Fine Flood Forecasts: Incorporating local data into global models through fine-tuning

Emil Ryd[1], **Grey Nearing**[2]
[1]University of Oxford
[2]Google Research
{emil.ryd@new.ox.ac.uk, gsnearing@google.com}

Abstract

Floods are the most common form of natural disaster and accurate flood forecasting is essential for early warning systems. Previous work has shown that machine learning (ML) models are a promising way to improve flood predictions when trained on large, geographically-diverse datasets. This requirement of global training can result in a loss of ownership for national forecasters who cannot easily adapt the models to improve performance in their region, preventing ML models from being operationally deployed. Furthermore, traditional hydrology research with physics-based models suggests that local data – which in many cases is only accessible to local agencies – is valuable for improving model performance. To address these concerns, we demonstrate a methodology of pre-training a model on a large, global dataset and then fine-tuning that model on data from individual basins. This results in performance increases, validating our hypothesis that there is extra information to be captured in local data. In particular, we show that performance increases are most significant in watersheds that underperform during global training. We provide a roadmap for national forecasters who wish to take ownership of global models using their own data, aiming to lower the barrier to operational deployment of ML-based hydrological forecast systems[1].

1 Introduction

Climate change is causing an increase in the intensity and frequency of heavy precipitation (Min et al., 2011), increasing river flooding (Alifu et al., 2022). Floods are already the most frequently occurring natural disaster, having affected 2.5 billion people between 1994 and 2013 (for Research on the Epidemiology of Disasters, 2015). The rate of floods has more than doubled since 2000 (World Meteorological Organization, 2021), and is expected to increase further due to anthropogenic climate change. The World Bank estimates that improving the flood early warning systems of developing countries to the standards of developed countries would save 23,000 lives per year on average (Hallegatte, 2012). In this paper, we focus specifically on the task of river-based flood prediction, and more specifically, on models that provide short-perm predictions of river flow volume.

Machine learning (ML) models trained on large global datasets have been shown to be effective at generating accurate flood forecasts (Kratzert et al., 2019a; Nearing et al., 2024). This type of model performs best when trained on data from a large and diverse set of geographical locations (Kratzert et al., 2024). However, it is difficult for local flood forecasting agencies to develop and train prediction models that require global datasets. Currently, such models are instead trained and operated by multinational organizations like Google (e.g., `g.co/floodhub`). This implies a potential loss of ownership for the national hydromet agencies who may have an institutional preference for using models calibrated or developed locally vs. global models provided by a third party. Furthermore, national hydromet agencies often have proprietary data for their own geographical region, which they may not wish

[1]Our code is available at `https://github.com/EmilRyd/Fine-Flood-Forecasts.git`

to share publicly, meaning that a global model trained by a third party would not have been trained on data from their local watersheds.

While training a model on as much data as possible makes sense from an ML perspective, it is contrary to intuitions derived from decades of research in the hydrological sciences, where individual basin calibration traditionally provides the best forecasts (Hrachowitz et al., 2013; Hirpa et al., 2018). Based on this intuition and the needs (discussed above) of many national and local flood forecasting agencies to use their own data and models, we propose fine-tuning a globally pre-trained model using data from individual basins. Fine-tuning is a well-known practice within the ML community, and for us serves the purpose of producing models aimed at high-quality local prediction while still capturing the advantages of large-sample training. The purpose of this paper is to highlight how this practice can help facilitate trust, reliability, and adoption of best-in-class flood forecasting tools.

What we describe and test in this paper is a relatively simple workflow facilitated by open, public ML-based hydrology models, which can be adopted by local agencies. Fine-tuning requires some amount of technical expertise, but does not require large computing resources. We provide the necessary open-source pre-trained global model as well as an outline of a fine-tuning procedure that agencies might use or adapt for their needs (see Appendix A for details).

2 Methodology

2.1 Data

As input data, we use the global Caravan dataset Kratzert et al. (2023), which contains atmospheric variables (e.g., precipitation, temperature) and catchment-specific hydrological variables (e.g. climatological soil moisture, elevation), along with globally sourced streamflow data, mostly contributed by national hydromet agencies. Currently, the Caravan dataset consists of 22,732 unique catchments (Färber et al., 2024), however to ensure continuity with previous work (Roi-Cohen and Morin, 2024; Kratzert et al., 2023) we trained on an older, reduced version of Caravan containing 6375 basins. Since fine-tuning on every basin in the dataset would be computationally expensive, we fine-tune on a random sample of 159 basins, see figure 1.

