

CATCHMENT TO MODEL SPACE MAPPING

LEARNING TRANSFER FUNCTIONS FROM DATA BY SYMBOLIC REGRESSION

Moritz Feigl¹, Robert Schweppe², Stephan Thober², Mathew Herrnegger¹, Luis Samaniego² and Karsten Schulz¹

¹Institute for Hydrology and Water Management (HyWa), University of Natural Resources and Life Sciences (BOKU), Vienna, Austria

²Department of Computational Hydrosystems, UFZ-Helmholtz Centre for Environmental Research, Leipzig, Germany

This study investigates automatic estimation of transfer functions for hydrological models.

This presentation aims to give some more detailed information in addition to the live presentation and is structured as follows:

1. Live presentation slide (used in the 2 min. talk)
2. Introduction
3. Study questions
4. Case study
5. Results
6. Conclusions

Catchment to model space mapping

Learning transfer functions from data by symbolic regression

Moritz Feigl¹, Robert Schweppe², Stephan Thober², Mathew Herrnegger¹, Luis Samaniego² and Karsten Schulz¹

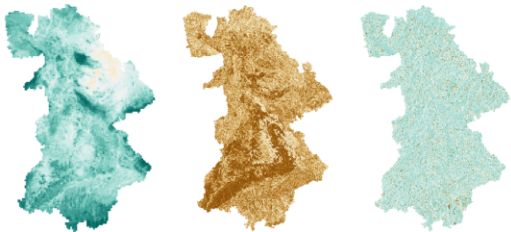
¹Institute for Hydrology and Water Management (HyWa), University of Natural Resources and Life Sciences (BOKU)

²Department of Computational Hydrosystems, UFZ-Helmholtz Centre for Environmental Research

contact: moritz.feigl@boku.ac.at

Problem

Distributed parameters are necessary for various types of models



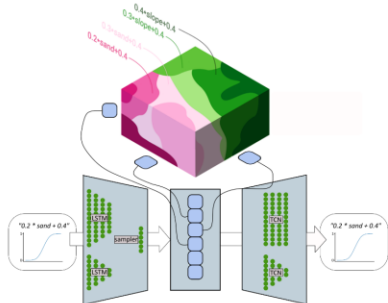
Potential model parameters

?

How do we estimate a spatially distributed parameter field?

Method

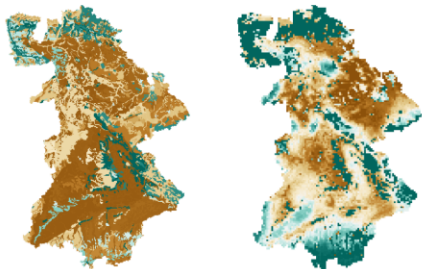
Deep learning model to enhance existing models



Function Space Optimization

New transfer functions

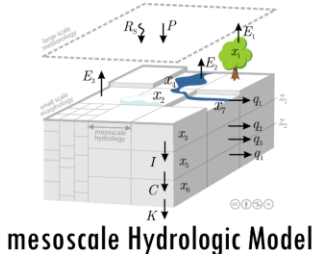
$$\text{Parameter 1} = \beta_1 \times \log(v_1) + \frac{v_2^3}{\beta_2 v_3}$$
$$\text{Parameter 2} = \beta_1 - \beta_2 v_1 + (v_3 - \beta_3)^2$$



Model parameters

Application

Case study



mesoscale Hydrologic Model



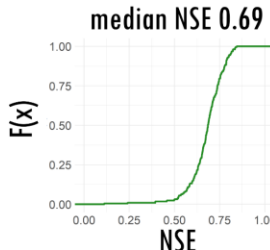
Training

5 years
7 gauges

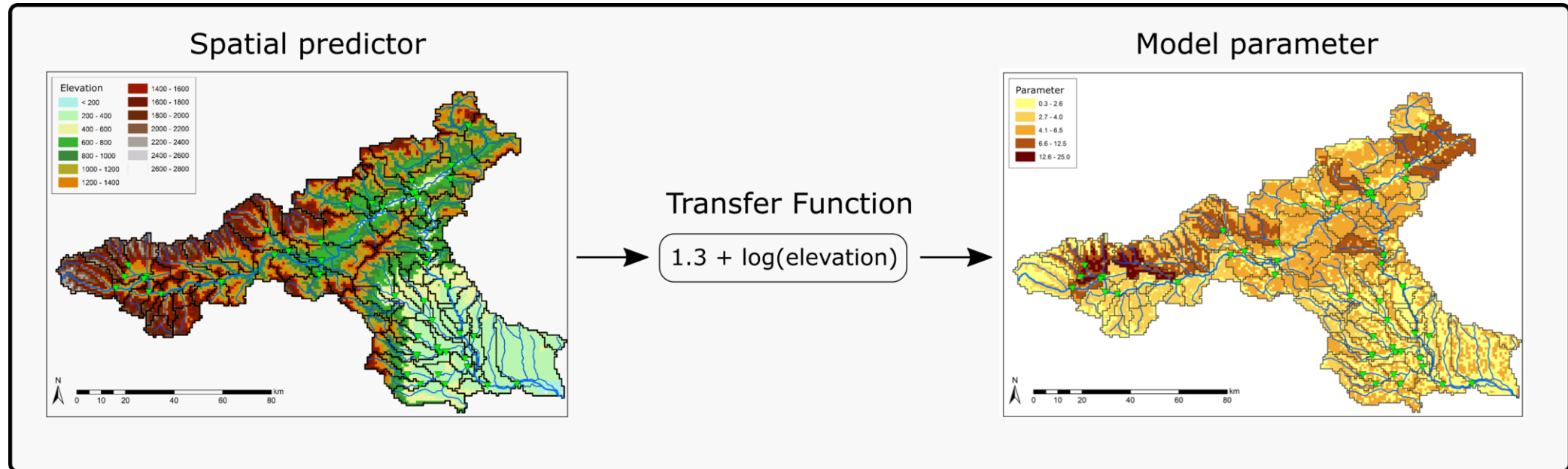


Validation

35 years
222 gauges



Introduction - Transfer Functions

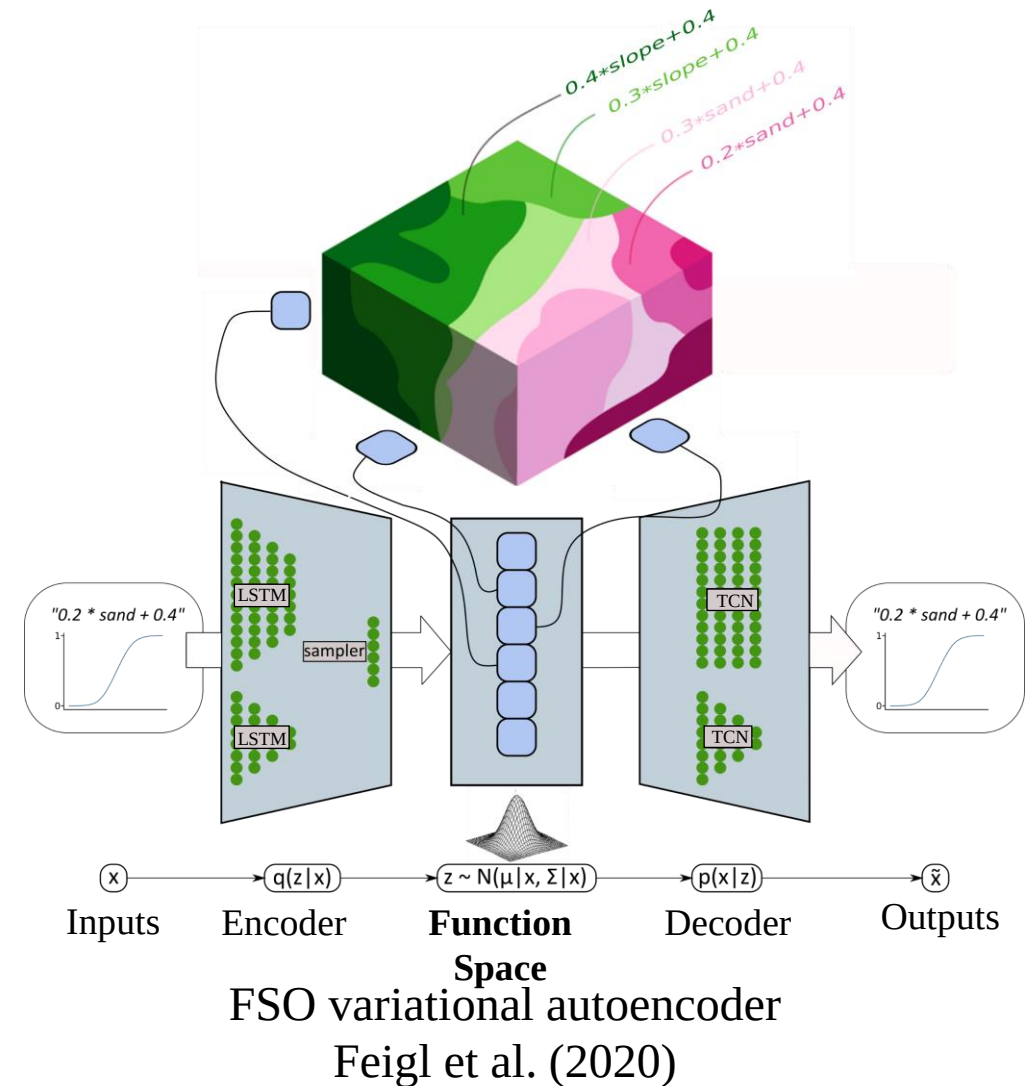


Transfer functions map geophysical catchment properties to distributed model parameters

