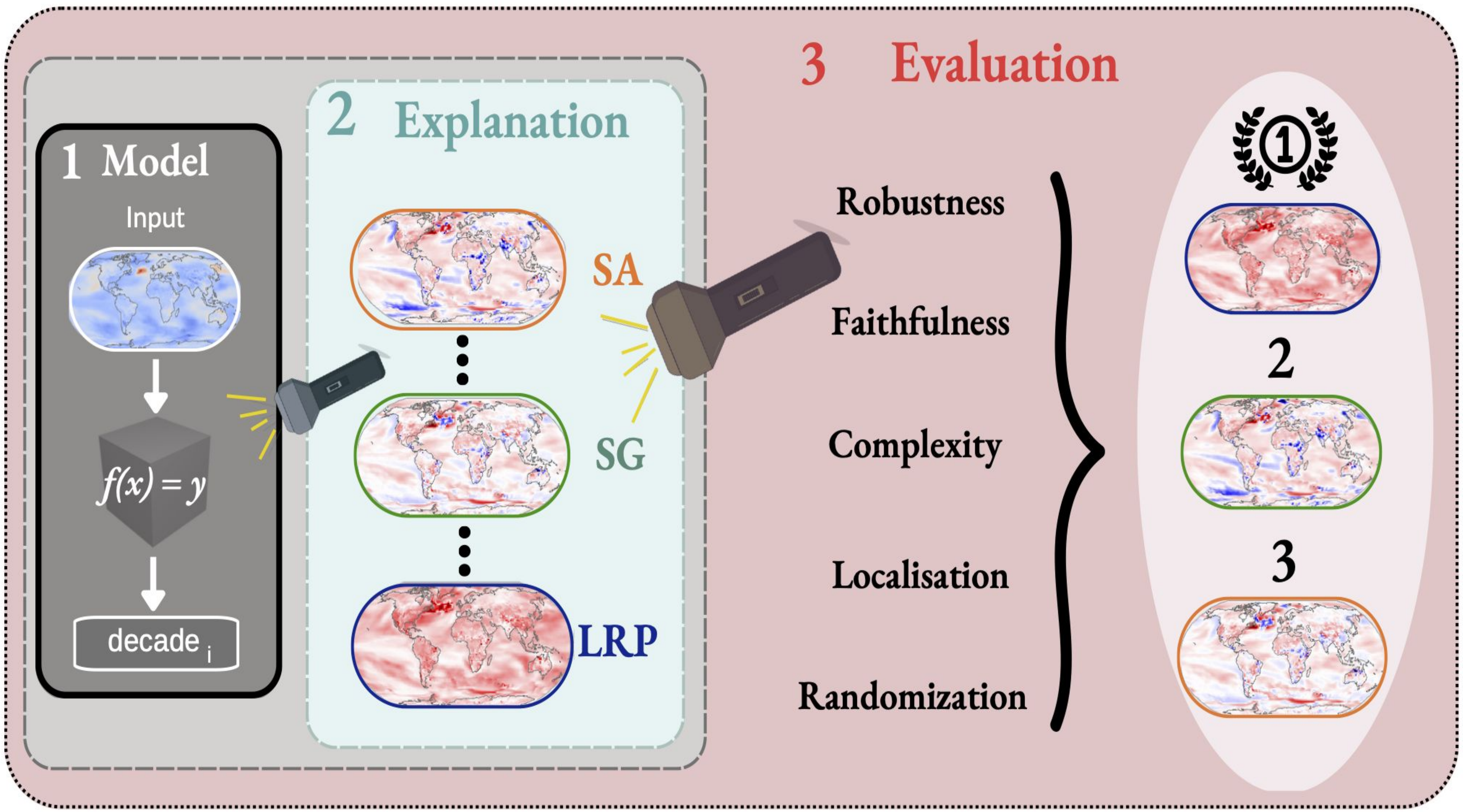
1. Introduction

The Challenge of XAI Method Selection

- ❖ Increasing number of methods with often no ground-truth
  - ➢ No performance measure
  - ➢ Choice by popularity or easy access (Krishna et. al. (2022))
- ❖ Since different explanations for the same network decision lead to different conclusions, trust and reliability becomes an issue

