Responses to comments by referee #1 (Elena Toth)

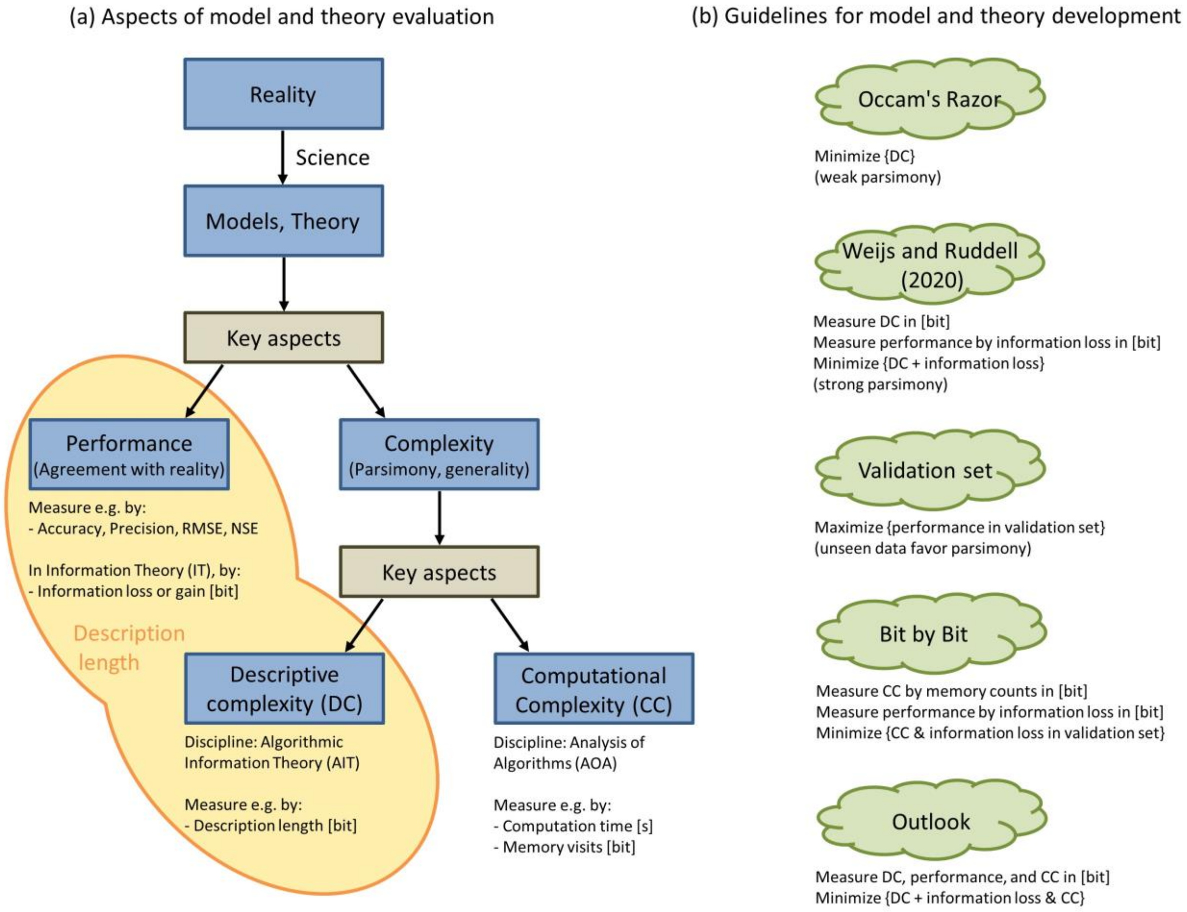
Dear Editor, dear Elena Toth,

We thank the first referee, Elena Toth, for the constructive review of our manuscript and the detailed comments, which will help us to sharpen our arguments. We will in the following reply to the comments point by point. The Referee comments are in blue.