Introduction - Function Space Optimization (FSO)

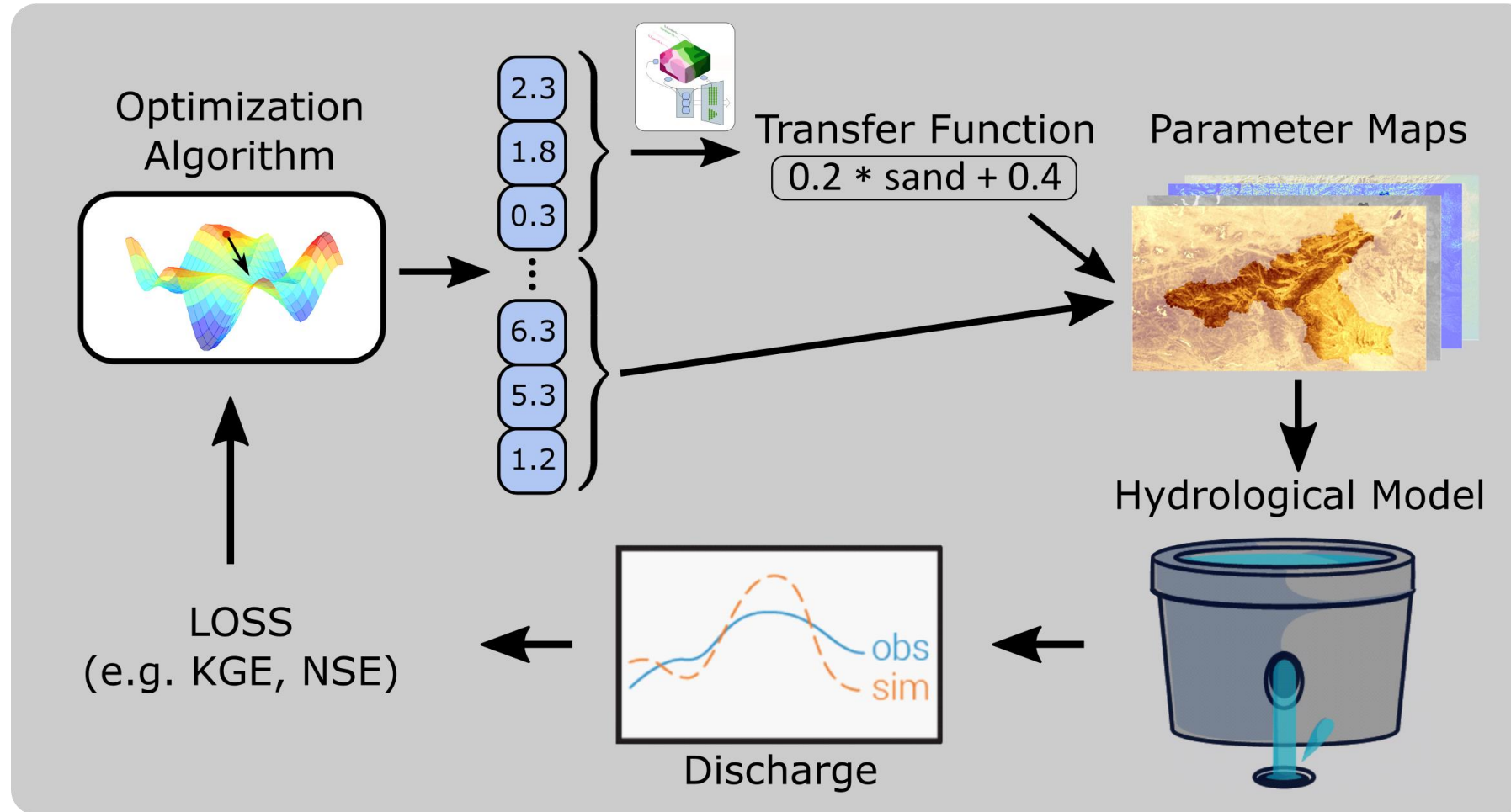
Feigl et al., 2020

- FSO: optimization method for transfer functions
- Uses a text generating Neural Network
- Transforms search into continuous problem
- Successfully tested on a single catchment



Introduction - Function Space Optimization (FSO)

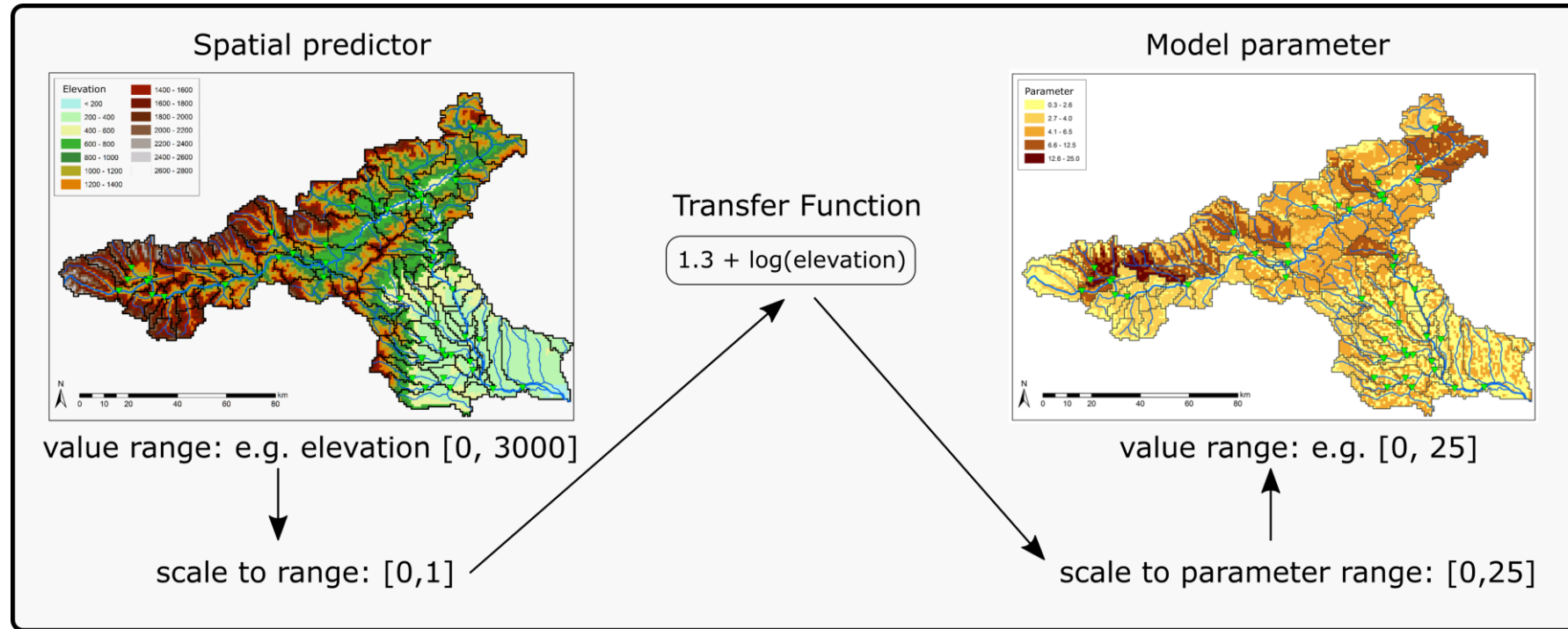
Function Space Optimization



Parts of the figure designed
by © Daniel Klotz

The FSO model optimization approach consists of the search for the best numerical parameters while simultaneously optimizing transfer functions using the VAE neural network.

