



## Probabilistic post-processing of CAMS radiation services

Hadrien Verbois\*, Vadim Becquet, Yves-Marie Saint-Drenan, Benoit Gschwind, Philippe Blanc, Marion Schroedter Homscheidt

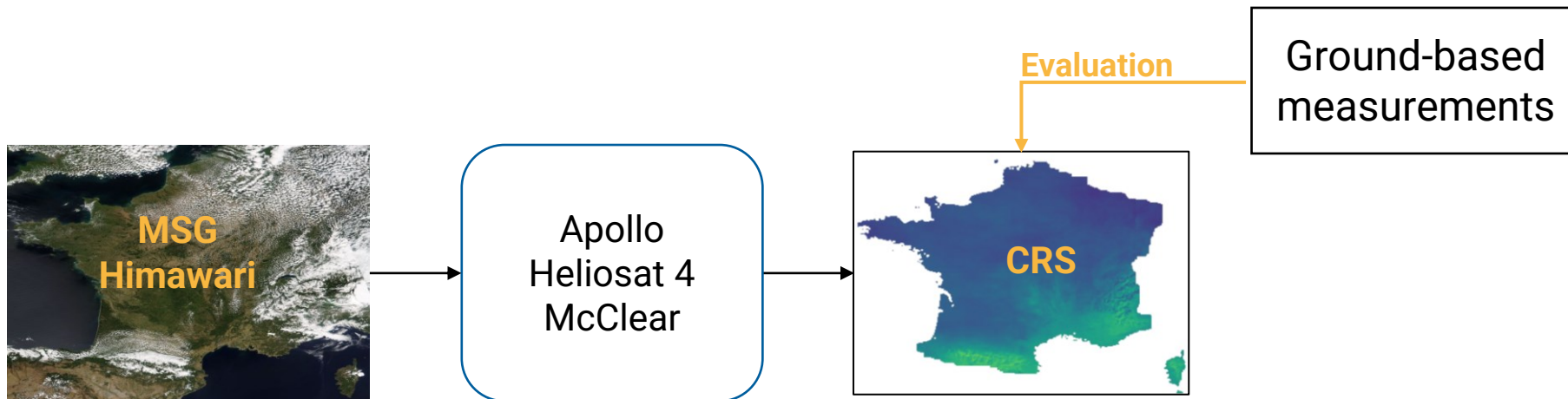


Deutsches Zentrum  
für Luft- und Raumfahrt  
German Aerospace Center

\*supported by HiDa in a guest stay at DLR and by SciDoSol in Mines Paris

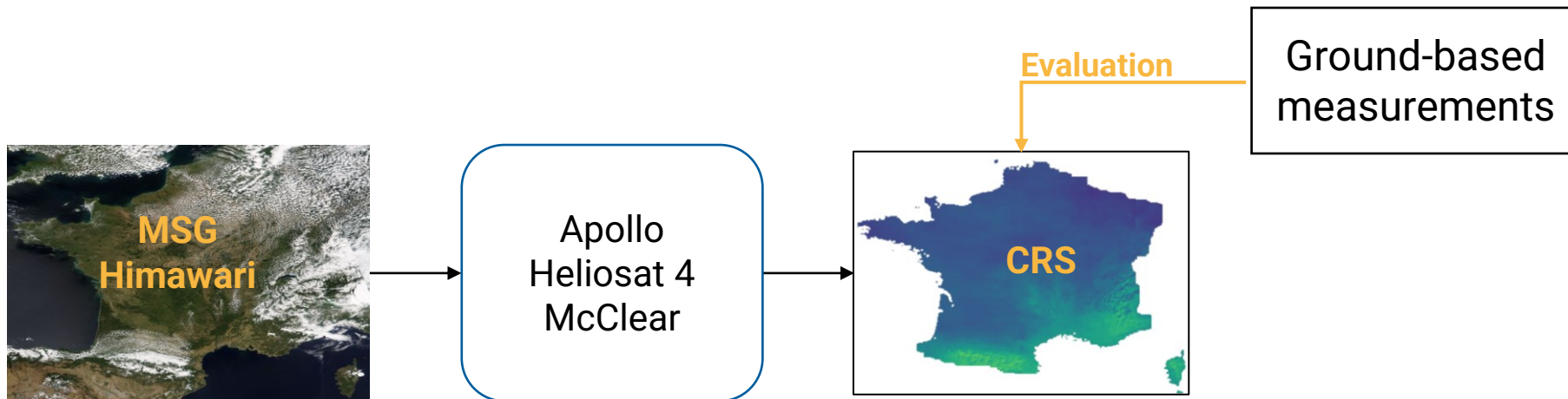
## Satellite retrieval methods

- Satellite-derived estimations of solar surface irradiance (SSI) are critical to many applications in the solar industry
- They are produced by *retrieval methods*, which infer SSI from Satellite image(s)
  - + Copernicus Radiation Service (CRS) is such a retrieval model
  - + It is state-of-the-art, and in *constant* evolution



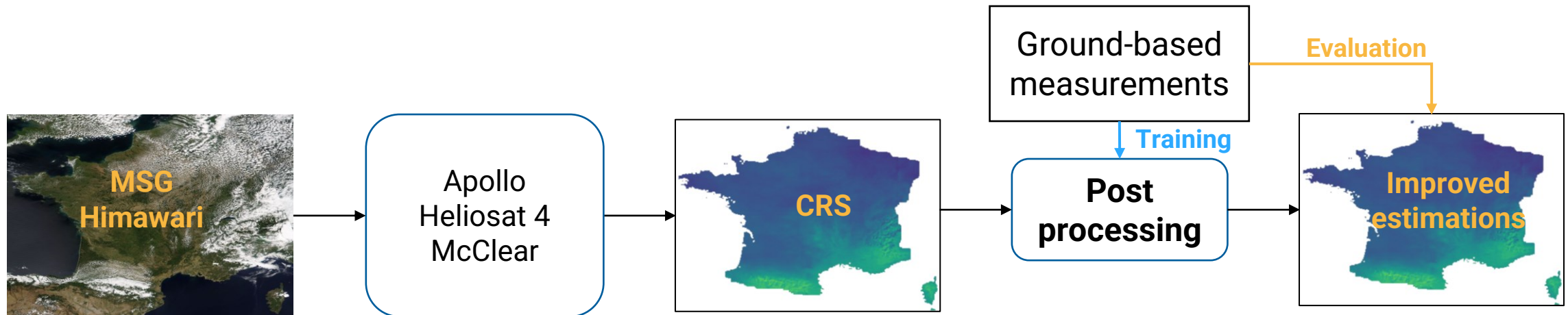
## Post-processing techniques

- Even state-of-the-art retrieval models exhibit random error in some (several) locations