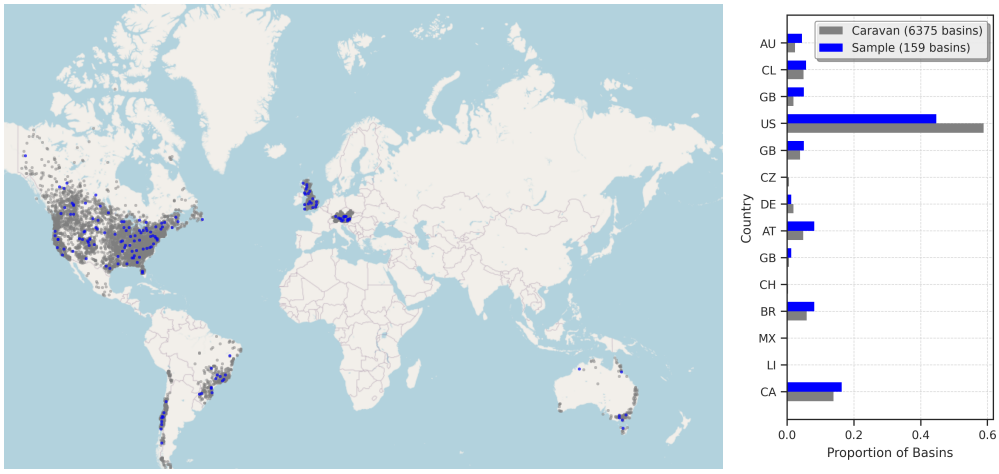

Figure 1: Left: world map showing all 6375 in the Caravan dataset (grey), and our 159 randomly sampled basins (blue). Right: bar chart showing how well each country is represented in our sample compared to the dataset as a whole.

2.2 Model architecture & training details

We pre-train a variant of a long short-term memory (LSTM) model, following Kratzert et al. (Kratzert et al., 2018), on all 6375 basins. LSTMs (Hochreiter and Schmidhuber, 1997) are a variant of recursive neural networks (RNNs) (Rumelhart et al., 1986), commonly used in temporal prediction tasks. LSTMs are suitable for streamflow prediction since they have an internal state space, which is similar to how we conceptualize physical systems such as watersheds (Kratzert et al., 2018). This allows LSTM memory cells to model hydrological dynamics such as reservoirs and storages (Kratzert et al., 2019b; Lees et al., 2022). Transformers Vaswani et al. (2017) have been tried, yet LSTMs remain the state-of-the-art within streamflow prediction with ML.The concepts outlined here are, in principle, applicable to most types of ML models.

Our LSTM model takes as input a time series of atmospheric and hydrological variables stretching 365 days back, and then outputs a single prediction of streamflow (i.e. volume of water flowing per unit time) for the following day, same as previous work (Kratzert et al., 2022; 2018; Nearing et al., 2024). We used the model configuration from Roi-Cohen and Morin (Roi-Cohen and Morin, 2024), and trained the model using the open-source Neuralhydrology library Kratzert et al. (2022). Training the 260K parameter model takes 30 wall clock hours on a single NVIDIA V100 GPU.

For each basin in our sample (see section 2.1), we swept over 50 sets of hyperparameters using Tree-structured Parzen estimator (Watanabe, 2023) from the hyperopt library (Bergstra et al., 2013), selecting the hyperparameters with the best validation set performance. To ensure that our results are robust across different model instances, we ran this procedure across 8 pre-trained model instances trained with different random seeds, generating 1272 data points in total. Further information on the model configuration and training setup is given in Appendix B.1.

2.3 Metrics

We focus on evaluating models using the Nash-Sutcliffe efficiency (NSE) and the Kling-Gupta efficency (KGE) metrics. These are arguably the main metrics used in assessing hydrological models, and are both related to standard correlation and bias metrics (Gupta et al., 2009; Knoben et al., 2019). These metrics measure the accuracy of streamflow predictions, and a score closer to 1 is better. However, these two metrics alone are generally not sufficient for a complete hydrological analysis, and a larger suite of performance metrics is reported in Appendix C.