Introduction - FSO parameter scaling



Parts of the figure designed
by © Daniel Klotz

All FSO TFs use scaled spatial predictors and are also scaled to the parameter range. To increase readability these scaling functions are not shown. However, it is important to keep them in mind, as even simple looking functions are in truth a bit more complex due to this scaling cascade. For example:

Scaling bounds:

$KSat \in [1.1, 1000] \text{ cm/day}$

$sand \in [0, 100] \%$

$KSat \text{ TFs range in}$

$trainig \in [-5.61, 8.48]$

Transfer function:

Simple: $KSat = sand - 0.7$

Full:

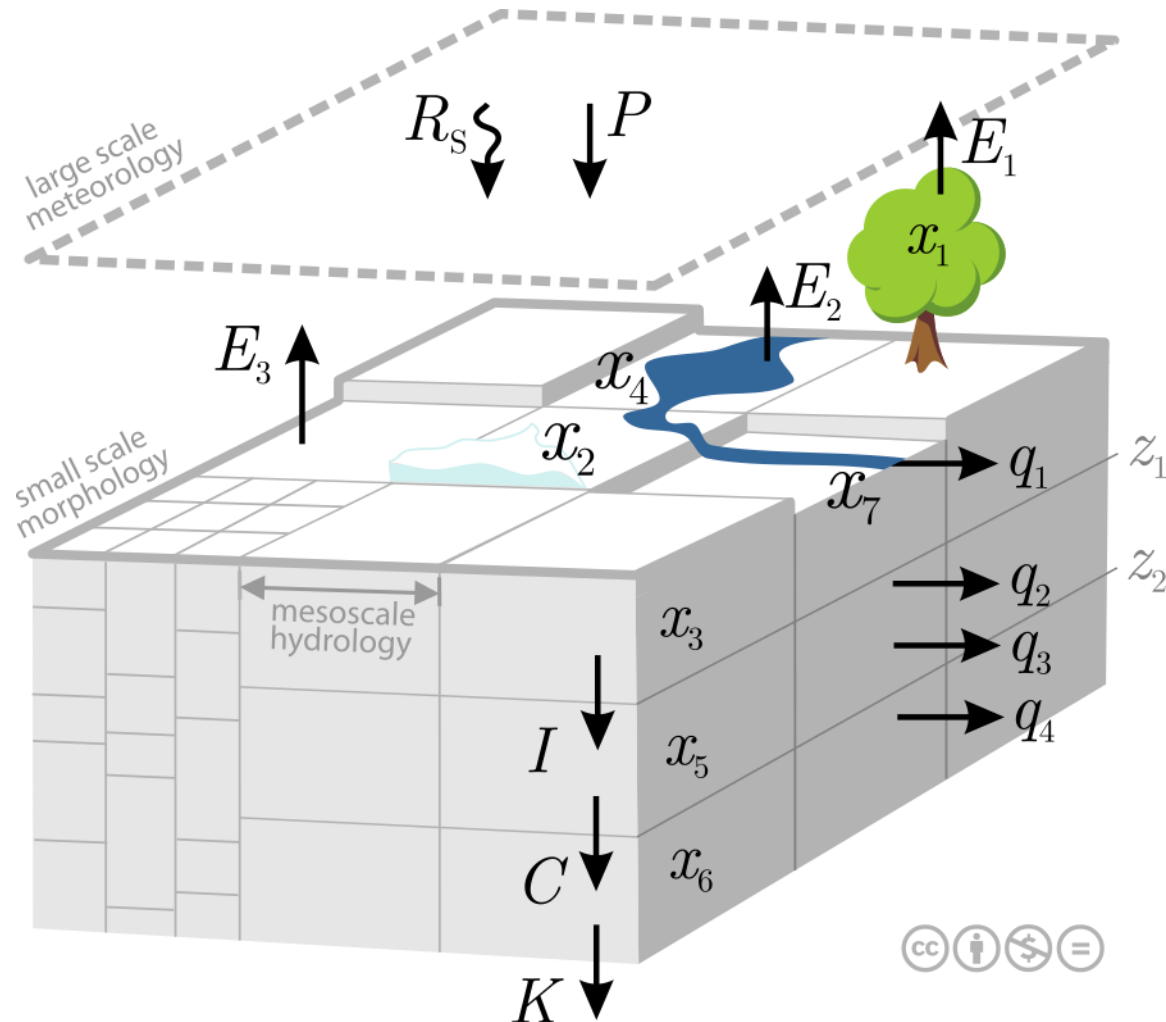
Research questions & Case study

1. Is it possible to use FSO to estimate transfer functions (TFs) in a complex distributed hydrological model that are transferable in space and time?
2. How do FSO estimated TFs differ from the original mHM functions in terms of performance and parameter fields?

Case study overview:

- Optimize TFs for 2 parameters of the mesoscale Hydrological Model
- Use 7 training basins and 222 validation basins to test transferability

Case study - mesoscale Hydrological Model (mHM)



- Developed by Samaniego et al. (2010)
- Spatially explicit distributed model
- Uses grid cells as primary units
- Defines parameter fields with the Multiscale Parameter Regionalization method (MPR)

Case study - Benchmark Study

Hydrol. Earth Syst. Sci., 21, 1769–1790, 2017
www.hydrol-earth-syst-sci.net/21/1769/2017/
doi:10.5194/hess-21-1769-2017
© Author(s) 2017. CC Attribution 3.0 License.



Hydrology and
Earth System
Sciences



A high-resolution dataset of water fluxes and states for Germany accounting for parametric uncertainty

Matthias Zink¹, Rohini Kumar¹, Matthias Cuntz^{1,2}, and Luis Samaniego¹

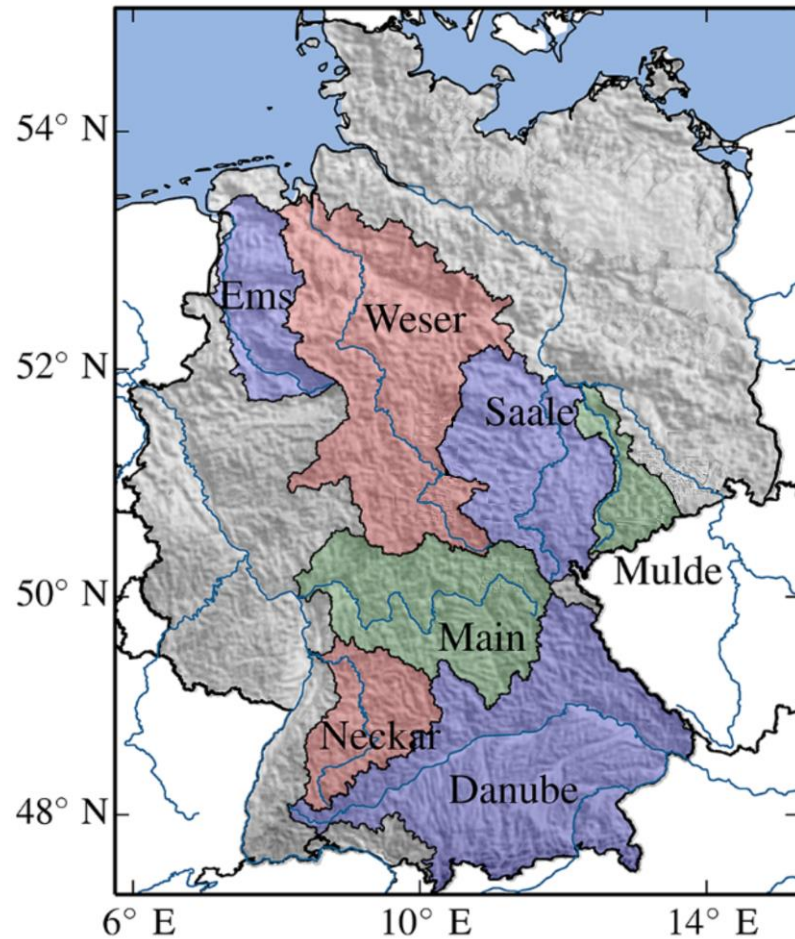
¹Helmholtz Centre for Environmental Research – UFZ, Department Computational Hydrosystems,
Permoserstraße 15, 04318 Leipzig, Germany

²INRA, Université de Lorraine, UMR1137 Ecologie et Ecophysiologie Forestières, Champenoux, France

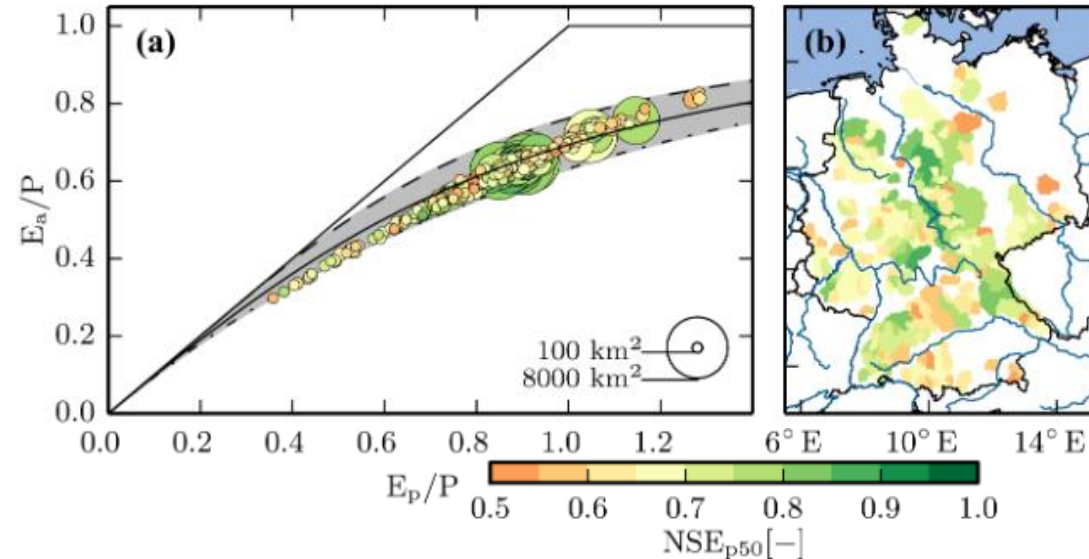
We compare our results to a study by Zink et al. (2017) using the same basins and time periods for training and validation.