Comment 1: The Technical Note addresses a very timely research question, covered also in a recent WRR debate on the role of Information Theory for helping to understand the complexity of Earth Systems. In particular, the Note attempts to provide some examples for testing/sustaining some of the ideas presented in the WRR debate, that includes also one contribution by two of the Authors (Weijs and Ruddell, 2020). I find the theme extremely interesting and I believe that proposing a way (using 'Strace') to calculate computational effort without the need to refer to the specific machine where the computations run is indeed brilliant and novel and worthy to be published. On the other hand, I confess that on one hand I find the Note a bit too theoretical and on the other hand I have some doubts on the soundness of the comparison that is presented. The note often refers to the Weijs and Ruddell (2020) paper but without well clarifying the relationship between such paper and the present work (what is the content of the previous paper and what is different here). A good part of the theoretical discussion, indeed a truly philosophical one, as such was the 'line' of the WRR debate, is repeated here, in a first part (three pages of introduction) that is definitely too long and too much theoretical for a technical note and may be substantially shortened, referring to the previous publication.

Reply 1: Thank you for this comment, which made us aware that we have to i) maker better clear the contribution of the paper, and ii) explain its connection to Weijs and Ruddell (2020).

The contribution of the paper is twofold: Firstly we make a practical suggestion on how to measure computational effort of computer-based models. Secondly, we embed this suggestion into the larger and more theoretical topic of model and theory building and related guiding principles such as Occam's Razor or the approach suggested by Weijs and Ruddell (2020). Apparently we haven't clearly explained the relation of the 'bit by bit' concept and the other concepts, and we also have not included a proper description of the classical 'validation set' approach and its relation to the other approaches. We suggest adding a figure and related discussion to the manuscript (see below) for clarification.


(a) Aspects of model and theory evaluation
Reality
Science
Models, Theory
Key aspects
Performance
(Agreement with reality)
Measure e.g. by:
- Accuracy, Precision, RMSE, NSE
In Information Theory (IT), by:
- Information loss or gain [bit]
Description length
Complexity
(Parsimony, generality)
Key aspects
Descriptive complexity (DC)
Discipline: Algorithmic Information Theory (AIT)
Measure e.g. by:
- Description length [bit]
Computational Complexity (CC)
Discipline: Analysis of Algorithms (AOA)
Measure e.g. by:
- Computation time [s]
- Memory visits [bit]
(b) Guidelines for model and theory development
Occam's Razor
Minimize {DC}
(weak parsimony)
Weijs and Ruddell (2020)
Measure DC in [bit]
Measure performance by information loss in [bit]
Minimize {DC + information loss}
(strong parsimony)
Validation set
Maximize {performance in validation set}
(unseen data favor parsimony)
Bit by Bit
Measure CC by memory counts in [bit]
Measure performance by information loss in [bit]
Minimize {CC & information loss in validation set}
Outlook
Measure DC, performance, and CC in [bit]
Minimize {DC + information loss & CC}

In short, we will discuss:

- Weijs and Ruddell (2020) extend Occam's Razor – which evaluates descriptive complexity only – by additionally measuring performance. Also, they measure both criteria in a unit [bit] and using a unified mathematics which allows a joint treatment and comparison between the two, including their inherent tension and complementarity.
- Classical validation set approaches seek to maximize model performance on unseen data. By using unseen data, general/parsimonious model approaches are favored, and overfitting is avoided. The approach by Weijs and Ruddell (2020) assures generality/parsimony by including descriptive complexity into their maximization function. For the latter we will include references to relevant literature from Algorithmic Information Theory.
- The bit by bit approach adds another key dimension of model complexity: computational complexity, measured by 'strace' in unit [bit]. It is important to note that computational complexity is neither a surrogate for measuring performance, nor for measuring descriptive complexity. In order to provide a comprehensive evaluation of a model, computational complexity can be either combined with the approach by Weijs and Ruddell (2020), OR with a validation set approach. In both cases, there is a guard for overfitting (descriptive complexity in the first, evaluation on unseen data in the second). In the current version of the

manuscript, we combined computational complexity with performance on a data set seen through calibration - as in Weijs and Ruddell (2020) - but did not add a descriptive complexity control. This is inconsistent with the above explanation. We therefore suggest that in a revised version of the paper, we will use computational complexity in combination with performance measured by information loss in a validation set approach. This is in accordance with the referee's suggestion, and readers from the hydrological community will find this easier to relate to as it follows established procedures of cal/val.