3 Results

3.1 Fine-tuning improvements

To evaluate the general performance improvements caused by fine-tuning, we compare the performance of the pre-trained model with its fine-tuned counterpart (see table 1). The results confirm our assumption (from (Kratzert et al., 2024)) that training a model on a single basin gives worse performance than training a model on a larger, multi-basin dataset. Although this is counterintuitive from a hydrological standpoint (where basin-specific calibration is traditionally crucial), it is a common occurrence across almost all applications of deep learning.

Table 1: Model performance comparison for fine-tuned, pre-trained, and single-basin trained model. Best score in bold, second best in italics.

Model	NSE (mean)	NSE (median)	KGE (mean)	KGE (median)
Single-basin [1]	0.358 (± 0.106)	0.541 (± 0.251)	0.331 (± 0.197)	0.609 (± 0.197)
Pre-trained	0.473 (± 0.019)	0.583 (± 0.016)	0.520 (± 0.007)	0.683 (± 0.010)
Fine-tuned	**0.541** (± 0.017)	**0.625** (± 0.013)	**0.599** (± 0.042)	**0.709** (± 0.015)

Fine-tuning provides a significant increase in prediction skill over the pre-trained model. We found an increase of 0.068 (14%) and 0.079 (15%) in the mean (over the 159 fine-tuned basins) NSE and KGE, respectively. The median increases 0.042 (7%) for NSE and 0.026 (4%) for KGE. This indicates that much of the improvement is happening at the tail end. The fact that the pre-trained median NSE and KGE are higher than the mean also implies that there is a significant tail end of underperforming basins, and as we show in section 3.2, improvement from fine-tuning is particularly large in these.

To put these improvements in context, Nearing et al. (Nearing et al., 2022) found an 8% improvement to median NSE scores in 539 basins within the continental United States by assimilating near-real-time streamflow data. Fine-tuning uses only historical data whereas data assimilation requires near real-time data and is therefore significantly more difficult and costly to implement (real-time data is harder to access and more difficult to ingest into ML workflows). Fine-tuning therefore represents a substantial performance gain using an approach that has less demanding data requirements than data assimilation. We use data assimilation as a point of reference because, as far as we are aware, no other published method for increasing performance of LSTM-based streamflow models has shown as much skill improvement.

3.2 Correlation with pre-trained skill

Apart from just knowing the overall skill improvements, it is desirable to know where or under what conditions fine-tuning is more or less valuable. We show the relationship between fine-tuning improvement and original prediction score from the pre-trained model in figure 2, which is negative. Hence, basins on which the pre-trained models has a lower prediction skill tend to benefit more from fine-tuning. The fraction of variance explained by this trend is 11.1%. Figure 2 also shows that improvement through fine-tuning is still noisy and not guaranteed – for 22% of (base model, basin) pairs, fine-tuning even impacts performance negatively. Looking at each basin individually (across all the models), the proportion of negative changes is lower at 14%.

As also seen in figure 2, for every country in the dataset, there is a net positive improvement from fine-tuning. The figure also shows a stronger negative relationship between the pre-trained score and the fine-tuning improvements. In particular, for the pre-trained model's worst-performing countries (Brazil & the US), fine-tuning improvements are substantially higher compared to the other countries. Future work will correlate fine-tuning skill changes with the geographical and hydrological characteristics of the watersheds.

