University of Stuttgart
Germany

SimTech

UNITE

To Physics-Inform or Physics-Ignore?

Evaluating the Balance of Physics-Based and Data-Driven Components in Hybrid Hydrological Models Using Information Theory

Manuel Álvarez Chaves

Eduardo Acuña Espinoza, Uwe Ehret and Anneli Guthke

SimTech
Junior Research Group for
Statistical Model-Data Integration

December 9th, 2024

AGU24
Washington, D.C. | 9–13 December 2024

Hello online viewer!

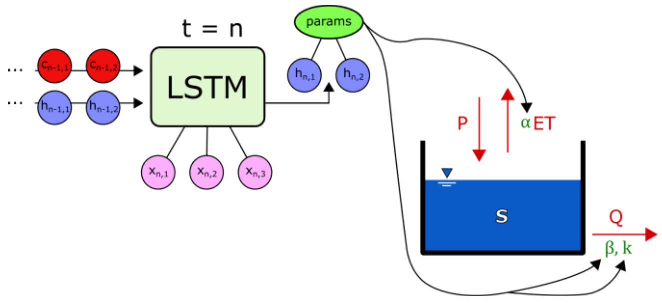
This is an alternative version of the slides that I have prepared for my talk, suitable for online and asynchronous viewing.

If you are at AGU and find these slides interesting, I would greatly enjoy speaking more about Hybrid Models and/or Information Theory during the conference. You can find me at my talk in:

- **Session:** H11E - Integrating Machine Learning and Physics-Based Models in Watershed
- **Location:** 143 A-C (Convention Center)
- **Date:** Monday, 9 of December, 2024
- **Time:** 9:02 – 9:12
- LINK

Hybrid models

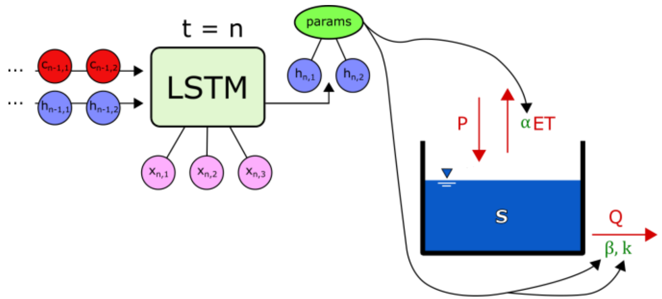
The premise



Hybrid models

The premise

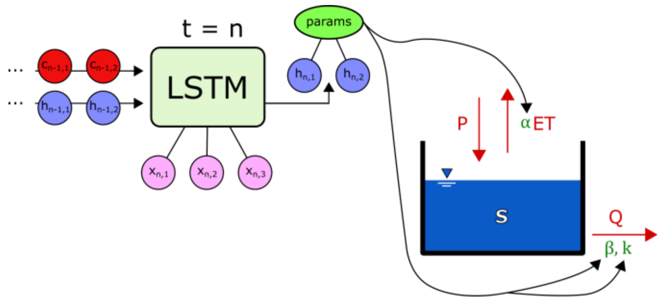
- There is a new paradigm of hybrid models in which “the model variables and/or specific components are replaced by neural networks”.



Hybrid models

The premise

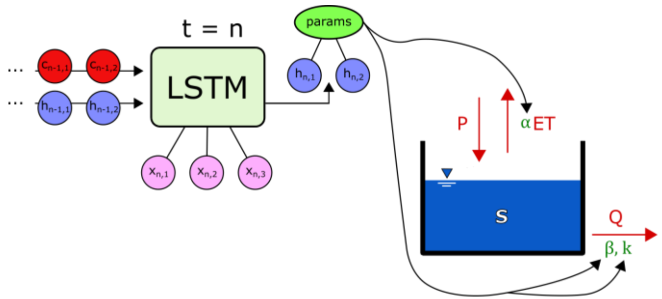
- There is a new paradigm of hybrid models in which “the model variables and/or specific components are replaced by neural networks”.
- Although this new paradigm brings along a lot of opportunities for better and more interpretable models, we also see a big **RISK** that neural networks will overwhelm any physical constraints.



Hybrid models

The premise

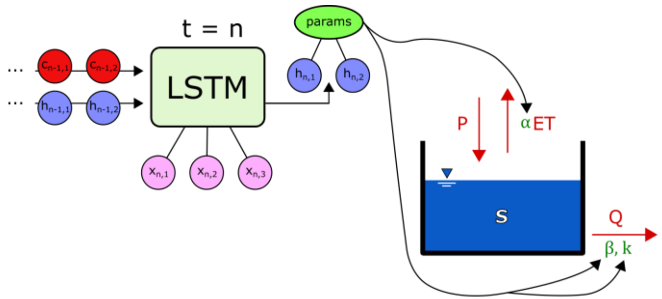
- There is a new paradigm of hybrid models in which “the model variables and/or specific components are replaced by neural networks”.
- Although this new paradigm brings along a lot of opportunities for better and more interpretable models, we also see a big **RISK** that neural networks will overwhelm any physical constraints.
- Can we identify this behavior?



Hybrid models

The premise

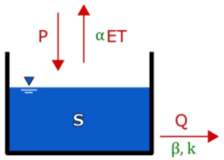
- There is a new paradigm of hybrid models in which “the model variables and/or specific components are replaced by neural networks”.
- Although this new paradigm brings along a lot of opportunities for better and more interpretable models, we also see a big **RISK** that neural networks will overwhelm any physical constraints.
- Can we identify this behavior?
- Yes we do! I’ll demonstrate…



Synthetic examples (toy models)

Model 1: True model

$\alpha = 0.8 \quad \beta = 1.2 \quad k = 26.0$



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^\beta}{k}$$

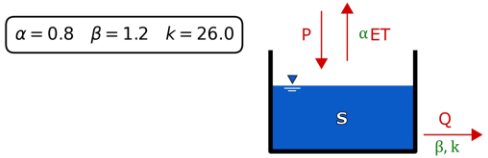
First, take a very simple hydrological model like the one on the left. This model has a set of mathematical equations, shown below, which define its behavior.

The model also has tunable parameters: α, β and k. These parameters allow for the model to behave differently.

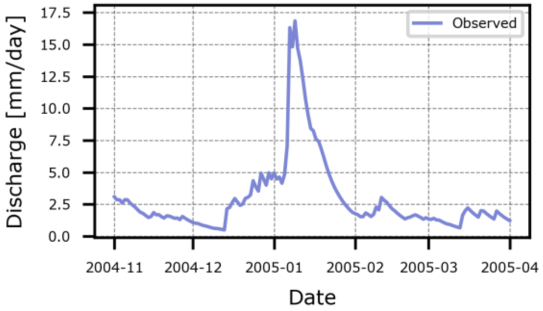
Normally these parameters would be used to fit the model to some data, but here we want to generate synthetic data. So we pick some parameters…

Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^\beta}{k}$$



First, take a very simple hydrological model like the one on the left. This model has a set of mathematical equations, shown below, which define its behavior.

The model also has tunable parameters: α, β and k. These parameters allow for the model to behave differently.

Normally these parameters would be used to fit the model to some data, but here we want to generate synthetic data. So we pick some parameters…

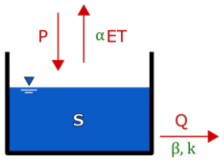
and drive some input precipitation data to generate a hydrograph.

The key point here is that we know the “true” model and parameters that were used to generate the “observed” data.

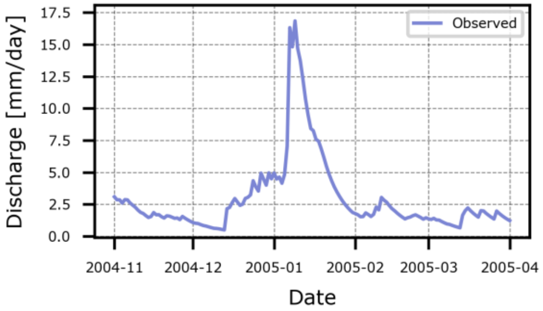
Synthetic examples (toy models)

Model 1: True model

$\alpha = 0.8 \quad \beta = 1.2 \quad k = 26.0$

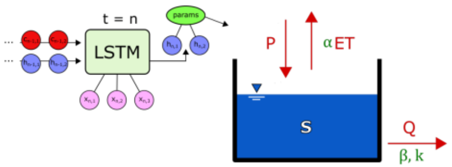


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^\beta}{k}$$

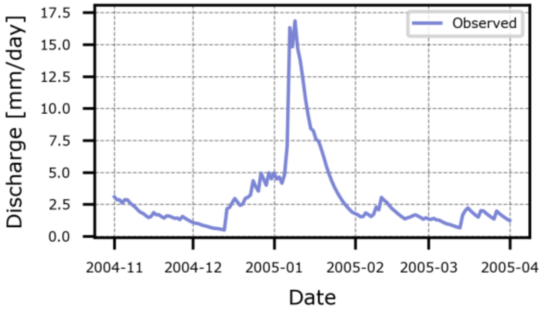


Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

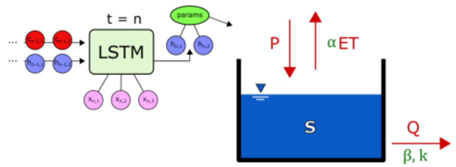


Now we replace the model’s fixed set of parameters with a neural network that will adjust the parameters at every time step.

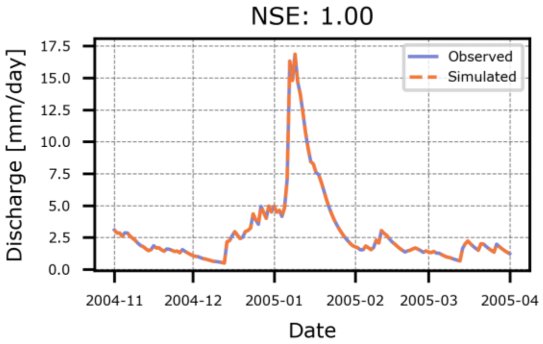
Using this alternative model to simulate…

Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$



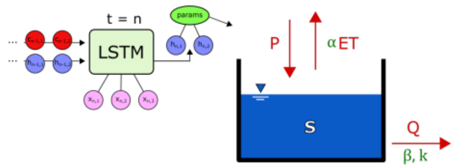
Now we replace the model’s fixed set of parameters with a neural network that will adjust the parameters at every time step.