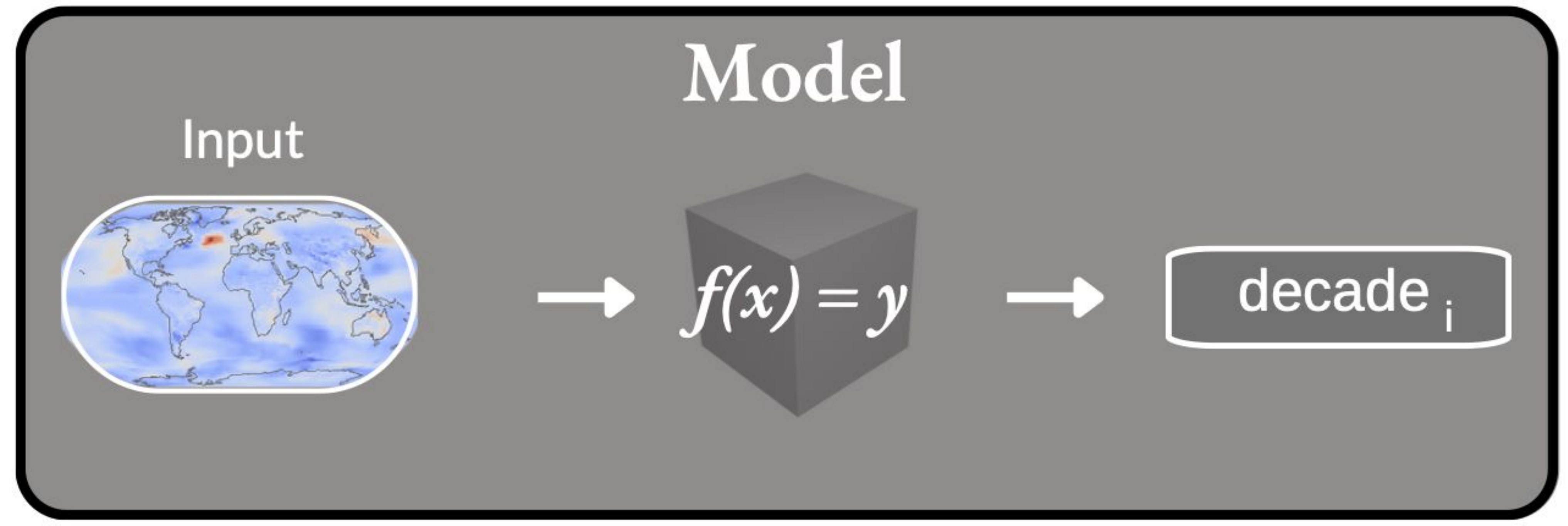
Bommer et. al., 2023

2. Overview



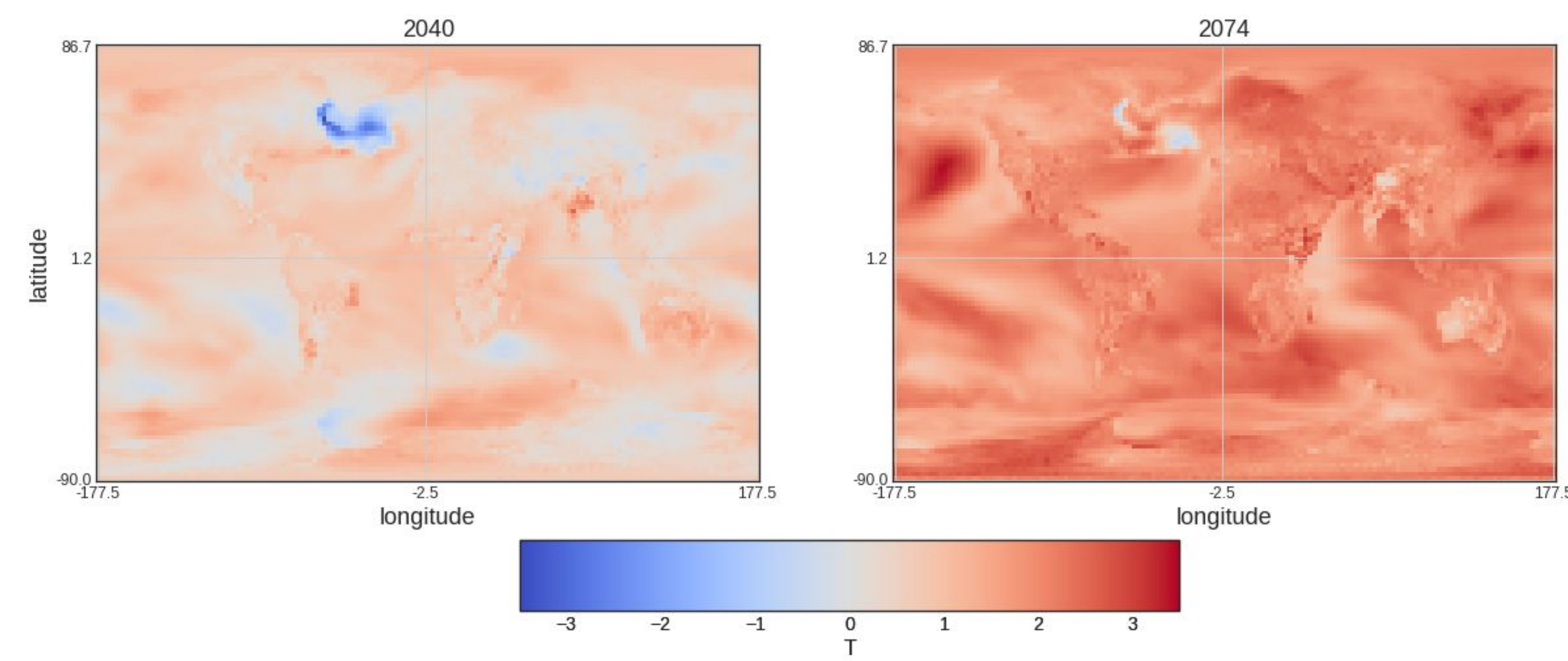
Schematic of the XAI evaluation procedure (Bommer et. al., 2023)