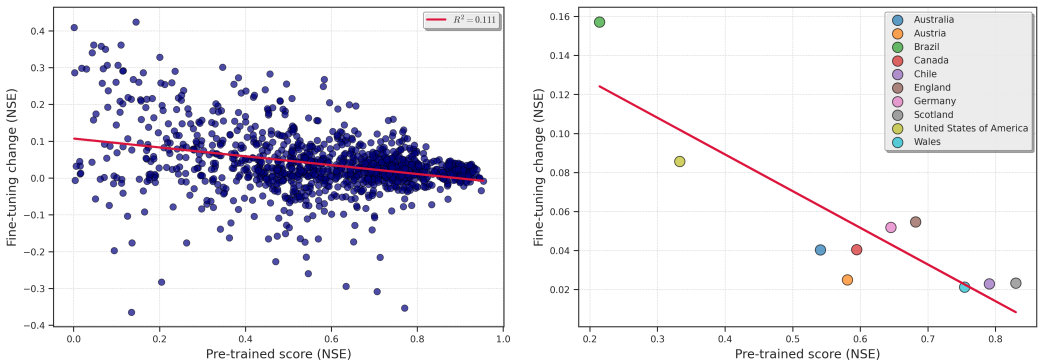

Figure 2: Correlation between fine-tuning improvements and pre-trained model skill across all basins (left) and aggregated across all countries (right). There is a negative correlation between pre-trained prediction skill and the improvement attained from fine-tuning.

[1]Only trained for 147 out of the 159 basins due to some basins not having any data within our pre-defined training set.

4 Conclusion & Discussion

We showed that fine-tuning a globally pre-trained model on individual basins leads to significant skill improvements, particularly in basins where a pre-trained global model struggles. Our results confirm the hypothesis – informed by traditional hydrology – that we can get more out of local data even when following the current best-practice for training ML-based hydrology models (i.e., training on large, diverse datasets). Furthermore, we found that basins and countries for which the pre-trained models struggle tend to gain the most from fine-tuning. Hence, we believe our method will be particularly helpful for national forecasters in regions that are currently underserved by global models. We provide a clear step-by-step guide for how to set up fine-tuning for forecasters in Appendix A.

In deciding whether to fine-tune a model, a matter that will likely be of particular importance to national forecasting agencies and hydrologists is the relationship between fine-tuning improvement and hydrological basin characteristics (e.g. aridity, area). By training a regression model on this task, one could find such relationship and hopefully also predict with accuracy for what basins fine-tuning will work. We expect that both of these questions will be important for national forecasters, and are questions left for future work.

Acknowledgments

The authors would like to thank Efrat Morin, Martin Gauch, Frederik Kratzert, and Moise Busogi for useful discussions and valuable advice. The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work. http://dx.doi.org/10.5281/zenodo.22558, and are especially grateful to Philip Stier and his lab for allowing us to use their compute resources in ARC.

References

Haireti Alifu, Yukiko Hirabayashi, Yukiko Imada, and Hideo Shiogama. Enhancement of river flooding due to global warming. *Scientific Reports*, 12(1):20687, Nov 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-25182-6. URL `https://doi.org/10.1038/s41598-022-25182-6`.

James Bergstra, Daniel Yamins, and David D Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 115–123. PMLR, 2013.

Centre for Research on the Epidemiology of Disasters. The human cost of natural disasters 2015: A global perspective. Technical report, 2015. URL `https://reliefweb.int/report/world/human-cost-natural-disasters-2015-global-perspective`.

C. Färber, H. Plessow, S. Mischel, F. Kratzert, N. Addor, G. Shalev, and U. Looser. Grdc-caravan: extending caravan with data from the global runoff data centre. *Earth System Science Data Discussions*, pages 1–28, 2024. doi: 10.5194/essd-2024-427.

Martin Gauch, Frederik Kratzert, Oren Gilon, Hoshin Gupta, Juliane Mai, Grey Nearing, Bryan Tolson, Sepp Hochreiter, and Daniel Klotz. In defense of metrics: Metrics sufficiently encode typical human preferences regarding hydrological model performance. *Water Resources Research*, 59(6):e2022WR033918, 2023. doi: https://doi.org/10.1029/2022WR033918. URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2022WR033918`. e2022WR033918 2022WR033918.

Hoshin V. Gupta, Harald Kling, Koray K. Yilmaz, and Guillermo F. Martinez. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *Journal of Hydrology*, 377(1):80–91, 2009. ISSN 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2009.08.003. URL `https://www.sciencedirect.com/science/article/pii/S0022169409004843`.

Stéphane Hallegatte. A cost effective solution to reduce disaster losses in developing countries: Hydro-meteorological services, early warning, and evacuation. Policy Research Working Paper 6058, World Bank, Washington, DC, May 2012. © World Bank. License: CC BY 3.0 IGO.