Using this alternative model to simulate…

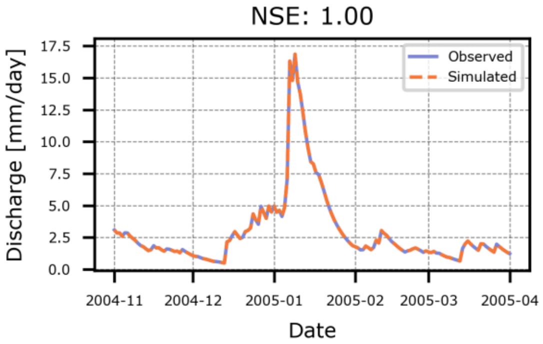
We get a perfect match!

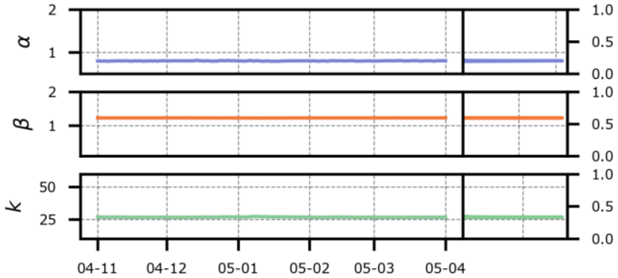
Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$





Now we replace the model's fixed set of parameters with a neural network that will adjust the parameters at every time step.

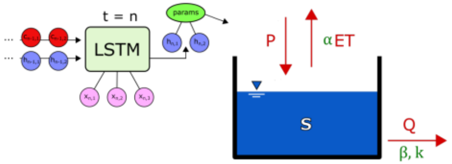
Using this alternative model to simulate…

We get a perfect match!

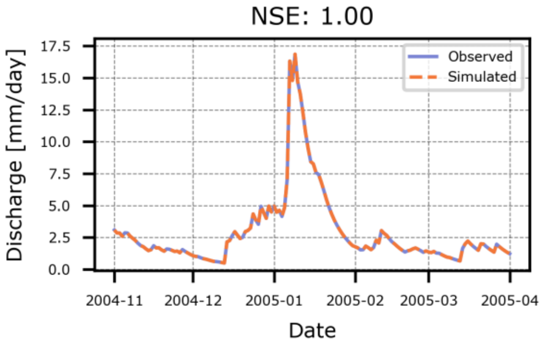
This happens because the neural network accurately identifies that it was coupled with the "true" model which had fixed parameters. Let's see what happens when it's not coupled with the true model.

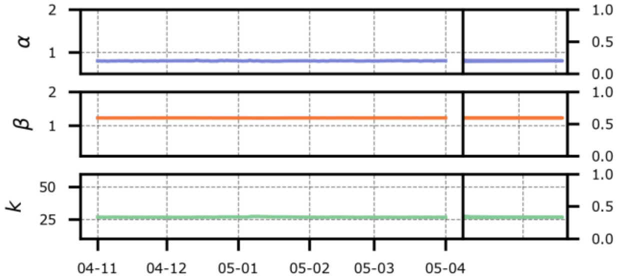
Synthetic examples (toy models)

Model 1: True model

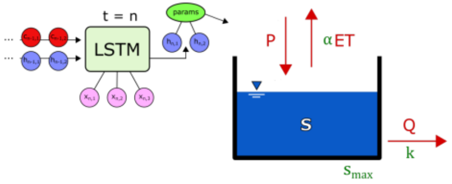


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

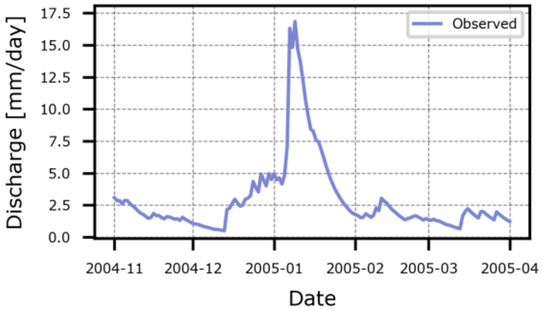




Model 2: Wrong process representation (correct architecture)



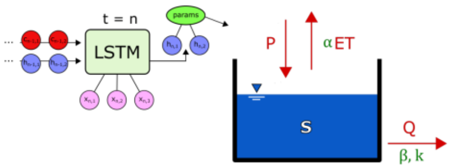
$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$



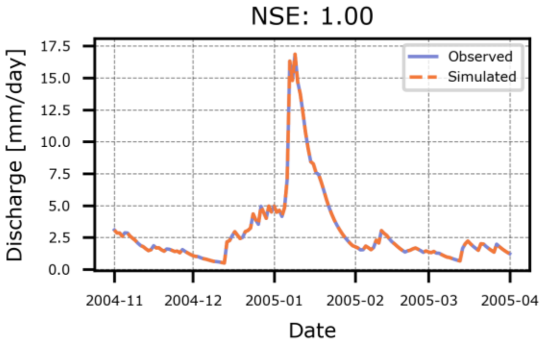
Model 2 is very similar but notice that the equations below are very different.

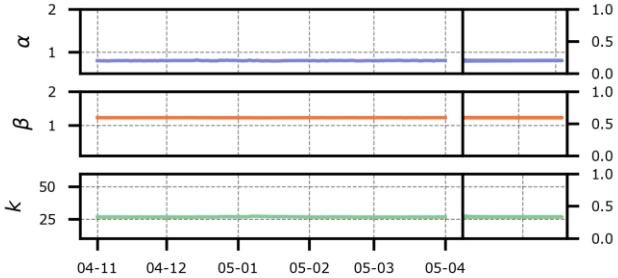
Synthetic examples (toy models)

Model 1: True model

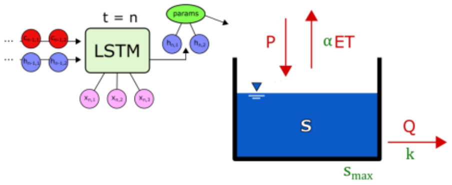


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

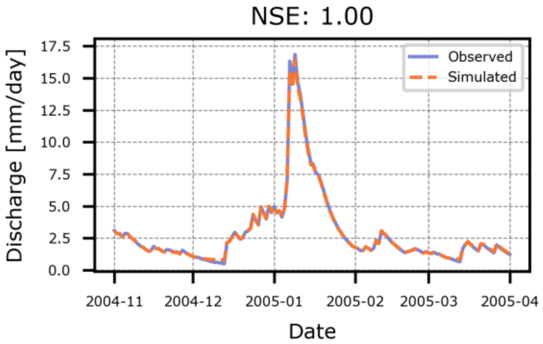




Model 2: Wrong process representation (correct architecture)



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$

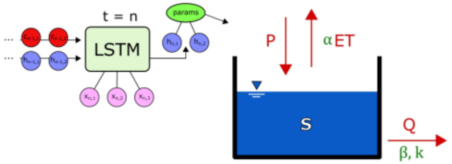


Model 2 is very similar but notice that the equations below are very different.

Nevertheless, when we simulate using this “wrong” model, we get a perfect match.

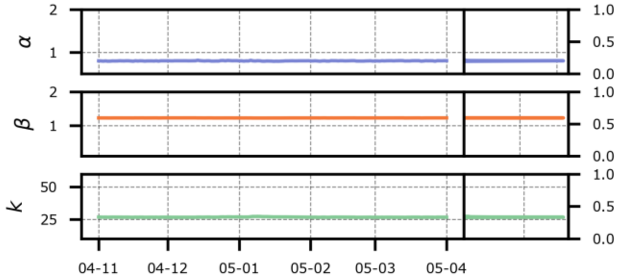
Synthetic examples (toy models)

Model 1: True model

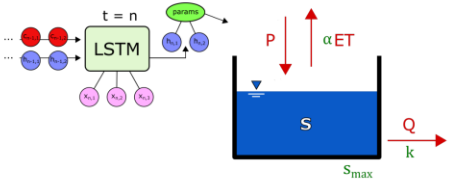


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

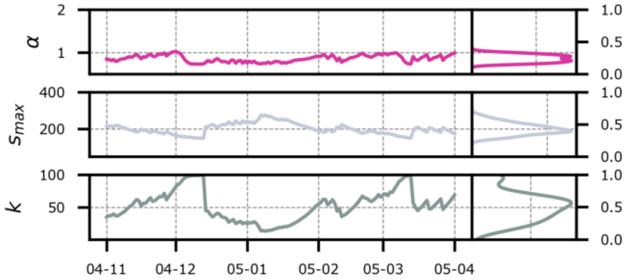
This happens because the neural network in the second model is able to adjust the parameters to match the behavior of the “true” model.



Model 2: Wrong process representation (correct architecture)

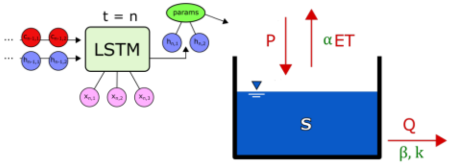


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$



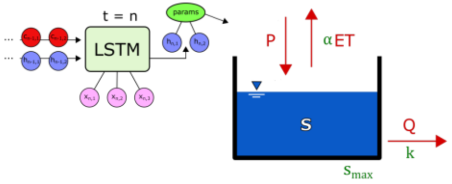
Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

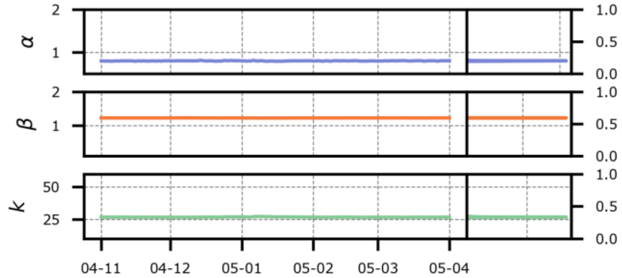
Model 2: Wrong process representation (correct architecture)

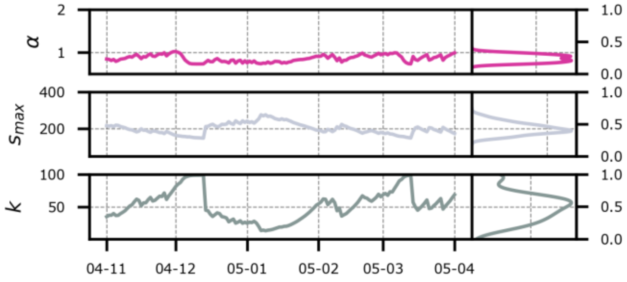


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$

This happens because the neural network in the second model is able to adjust the parameters to match the behavior of the "true" model.