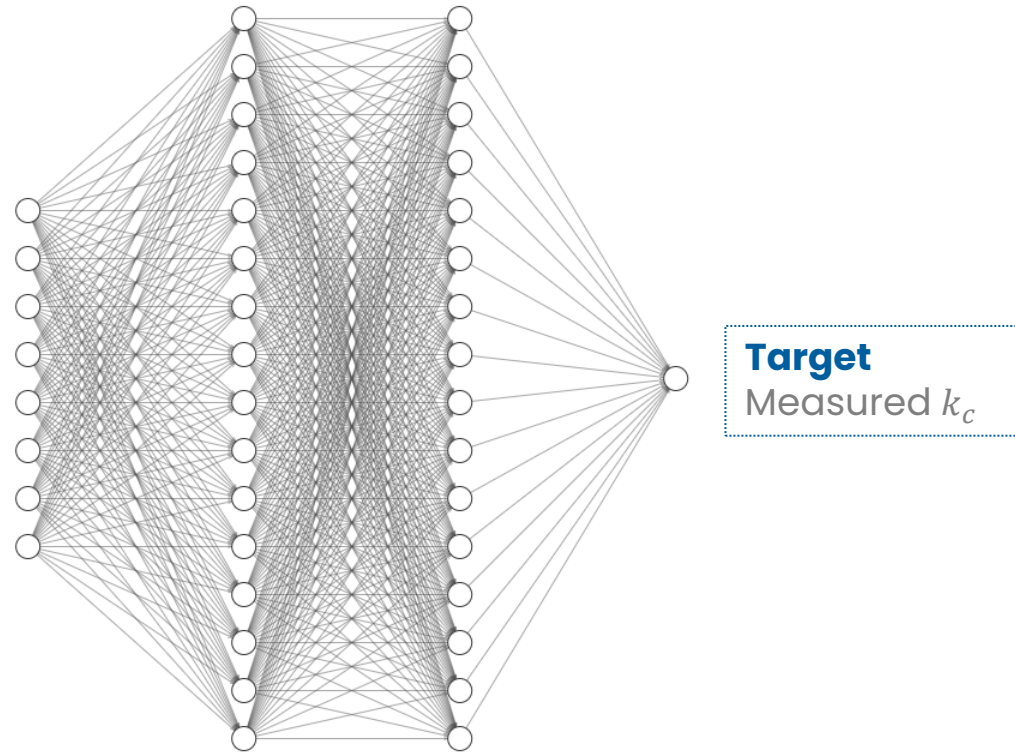
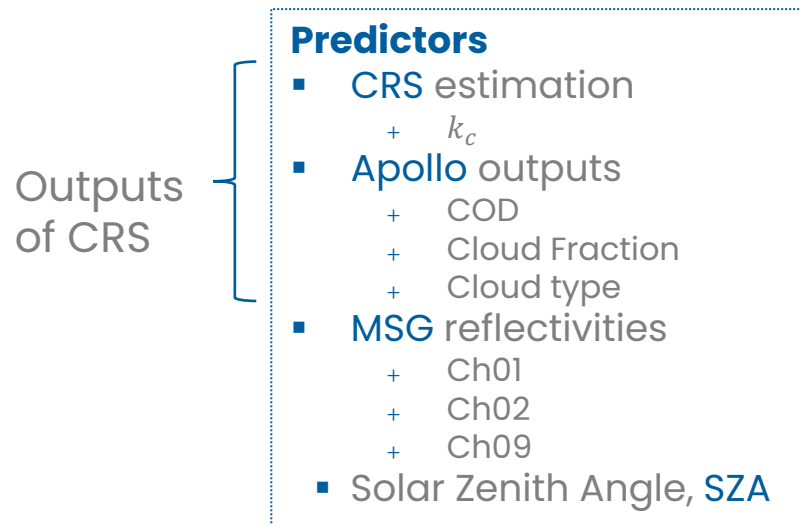
## Post-processing techniques

- Even state-of-the-art retrieval models exhibit random error in some (several) locations
- They can thus be coupled with **post-processing techniques**, which use ground measurements to correct the satellite-based estimations
- In this work, we look into the feasibility of a **global, operational, and probabilistic** post-processing of CRS



### The post-processing algorithm

- We first implemented a deterministic post-processing algorithm for CRS, **ML model**:



#### The network

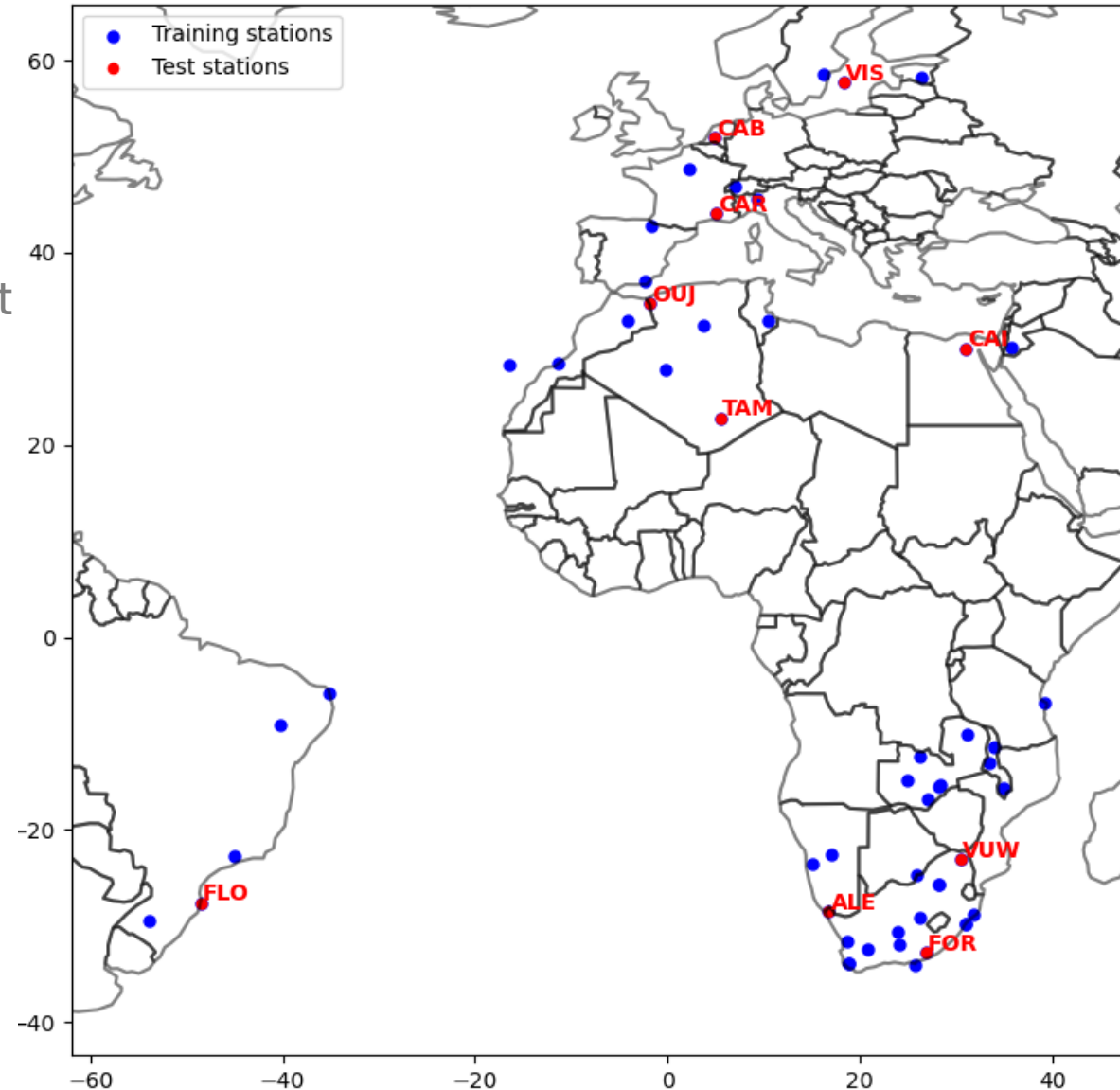
- Cost function: mean squared error (MSE)
- RMSprop algorithm for *gradient descent*
- Early stopping for regularization



## Methods: deterministic post-processing

### The data

- CRS data downloaded from soda
- MSG channels 1, 2, and 9 obtained from Meteosat
- Ground measurements from PVPS:
  - + 55 stations across all MSG field of view
  - + Went through thorough QC
  - + Using libinsitu to access the data
- We enforce **spatiotemporal separation** between training and testing set
  - + 10 stations reserved for testing
  - + 1 year reserved for testing (2018)