3. Preliminaries - Setup



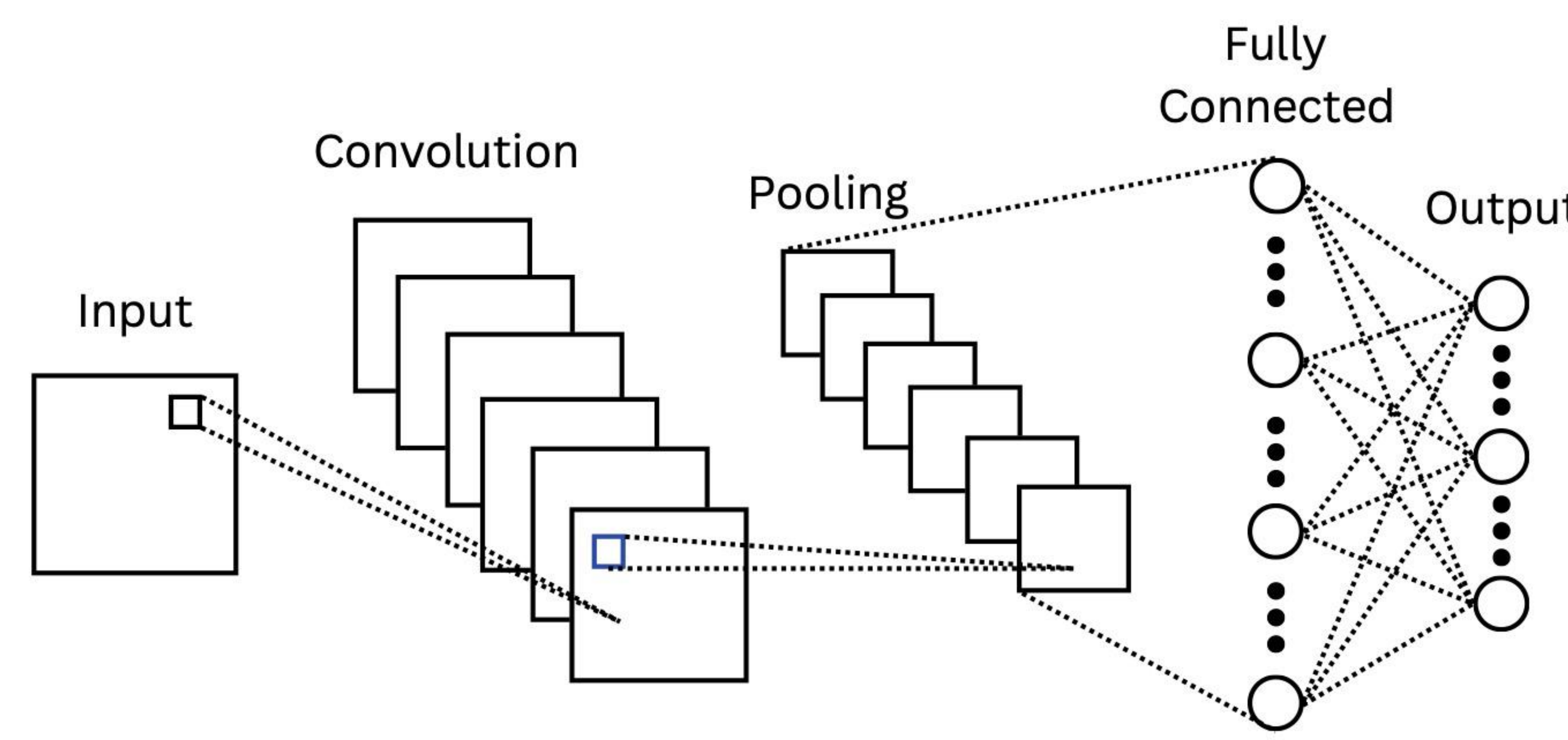
Task

- **Classification** of annual temperature maps based on their decade (Bommer et. al., 2023)



Data

- Model data from CESM 1 (Hurrell et al. (2013))
- Standardized, annual, 2-m air temperature (T2m) temperature maps from 1920-2080



Network

- Convolutional neural network (CNN)
- 20 classes from 1900-2100 (see also Labe and Barnes, (2021))

Loading

- Batch of preprocessed data and pre-trained network as kaggle dataset

3. Preliminaries - Explanations

Generate XAI Methods

- ❖ Six *local* explanation methods from *'tf_explain'* package
  - ➢ VanillaGradients, SmoothGrad, GradientsInput, Integrated Gradients, Occlusion Sensitivity, GradCam

```python
# Generate several explanation methods with Quantus.
xai_methods ={"VanillaGradients": {},
              "IntegratedGradients": {},
              "SmoothGrad": {},
              "GradientsInput": {},
              "OcclusionSensitivity": {"window": (1, 5, 6)},
              "GradCAM": {"gc_layer": "conv2d", "shape": (1, 95, 144)}
              }

explanations = {}
for method, kwargs in xai_methods.items():
    a_batch = quantus.explain(model=model,
                              inputs=x_batch[samples,:,:,:],
                              targets=y_batch[samples],
                              **{**{"method": method}, **kwargs})
```

ICLR International Conference On Learning Representations

4. XAI Evaluation - Properties

Measure Explanation Quality

- ➢ **Faithfulness** (↑) quantifies to what extent explanations follow the predictive behaviour of the model, asserting that more important features affect model decisions more strongly e.g., (Bach et al., 2015; Dasgupta et al., 2022).
- ➢ **Robustness** (↓) measures to what extent explanations are stable/ similar when subjected to slight input perturbations, assuming an approximately constant model output e.g., (Alvarez-Melis et al., 2018; Yeh et al., 2019).
- ➢ **Randomisation** (↓) tests to what extent explanations deteriorate as labels or model parameters gets randomised e.g., (Adebayo et. al., 2018); Sixt et al., 2020).
- ➢ **Localisation** (↑) tests if the explainable evidence is centred around a region of interest, e.g., defined through a bounding box, a segmentation mask or a cell within a grid e.g., (Zhang et al., 2018; Arras et al., 2021).
- ➢ **Complexity** (↓) captures to what extent explanations are concise, i.e., that few features are used to explain a model prediction e.g., (Chalasani et al., 2020; Bhatt et al., 2020).