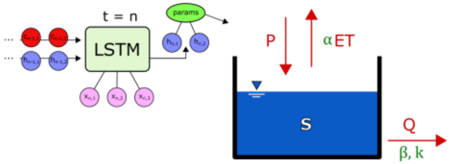
This tells us that, even though the behavior and response of both models is the same; the latter is making accurate predictions not because of it's physical structure, but because the neural network is compensating for its deficiencies.





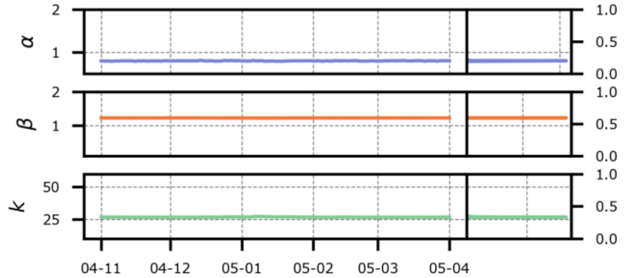
Synthetic examples (toy models)

Model 1: True model

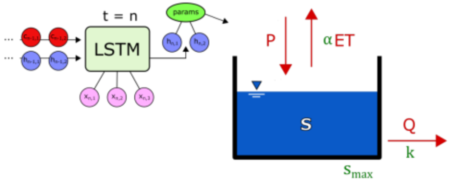


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

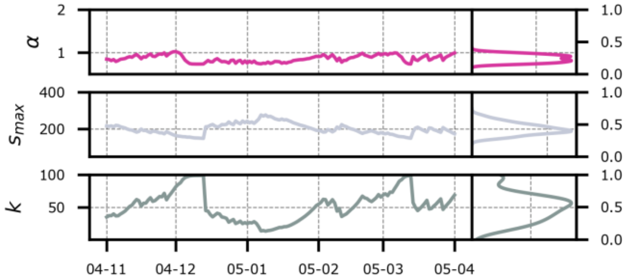
This result is very easy to identify by simply looking at the parameters: horizontal lines vs. wiggly lines. But we would like to put a number to it.



Model 2: Wrong process representation (correct architecture)

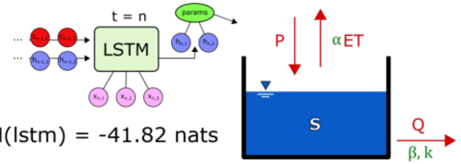


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$



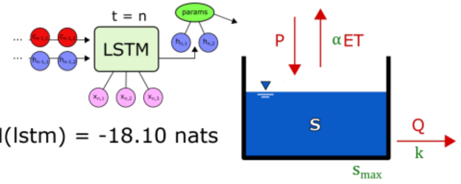
Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

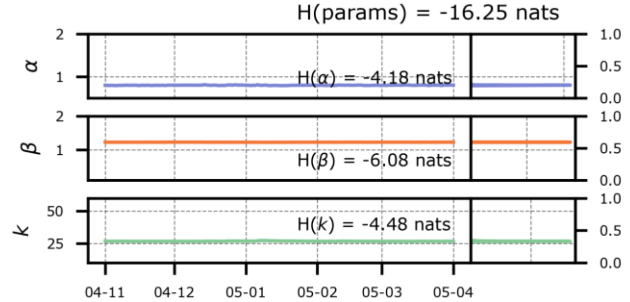
Model 2: Wrong process representation (correct architecture)

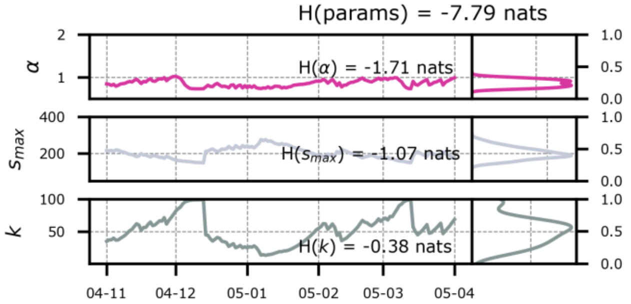


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$

This result is very easy to identify by simply looking at the parameters: horizontal lines vs. wiggly lines. But we would like to put a number to it.

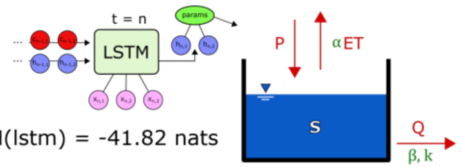
Here's where Information Theory comes in. The entropy of a variable H(X) can be used to identify it's overall variability, so we quantify the entropy of model the models dynamic parameters and the internal states of the neural network.





Synthetic examples (toy models)

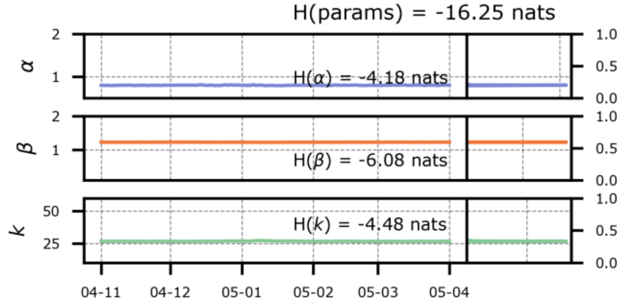
Model 1: True model



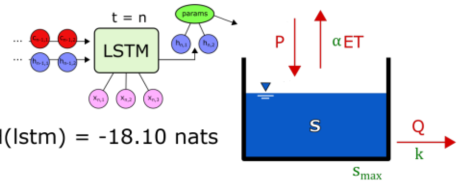
H(lstm) = -41.82 nats

$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

Don't mind the negative numbers, this is just an artifact of calculating entropy for continuous variables.

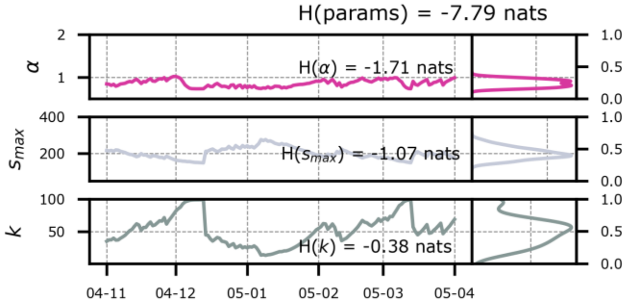


Model 2: Wrong process representation (correct architecture)



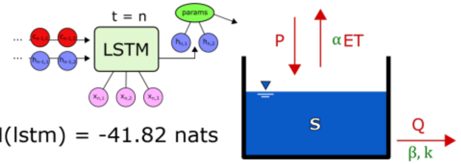
H(lstm) = -18.10 nats

$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$



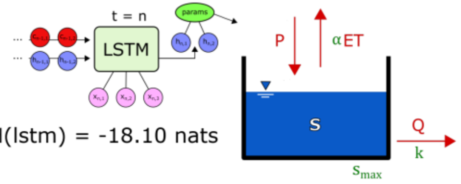
Synthetic examples (toy models)

Model 1: True model



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S^{\beta}}{k}$$

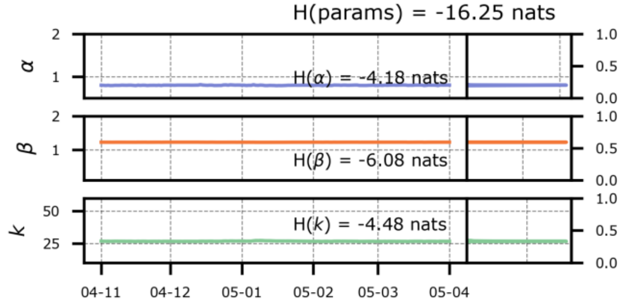
Model 2: Wrong process representation (correct architecture)

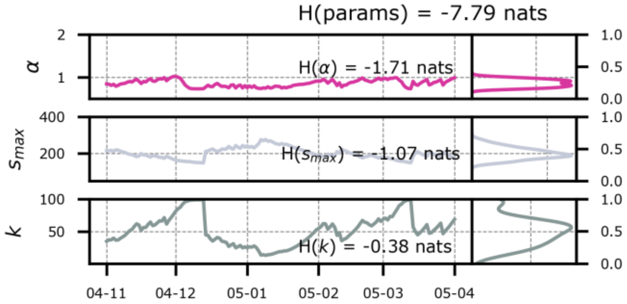


$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{S}{k} \qquad S \leq S_{max}$$

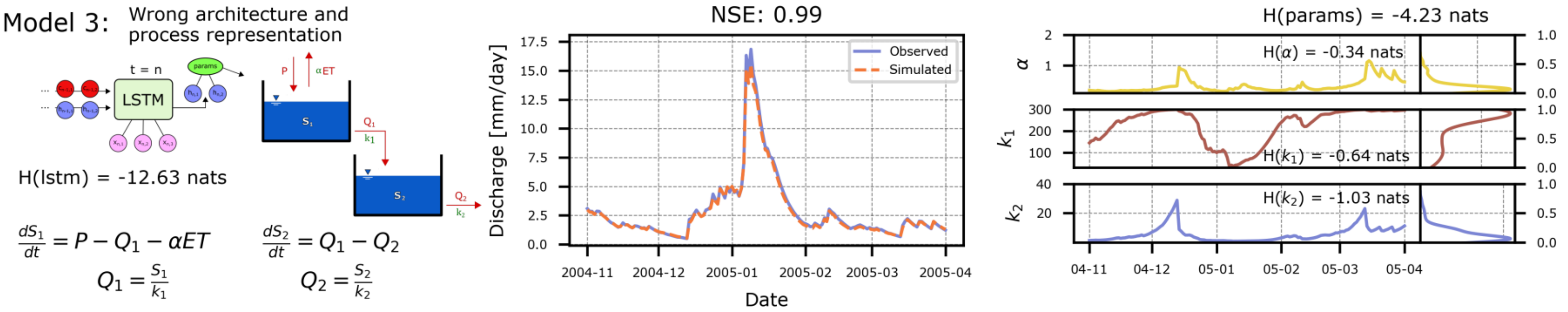
Don’t mind the negative numbers, this is just an artifact of calculating entropy for continuous variables.