Case study - Benchmark Study

- Optimizing mHM 100 times with 2000 iterations
- Using 7 gauging stations
- Validate 100 parameter sets on 220 Basins



Training basins
(Zink et al., 2017)



Validation basins
(Zink et al., 2017)

1. Run FSO 5 times to estimate TFs in 7 large training basins in training time (2000-2004) and assess their performance in test time (1965-1999).
2. Choose best FSO run by comparing the performance in 20 validation basins (full time series) that were sampled from all 222 validation basins. Stratified sampling was used so that the sampled basins are uniformly distributed across the budyko curve.
3. Comparing the best FSO Run with the median performance of Zink et al. (2017) in all 222 validation basins.
4. Assess non-FSO mHM parameter results.
5. Assess FSO Parameter fields and their difference compared to the original mHM functions.

Case study - Technical details

Important for completeness, but you can also skip them!

FSO Runs:

- Optimizer: Runs 1-4 Shuffled Complex Evolution (SCE), Run 5 DDS
- Estimate 2 sensitive TFs (Ksat and FieldCap) while optimizing 59 non-FSO mHM parameters
- Loss function for training and comparison is a power mean of NSE and log NSE:

$$Loss_{basin} = \left(\frac{1}{2} \sum_i^2 \phi_i^6 \right)^{\frac{1}{6}}, \text{ with } \phi_1 = NSE \text{ and } \phi_2 = \log NSE$$

- The FSO optimization loss for all basins is the weighted mean loss, with weights = $1 - Loss_{basin}$.

Zink et al.(2017) benchmark (important for interpretation!):

- Estimated 100 parameter sets in each of the 7 large training basins in training time (2000-20004) with 2000 DDS iterations (i.e. 700 parameter sets)
- Applied each set in the other 6 basins in validation time (1965-1999) and keep only those with $NSE \geq 0.65$.
- Apply mHM with these best performing 100 parameter sets on the 222 validation basins.

Zink et al. results → NSE values of 100 independent mHM parameter sets

We compare the median NSE from these 100 runs with our results. → Difficult to interpret, but gives a general intuition of what can be considered “good” or “bad” performance.

Case study - Technical details

FSO Runs:

- Optimizer: Runs 1-4 Shuffled Complex Evolution (SCE), Run 5 DDS
- Estimate 2 sensitive TFs (Ksat and FieldCap) while optimizing 59 non-FSC
- Loss function for training and comparison is a power mean of NSE and log

From the previous test (EGU20, AGU20) we already knew that the DDS converged to a local minimum very fast (<500 iterations) and did not seem to be able to adequately solve this optimization problem. Therefore, we switched to the SCE and kept one DDS run for comparison).

$$Loss_{basin} = \left(\frac{1}{2} \sum_i^2 \phi_i^6 \right)^{\frac{1}{6}}, \text{ with } \phi_1 = NSE \text{ and } \phi_2 = \log NSE$$

- The FSO optimization loss for all basins is the weighted mean loss, with weights = $1 - Loss_{basin}$.

Zink et al.(2017) benchmark (important for interpretation!):

- Estimated 100 parameter sets in each of the 7 large training basins in training time (2000-20004) with 2000 DDS iterations (i.e. 700 parameter sets)
- Applied each set in the other 6 basins in validation time (1965-1999) and keep only those with $NSE \geq 0.65$.
- Apply mHM with these best performing 100 parameter sets on the 222 validation basins.

Zink et al. results → NSE values of 100 independent mHM parameter sets

We compare the median NSE from these 100 runs with our results. → Difficult to interpret, but gives a general intuition of what can be considered “good” or “bad” performance.

Case study - Technical details

FSO Runs:

- Optimizer: Runs 1-4 Shuffled Complex Evolution (SCE), Run 5 DDS
- Estimate 2 sensitive TFs (Ksat and FieldCap) while optimizing 59 non-FSO mHM parameters
- Loss function for training and comparison is a power mean of NSE and log NSE:

$$Loss_{basin} = \left(\frac{1}{2} \sum_i^2 \phi_i^6 \right)^{\frac{1}{6}}, w$$

As this is the first study that automatically estimated TFs for a large set of basins, we wanted to keep it simple and only chose to estimate two sensitive TFs .

- The FSO optimization loss for all basins is the weighted mean loss, with weights $w_i = 1 / Loss_{basin}$.

Zink et al.(2017) benchmark (important for interpretation!):

- Estimated 100 parameter sets in each of the 7 large training basins in training time (2000-20004) with 2000 DDS iterations (i.e. 700 parameter sets)
- Applied each set in the other 6 basins in validation time (1965-1999) and keep only those with $NSE \geq 0.65$.
- Apply mHM with these best performing 100 parameter sets on the 222 validation basins.

Zink et al. results → NSE values of 100 independent mHM parameter sets

We compare the median NSE from these 100 runs with our results. → Difficult to interpret, but gives a general intuition of what can be considered “good” or “bad” performance.

Case study - Technical details

FSO Runs:

This loss function results in a model that has equally good NSE and log NSE performance.

- Optimizer: Runs 1-4 Shuffled Complex Evolution (SCE), Run 5 DDS
- Estimate 2 sensitive TFs (Ksat and FieldCap) while optimizing 59 non-FSO mHM parameters
- Loss function for training and comparison is a power mean of NSE and log NSE:

$$Loss_{basin} = \left(\frac{1}{2} \sum_i^2 \phi_i^6 \right)^{\frac{1}{6}}, \text{ with } \phi_1 = NSE \text{ and } \phi_2 = \log NSE$$

- The FSO optimization loss for all basins is the weighted mean loss, with weights = $1 - Loss_{basin}$.

Zink et al.(2017) benchmark

This formulation is taken from Zink et al. (2017) and is technically only correct for $NSE/\log NSE > 0$. In case that $NSE/\log NSE < 0$: $\text{sgn}(\phi_i) * \phi_i^6$ should be used inside the sum.

- Estimated 100 parameter sets using 7000 DDS iterations (i.e. 7000000 iterations) on training time (2000-20004) with 2000
- Applied each set in the validation step only those with $NSE \geq 0.65$.
- Apply mHM with these best performing 100 parameter sets on the 222 validation basins.

Zink et al. results → NSE values of 100 independent mHM parameter sets

We compare the median NSE from these 100 runs with our results. → Difficult to interpret, but gives a general intuition of what can be considered “good” or “bad” performance.

Case study - study basins

7 Training basins, 220 Validation Basins

Resolution:

Spatial predictors: 100 x 100 m

Model grid: 4 x 4 km

Spatial predictors:

Mean sand percentage (sand)

Mean clay percentage (clay)

Mineral bulk density

Aspect

Terrain slope

Elevation

Time series:

7 & 220 gauging stations

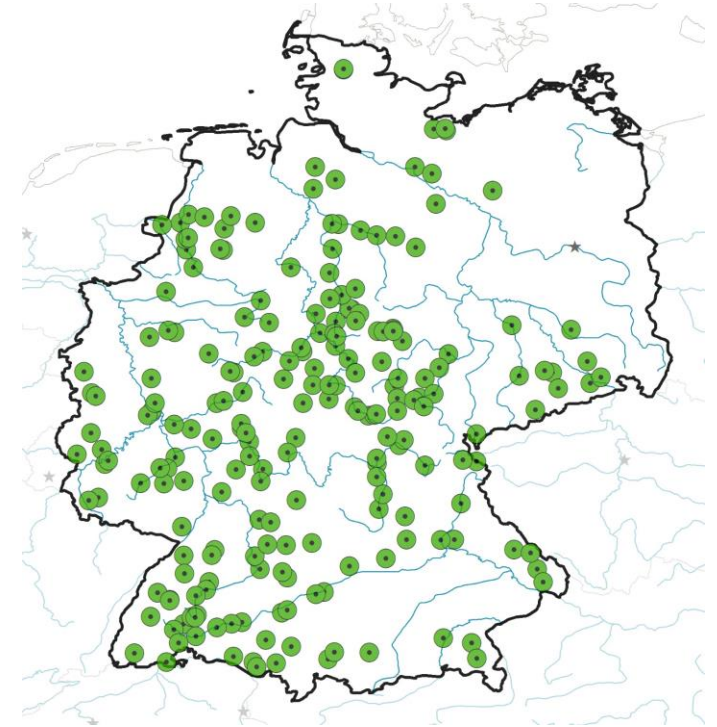
Calibration: 2000-2004

Validation: 1965-1999

Spin-up: 5 years



7 Training basins

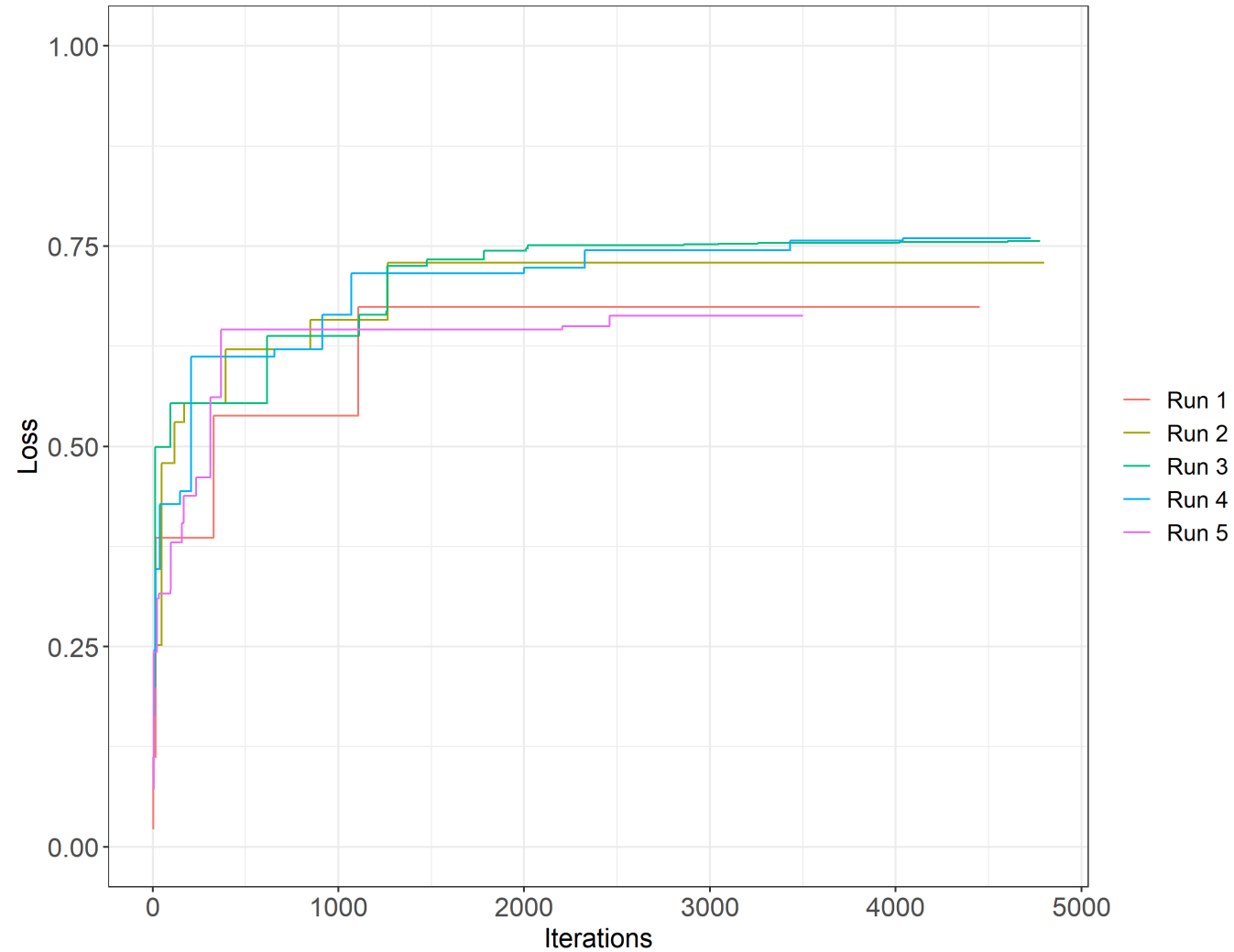


220 Validation basins

Results – optimization

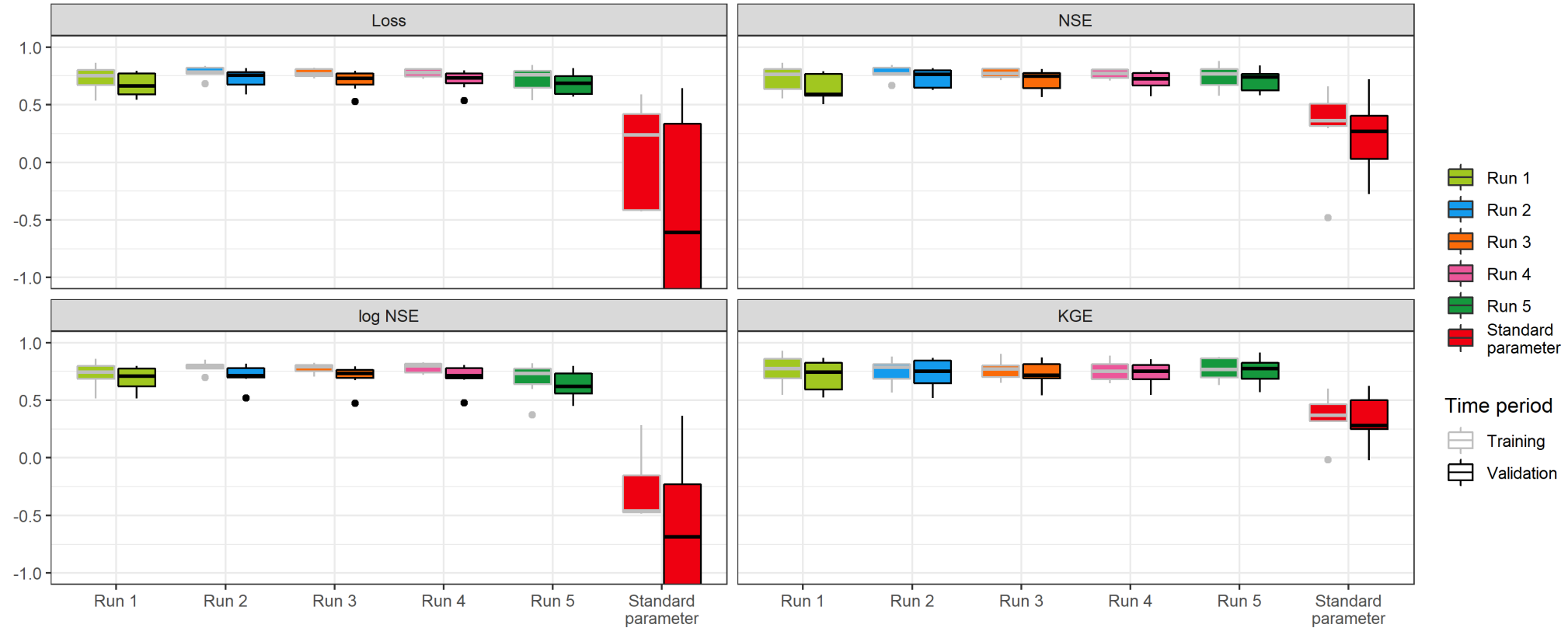


7 Training basins



Overview of loss per iteration of all FSO
optimization runs during training

Results – 7 Training Basins



KSat

Run 1 **dem** - 3.081 * (**bd** + cosh(**slope**))

Run 2 **aspect** - 2.96 * **bd** - 3.078

Run 3 -3.135 - cosh(**bd**)

Run 4 log10(**aspect**)-3.115

Run 5 (0.385/(**clay***asin(**sand**) – 1.0)) * 3.828 - (**clay** + cos(**dem**))

FieldCap

cos(**clay**)/**bd** + tanh(**slope**)