The Meta-Evaluation Problem in Explainable AI: Identifying Reliable Estimators with MetaQuantus (Hedström et al., 2023b)

4. XAI Evaluation - Quantus

Goals & Applications

- Quantus is an XAI toolkit for responsible evaluation of neural network explanations, for ML practitioners
- Quantus has been used for various healthcare applications [1,2,3,4], XAI optimisation [5], climate science [6, 7, 8]

Library Content

- Providing 30+ metrics in 6 categories for XAI evaluation with tutorials and API reference
- Supporting different data types (image, time-series, tabular/ NLP) and ML frameworks models (PyTorch and Tensorflow)
- Additional built-in XAI methods support

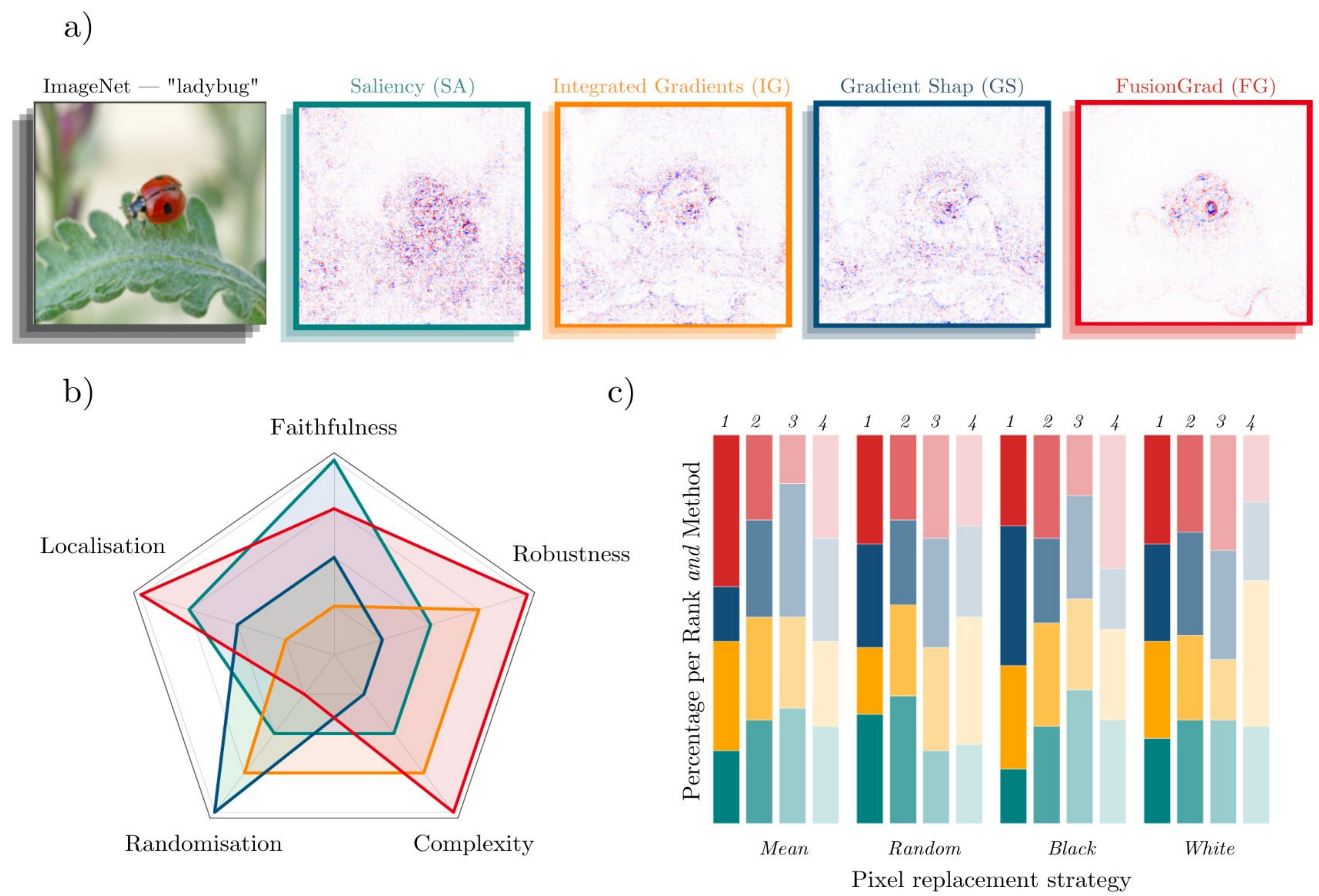


Figure: a) Simple qualitative comparison of XAI methods is often not sufficient to distinguish which gradient-based method — Saliency, Integrated Gradients, GradientShap or FusionGrad is preferred. With Quantus, we can obtain richer insights on how the methods compare b) by holistic quantification on several evaluation criteria and c) by providing sensitivity analysis of how a single parameter, e.g., pixel replacement strategy of a faithfulness test influences the ranking of XAI methods.

4. XAI Evaluation - How to run

Code Snippet

❖ Evaluate XAI methods in a one-liner or compute scores with **quantus.evaluate()**

```
[ ]
import quantus

# Alternative 1. Evaluate the Gradient explanations in a one-liner - by calling the intialised metric!
# For evaluation, a model, input, labels and explanations are needed, using the same samples as loaded before.
quantus.Sparseness()(model=model,
                     x_batch=x_batch_samples,
                     y_batch=y_batch_samples,
                     a_batch=explanations["VanillaGradients"])
```

```
Warnings and information:
 (1) The Sparseness metric is likely to be sensitive to the choice of normalising 'normalise' (and 'normalise_func') and if taking absolute v
 (2) If attributions are normalised or their absolute values are taken it may destroy or skew information in the explanation and as a result,
 (3) Make sure to validate the choices for hyperparameters of the metric (by calling .get_params of the metric instance).