The main result here is that the entropy of the predicted parameters and state of the neural network is much lower for Model 1, the “true” model, than Model 2, the “wrong” model.



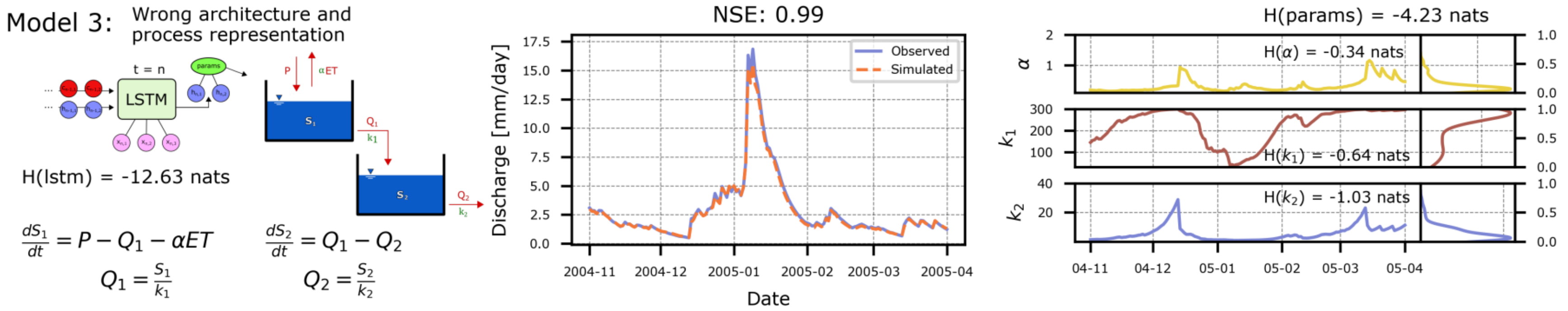


Synthetic examples (toy models)



And basically, this becomes our playground. Here’s a third model which again is “wrong” but is able to achieve perfect simulations. But we can measure how much the neural network compensates for it’s “wrongness”.

Synthetic examples (toy models)

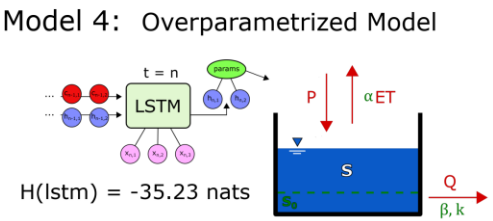


And basically, this becomes our playground. Here's a third model which again is "wrong" but is able to achieve perfect simulations. But we can measure how much the neural network compensates for it's "wrongness".

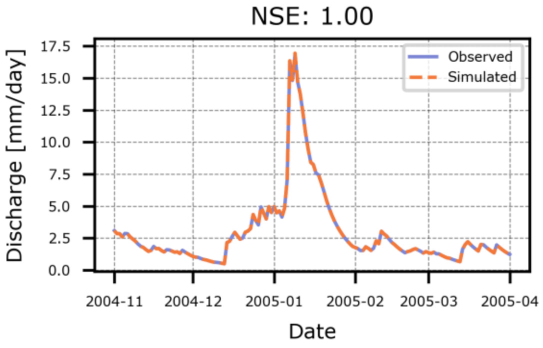
In this case the neural network has to compensate for more because this model is even more different than the "true" model when compared to Model 2.

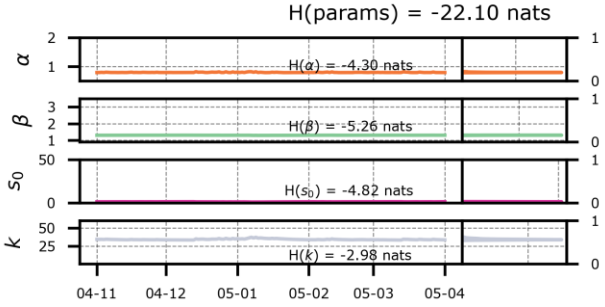
Synthetic examples (toy models)

To close our examples, here's the exception.



$$\frac{dS}{dt} = P - Q - \alpha ET$$
$$Q = \frac{(S - S_0)^{\beta}}{k}$$



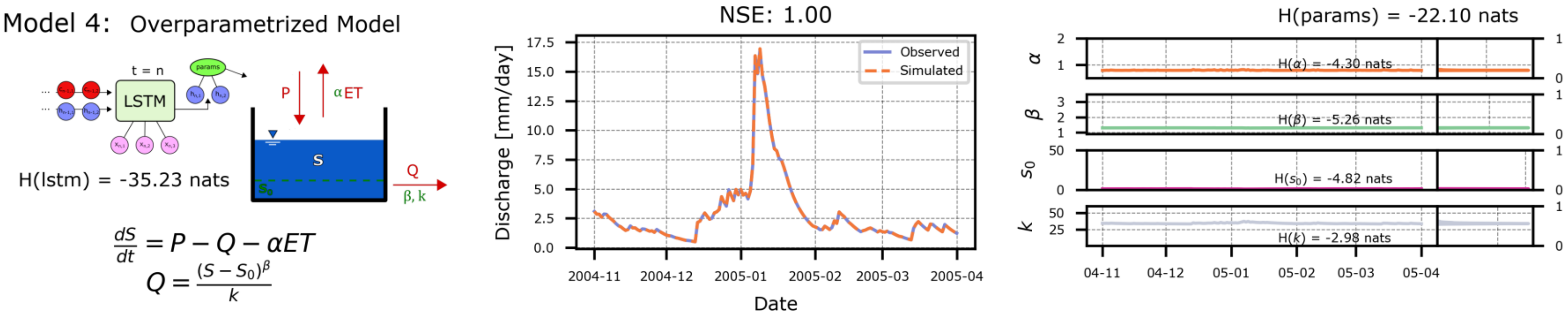


Synthetic examples (toy models)

To close our examples, here’s the exception.

This model has four tunable parameters instead of the three that the “true” model has. Moreover, if you take a look at the equations, if the neural network drives the value of S_0 to 0, suddenly this model becomes the “true” model. This results in our measurement of entropy in the parameters to gives a wrong conclusion.

Here -22.10 nats are less than the -16.25 nats that the true model had.

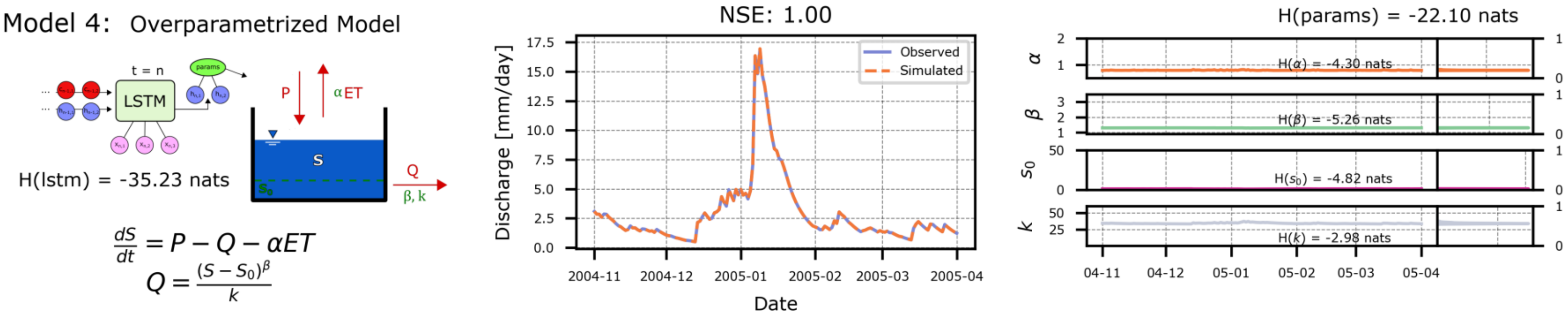


Synthetic examples (toy models)

But the true story is told by the entropy of the states in the neural network.

In this case we see that there is some additional activity in the neural network which reports a larger entropy than the -41.25 nats of the “true” model.

Ultimately this strengthens our proposed method because we can focus on measuring entropy only in one place, and it allows us to compare models with different numbers of tunable parameters.



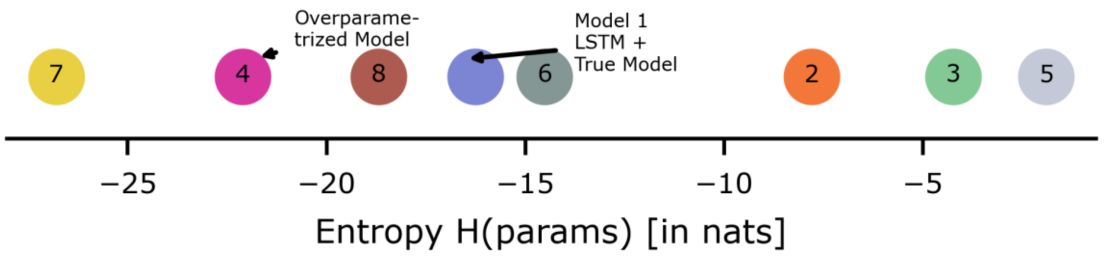
Synthetic examples (toy models)

Summary

To summarize our analysis we made an additional four examples, similar to those shown before, and put all their measurements of entropy in a single line.

By using the measurement of the parameters we get the arrangement shown below which is messy and doesn’t lead us to any insights because of the reasons explained before.

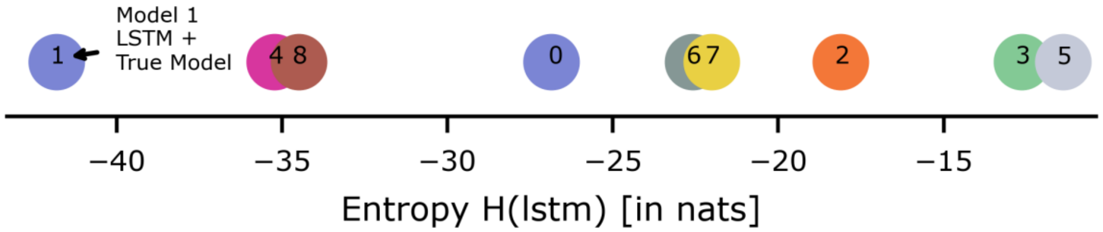
Instead….