## Results: deterministic performances

---

### Metrics

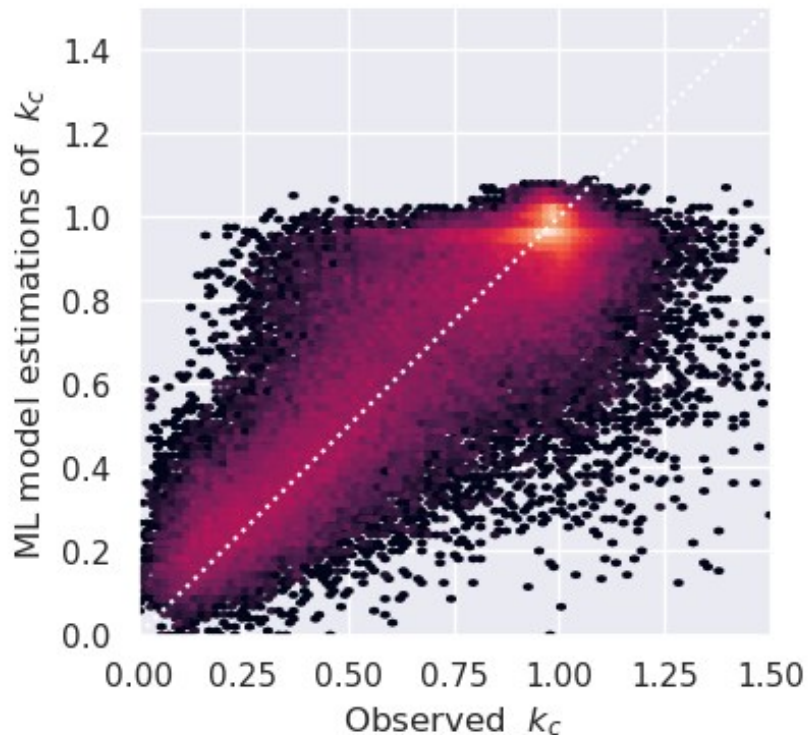
- For the sake of simplicity, we focus here on RMSE:

$$RMSE = \sqrt{\sum_{t=1..n} (I_{ref} - I_{estim})^2}$$

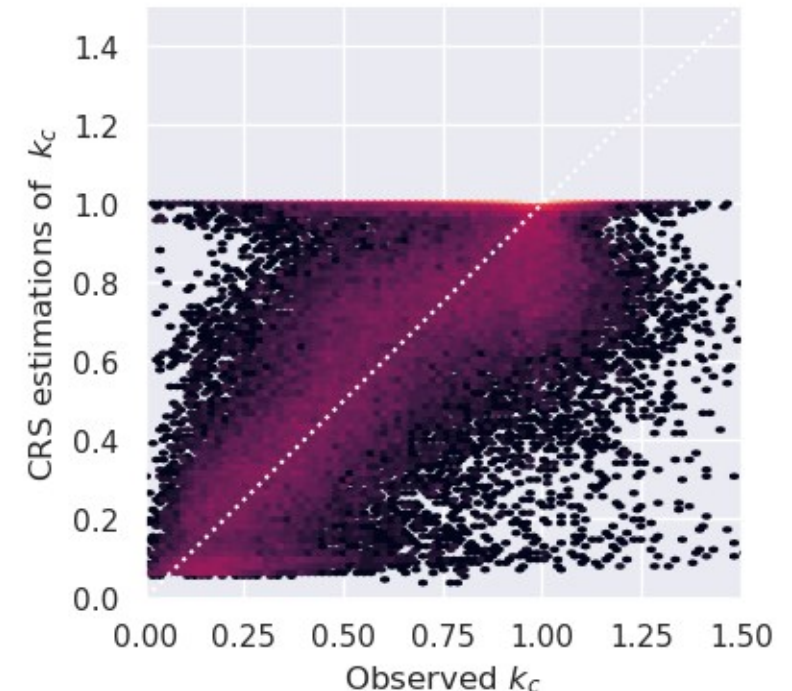
## Results: deterministic performances

### Overall performance of ML-all

- The performances are averaged over the 10 test stations and 1 test year:
  - + Significant reduction of RMSE
  - + Joint distributions of ML model with observations looks *better* than that of CRS with observation



Model	RMSE
ML model	76 W.m <sup>-2</sup>
CRS	88 W.m <sup>-2</sup>

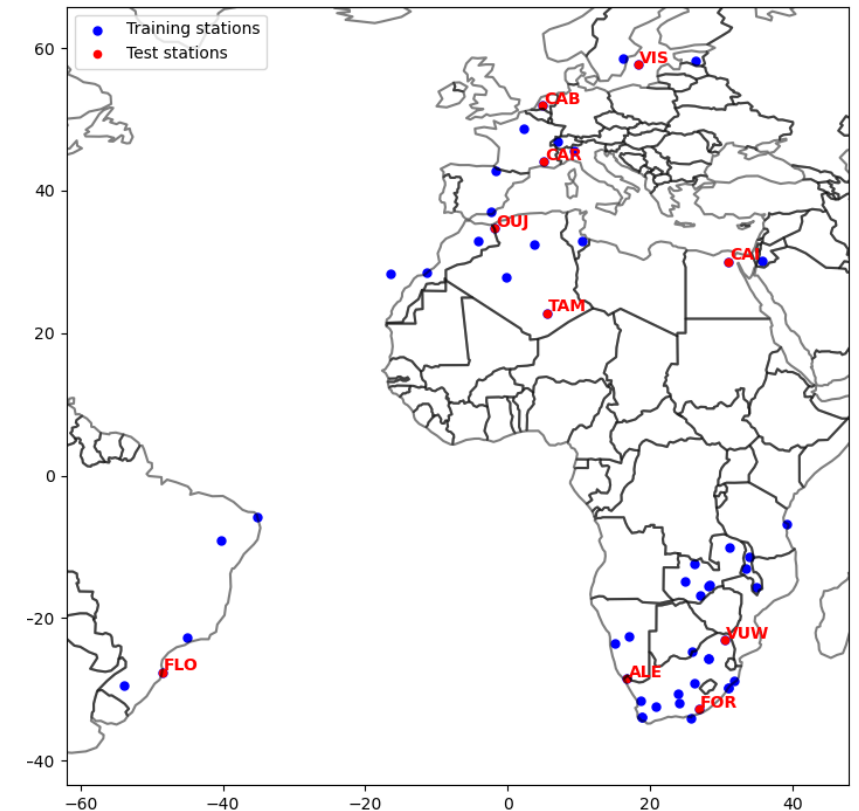
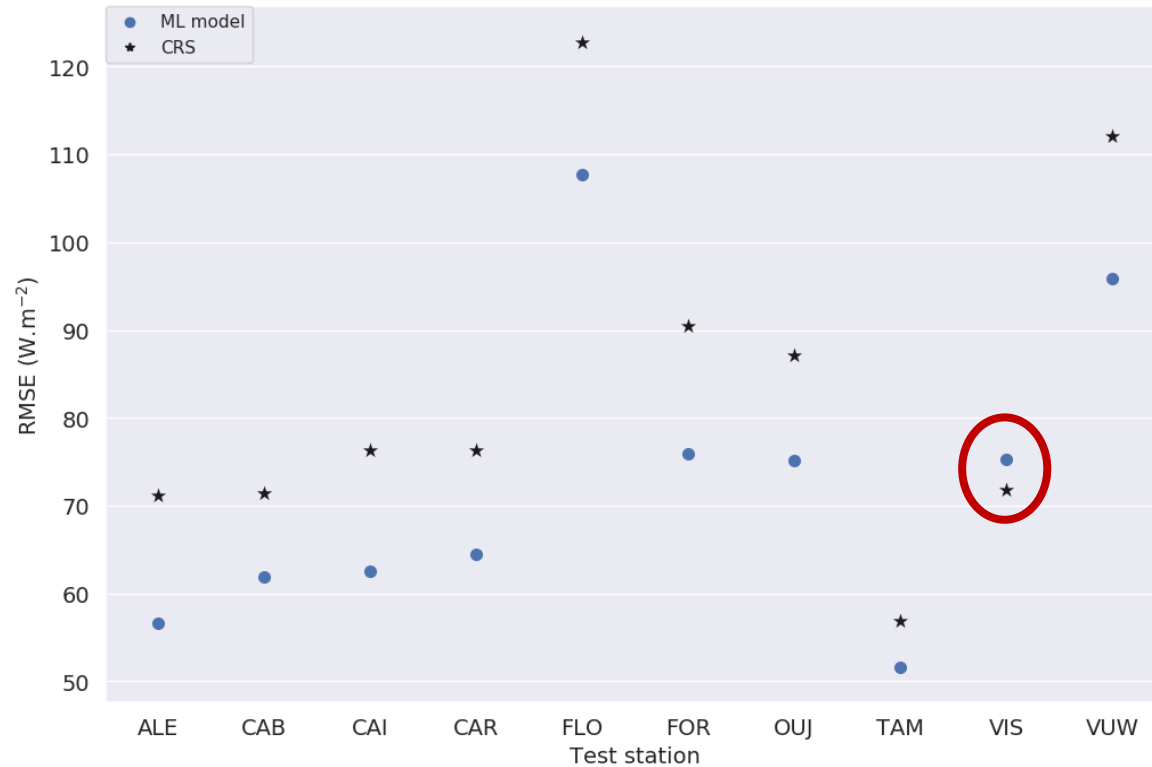




