Benchmarking Transformers on Spatio-Temporal River Water Temperature Modeling: Implementation and Reproducibility Notes*

Linlin Jia[1][0000−0002−3834−1498], Benjamin Fankhauser[1][0000−0002−7982−2669], Vidushi Bigler[2][0000−0001−6043−8264], and Kaspar Riesen[1][0000−0002−9145−3157]

[1] Institute of Computer Science, University of Bern, Bern, Switzerland
linlin.jia@unibe.ch
[2] Institute for Optimisation and Data Analysis, Bern University of Applied Sciences, Biel, Switzerland

Abstract. This paper presents the implementation and reproducibility notes for the ICPR 2026 paper *Benchmarking Transformers on Spatio-Temporal River Water Temperature Modeling*. We release `swiss-river-network-benchmark`, an open-source Python package that bundles three river-graph datasets, eight reference model architectures, and the complete model selection pipeline behind a typed `srn` command-line interface. We also release an interactive workbench that allows hydrologists to explore the data, compare models, bring their own data or model, and analyze predictions without writing code.

Keywords: Reproducible research · Benchmark · Transformers · Graph neural networks · Hydrology · Time-series forecasting.

1 Introduction

River water temperature governs aquatic ecology, water management, and climate-impact assessment [1,2]. In the ICPR 2026 paper [3], the authors provided the first systematic comparison of Transformer [4] versus LSTM [5] models for this spatio-temporal task for both nowcasting and multi-horizon forecasting. This companion paper describes the open-source library we release, namely `swiss-river-network-benchmark`, to bridge the gap between such scientific machine learning studies and the hydrologists who could benefit from them. The library spans eight model architectures and three datasets, the same as in paper [3]. Building directly on that library, we then develop an interactive workbench (Fig. 3) that turns the repository into a point-and-click tool a hydrologist can use without writing a single line of code, so that they may explore the river network on a map, compare the benchmarked models, bring their own data or their own trained model, run and stream predictions, and analyze the results.

* Supported by Swiss National Science Foundation (SNSF) Grant Nr. PT00P2_206252. Data are kindly provided by *the Federal Office for the Environment, MeteoSwiss, and Canton of Zurich.*

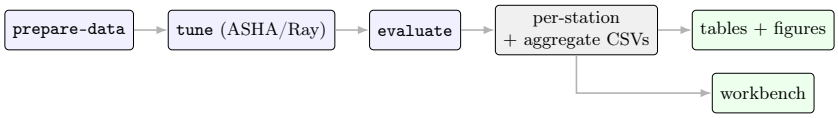


Fig. 1. The library pipeline. Four `srn` commands turn raw data into the committed CSVs, from which every table, figure, and the interactive workbench are regenerated on a CPU; only `tune` and `evaluate` need a GPU.

The remainder of the paper is organized as follows: Section 2 presents the library, including its pipeline, dataset processing, implementation, selection and evaluation of the models. Section 3 then describes the interactive workbench in detail. Section 4 reports our own evaluation of the library, covering a multi-aspect comparison of the models and the sensitivity of the results to the main design choices Finally, Section 5 concludes the paper and discusses future work. All of the material is released under the MIT license, with the code available at https://github.com/jajupmochi/swiss-river-network-benchmark and a hosted interactive demo of the workbench on Hugging Face Spaces at https://huggingface.co/spaces/jajupmochi/swiss-river-network-benchmark.

2 The `swiss-river-network-benchmark` library

The library packages the datasets, models and the full experimental pipeline behind a single installable package, `swissrivernetwork`, and a typed `srn` command-line interface, with an auto-generated API reference. It is detailed in the following sections.

2.1 Framework and pipeline

The library pipeline chains four commands, as shown in Fig. 1. First, `prepare-data` builds the three datasets. Then, `tune` runs an ASHA [6] hyper-parameter search on Ray Tune [7] per model-dataset pair. After that, `evaluate` scores the tuned checkpoints, and `sweep` finally varies the history-window length. Every stage writes its artifacts to disk, in the form of graphs, checkpoints, and per-station and aggregate CSVs, so that a later stage or an external reader can resume from the committed outputs without repeating the expensive ones. Every public module, class and function carries a docstring, from which a browsable API reference is generated at build time with `mkdocstrings` and published on the documentation site. That means the library can also be driven programmatically, not only from the command line.