Feyera A. Hirpa, Peter Salamon, Hylke E. Beck, Valerio Lorini, Lorenzo Alfieri, Ervin Zsoter, and Simon J. Dadson. Calibration of the global flood awareness system (glofas) using daily streamflow data. *Journal of Hydrology*, 566:595–606, 2018. ISSN 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2018.09.052. URL https://www.sciencedirect.com/science/article/pii/S0022169418307467.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

M. Hrachowitz, H.H.G. Savenije, G. Blöschl, J.J. McDonnell, M. Sivapalan, J.W. Pomeroy, B. Arheimer, T. Blume, M.P. Clark, U. Ehret, F. Fenicia, J.E. Freer, A. Gelfan, H.V. Gupta, D.A. Hughes, R.W. Hut, A. Montanari, S. Pande, D. Tetzlaff, P.A. Troch, S. Uhlenbrook, T. Wagener, H.C. Winsemius, R.A. Woods, Zehe E., and C. Cudennec. A decade of predictions in ungauged basins (pub)—a review. *Hydrological Sciences Journal*, 58(6):1198–1255, 2013. doi: 10.1080/02626667.2013.803183. URL https://doi.org/10.1080/02626667.2013.803183.

W. J. M. Knoben, J. E. Freer, and R. A. Woods. Technical note: Inherent benchmark or not? comparing nash–sutcliffe and kling–gupta efficiency scores. *Hydrology and Earth System Sciences*, 23(10):4323–4331, 2019. doi: 10.5194/hess-23-4323-2019. URL https://hess.copernicus.org/articles/23/4323/2019/.

F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018. doi: 10.5194/hess-22-6005-2018. URL https://hess.copernicus.org/articles/22/6005/2018/.

F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, and G. Nearing. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, 23(12):5089–5110, 2019a. doi: 10.5194/hess-23-5089-2019. URL https://hess.copernicus.org/articles/23/5089/2019/.

F. Kratzert, M. Gauch, D. Klotz, and G. Nearing. Hess opinions: Never train a long short-term memory (lstm) network on a single basin. *Hydrology and Earth System Sciences*, 28(17):4187–4201, 2024. doi: 10.5194/hess-28-4187-2024. URL https://hess.copernicus.org/articles/28/4187/2024/.

Frederik Kratzert, Mathew Herrnegger, Daniel Klotz, Sepp Hochreiter, and Günter Klambauer. *NeuralHydrology – Interpreting LSTMs in Hydrology*, page 347–362. Springer International Publishing, 2019b. ISBN 9783030289546. doi: 10.1007/978-3-030-28954-6_19. URL http://dx.doi.org/10.1007/978-3-030-28954-6_19.

Frederik Kratzert, Martin Gauch, Grey Nearing, and Daniel Klotz. Neuralhydrology — a python library for deep learning research in hydrology. *Journal of Open Source Software*, 7(71):4050, 2022. doi: 10.21105/joss.04050. URL https://doi.org/10.21105/joss.04050.

Frederik Kratzert et al. Caravan-a global community dataset for large-sample hydrology. *Scientific Data*, 10(1):61, 2023.

T. Lees, S. Reece, F. Kratzert, D. Klotz, M. Gauch, J. De Bruijn, R. Kumar Sahu, P. Greve, L. Slater, and S. J. Dadson. Hydrological concept formation inside long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 26(12):3079–3101, 2022. doi: 10.5194/hess-26-3079-2022. URL https://hess.copernicus.org/articles/26/3079/2022/.

Seung-Ki Min, Xuebin Zhang, Francis W. Zwiers, and Gabriele C. Hegerl. Human contribution to more-intense precipitation extremes. *Nature*, 470(7334):378–381, Feb 2011. ISSN 1476-4687. doi: 10.1038/nature09763. URL https://doi.org/10.1038/nature09763.

G. S. Nearing, D. Klotz, J. M. Frame, M. Gauch, O. Gilon, F. Kratzert, A. K. Sampson, G. Shalev, and S. Nevo. Technical note: Data assimilation and autoregression for using near-real-time streamflow observations in long short-term memory networks. *Hydrology and Earth System Sciences*, 26(21):5493–5513, 2022. doi: 10.5194/hess-26-5493-2022. URL https://hess.copernicus.org/articles/26/5493/2022/.