Synthetic examples (toy models)

Summary

If we arrange them by the entropy of the neural network, we get this very logical sequence in which the “true” model has the least entropy and all other models follow based on how closely they match the structure of the “true” model.

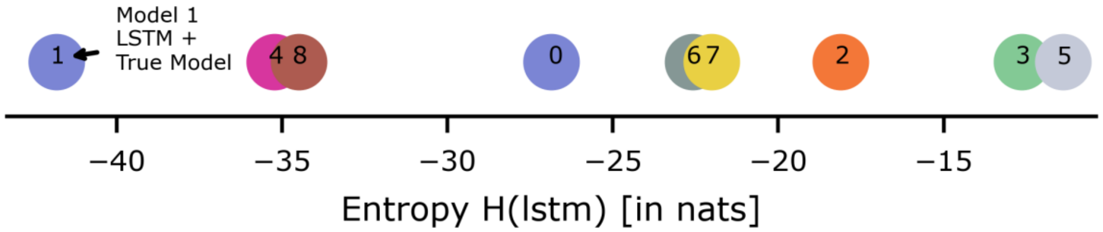


Synthetic examples (toy models)

Summary

If we arrange them by the entropy of the neural network, we get this very logical sequence in which the "true" model has the least entropy and all other models follow based on how closely they match the structure of the "true" model.

As we mentioned before, Model 4 has some semblance of the "true" model in its architecture so it sits very close to the "true" model in our arrangement.



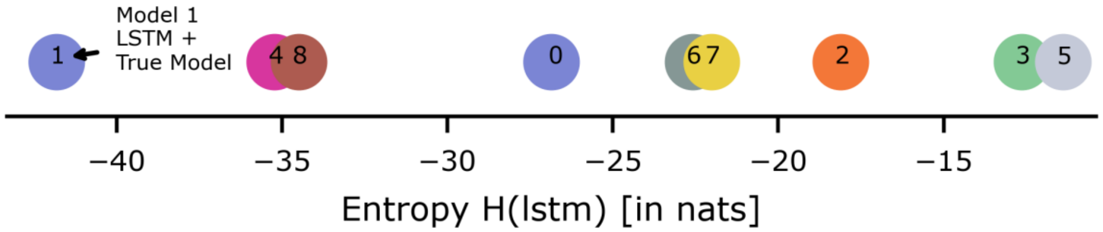
Synthetic examples (toy models)

Summary

If we arrange them by the entropy of the neural network, we get this very logical sequence in which the “true” model has the least entropy and all other models follow based on how closely they match the structure of the “true” model.

As we mentioned before, Model 4 has some semblance of the “true” model in its architecture so it sits very close to the “true” model in our arrangement.

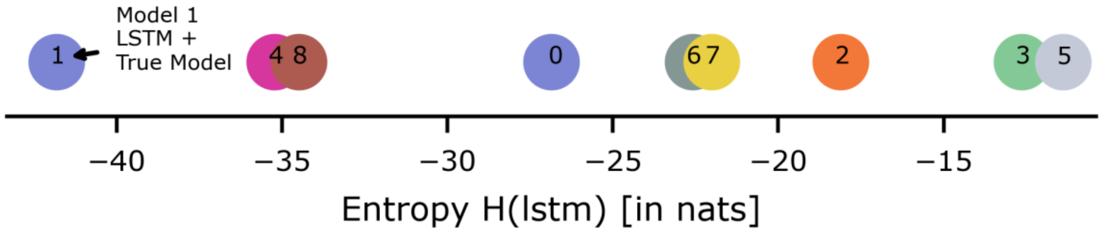
And Model 3 has a completely different architecture and set of equations and parameters, so it sits furthes away from the “true” model.



Synthetic examples (toy models)

Summary

Additionally, we have the benefit that we can put the neural network by itself in this axis as Model 0.

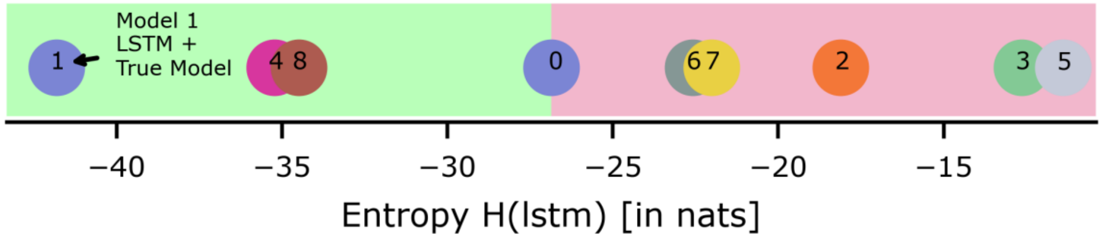


Synthetic examples (toy models)

Summary

Additionally, we have the benefit that we can put the neural network by itself in this axis as Model 0.

As you can see, the neural network creates a divide between models. Establishing the neural network by itself as a baseline, we can use this divide to distinguish between cases in which the added “physical component” of the model indeed helped or cases in which the added “physics” made the task of the neural network more difficult.



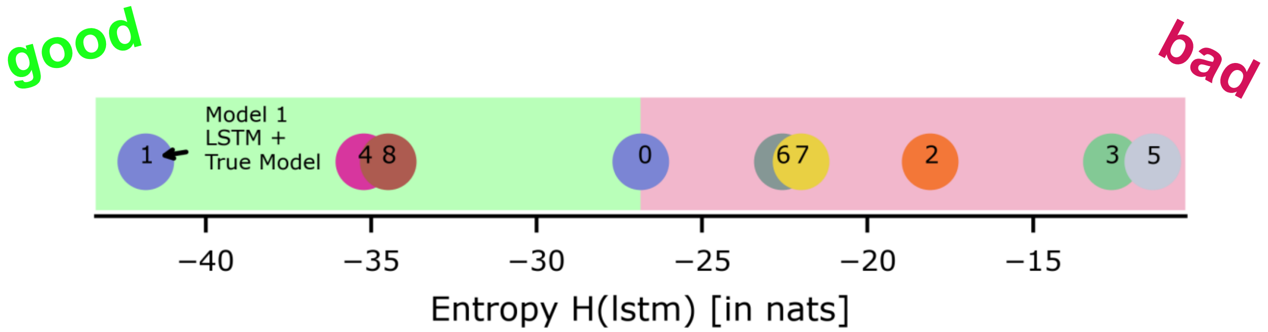
Synthetic examples (toy models)

Summary

Additionally, we have the benefit that we can put the neural network by itself in this axis as Model 0.

As you can see, the neural network creates a divide between models. Establishing the neural network by itself as a baseline, we can use this divide to distinguish between cases in which the added “physical component” of the model indeed helped or cases in which the added “physics” made the task of the neural network more difficult.

A dividing line between adding “good” or “bad” physics.



That was a lot, but hopefully you found that to be an exciting result :)

That was a lot, but hopefully you found that to be an exciting result :)

Of course, in real life there is not such thing as a “true” model. So what would happen in a case study using a large sample dataset?

That was a lot, but hopefully you found that to be an exciting result :)

Of course, in real life there is not such thing as a “true” model. So what would happen in a case study using a large sample dataset?

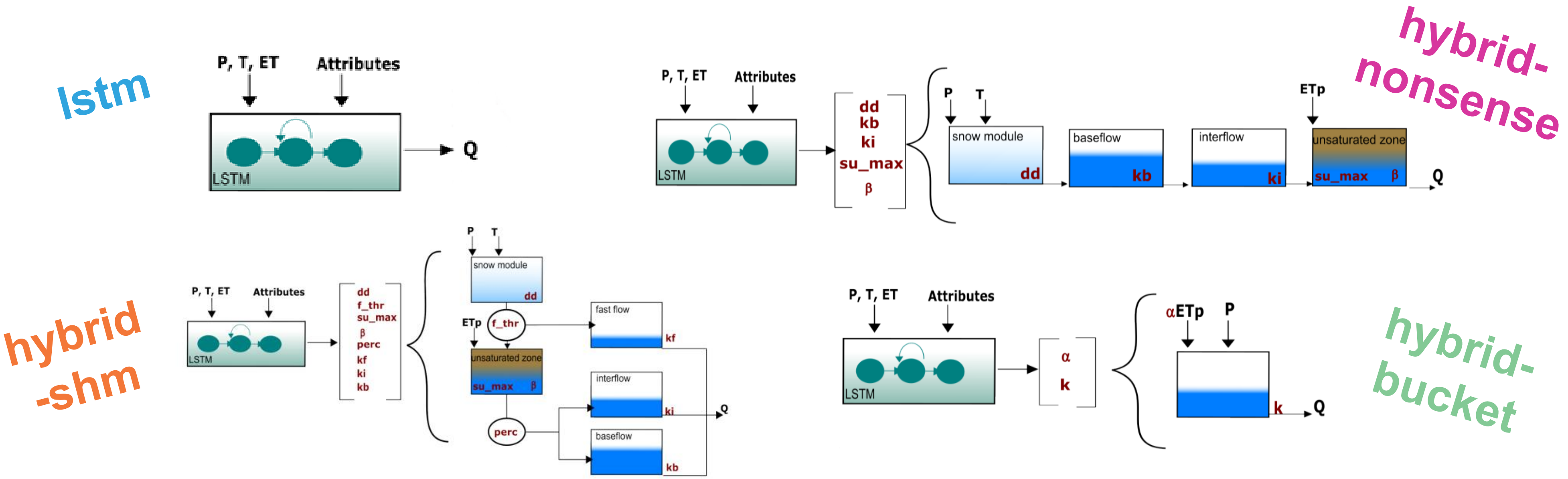
In the next slide, we show the four models that were used as part of a study in the CAMELS-GB dataset. One of these models is a purely data-driven LSTM and the other three are hybrid models with different physics-components.

In the highlighted paper we focused mostly in performance and interpretability. We concluded that if you focus in only performance, all of the proposed models were equivalent. Now, with our proposed approach, we can evaluate truly how much we are benefitting from adding this physics-based components.