# Results: deterministic performances

## Station-level analysis

- It is insightful to look at station-level improvements
  - + RMSE is reduced every in locations except in Visby (an island in the Baltic sea)



## Results: deterministic performances

---

### Summary

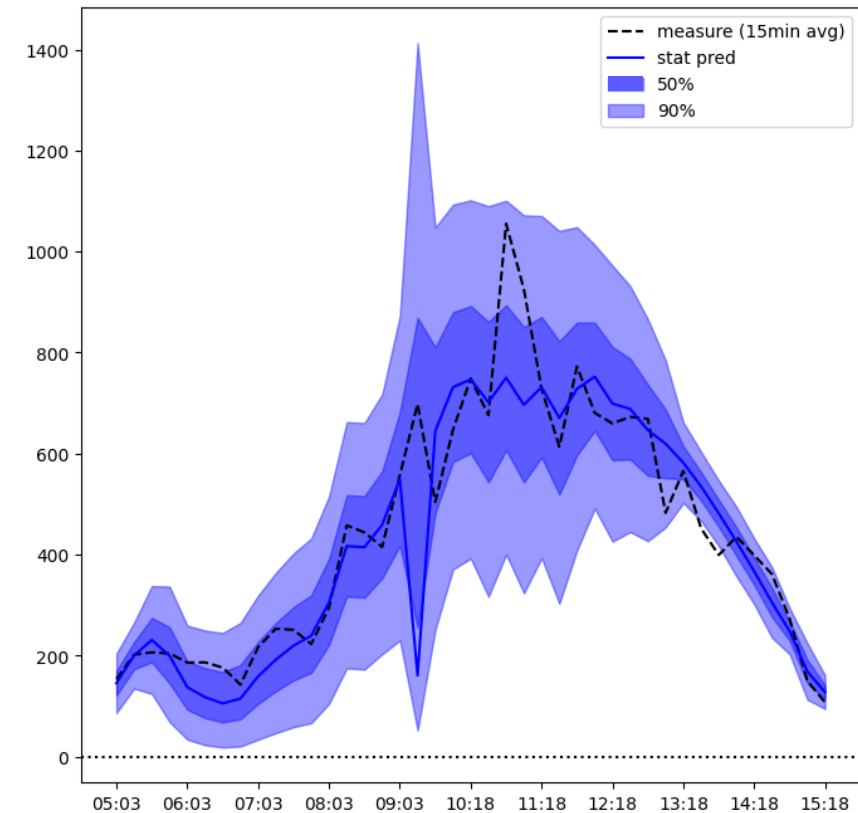
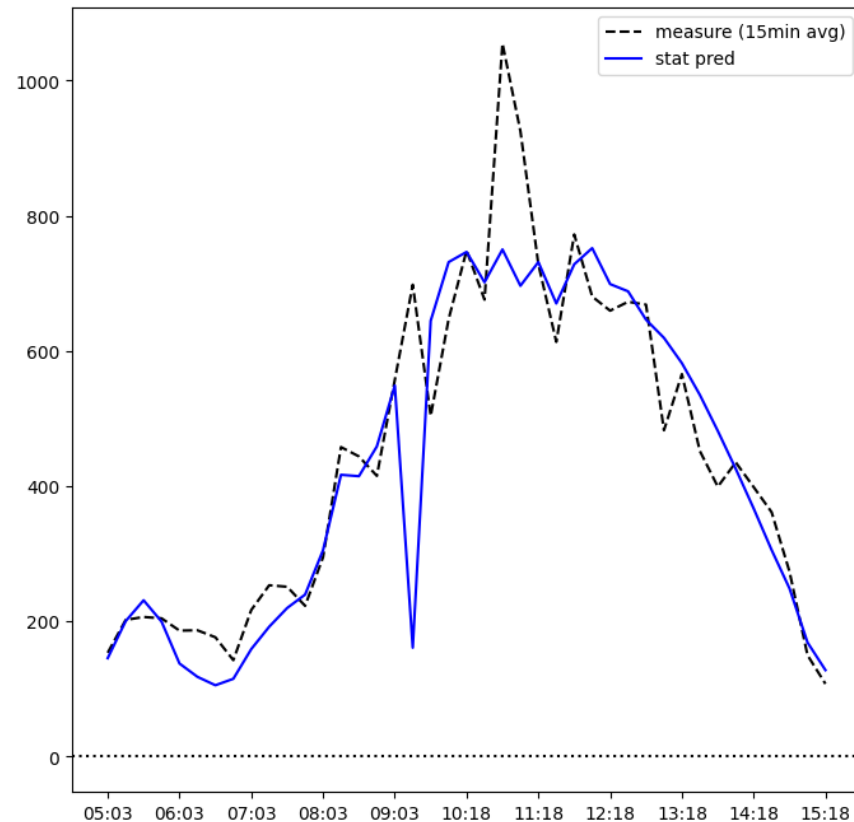
- The post-processing improves CRS RMSE
  - + on average
  - + But not in all test stations

### What can we do?

- Improve the model
- Take a **probabilistic** approach

## Probabilistic estimations

- Instead of providing a single best-guess estimation of SSI, we provide a probability distribution function (PDF) for the SSI
  - + «Provide the best estimate and an estimate of the uncertainty»