 (4) For further information, see original publication: Chalasani, Prasad, et al. Concise explanations of neural networks using adversarial t
 (5) To disable these warnings set 'disable_warnings' = True when initialising the metric.

[0.6679884922790988,
 0.4523420148821614,
 0.5321293354048773,
 0.40148710064506854]
```

Learn more: Paper at JMLR V24, Code at Github and API documentation

ICLR International Conference On Learning Representations

9

5) XAI Method Selection - 1/4

1. Choose evaluation properties for the task

```python
# Initialise the Quantus evaluation metrics.
metrics = {
    "Robustness": quantus.AvgSensitivity(
        nr_samples=2,
        lower_bound=0.2,
        norm_numerator=quantus.norm_func.fro_norm,
        norm_denominator=quantus.norm_func.fro_norm,
        perturb_func=quantus.perturb_func.uniform_noise,
        similarity_func=quantus.similarity_func.difference,
        abs=True,
        normalise=False,
        aggregate_func=np.mean,
        return_aggregate=True,
        disable_warnings=True,
    ),
    "Faithfulness": quantus.FaithfulnessCorrelation(
        nr_runs=10,
        subset_size=224,
        perturb_baseline="black",
        perturb_func=quantus.baseline_replacement_by_indices,
        similarity_func=quantus.similarity_func.correlation_pearson,
        abs=True,
        normalise=False,
        aggregate_func=np.mean,
        return_aggregate=True,
        disable_warnings=True,
    ),
```

```python
    "Localisation": quantus.RelevanceRankAccuracy(
        abs=True,
        normalise=False,
        aggregate_func=np.mean,
        return_aggregate=True,
        disable_warnings=True,
    ),
    "Complexity": quantus.Sparseness(
        abs=True,
        normalise=False,
        aggregate_func=np.mean,
        return_aggregate=True,
        disable_warnings=True,
    ),
    "Randomisation": quantus.ModelParameterRandomisation(
        layer_order="independent",
        similarity_func=quantus.ssim,
        return_sample_correlation=True,
        abs=True,
        normalise=False,
        aggregate_func=np.mean,
        return_aggregate=True,
        disable_warnings=True,
    ),
}
```