Case Study: CAMELS-GB

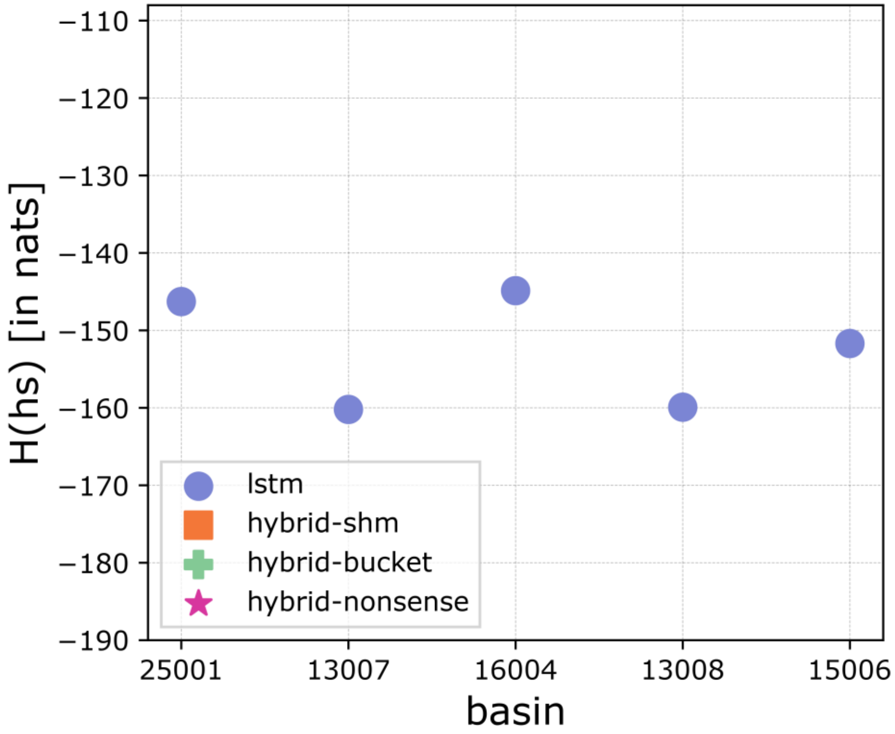
Setup

- Acuña Espinoza, E., Loritz, R., Álvarez Chaves, M., Bäuerle, N., & Ehret, U. (2024). **To bucket or not to bucket?** Analyzing the performance and interpretability of hybrid hydrological models with dynamic parameterization. *Hydrology and Earth System Sciences*, 28(12), 2705–2719. https://doi.org/10.5194/hess-28-2705-2024



Why we want to use hybrid models?

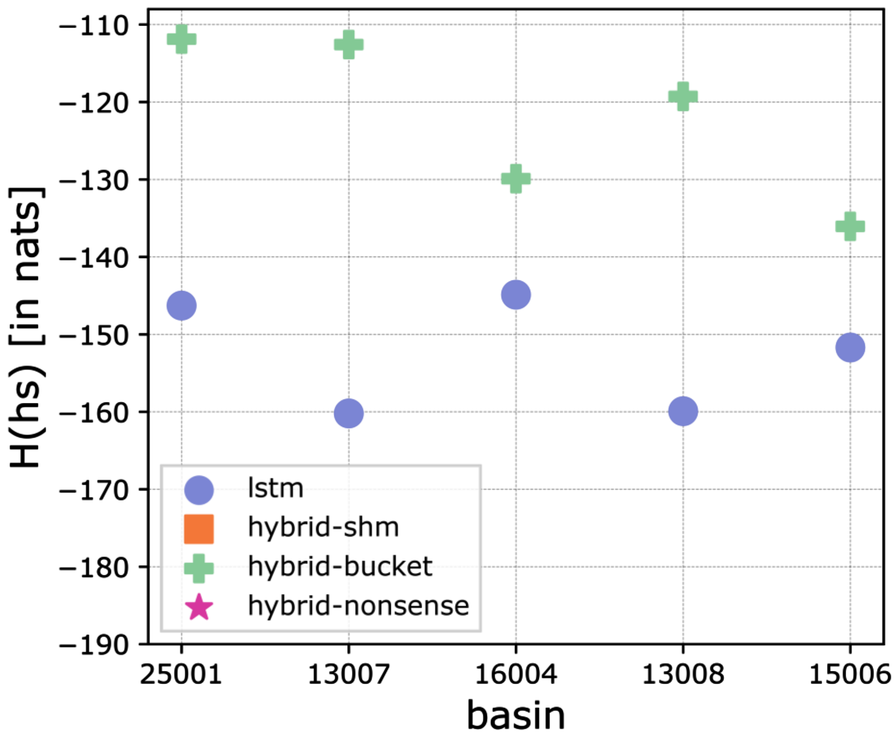
Analyzing the entropy of competing models



First, we start by showing the calculated entropy of the LSTM (neural network) in five catchments in the dataset. As explained before, this can be see as our baseline.

Why we want to use hybrid models?

Analyzing the entropy of competing models

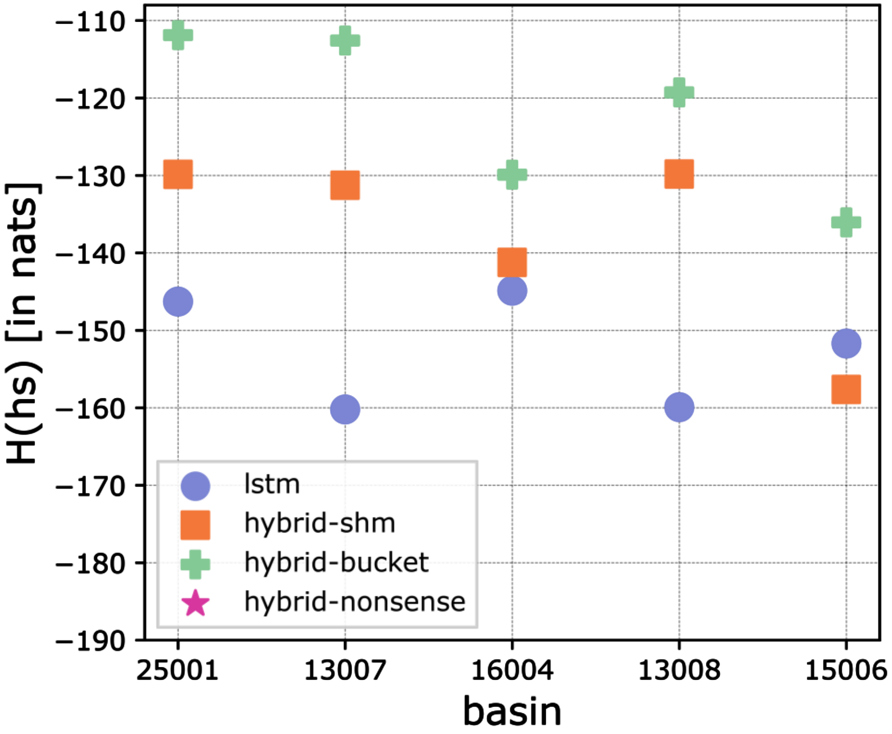


First, we start by showing the calculated entropy of the LSTM (neural network) in five catchments in the dataset. As explained before, this can be see as our baseline.

Now we add the “hybrid-bucket” model. As expected, the model doesn’t really benefit from adding something as simple as a bucket (single reservoir). Therefore entropy increases.

Why we want to use hybrid models?

Analyzing the entropy of competing models



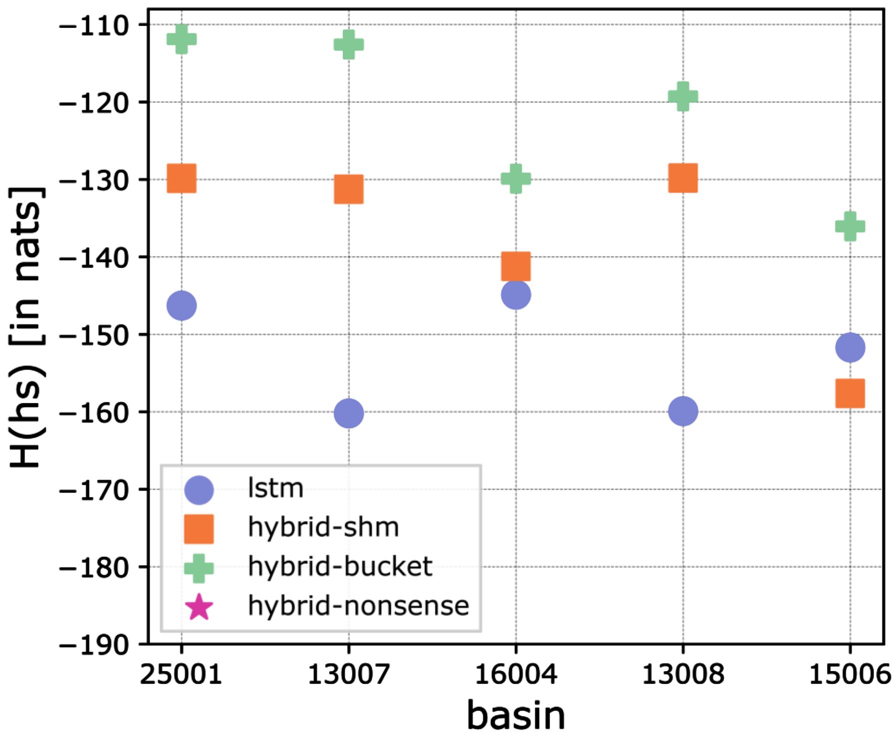
First, we start by showing the calculated entropy of the LSTM (neural network) in five catchments in the dataset. As explained before, this can be see as our baseline.

Now we add the "hybrid-bucket" model. As expected, the model doesn't really benefit from adding something as simple as a bucket (single reservoir). Therefore entropy increases.

Next we add "hybrid-shm". SHM is a proper hydrological model that we would use in practice, but we see that it's only in one case (catchment 15006) were it is providing any benefit.

Why we want to use hybrid models?

