A Neural Ordinary Differential Equations Approach for 2D Flow Properties Analysis of Hydraulic Structures

Sebastian Eilermann [1], Lisa Lüddecke [2], Michael Hohmann [1], Bernd Zimmering [1], Mario Oertel [2], Oliver Niggemann [1]

[1] *Institute for Artificial Intelligence (HSU-AI), Helmut-Schmidt-University Hamburg*

[2] *Hydraulic Engineering Section, Civil Engineering, Helmut-Schmidt-University Hamburg*

{*firstname.lastname*}*@hsu-hh.de*

Editors: Cecília Coelho, Bernd Zimmering, M. Fernanda P. Costa, Luís L. Ferrás, Oliver Niggemann

Abstract

In hydraulic engineering, the design and optimization of weir structures play a critical role in the management of river systems. Weirs must efficiently manage high flow rates while maintaining low overfall heights and predictable flow behavior. Determining upstream flow depths and discharge coefficients requires costly and time-consuming physical experiments or numerical simulations. Neural Ordinary Differential Equations (NODE) can be capable of predicting these flow features and reducing the effort of generating experimental and numerical data. We propose a simulation based 2D dataset of flow properties upstream of weir structures called FlowProp. In a second step we use a NODE-based approach to analyze flow behavior as well as discharge coefficients for various geometries. In the evaluation process, it is evident that the aforementioned approach is effective in describing the headwater, overfall height and tailwater. The approach is further capable of predicting the flow behavior of geometries beyond the training data.

Project page and code: https://github.com/SEilermann/FlowProp

Keywords: Neural Ordinary Differential Equations, Hydraulic Engineering, Benchmark Dataset, Discharge Coefficients

1. Introduction

The use of machine learning in the field of fluid dynamics has gained increasing attention, particularly for its ability to model complex systems where traditional numerical simulations can be computationally expensive. In the specific case of hydraulic engineering, predicting flow behavior over different geometric configurations is crucial for the design and optimization of structures such as weirs and dams. This study explores the potential of Neural Ordinary Differential Equations (NODEs) (Chen et al., 2018), a relatively recent architecture in deep learning, to tackle the problem of fluid flow prediction over varying weir geometries.

Weirs, as hydraulic structures, are essential for regulating flow in open channels. Their efficiency is typically characterized by the discharge coefficient c_D, which represents the relationship between the discharge Q and the geometric properties of the structure (Aigner and Bollrich, 2015). Traditional research methods for calculating c_D often rely on scaled physical model tests or computational fluid dynamics (CFD) simulations, which can be limited by cost and time (Oertel and Bung, 2014). Recent work has shown that machine learning models, particularly fully connected neural networks (FC-NNs), can predict hydraulic properties with reasonable accuracy (Fatahi-Alkouhi et al., 2024; Iqbal and Ghani, 2024). However, such models often struggle with generalization to out-of-distribution (OOD) geometries, i.e., shapes that deviate from those seen in training.

In this paper, we propose the use of a NODE-based approach to enhance the prediction of waterlines and c_D across a diverse set of weir geometries. NODEs offer a principled way to model time-dependent or continuous systems by learning the underlying differential equations governing the system's dynamics (Chen et al., 2018). This makes them particularly suitable for our task, where fluid flow is governed by equations but the exact solution is complicated by complex geometrical boundaries. We hypothesize that NODEs, by solving such systems as an ODE, can generalize better to previously unseen geometries compared to FC-NNs.

To validate this, we present a series of experiments where NODE architectures are trained on our custom dataset, FlowProp, which includes various weir shapes, ranging from trapezoidal forms to more irregular, non-polygonal geometries. We systematically compare NODEs to traditional FC-NN architectures and evaluate their reconstruction of waterlines as well as their ability to predict c_D values. Our results demonstrate that NODEs outperform FC-NNs, particularly in scenarios involving novel or complex geometries, showcasing their potential for applications in hydraulic design.

The contributions of this paper are threefold: (1) we introduce and optimize NODE architectures specifically for the task of fluid flow reconstruction and discharge coefficient prediction; (2) we demonstrate the superior generalization capabilities of NODEs on out-of-distribution weir geometries; and (3) we provide a new dataset, FlowProp, designed to benchmark models on realistic hydraulic tasks involving diverse geometric configurations.

The remainder of this paper is structured as follows: In Section 2, we review related work in the application of machine learning to hydraulic systems. Method and dataset generation are presented in Section 3. Section 4 details the experiments conducted including results and discussions, followed by conclusions and future work in Section 5.