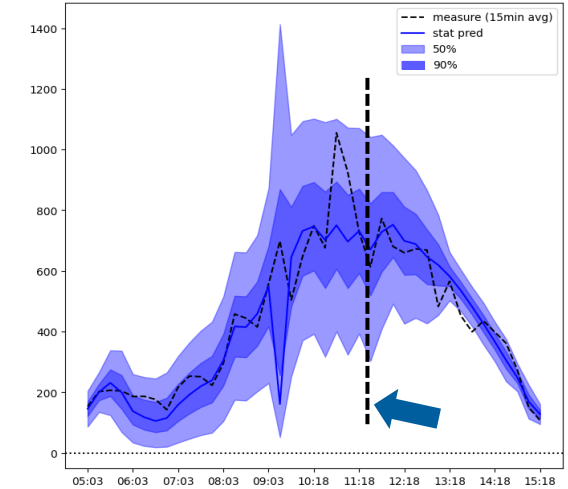
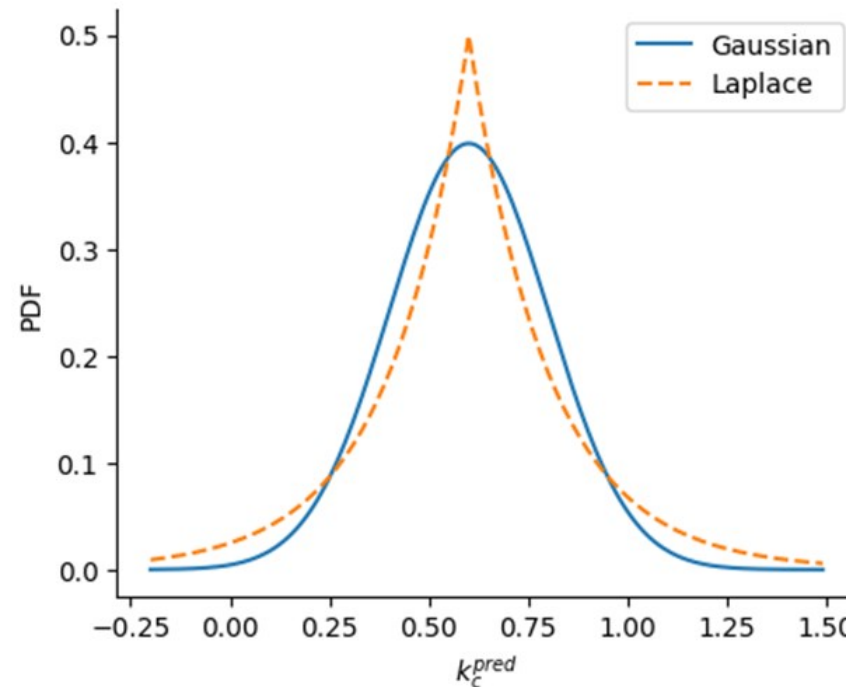
# Methods: probabilistic estimations

## Probabilistic estimations

- In this work, we proposed to use a **parametric framework**
  - + We assume the estimation PDF follows a particular distribution
    - The choice of a distribution for  $p$  is critical: we test Gaussian and Laplace distributions

### + Gaussian

- $P(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
- $E[X] = \mu$
- $var(X) = \sigma^2$
- *Kurtosis* = 0

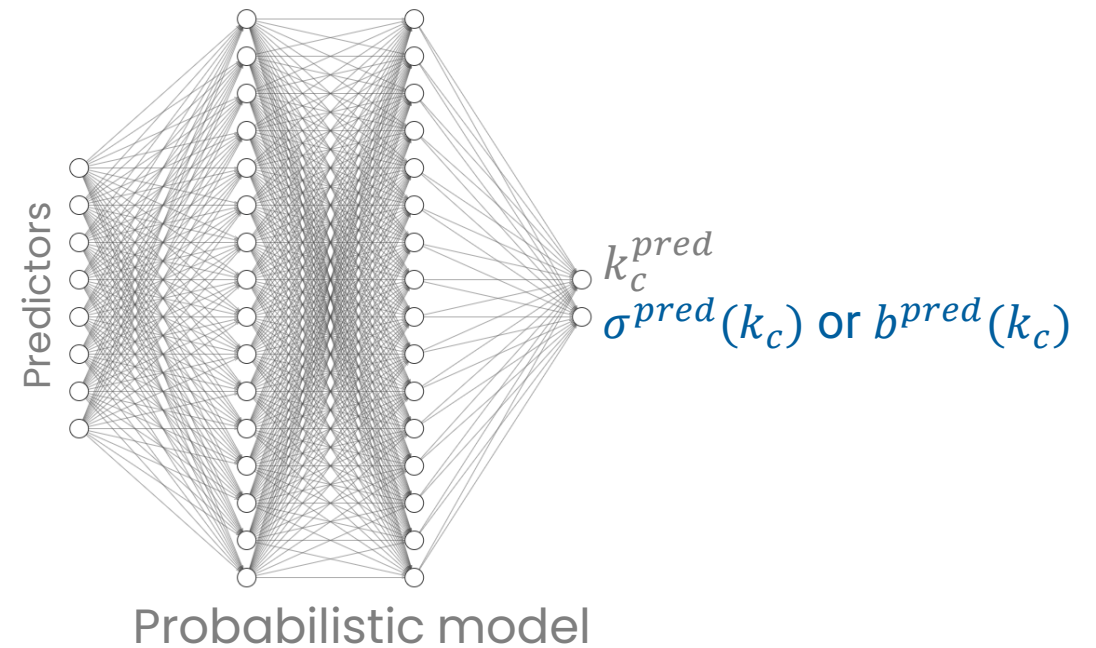
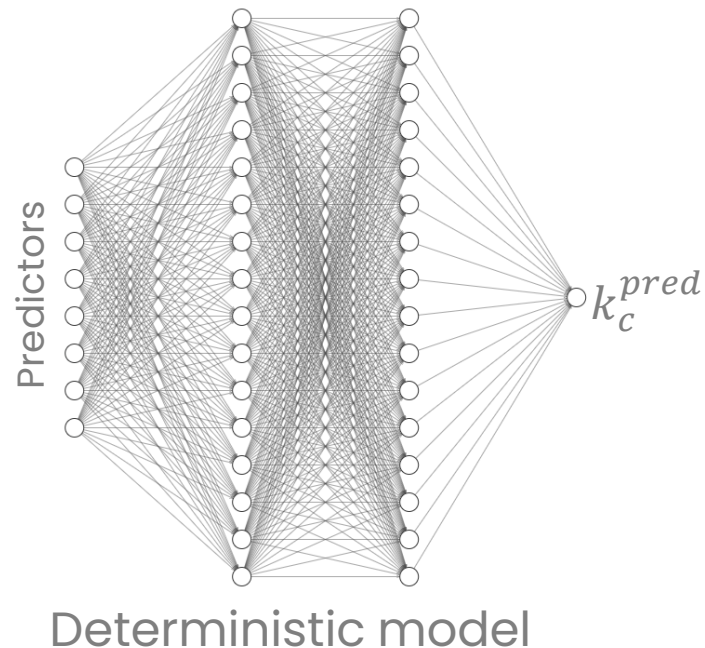


### + Laplace

- $P(x, \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$
- $E[X] = \mu$
- $var(X) = 2b^2$
- *Kurtosis* = 3