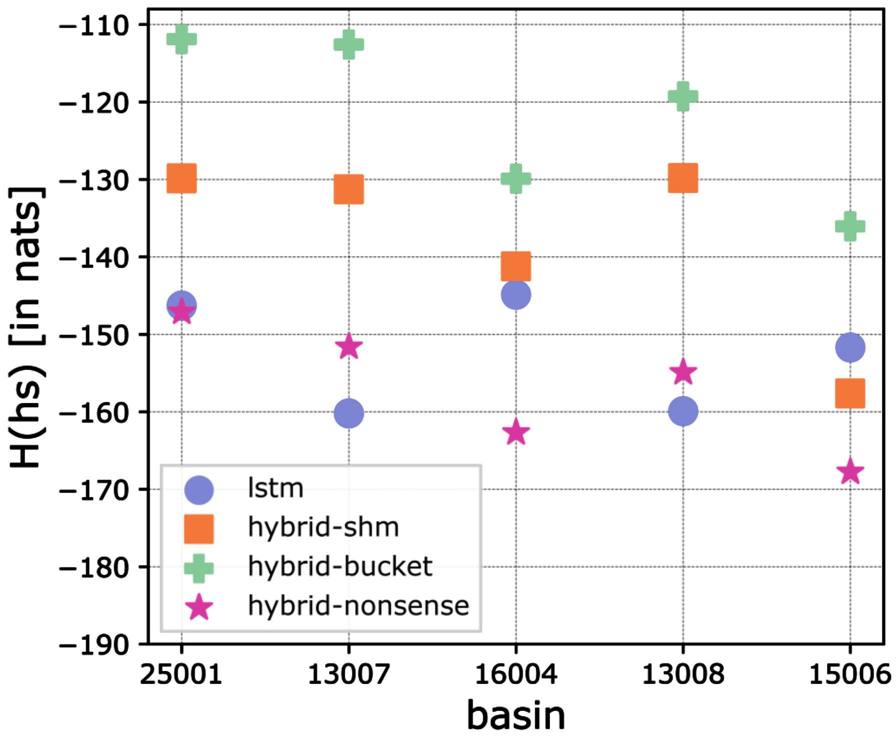
Analyzing the entropy of competing models



Finally we add the “hybrid-nonsense” model.

Why we want to use hybrid models?

Analyzing the entropy of competing models



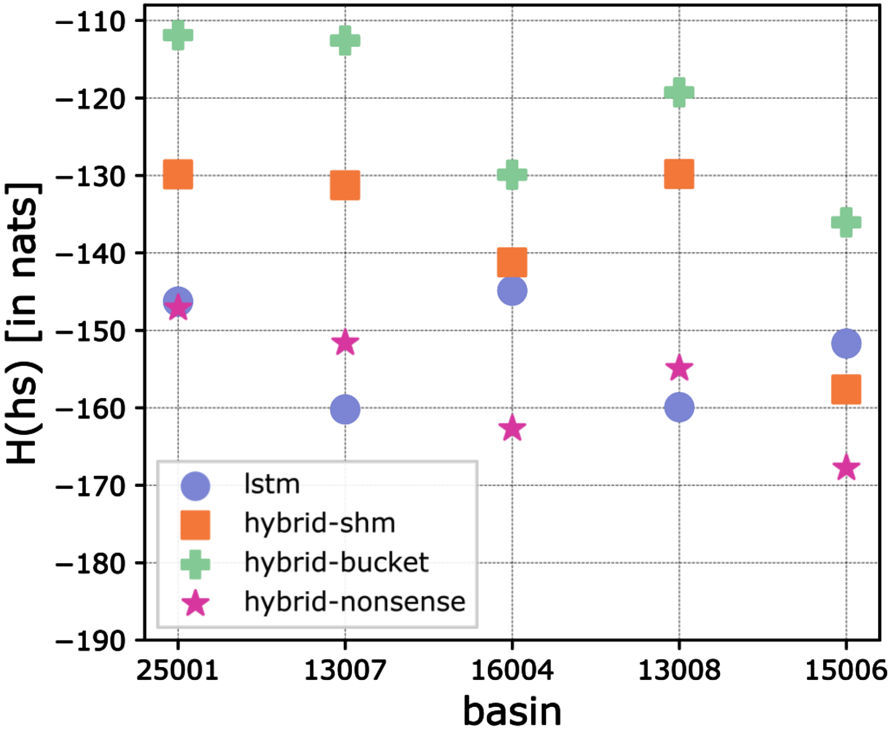
Finally we add the “hybrid-nonsense” model.

Surprisingly, we see that there are at least two cases were the “nonsense” model beats the baseline and in all of the shown cases it is better than SHM.

The reason for this is the “nonsese” is not exactly nonsense.

Why we want to use hybrid models?

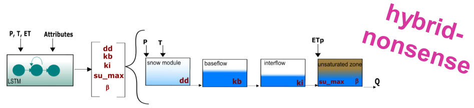
Analyzing the entropy of competing models



Finally we add the “hybrid-nonsense” model.

Surprisingly, we see that there are at least two cases were the “nonsense” model beats the baseline and in all of the shown cases it is better than SHM.

The reason for this is the “nonsese” is not exactly nonsense.

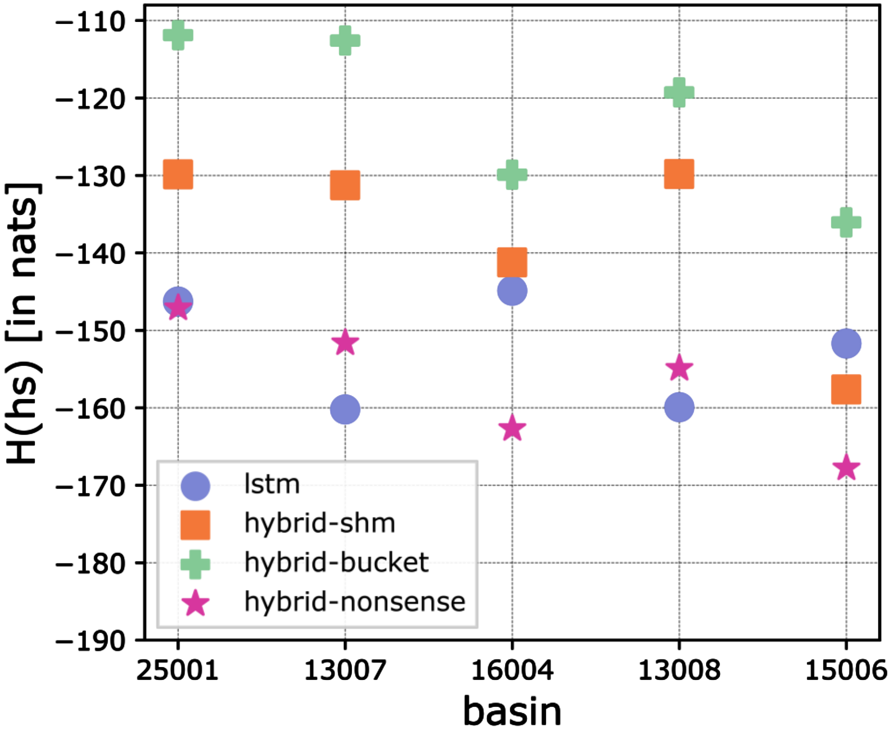


If you look at the labels the model is definitely nonsense but, ignoring the labels, the model is only a train of lagged reservoirs holding precipitation before releasing it as discharge.

In that way, it makes sense that nonsense is better than SHM because, in this dataset, the rainfall-runoff behavior is more complex than a bucket but less complex than something like SHM.

Why we want to use hybrid models?

Analyzing the entropy of competing models

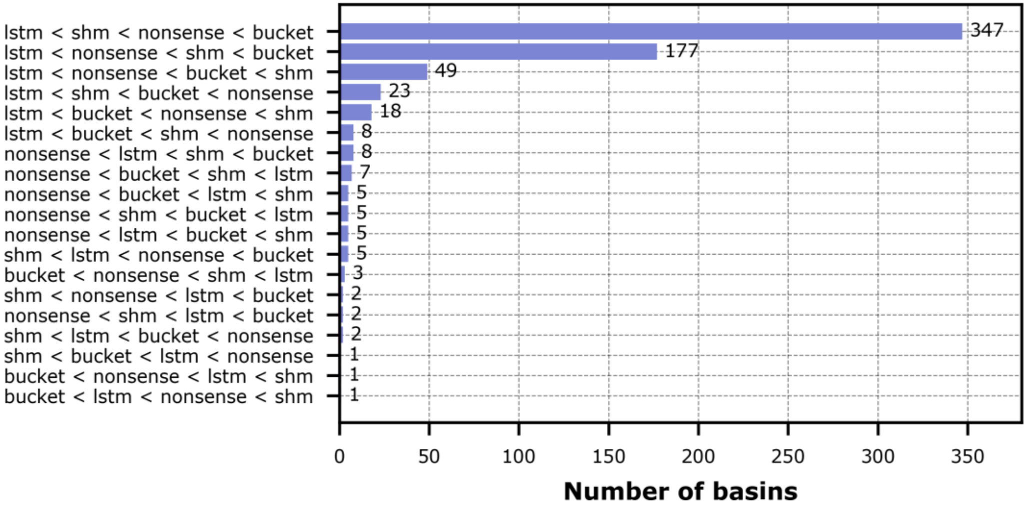


In that way, it's understandable that nonsense is better than SHM because, in this dataset, the rainfall-runoff processes are more complex than those described by a simple bucket but les complex than something like SHM.

Why we want to use hybrid models?

Analyzing the entropy of competing models

Taking a look at all catchments in the dataset, we come to the same conclusion.

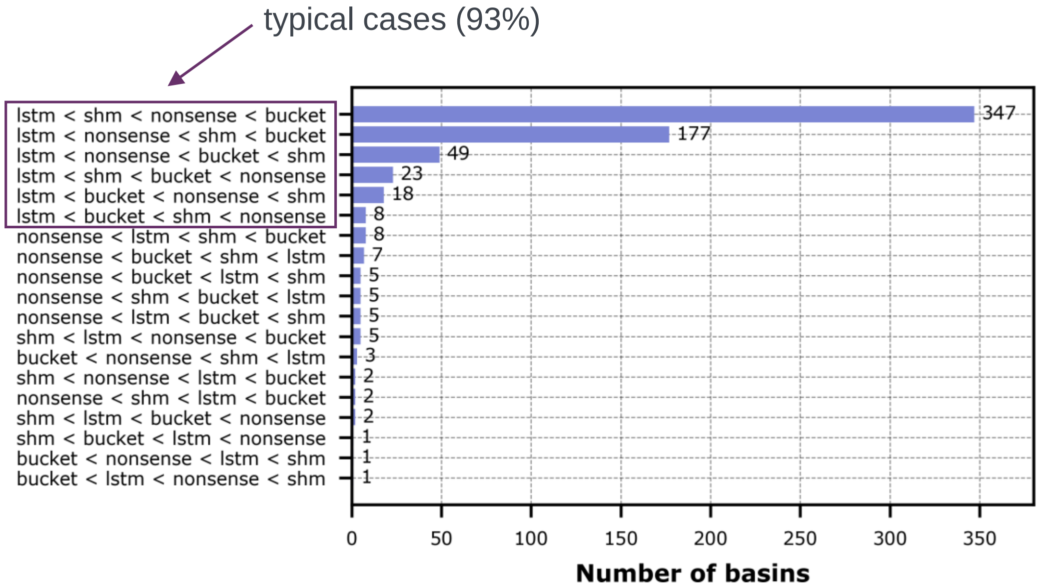


Why we want to use hybrid models?