2.2 Dataset processing

The library accepts any river network described by a set of monitoring stations, an aligned air- and water-temperature time series for each station, and a directed

Table 1. Statistics of the three released datasets following [3].

Dataset	# Stations	# Years	Training Range (Y)	Testing Range (Y)	km^2/Station	Missing Water Temp.
Swiss-1990	28	30	1990-2012	2013-2020	1596	No
Swiss-2010	63	10	2010-2017	2018-2020	1039	Yes
Zurich	15	13	2009-2019	2020-2022	74	Yes

Table 2. Feature matrix of the four model architectures, each instantiated with an LSTM and a Transformer backbone.

Class	Station Emb.	River Graph	Mechanism
Isolated (baseline)	✗	✗	none (one model per station)
Embedded	✓	✗	learnable station identity
Graphlet	✗	✓	neighbours' predictions concatenated to the input
ST-GNN	✓	✓	message passing on the river graph

edge list that follows the downstream flow. Its `data_preparation` module turns such raw records into the released artifacts. It builds the river network as a directed graph, aligns the two temperature series, drops short or empty sequences, and writes fixed train, validation and test splits. At training time, the `dataset` module reads these files and exposes windowed datasets in the `torch` format, where a sliding history window of air temperature is the input and the water temperature is the target. The missing ground truth is masked wherever a station has gaps.

We use three such datasets as the benchmark in this work, which are shown in Table 1. They differ in extent, length and completeness. `swiss-1990` spans 28 stations over three decades with no missing ground truth, `swiss-2010` covers more of the country with 63 stations but has gaps, and `zurich` is a smaller, denser urban network. In every case, the stations form a directed river graph whose edges follow the downstream flow, and air temperature is the only input signal. All three datasets, together with the trained models, ship with the library and can be explored interactively in the workbench as detailed in Section 3.

2.3 Implementation of the models under study

For K stations, each station s_k carries an air-temperature input series $A_k^{(t)} = (a_k^{(1)}, \ldots, a_k^{(t)})$, and the task is to predict the water temperature $\hat{w}_k^{(t')}$ through a learned model $\mathcal{M}_k$ and a linear head $\mathcal{L}_k$,

$$\hat{w}_k^{(t')} = \mathcal{L}_k\left(\mathcal{M}_k\left(A_k^{(t)}\right)\right), \tag{1}$$

where $t' = t$ is nowcasting and $t' > t$ forecasting. The four model architectures shown in Table 2 are each instantiated with an LSTM and a Transformer backbone, where the latter is equipped with *learnable*, *sinusoidal* or RoPE [8]

positional encodings. The *Embedded* and *ST-GNN* architectures share a single model across stations via learnable station embeddings, while *Graphlet* and *ST-GNN* encode river connections with graph structure.

2.4 Model selection and evaluation methods

The aforementioned models are selected and evaluated with a standard train, validation, and test protocol on the fixed splits as in Table 1:

- *Hyper-parameter search.* For each model-dataset pair, the hyper-parameters are chosen by a random search that is scheduled with ASHA [6] on Ray Tune [7] and driven by the validation loss. Across the whole benchmark, this amounts to 13,200 trained models. An RTX 4090 GPU was used for training, and an RTX 3070 for inference.
- *Model Selection.* Within each search, the best trial and its checkpoint are chosen by the validation mean-squared error alone, while the test period is kept untouched until the final evaluation.
- *Model Evaluation.* The selected checkpoint is then scored on the held-out test period with RMSE, MAE and the Nash–Sutcliffe efficiency, reported both per station and aggregated across the network.

Since every stage writes its artifacts to disk, the committed per-station and aggregate CSVs regenerate all of the result tables and figures on a personal CPU in seconds, reproducing the reported cells without a GPU or the trained checkpoints, and the same CSVs drive the interactive workbench, which will be detailed in Section 3.