### Probabilistic estimations

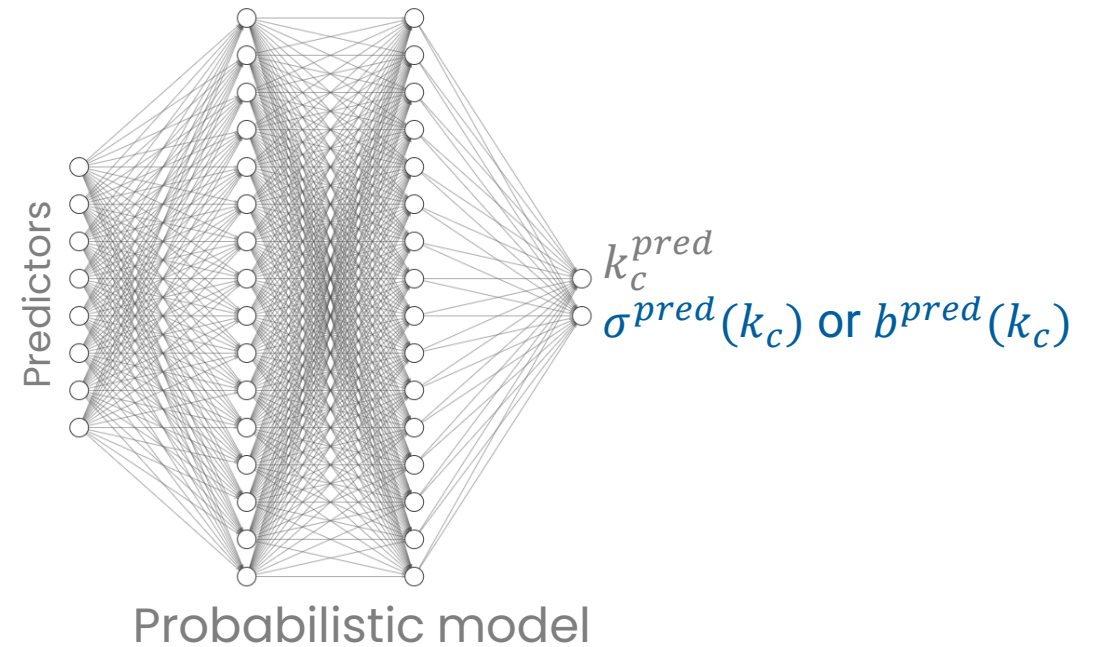
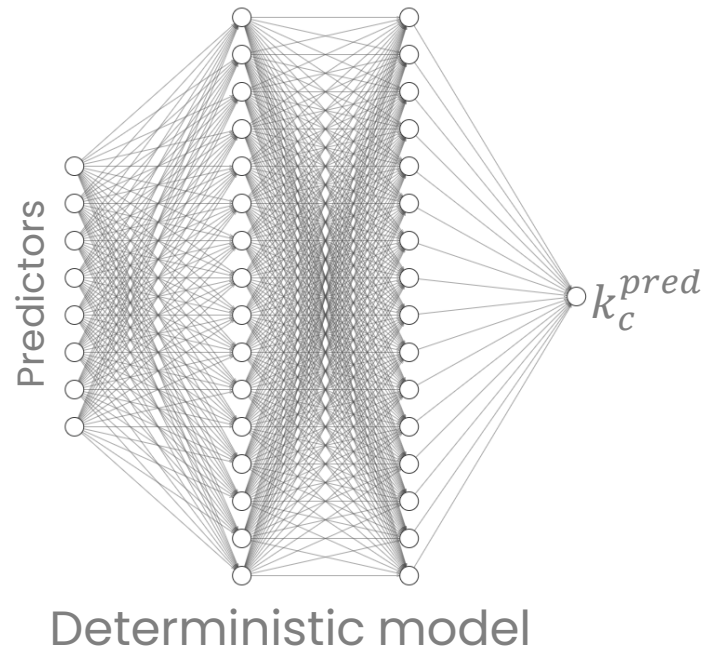
- In this work, we proposed to use a **parametric framework**
  - + We assume the estimation PDF follows a particular distribution
    - The choice of a distribution for  $p$  is critical: we test Gaussian and Laplace distributions
  - + We use a neural network to estimate the parameters of that distribution





### Probabilistic estimations

- In this work, we proposed to use a **parametric framework**
  - + We assume the estimation PDF follows a particular distribution
    - The choice of a distribution for  $p$  is critical: we test Gaussian and Laplace distributions
  - + We use a neural network to estimate the parameters of that distribution
  - + We built the cost function using the **Maximum Likelihood Estimator (MLE)**



## Methods: probabilistic estimations

### The Maximum Likelihood Estimator (MLE)

- Maximizes the probability  $P$  of the observations given  $k_c^{pred}(t, x)$  and  $\sigma^{pred}(t, x)$ :

$$P = p(k_c^{meas}(t, x) | k_c^{pred}(t, x), \sigma^{pred}(t, x) \text{ for } t \text{ in } 1..T \text{ and } x \text{ in } 1..L)$$

- Assuming  $p$  are iid:

$$P = \prod_{\substack{t=1, \\ x=1}}^{T,L} p(k_c^{meas}(t, x) | k_c^{pred}(t, x), \sigma^{pred}(t, x))$$

- Assuming the PDF is Gaussian:

$$P = \prod_{\substack{t=1, \\ x=1}}^{T,L} \frac{1}{\sqrt{2\pi\sigma^{pred}(t, l)^2}} e^{-\frac{(k_c^{meas}(t, l) - k_c^{pred}(t, l))^2}{2\sigma^{pred}(t, l)^2}}$$

- Taking the log and removing constants:

*Log likelihood*

$$= \sum_{\substack{t=1..T \\ x=1..L}} \log(\sigma_{pred}(t, x)) + \frac{(k_c^{meas}(t, x) - k_c^{pred}(t, x))^2}{2\sigma_{pred}^2(t, x)}$$

- Assuming the PDF follows a Laplace distribution:

$$P = \prod_{\substack{t=1, \\ x=1}}^{T,L} \frac{1}{2b^{pred}} e^{-\frac{|k_c^{meas}(t, x) - k_c^{pred}(t, x)|}{b^{pred}}}$$

- Taking the log and removing constants:

*Log likelihood*

$$= \sum_{\substack{t=1..T \\ x=1..L}} \log(b_{pred}(t, x)) + \frac{|k_c^{meas}(t, x) - k_c^{pred}(t, x)|}{b^{pred}(t, x)}$$

## Methods: probabilistic estimations

### The Maximum Likelihood Estimator (MLE)

- Maximizes the probability  $P$  of the observations given  $k_c^{pred}(t, x)$  and  $\sigma^{pred}(t, x)$ :

+ Assuming the PDF is Gaussian:

*Log likelihood*

$$= \sum_{\substack{t=1..T \\ x=1..L}} \log(\sigma_{pred}(t, x)) + \frac{(k_c^{meas}(t, x) - k_c^{pred}(t, x))^2}{2\sigma_{pred}^2(t, x)} \quad \text{MSE}$$

+ Assuming the PDF follows a Laplace distribution:

*Log likelihood*

$$= \sum_{\substack{t=1..T \\ x=1..L}} \log(b_{pred}(t, x)) + \frac{|k_c^{meas}(t, x) - k_c^{pred}(t, x)|}{b_{pred}(t, x)} \quad \text{MAE}$$

- MLE estimates *tweak* the MSE or MAE loss to incorporate the predicted uncertainty  $\sigma^{pred}(t, x)$  or  $b^{pred}(t, x)$

## Methods: probabilistic estimations

### Summary

- We train two probabilistic models:
  - + Same network architecture and predictors as the deterministic model **ML model**
  - + The only difference is the cost function



## Results: probabilistic estimations

---

### Metrics

- Validation of probabilistic estimations is not as straightforward as that of deterministic estimations
  
- It requires to look into three main aspects:
  - + The accuracy
  - + The reliability
  - + The sharpness and resolution



### The accuracy

- This is the same concept as for deterministic estimations
  - + Measured by e.g. RMSE
  - + Should be compared to that of deterministic models