5) XAI Method Selection - 2/4

1. Choose evaluation properties for the task
2. **Calculate scores for all methods and each property**

	Robustness	Faithfulness	Localisation	Complexity	Randomisation
VanillaGradients	0.404582	0.025221	0.034769	0.678957	0.255716
IntegratedGradients	0.449772	0.035522	0.029630	0.380961	0.053904
SmoothGrad	0.383887	0.012234	0.040590	0.692202	0.082834
GradientsInput	0.381192	0.056342	0.038549	0.663006	0.295150
OcclusionSensitivity	0.166494	0.002010	0.026909	0.164859	0.711957
GradCAM	0.168093	0.010495	0.014361	0.150166	0.999771

5) XAI Method Selection - 3/4

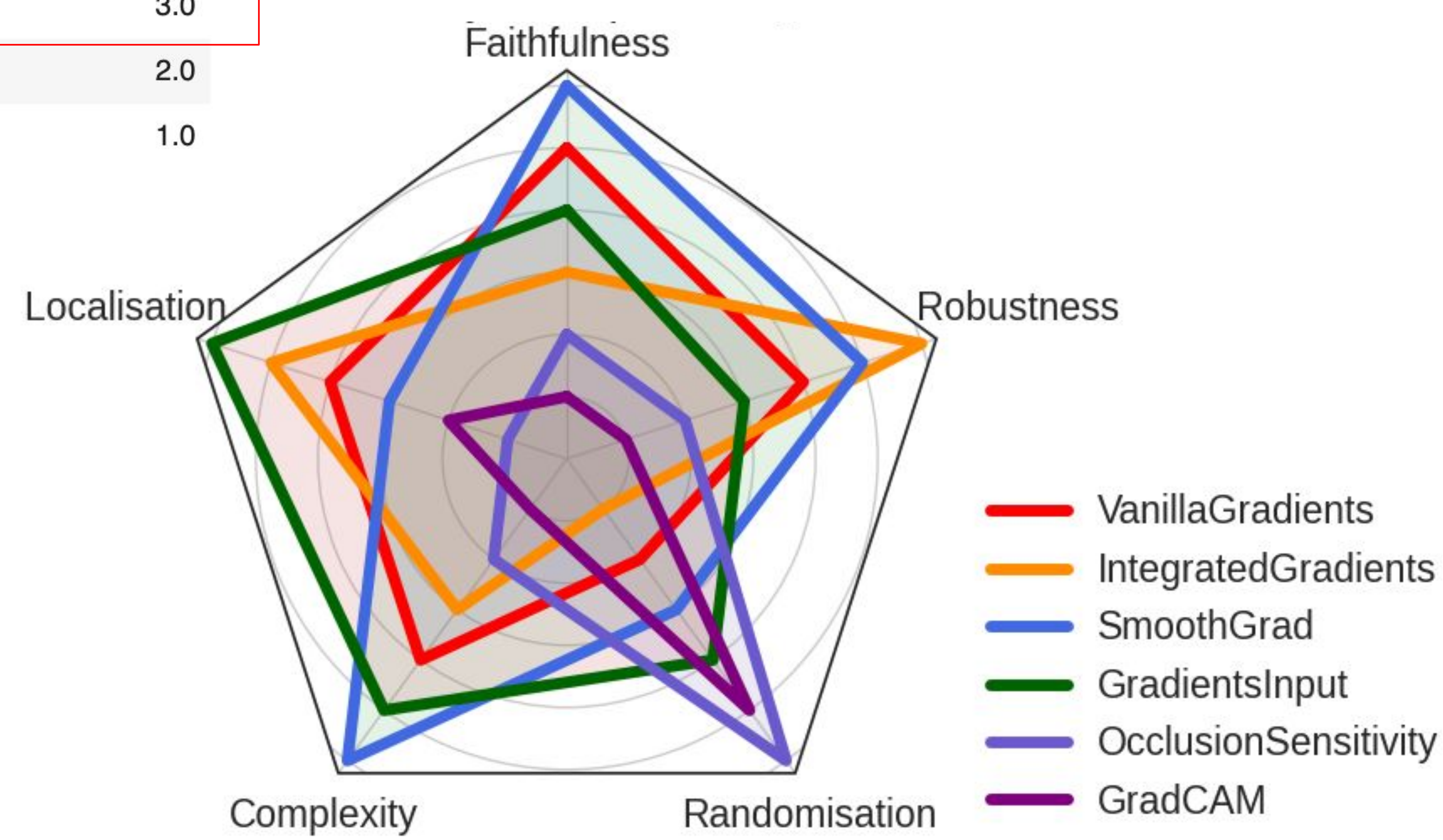
1. Choose evaluation properties for the task
2. Calculate scores for all methods and each property
3. **Rank explanation methods**