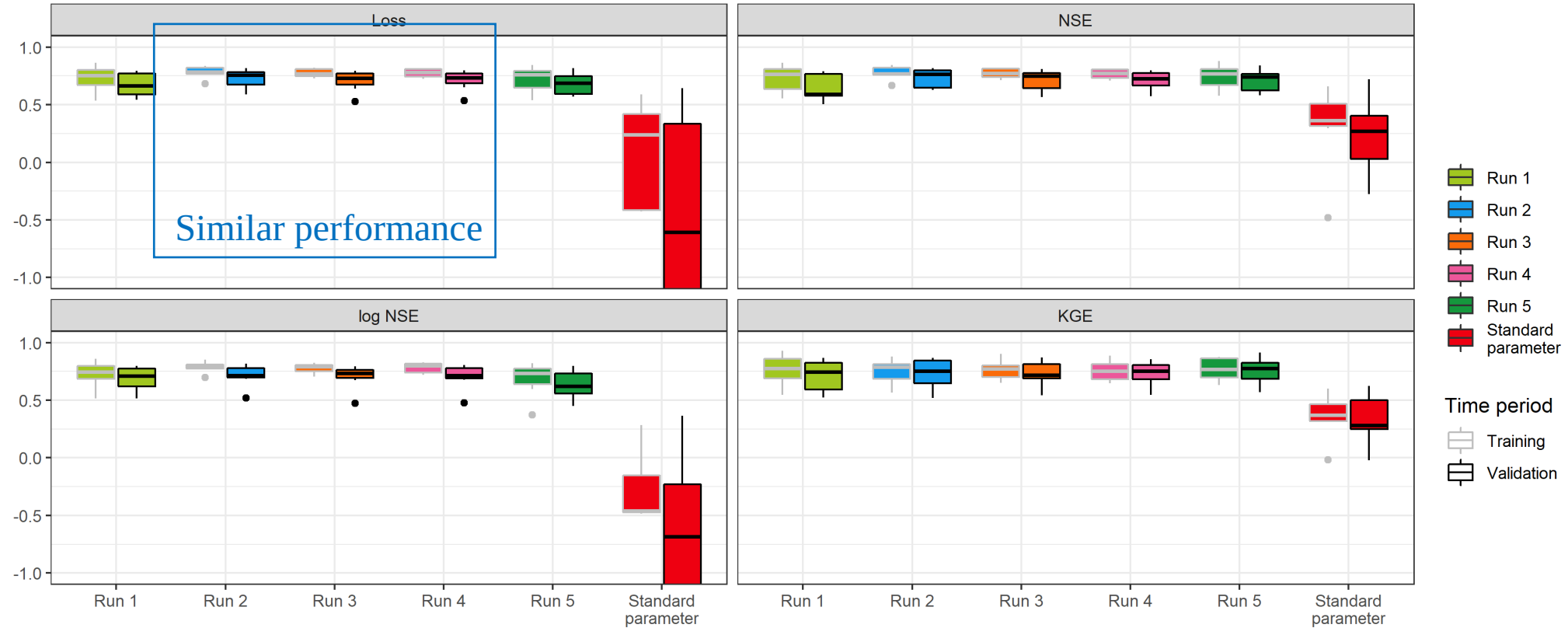
cos(cosh(**dem**) * cosh(**dem**))^2.95

Ksat - (2.84 + **ThetaS***cosh(**bd**))

aspect - **aspect**-3.165

slope

Results – 7 Training Basins



KSat

- Run 1 $\text{dem} - 3.081 * (\text{bd} + \cosh(\text{slope}))$
- Run 2 $\text{aspect} - 2.96 * \text{bd} - 3.078$
- Run 3 $-3.135 - \cosh(\text{bd})$
- Run 4 $\log_{10}(\text{aspect}) - 3.115$
- Run 5 $(0.385 / (\text{clay} * \sin(\text{sand}) - 1.0)) * 3.828 - (\text{clay} + \cos(\text{dem}))$

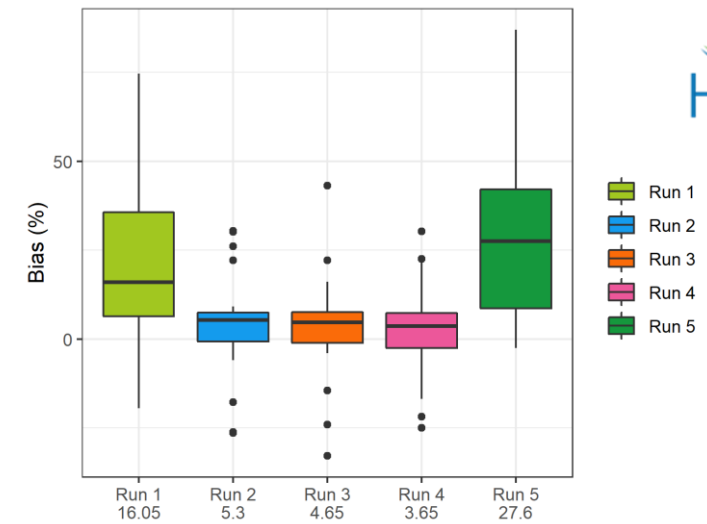
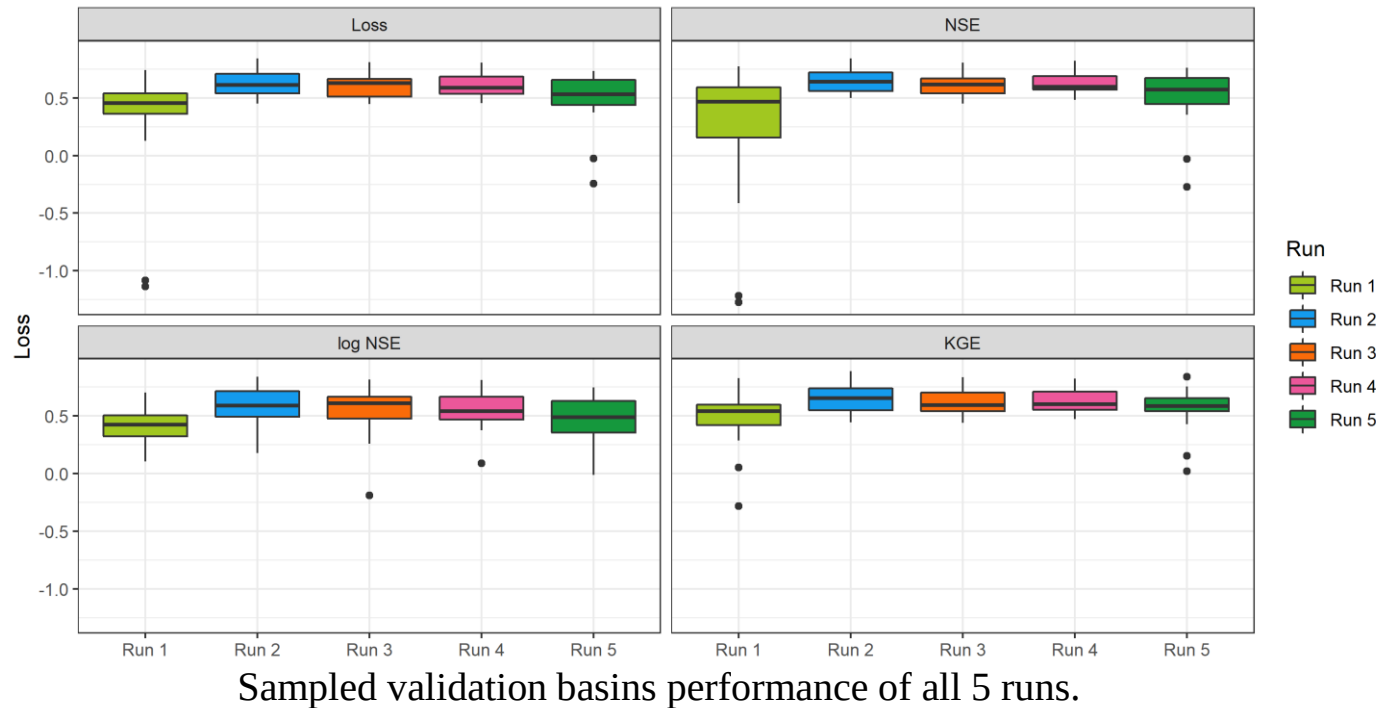
FieldCap

- $\cos(\text{clay}) / \text{bd} + \tanh(\text{slope})$
- $\cos(\cosh(\text{dem}) * \cosh(\text{dem}))^{2.95}$
- $\text{Ksat} - (2.84 + \text{ThetaS} * \cosh(\text{bd}))$
- $\text{aspect} - \text{aspect} - 3.165 \sim 0.1 \text{ (scaled values)}$
- slope

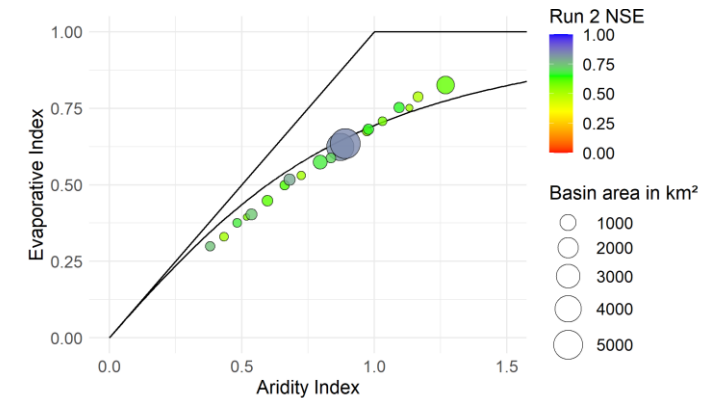
Very different functions!

We did not allow for constant values, but it still managed to generate them.