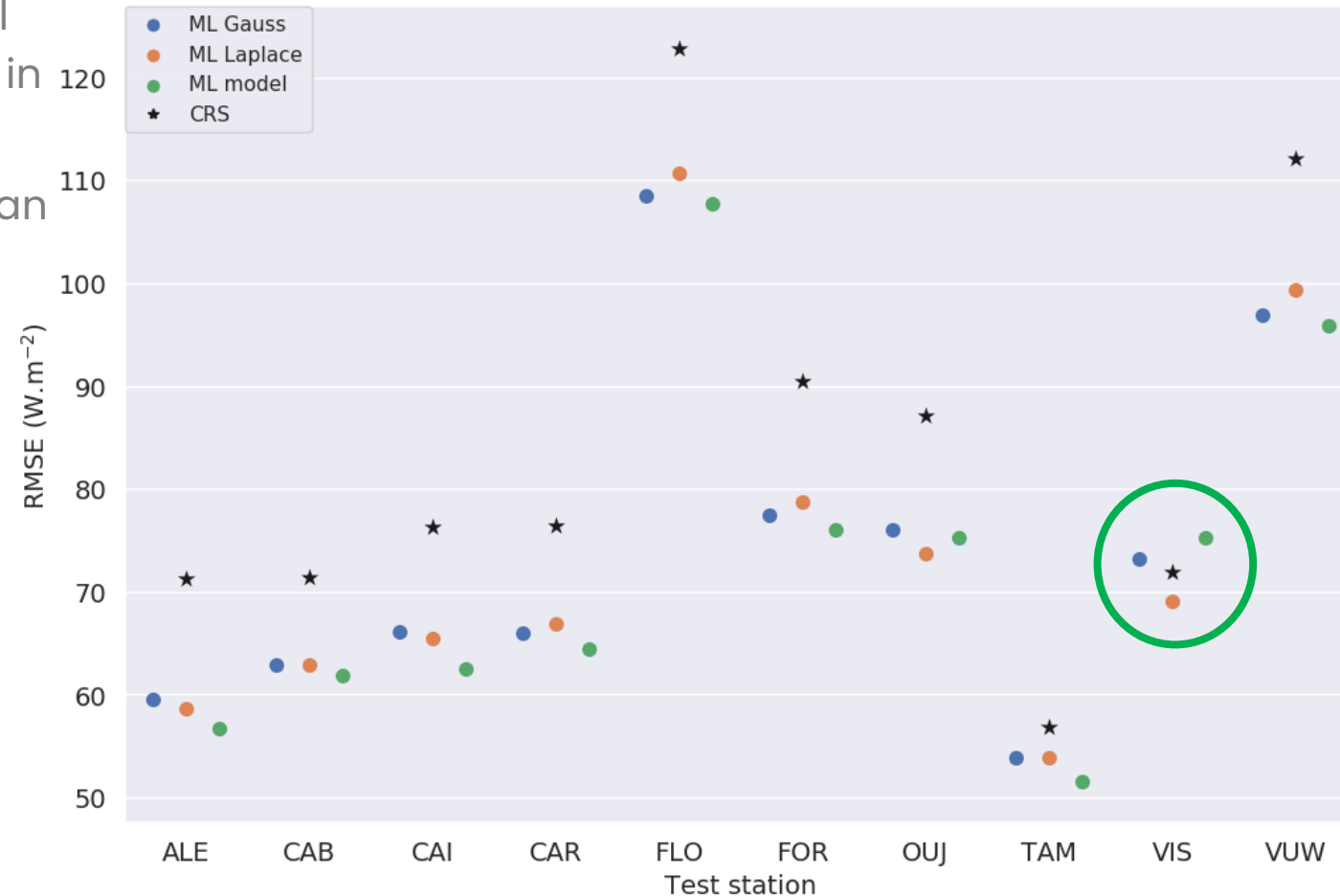
Model	RMSE
CRS	88 W.m <sup>-2</sup>
ML model	76 W.m <sup>-2</sup>
ML Gauss	78 W.m <sup>-2</sup>
ML Laplace	78 W.m <sup>-2</sup>

- The performances ML Gauss and ML Lapalce are similar to those of ML model
  - + Their RMSE is only slightly higher than that of ML model
  - + But still significantly lower than that of CRS

## Results: probabilistic performances

### The accuracy

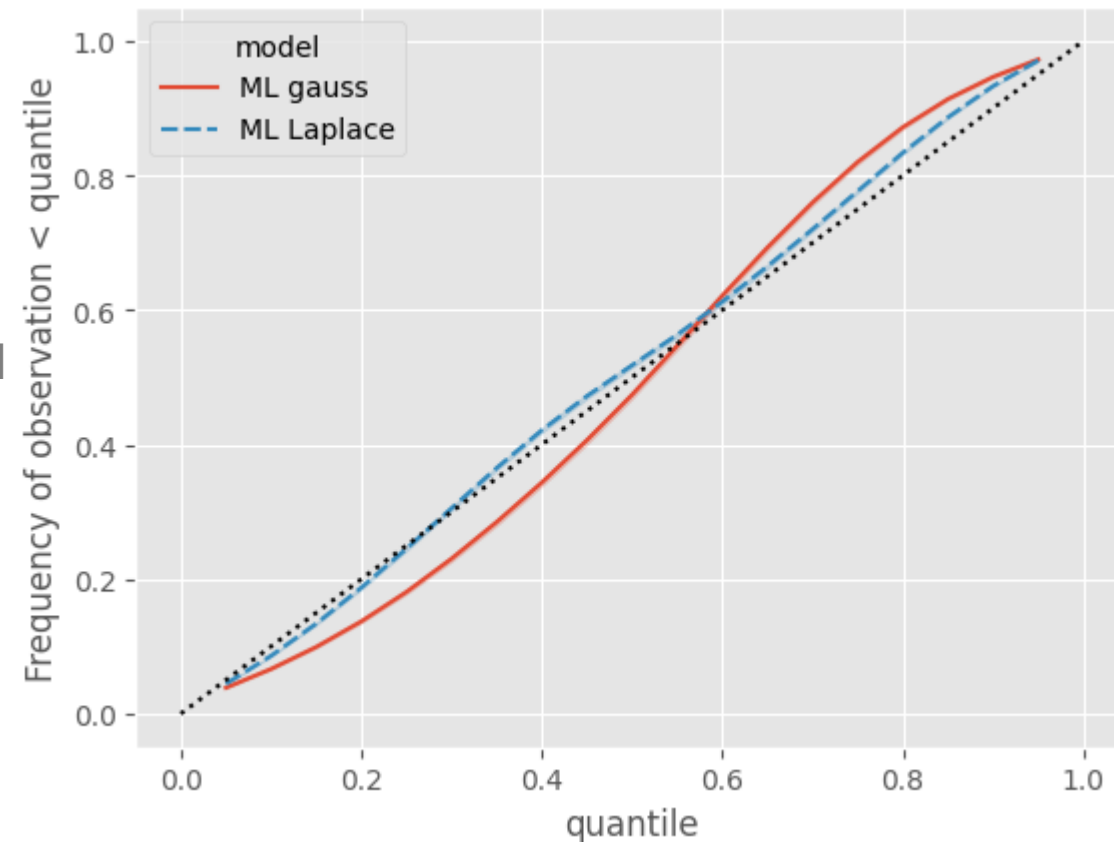
- Can also be assessed at the station level
  - + ML Gauss performs similarly to ML model and outperform CRS everywhere except in VIS
  - + ML Laplace has a slightly higher RMSE than ML model in several station
  - + BUT it outperforms CRS in all stations
- Probabilistic framework may help increase robustness