Grey Nearing, Deborah Cohen, Vusumuzi Dube, Martin Gauch, Oren Gilon, Shaun Harrigan, Avinatan Hassidim, Daniel Klotz, Frederik Kratzert, Asher Metzger, Sella Nevo, Florian Pappenberger, Christel Prudhomme, Guy Shalev, Shlomo Shenzis, Tadele Yednkachw Tekalign, Dana Weitzner, and Yossi Matias. Global prediction of extreme floods in ungauged watersheds. *Nature*, 627(8004):559–563, Mar 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07145-1. URL https://doi.org/10.1038/s41586-024-07145-1.

O. Roi-Cohen and E. Morin. Global vs. regional training of deep learning streamflow prediction models using the caravan data set for interpreting high vs. low flow prediction (poster). Flood Forecasting Meets Machine Learning Google Workshop, January 22-23 2024. Virtual.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. Technical Report ICS 8506, Institute for Cognitive Science, University of California, San Diego, California, 1986.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Lukasz Kaiser, Illia Polosukhin, Andreas Kaiser, Sharan Narang, Geoffrey Shazeer, and Yoshua Bengio. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008. Curran Associates, Inc., 2017. URL https://arxiv.org/abs/1706.03762.

Shuhei Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *ArXiv*, abs/2304.11127, 2023.

World Meteorological Organization. State of climate services 2021: Water. Report WMO-No. 1278, World Meteorological Organization, Geneva, 2021. URL https://library.wmo.int/index.php?lvl=notice_display&id=21963.

A Fine-tuning guide for forecasters

Here is a step-by-step guide for national forecasters and other regional operators who wish to fine-tune global models on their own data. An even more detailed guide can be found in our repository `https://github.com/EmilRyd/Fine-Flood-Forecasts.git`.

1. **Download Caravan:** Information on the Caravan dataset can be found on Github at `https://github.com/kratzert/` and the actual dataset is downloaded from Zenodo at `https://zenodo.org/records/14673536`
2. **(Optional): Add your own data**: If the basin data you want to fine-tune on is already in Caravan, then you can skip this step. Otherwise, add the basins (together with streamflow labels) to the Caravan dataset. Please note that you do not need to add this to the public dataset, you can just format your data into your local copy of Caravan and run it there. A complete tutorial for formatting your data into Caravan is at `https://github.com/kratzert/Caravan/wiki/Extending-Caravan-with-new-basins`.
3. **Fine-tune on your data:** Follow the instructions in our repository to fine-tune your pre-trained model on your chosen basins. Our code automatically downloads a global pre-trained model which we have open-sourced, and provides all the necessary code to fine-tune this model on any individual basins (either already in Caravan or that you add into the Caravan format).

B Training details

B.1 Model configuration and pre-training setup

We use the Neuralhydrology library (Kratzert et al., 2022) to pre-train 8 instances of our base model. We follow very closely the model configuration used by Roi-Cohen and Morin (Roi-Cohen and Morin, 2024), changing only the batch size from 128 to 256 to speed up training. Our base model is an LSTM with a single hidden layer of size 256, with a linear embedding layer of 10 neurons for the static catchment attributes, and a linear head (see the Neural Hydrology documentation for more details). We trained our model for 40 epochs, with a learning rate of 5×10^{-5} for epochs $1 - 30$, and 5×10^{-6} for epoch $31 - 40$. We used NSE as the loss. The precise model configuration is in the model config file in our repository (after downloading the pre-trained model from Hugging Face) for full details.

B.2 Fine-tuning setup

We randomly sampled 159 basins from our full dataset. For each basin, we then performed a hyperparameter sweep over a set of fine-tuning parameters using the Tree-Structured Parzen Estimator (Watanabe, 2023) within the hyperopt library (Bergstra et al., 2013). We ran 50 evaluations for every basin. We followed the same procedure for our single-basin trained models, with some additional hyperparemeters to vary the model size to avoid overparametrizing our model. Our hyperparameter search space is:

Hyperparameter	**Fine-tuning**	**Training (single basin)**
Epochs	$[1, 40]$	$[1, 40]$
Learning Rate (epoch 1-20)	$[10^{-5}, 10^{-3}]$	$[10^{-5}, 10^{-3}]$
Learning Rate (epoch 21-40)	$[10^{-6}, 10^{-4}]$	$[10^{-6}, 10^{-4}]$
Loss [2]	{NSE, MSE, RMSE}	{NSE, MSE, RMSE}
Fine-tuning modules	{Full, Head}	N/A
Hidden size	N/A	{16, 32, 64}

Table 2: Hyperparameter search space for fine-tuning.