3 An Interactive Workbench for Hydrologists

A central goal and the main practical benefit is to make the benchmark usable by hydrologists, not only machine learning researchers. The interactive *workbench* achieves this goal by turning a Python repository into a point-and-click tool, in the form of a Gradio app. It runs identically on the Hugging Face Space and locally via `srn app gradio`, over the released CSVs and the user's own inputs. A double-click launcher (`scripts/launch-workbench.*`) installs everything and opens the app for non-programmers. Fig. 2 illustrates its architecture and Fig. 3 demonstrates two snapshots of the workbench in use. We describe its main modules below:

Explore. The *Explore* module places the monitoring stations on a Swiss basemap together with the river-network graph, plots each station's air and water series and their day-of-year climatology, shows a data-coverage heatmap, and renders the network in 3D, so that data quality and seasonality can be judged before any model is run.

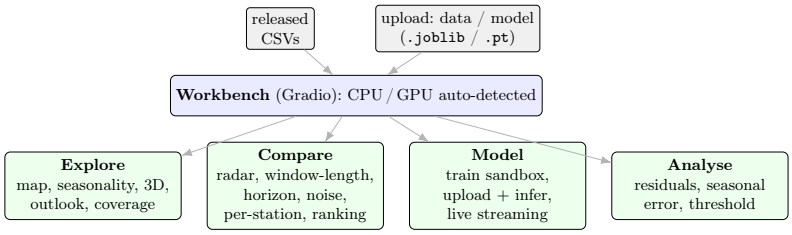
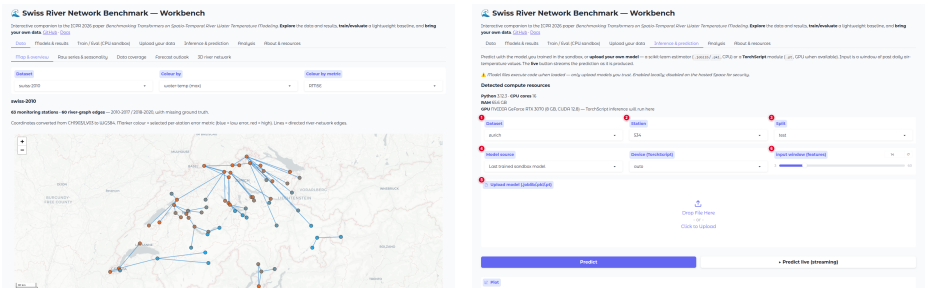


Fig. 2. Architecture of the interactive workbench. Released CSVs and user uploads (data or a model) feed a single Gradio app.



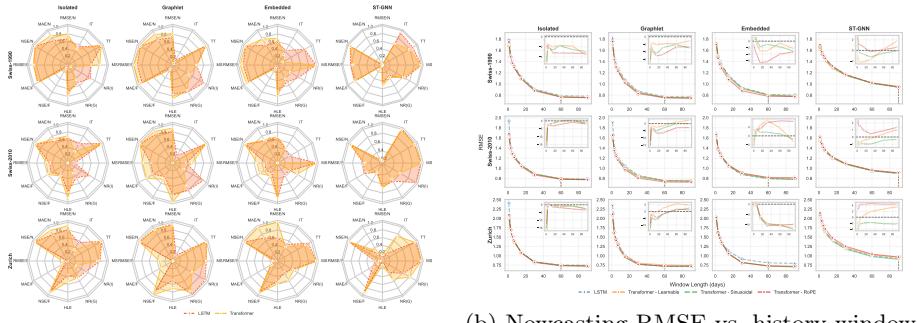
(a) Explore the river network on a map (b) Bring your own model and predict live

Fig. 3. The workbench in use. (a) A hydrologist explores the river network and station data on a Swiss basemap; (b) a user uploads their own model and streams predictions.