- We will further describe an approach - and suggest it for future research - where computational complexity is combined with the approach by Weijs and Ruddell (2020). This avoids the need for splitting available data into cal and val subsets, but still favors general/parsimonious models. In addition, as all criteria can be measured in unit [bit], this allows a joint treatment and comparison of all key aspects of model evaluation.

We hope that by the figure and the explanation, the relation between the different guiding principles, and their relation to the scientific process of model/theory building/evaluation becomes more clear. We suggest keeping the existing in-depth introduction to the theory, as placing the bit by bit approach into this framework is one main goal of the paper.

Comment 2: But my main concern is that the comparison of the models is not fair, since they do not make use of the same information and this is instead crucial in a work focussing on information theory. Looking at Table 1, last column, we may see that the data used for running the models are not the same. In fact, the bucket models (Models 02 and 05) do not use any streamflow data in any way for the simulation but only for calibrating the parameters. The same holds for the ANN (Model-08) since only P is provided in input. On the other hand, the autoregressive model (Model-07) uses only past streamflow values as input. It is well known that for a short lead time (the models are here used as simulation models, with lead-time equal to one), the recent measures of the streamflow (Q) is much more informative than the rainfall values, that in real-time flow forecasting become more and more important when the lead-time increases, since Q encapsulates a lot of useful information on the catchment behaviour, and it may be seen as a very good approximation of the catchment conditions. Therefore it was easily predictable that the autoregressive model (Model-07) would have outperformed the other models, independently of its complexity, due to the different input information they use. Thus, leaving aside the analysis of the performances (expected, due to the setting up of the models), the interesting part of the results is the analysis of the complexity. Section 3.2 and Figure 3 show that, a part from Model-03 and Model-08 (ANN), all the other complexities are very close. The reason for the high computational complexity of Model-03 is the excessively (and not necessary) fine time step. The reason for the high computational complexity of Model-08, that is the Artificial Neural Network, may certainly be inherent in the structure of these kind of models, that tend to have a relatively high number of parameters (but the internal parameters are in some cases not all influential and since, despite the high number of parameters, ANNs generally work very well on independent data, they cannot be blamed of overfitting/overtraining). But in addition, in this case the ANN model is not only fed by the "wrong" input (P instead of Q), but its architecture is also certainly more complex than needed: why using 10 hidden nodes? If it were used, as it should, in a way that is consistent with the regressive model, it should be fed by the last streamflow values rather than (or in addition to) some past rainfall (needed especially if considering longer lead-times) and it would perform much better than now. And probably a few hidden nodes would be more than enough (as proved in many previous works where such models rival with more complex conceptual models in forecasting/updating mode), so its complexity would be less. Due to the potential of the bit-by-bit concept, and the utility to be able to measure computational complexity through 'Strace', I do encourage the Authors to perform and present a more fair comparison and then focussing and explaining the differences, in performances and complexity, found in models that use indeed the same input information content (and have the most parsimonious structure that is possible).

Reply 2: Thank you for this comment, which indicates that we need to better explain the take-home messages from the model comparison. The two key messages are that

- computational complexity as measured by 'strace' is sensitive to ALL computational aspects of a model: the amount of required forcing data, the size of the model itself, its time stepping, spatial resolution, numerical scheme, etc. ,
- 'strace' can be applied to ANY computer-based model, i.e. it is general within its range of application (evaluation of computer-based models).