Results – Sampled validation basins



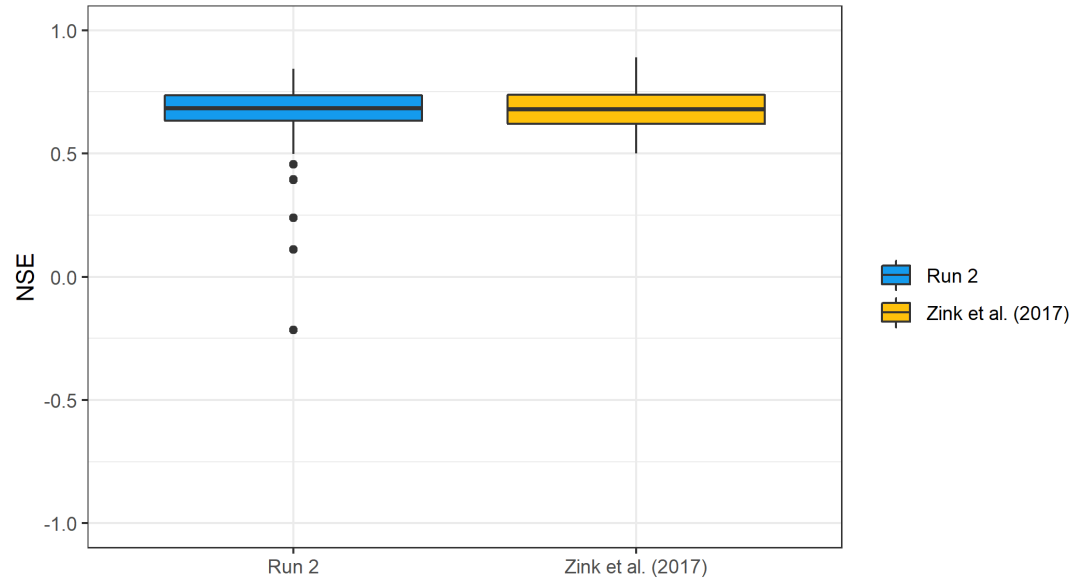
Percentage bias of all sampled validation basins with corresponding median values on the x-axis.



Budyko curve position of sampled validation basins and Run 2 performance

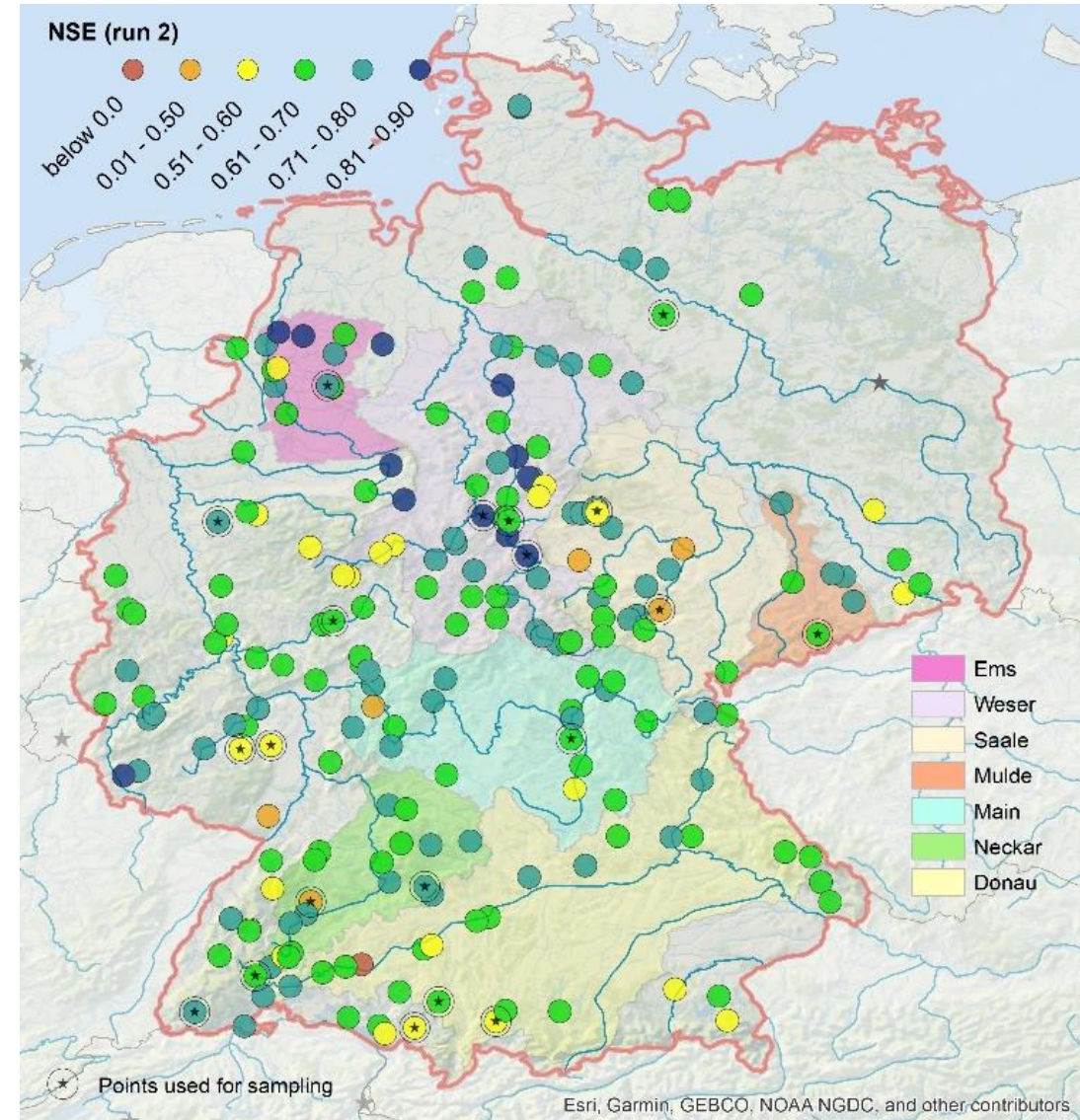
From this we can conclude that run 2, 3, 4 are very similar, but run 2 has the best performance.
→ Focus on Run 2

Results – Validation basins

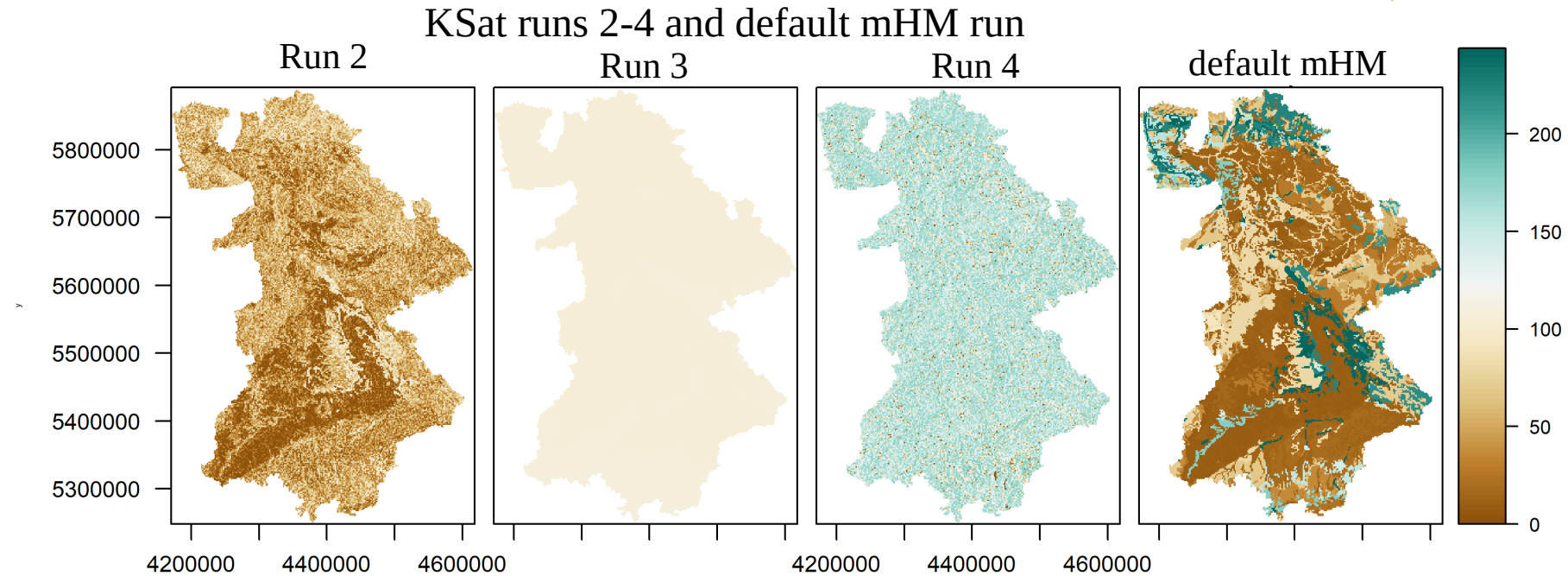


Validation basin NSE of FSO Run 2 and Zink et al. (2017) benchmark.