Compare. The *Compare* module reproduces the paper's results on demand from the committed CSVs, including the multi-aspect radar chart comparison, the history-window and forecasting-horizon curves, the noise-robustness curves, the per-station error, and a ranking of the eight models. Every panel is regenerated on a CPU in seconds, so a reader can reproduce a figure without a GPU or the trained checkpoints.

Bring your own data and model. Users are not limited to the bundled data and models: any time series in a common format (CSV/TSV/Excel/JSON/Parquet) is auto-routed to the matching view with schema validation, and a trained model, either a scikit-learn estimator (`.joblib`) or a TorchScript module (`.pt`), can be uploaded to run on the released data.

Training, evaluation and prediction. Training is a one-command operation, where the pipeline of Section 2 trains any of the eight models, or a user's own architecture, with a single command. The workbench then makes any trained model usable for evaluation and prediction. A fitted benchmark or an uploaded model, or a lightweight baseline is streamed *live* against the observations and scored with RMSE, MAE and Nash–Sutcliffe efficiency, on a detected GPU or the CPU.



(a) Multi-aspect model comparison

(b) Nowcasting RMSE vs. history window

Fig. 4. Evaluation of the library, reproducing two figures from the accompanying ICPR paper [3]. (a) A multi-aspect comparison (normalized to $[0, 1]$, where larger is better) of the LSTM and Transformer variants across the four model architectures and the three datasets; (b) Nowcasting RMSE falls steeply as the history window grows and then saturates at a method-dependent point.

Business intelligence, analysis and visualization. Beyond raw comparison, an analysis layer targets decision-making: residual bias and its distribution, mean error by calendar month, and the number of days per year a station exceeds an ecological threshold. This information gives a compact business-intelligence view of the data and the models' behavior.

4 Evaluation of the library

We evaluate the library along several axes, starting from the comparison it is meant to reproduce and then turning to the choices that most affect the results.

Multi-aspect comparison. Figure 4(a) reproduces the multi-aspect comparison of the eight models across the three datasets. No single model dominates every aspect: the ranking shifts with the dataset, the metric, and the forecasting horizon, so reporting the full multi-aspect view rather than a single headline number is what lets a practitioner pick the model that fits their own priorities.

Parameter sensitivity. Two conditions dominate the accuracy of every model: the history-window length and the forecast horizon. In nowcasting, a short history window inflates the RMSE by more than 100%, and the error saturates only beyond a method-dependent point of about 60 days for the *Isolated*, *Graphlet* and *Embedded* models and 90 days for the ST-GNN. In forecasting, the error grows with the horizon. Extending it from one to seven days raises the RMSE by up to about 0.7 °C. Figure 4(b) plots the history-window curve, and the released sweep CSVs let readers re-examine both effects directly.

5 Conclusion and Future Work

This paper presents the implementation and reproducibility of the ICPR benchmark [3]. It releases `swiss-river-network-benchmark`, with a reproducible pipeline which regenerates every reported result, and one code-free workbench opening the data, the models, and their predictions to hydrologists. Together, they turn an one-off study into a benchmark that domain experts can inspect, reuse, and extend.

The future work includes a already-in-building complete system around this benchmark in two directions. First, we are adding more models and more datasets to broaden the comparison. Second, we are turning the workbench into a general platform that brings the same code-free workflow to researchers and practitioners outside computer science, and extends it from river temperature to other applied domains.

References

1. Caissie, D.: The thermal regime of rivers: a review. Freshwater biology **51**(8), 1389–1406 (2006)
2. Michel, A., Brauchli, T., Lehning, M., Schaefli, B., Huwald, H.: Stream temperature and discharge evolution in switzerland over the last 50 years: annual and seasonal behaviour. Hydrology and Earth System Sciences **24**(1), 115–142 (2020)
3. Jia, L., Fankhauser, B., Bigler, V., Riesen, K.: Benchmarking transformers on spatio-temporal river water temperature modeling. In: International Conference on Pattern Recognition (ICPR) (2026)
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
6. Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., Talwalkar, A.: A system for massively parallel hyperparameter tuning. In: Proceedings of Machine Learning and Systems (MLSys) (2020)
7. Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118 (2018)
8. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: RoFormer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)