Analyzing the entropy of competing models

Taking a look at all catchments in the dataset, we come to the same conclusion.

In 93% of the catchments in the dataset, adding a physical-component to our model provided no benefits when compared to a data-driven baseline.



Why we want to use hybrid models?

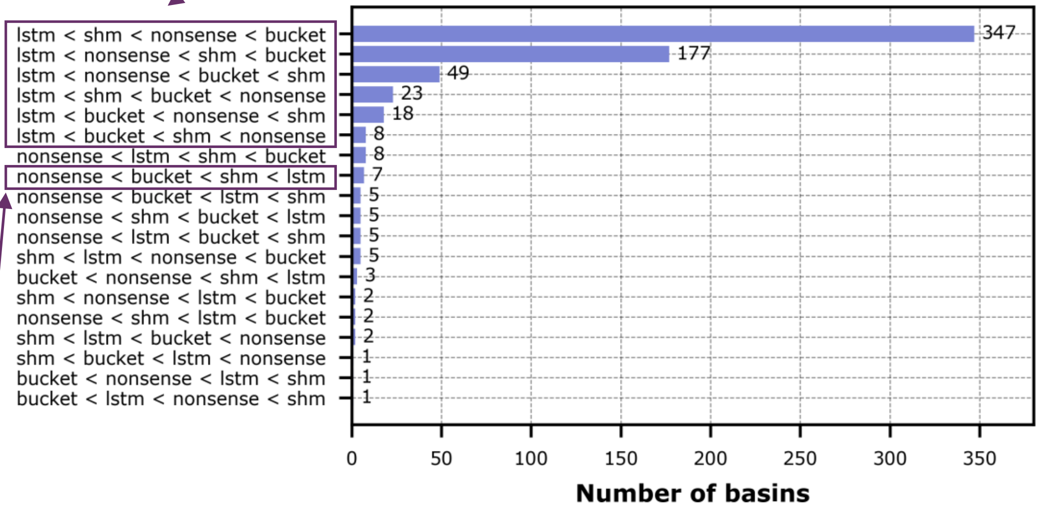
Analyzing the entropy of competing models

Taking a look at all catchments in the dataset, we come to the same conclusion.

In 93% of the catchments in the dataset, adding a physical-component to our model provided no benefits when compared to a data-driven baseline.

In at least 1% of the catchments, adding physics indeed helped.

typical cases (93%)



“physics help” (1%)

Why we want to use hybrid models?

Analyzing the entropy of competing models

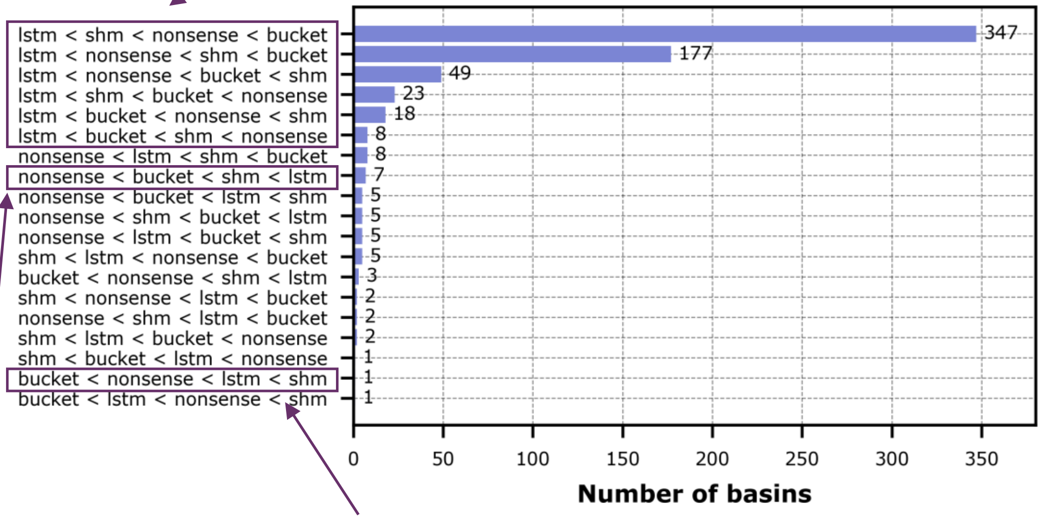
Taking a look at all catchments in the dataset, we come to the same conclusion.

In 93% of the catchments in the dataset, adding a physical-component to our model provided no benefits when compared to a data-driven baseline.

In at least 1% of the catchments, adding physics indeed helped.

But there are cases where the physics that helped are not exactly the ones you think. For example, here the model the you would use in practice is below the baseline and two “toy models” are above.

typical cases (93%)



“physics help” (1%)

Why we want to use hybrid models?

Analyzing the entropy of competing models

Taking a look at all catchments in the dataset, we come to the same conclusion.

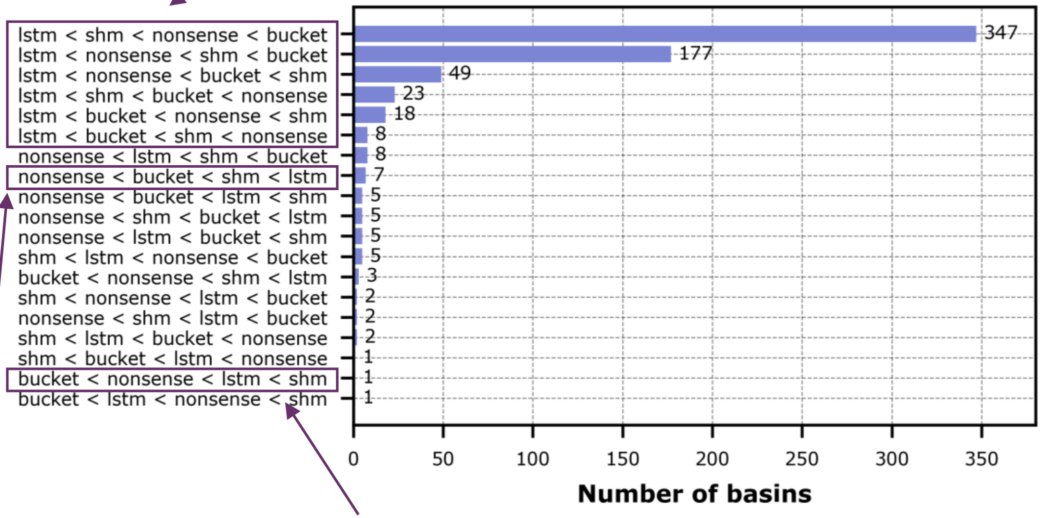
In 93% of the catchments in the dataset, adding a physical-component to our model provided no benefits when compared to a data-driven baseline.

In at least 1% of the catchments, adding physics indeed helped.

But there are cases where the physics that helped are not exactly the ones you think. For example, here the model the you would use in practice is below the baseline and two “toy models” are above.

Maybe in this cases you could...

typical cases (93%)



“physics help” (1%)

Why we want to use hybrid models?

Analyzing the entropy of competing models

Taking a look at all catchments in the dataset, we come to the same conclusion.

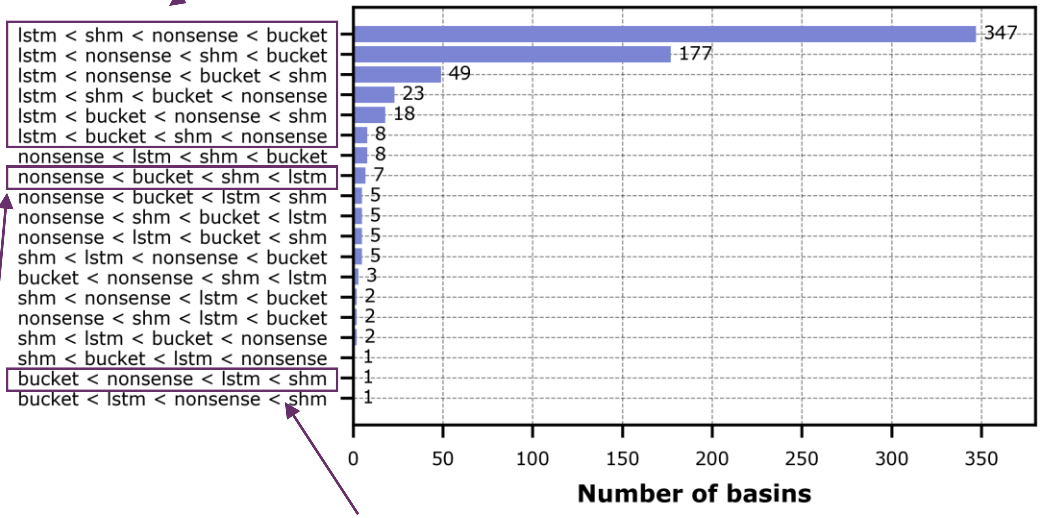
In 93% of the catchments in the dataset, adding a physical-component to our model provided no benefits when compared to a data-driven baseline.

In at least 1% of the catchments, adding physics indeed helped.

But there are cases where the physics that helped are not exactly the ones you think. For example, here the model the you would use in practice is below the baseline and two “toy models” are above.

Maybe in this cases you could...

typical cases (93%)



“physics help” (1%)

“skip the hydrologist” (0.2%)

Conclusions

Some last words

- Depending on the approach, physical constraints in hybrid models are ignored.
- How much of the performance of a hybrid model comes from the data-driven or the physics-based component can be quantified.
- **Information Theory** provides a powerful tool for analysing models across the spectrum of physics-based, data-driven and anything in between.

Thank you!

Manuel Álvarez Chaves

email: manuel.alvarez-chaves@simtech.uni-stuttgart.de

phone: +49 (0) 711 685-60130

Research Group on Statistical Model-Data Integration (SMDI)

SC SimTech

University of Stuttgart

Universitätsstraße 32, 227a, Stuttgart, Germany