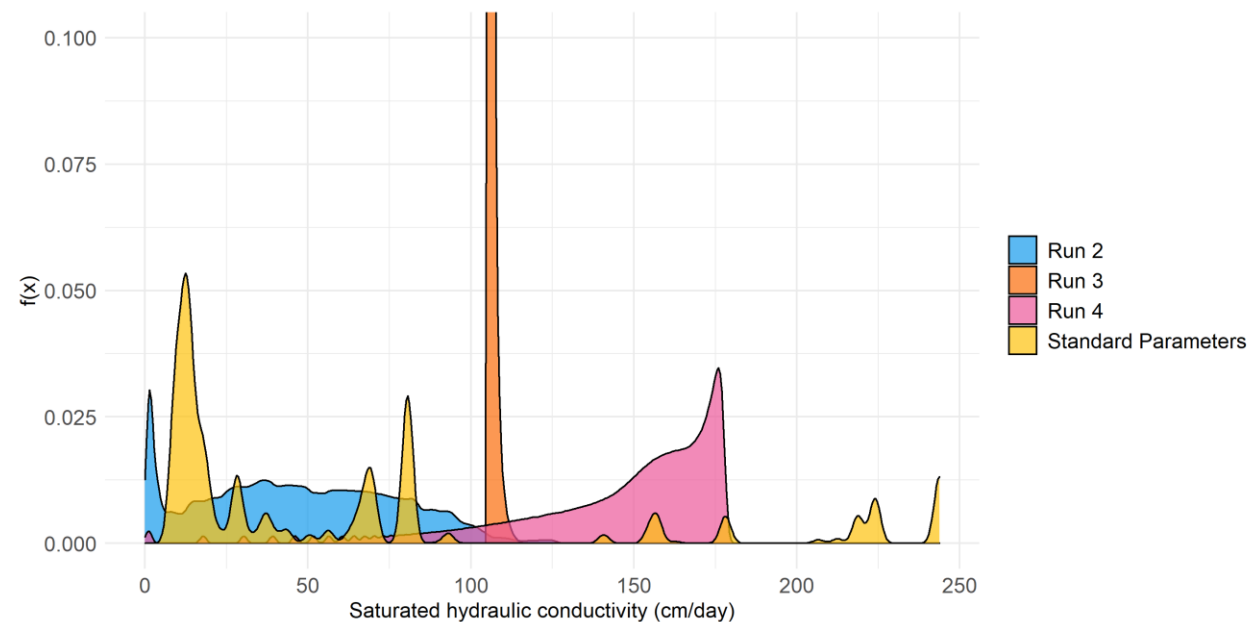
One FSO parameter set performs equally well as the median of 100 optimized original mHM parameter sets.



Results – Parameter fields KSat

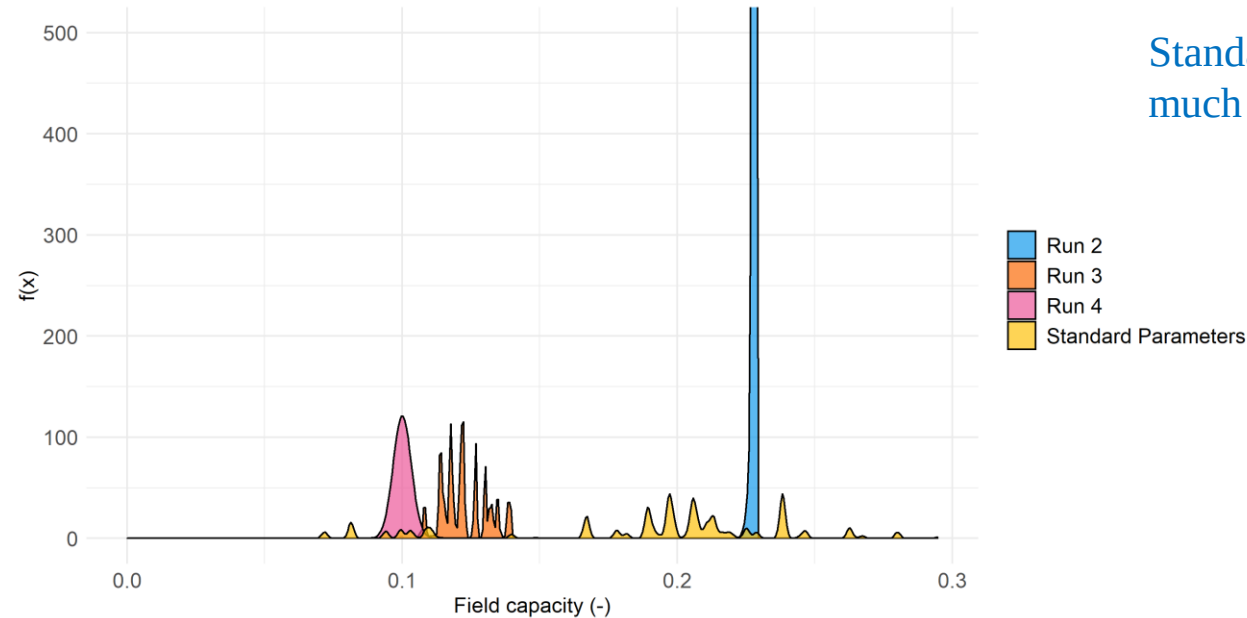
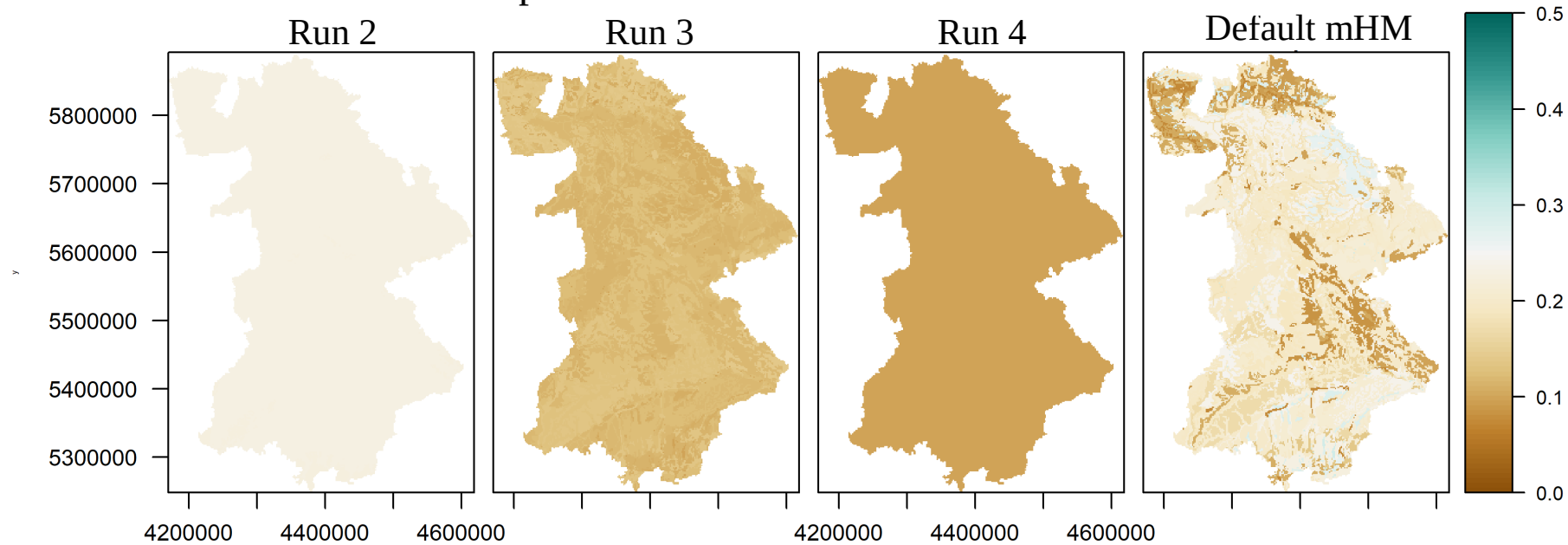


Very different fields.
Run 2 and default show
somewhat similar
spatial patterns.



Results – Parameter fields FieldCap

FieldCap runs 2-4 and default mHM run



Standard parameter have a much larger variability.

- FSO is able to estimate TFs that can be used to transfer model parameters across time and space!

Our main goal!

- FSO works with a very minimalistic setting - only 5 years data of 7 gauging stations:

training median NSE = 0.77

PUB median KGE = 0.69

- Different TFs and resulting parameter fields lead to similar results -> physical interpretation is difficult at this point.
- Larger number of training set will most likely significantly increase performance and transferability.
- Regularization with additional boundary condition (e.g. ET) seems to be necessary to derive TFs that produce parameter fields that can more robustly reflect the physical properties.

References

Feigl, M., Herrnegger, M., Klotz, D., & Schulz, K. (2020). Function Space Optimization: A symbolic regression method for estimating parameter transfer functions for hydrological models. *Water resources research*, 56(10), <https://doi.org/10.1029/2020WR027385>

Samaniego, L., Kumar, R., & Attinger, S. (2010). Multiscale parameter regionalization of a grid-based hydrologic model at the mesoscale. *Water Resources Research*, 46(5).
<https://doi.org/10.1029/2008WR007327>

Zink, M., Kumar, R., Cuntz, M., & Samaniego, L. (2017). A high-resolution dataset of water fluxes and states for Germany accounting for parametric uncertainty. *Hydrology and Earth System Sciences*, 21(3), <https://doi.org/10.5194/hess-21-1769-2017>