## Results: probabilistic performances

### Reliability

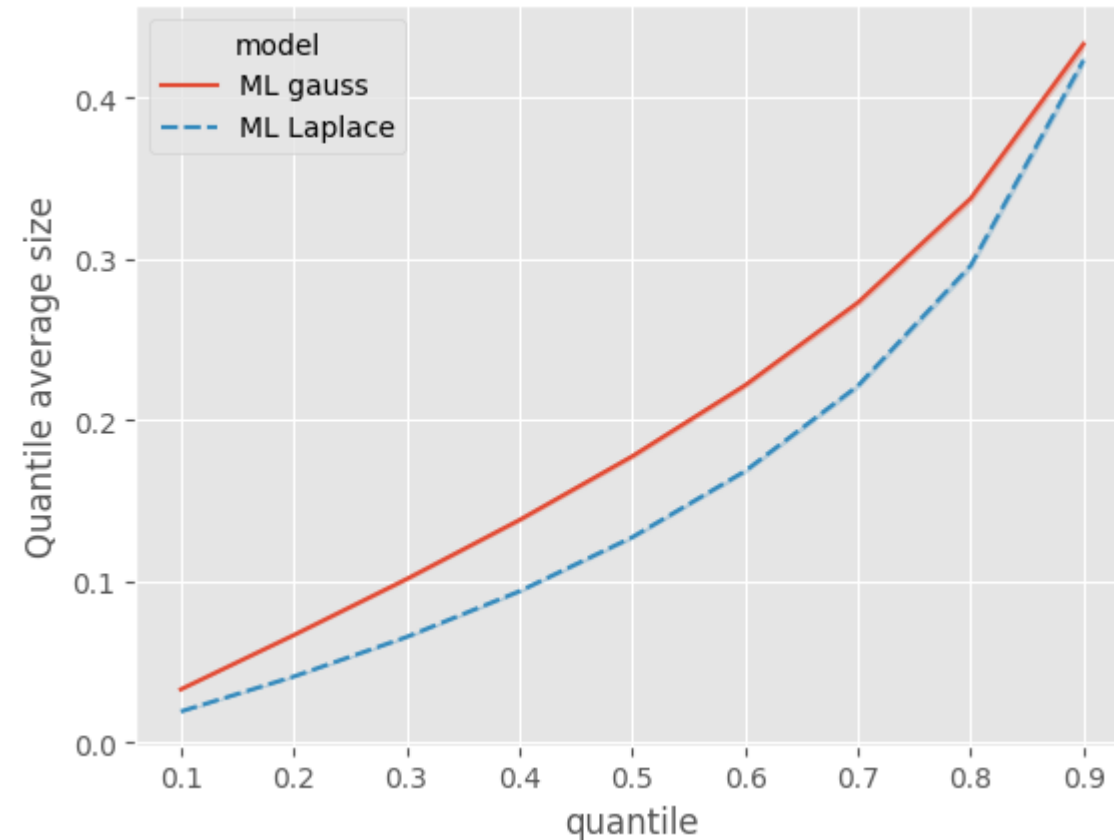
- It measures the statistical consistency between the forecasts and the observations.
  - + In other words, the observations should fall  $q\%$  of the time under the  $q$  quantile
- Gaussian-based models are not well calibrated
  - + Low quantiles are too low, high quantiles too high
  - + Model is underconfident
- Laplace-based models are much better calibrated
  - + Higher kurtosis better fits our usecase



## Results: probabilistic performances

### Sharpness

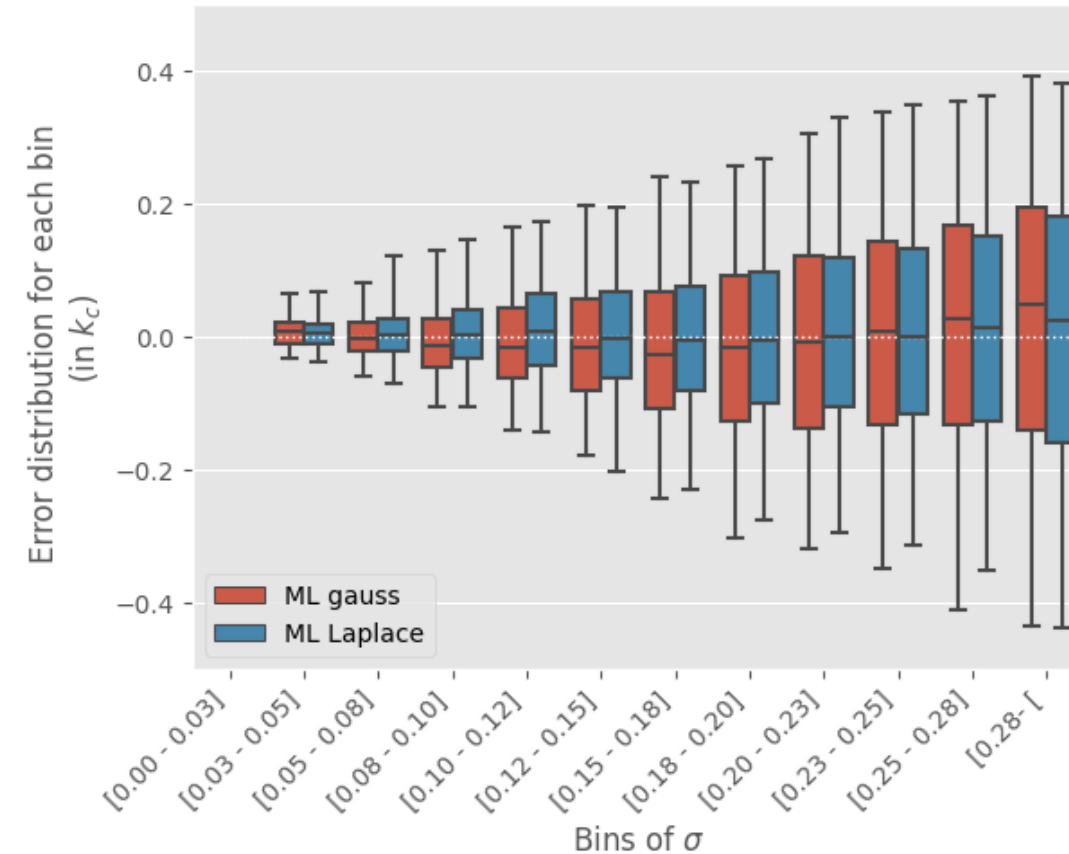
- Sharpness refers to the concentration of the predictive distributions
  - + It evaluates how informative the forecasts are
  - + It can be measured by the average width of the quantiles
- ML Laplace is significantly sharper than ML Gauss



## Results: probabilistic performances

### Resolution

- Resolution measures the capacity of a forecasting model to issue forecasts that are case-dependent.
  - + There is a lack of consensus in the scientific community on how to measure it
  - + And even on how to precisely define it
- The **spread-skill relationship** provides some information about the resolution
  - + We group the data in bins of  $\sigma^{pred}$  and look at the error distribution within each bin/group
- There is a clear spread-skill relationship for both





## Conclusion

---

- Parametric probabilistic models perform as well as deterministic model in terms of RMSE
  - + And *might* even be **more robust**
- ...and are able to **quantify the uncertainty** of the post-processed estimations
  - + Provided the *right* distribution is used
- The parametric framework is flexible and allows further developments
  - + Incorporation of **physical priors**
  - + Estimation of **uncertainty only**
- **Applications** of probabilistic estimations still to be explored
  - + Forecasting (especially when data fusion is involved)
  - + Ramp alerts
  - + TMY
  - + ... ?



Thank you

Questions?



Deutsches Zentrum  
für Luft- und Raumfahrt  
German Aerospace Center