[2]Here, "Loss" only changes the loss that is calculated during gradient descent (for fine-tuning), but the parameter sweep still always evaluates against the NSE value on the validation set to find

For "Fine-tuning modules", "Full" means that we fine-tune all the weights from the pre-trained model (except the initial embedding layer for the statics attributes), and "Head" means that fine-tune only the head, which is just a linear layer. We fine-tune each of our 8 base models on the 159 basins, and then average over the fine-tuning values for each of the basins.

C More results

Above we reported NSE and KGE scores of our various models. These metrics measure correlations (in some sense) between simulated and predicted hydrographs, but do not capture all important aspects of hydrological prediction. For a more complete picture, we report several other common performance metrics. For more details on these metrics, see Gauch et al (Gauch et al., 2023).

Metric	**Single-basin**	**Pre-trained**	**Fine-tuned**
NSE (mean)	0.358 ± 0.106	0.473 ± 0.019	$\mathbf{0.541}$ ± 0.014
NSE (median)	0.541 ± 0.025	0.582 ± 0.016	$\mathbf{0.627} \pm 0.008$
MSE (mean)	2.338 ± 0.297	2.227 ± 0.020	$\mathbf{2.016} \pm 0.098$
MSE (median)	1.265 ± 0.117	1.080 ± 0.076	$\mathbf{0.963} \pm 0.021$
RMSE (mean)	1.271 ± 0.070	1.220 ± 0.006	$\mathbf{1.153} \pm 0.023$
RMSE (median)	1.125 ± 0.068	1.039 ± 0.037	$\mathbf{0.982} \pm 0.021$
KGE (mean)	0.331 ± 0.197	0.520 ± 0.007	$\mathbf{0.599} \pm 0.026$
KGE (median)	0.609 ± 0.022	0.683 ± 0.010	$\mathbf{0.711} \pm 0.007$
Pearson r (mean)	0.708 ± 0.017	0.758 ± 0.002	$\mathbf{0.773} \pm 0.005$
Pearson r (median)	0.740 ± 0.015	0.795 ± 0.005	$\mathbf{0.804} \pm 0.004$
α-NSE (mean)	0.728 ± 0.021	$\mathbf{0.881} \pm 0.009$	0.842 ± 0.006
α-NSE (median)	0.749 ± 0.035	$\mathbf{0.880} \pm 0.005$	0.868 ± 0.013
β-NSE (mean)	0.053 ± 0.027	0.059 ± 0.006	$\mathbf{0.033} \pm 0.005$
β-NSE (median)	$\mathbf{0.004} \pm 0.005$	0.029 ± 0.003	$\mathbf{0.004} \pm 0.001$
β-KGE (mean)	1.228 ± 0.199	1.173 ± 0.009	$\mathbf{1.085} \pm 0.026$
β-KGE (median)	$\mathbf{1.004} \pm 0.042$	1.041 ± 0.004	1.007 ± 0.016
Peak timing (mean)	0.750 ± 0.043	0.647 ± 0.012	$\mathbf{0.609} \pm 0.011$
Peak timing (median)	0.625 ± 0.040	0.537 ± 0.009	$\mathbf{0.500} \pm 0.012$
Missed-Peaks (mean)	0.605 ± 0.014	0.559 ± 0.005	$\mathbf{0.548} \pm 0.004$
Missed-Peaks (median)	0.591 ± 0.020	0.546 ± 0.008	$\mathbf{0.537} \pm 0.007$
Peak-MAPE (mean)	52.10 ± 1.40	46.71 ± 0.60	$\mathbf{45.54} \pm 0.48$
Peak-MAPE (median)	50.40 ± 1.77	45.18 ± 0.61	$\mathbf{44.18} \pm 0.61$

Table 3: Performance metrics with mean $\pm$ std/$\sqrt{n}$ and median $\pm$ MAD/$\sqrt{n}$ (mean absolute deviation)

the best set of fine-tuning hyperparameters. In our public repository, however, the loss is set to be the same on both the validation and training set.