	Complexity	Faithfulness	Localisation	Robustness	Randomisation
VanillaGradients	5.0	4.0	4.0	2.0	4.0
IntegratedGradients	3.0	5.0	3.0	1.0	6.0
SmoothGrad	6.0	3.0	6.0	3.0	5.0
GradientsInput	4.0	6.0	5.0	4.0	3.0
OcclusionSensitivity	2.0	1.0	2.0	6.0	2.0
GradCAM	1.0	2.0	1.0	5.0	1.0

5) XAI Method Selection - 4/4

1. Choose evaluation properties for the task
2. Calculate scores for all methods and each property
3. Rank explanation methods
4. **Choose best ranked explanation method**

	Complexity	Faithfulness	Localisation	Robustness	Randomisation
VanillaGradients	5.0	4.0	4.0	2.0	4.0
IntegratedGradients	3.0	5.0	3.0	1.0	6.0
SmoothGrad	6.0	3.0	6.0	3.0	5.0
GradientsInput	4.0	6.0	5.0	4.0	3.0
OcclusionSensitivity	2.0	1.0	2.0	6.0	2.0
GradCAM	1.0	2.0	1.0	5.0	1.0



ICLR International Conference On Learning Representations

Resources

Github

- https://github.com/understandable-machine-intelligence-lab/Quantus
- https://github.com/philine-bommer/Climate_X_Quantus

References

- Labe and Barnes (2021)
  https://agupubs.onlinelibrary.wiley.co<m/doi/full/10.1029/2021MS002464
- Bommer et al. (2023), https://arxiv.org/abs/2303.00652
- Hedström et al (2023a) https://jmlr.org/papers/v24/22-0142.html
- Hedström et al (2023b) https://arxiv.org/abs/2302.07265

understandable-machine-intelligence-lab/**Quantus**

Quantus is an eXplainable AI toolkit for responsible evaluation of neural network explanations

14 Contributors · 8 Used by · 3 Discussions · 333 Stars · 52 Forks

understandable-machine-intelligence-lab/Quantus: Quantus is an eXplainable AI toolkit for responsible evaluation of neural network...

Quantus is an eXplainable AI toolkit for responsible evaluation of neural network explanations - GitHub - understandable-machine-intelligence-lab/Quantus: Quantus is an eXplainable AI toolkit for r...

GitHub

philine-bommer/
Climate_X_Quantus

This repository contains the code and packages to reproduce all experiments in "Finding the right XAI Method — A Guide...

1 Contributor · 0 Issues · 0 Stars · 0 Forks

philine-bommer/Climate_X_Quantus: This repository contains the code and packages to reproduce all experiments in "Finding the right XAI...

This repository contains the code and packages to reproduce all experiments in "Finding the right XAI Method --- A Guide for the Evaluation and Ranking of Explainable AI Methods in Climate...

GitHub

ICLR
International Conference On Learning Representations