The main goal of the model comparison is therefore to demonstrate is applicability across a range of different model concepts (therefore we included bucket models, autoregressive models, ANN models), and its sensitivity to different computational aspects (therefore we included variations of a bucket model in terms of time stepping). It is therefore neither required nor useful to establish models that all rely on the same input data, because the bit by bit approach might be used for exactly the purpose of comparing two models relying on different input data. In the light of the two key take-home messages mentioned above, we hope it also becomes clear that the actual performance of the models, and their performance comparison is not central to this paper, and that they might even distract from the main messages. Nevertheless we see the referee's point that the ANN used in the study was set up in a far-from-optimal manner, which obviously appears as an unfair treatment compared to the other models.

We therefore suggest to include into a revised version of the manuscript:

- A better explanation of the purpose and key messages of the model comparison
- Replacing the current ANN by a more appropriate one with less nodes

- As explained in reply 1, we will calibrate the models in a calibration subset of the data, and present results for a validation subset.

Specific comments

Comment 3: Abstract: ll.12-20: may be summarised.

Reply 3: Respectfully we disagree. A brief introduction of the main terms in the abstract helps to put the 'bit by bit' concept into perspective, and helps conveying the two main goals of the manuscript (see our reply to comment 1). We will however re-read and potentially re-write the abstract in the light of the added figure and changed experimental design (see our reply to comment 1).

Comment 4: Pages 2 to 4 may be summarised in one page, referring to Wejis and Ruddell (2020) for the philosophical discussion.

Reply 4: Please see our reply to comment 3.

Comment 5: Eq. 1: I would suggest to move eq 1 inside Table 1 (Model-07 row)

Reply 5: Agreed. We will do so in a revised version of the manuscript.

Comment 6: ll. 155-158: actually I would have found very interesting an evaluation of out-of-sample performance of the models, since this is indeed crucial for data-driven models and it would be very useful to understand what each model is able to infer on the behavior of the basin on independent data, to analyse their generalisation ability.

Reply 6: In a revised version of the manuscript, we will fit the models in a calibration period, and present results for a validation period. Please see also our related reply to comment 1.

Comment 7: Second part of Section 2.4.1: I think that more detail on the meaning and computation of entropy is necessary, since it is a ’niche’ not widely known to the readers.

Reply 7: Agreed. We will add the entropy equation, a brief explanation, and references to relevant literature to a revised version of the manuscript.

Comment 8: ll. 266-269: can you explain the differences in computational complexity between Models 00 and 01? I would have expected their complexity to be practically null for both, since they do not need to make computations at each step. . .

Reply 8: The computational complexity of these model arises from preparing the output, and actually writing the output. The difference between the two is that for the first, a single number is written into the output array, for the second the observed timeseries is read, and then written into the output array. The related Matlab code is:

Model_00	Model_01

`% assign the mean flow`
`q_host_mean = 4.5998;`
 	
`% make the prediction for each time step`
`output_00 = zeros(87650,1) + q_host_mean;`
	`% load the observed Q data`
	`load q_host`
	
	`% copy the input ('q_host') to the output ('output')`
	`output_01 = q_host;`

So the effort is really low, but nevertheless an array of double-precision numbers needs to be copied. Interestingly, as the results show, the related effort is higher than for model_04, even if the latter has more elaborate code. The reason is that the latter acts on integer-precision variables.

Code of model_04:

```
% load the input data
load p_ebni_int

% get parameters
len = length(p_ebni_int); % length of the data set

% hydrological model setup
K = 55;                         % retention constant = mean transit time [h]
qsim = int8(zeros(len,1));      % reservoir discharge [mm/h]
S = int8(0);                    % initialize the reservoir fill level [mm]

% loop over time
for t = 2 : len
    S = S + p_ebni_int(t);              % storage change due to rainfall input
    qsim(t) = S / K;    % discharge as f(storage volume)
    S = S - qsim(t);                % storage change due to discharge
end

% convert the discharge from [mm/h] into [m³/s]
output_04 = qsim * 31.8888888;
```

Yours sincerely,

Uwe Ehret, on behalf of all co-authors