2. State of the Art

2.1. Hydraulic Structures and Discharge Coefficients

A weir is defined as a barrier structure constructed with the intention of impounding river systems for a variety of reasons, including, but not limited to, the generation of electricity and the regulation of water levels (Strobl and Zunic, 2006). However, the construction of weirs also has the effect of disrupting the natural flow of rivers, which can potentially result in flooding in the event of high Q caused by extreme weather conditions. Therefore, an efficient design for discharging large Q but still achieving minimal overfall heights is a significant objective of hydraulic engineers. A key performance metric for weirs is the c_D, which represents the efficiency of discharging water over the structure. The Q over a weir can be described by the classical weir flow equation (Bashiri, 2016):

$$Q = \frac{2}{3} \cdot c_D \cdot b \cdot \sqrt{2g} \cdot H_s^{\frac{3}{2}} \tag{1}$$

where b is the weir width (river width at the location of the weir), H_s is the total upstream head, and g is the gravitational acceleration. The c_D encapsulates the effects of the weir geometry and hydraulic parameters such as Q, approaching flow velocity (v_s), surface roughness, tailwater and more (Aigner and Bollrich, 2015).

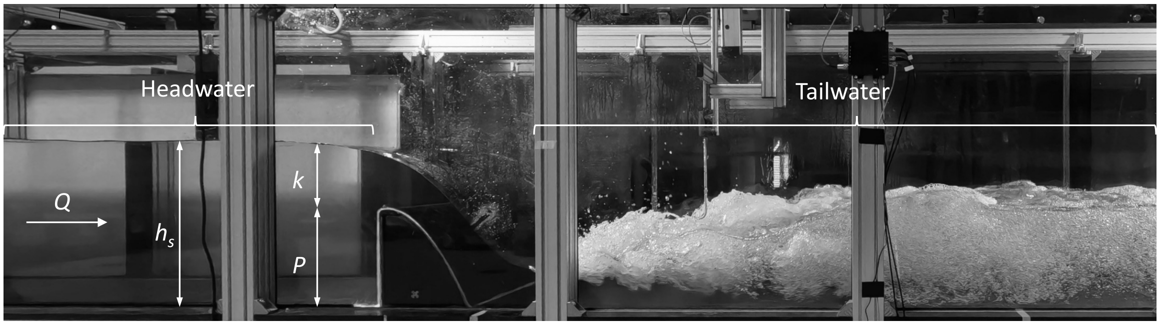

Figure 1: Example of flow behavior over an exemplary linear weir geometry with the weir height P and the overfall height k in a scaled physical model test with flow direction from left to right.

Traditionally, c_D has been determined using empirical equations derived from experimental data or CFD simulations. Both research methods are time-consuming and costly, requiring the use of laboratory resources or high-performance computing (Oertel and Bung, 2014). Although scaled physical model tests in hydraulic laboratories allow for the collection of data for varying Q in a relatively short period of time, the variation of geometries is a significant cost and time investment. Moreover, the results for small Q may be influenced by scale effects, which can lead to inaccurate results (Shen and Oertel, 2023). In numerical simulations, the variation of geometries can be readily accomplished, and simulations at the prototype scale can eliminate scale effects. However, the calculation of complex structures with a fine computing mesh for a single Q can take up to several days (Oertel and Bung, 2014).

To provide a more precise calculation of c_D, the following equation, derived from the continuity equation, is commonly used (Ferguson, 1986):

$$c_D = \frac{3 \cdot Q}{2 \cdot b \cdot \sqrt{2 \cdot g} \cdot \left(\frac{\left(\frac{Q}{b \cdot h_s}\right)^2}{(2 \cdot g)} + k \right)^{3/2}} \tag{2}$$

where h_s represents the upstream flow depth (water level) and P is the weir height. This equation takes into account the upstream flow conditions and the geometric properties of the weir. The derivation of this formula from the continuity equation can be found in Appendix C.

All hydraulic properties that can be analyzed in the tailwater of a weir structure are highly dependent on the downstream flow conditions (Besser and Oertel, 2024). In contrast to the headwater, the tailwater is not subject to significant regulation; rather, it is adjusted to align with the prevailing natural discharge conditions. Given the inherent difficulty in establishing the downstream water level as a computational boundary condition within a numerical simulation in the absence of knowledge regarding natural flow conditions, this study will concentrate on the analysis of upstream flow parameters, rather than characteristics of the downstream flow (Thorenz, 2024).

While CFD simulations remain the benchmark for high-precision calculations, their computational demands highlight the need for more efficient methods to estimate c_D across diverse weir geometries.

2.2. Artificial Intelligence in Hydraulics

Artificial intelligence in various forms has already been implemented in the field of research related to hydraulic engineering and, more specifically, hydraulic structures. Existing studies have dealt with the estimation of scour depth and location downstream of hydraulic structures using a hybrid artificial intelligence model with binary particle swarm optimization (PSO) algorithm and support vector regression (SVR) (Salih et al., 2020) or multi-output descriptive neural network (DNN) (Guven, 2011). Other studies used FC-NNs to estimate velocities in ventilated step spillways (Valero and Bung, 2016) or to predict the friction factor of open channel flow (Yuhong and Wenxin, 2009). Furthermore, in (Fatahi-Alkouhi et al., 2024; Iqbal and Ghani, 2024) the c_D of modified semi-cylindrical structures and piano key weirs is estimated by prediction with different several models like support vector machines, simple feed forward neural networks and multivariate adaptive regression splines.

2.3. Neural Ordinary Differential Equations

The concept of NODEs, as initially proposed by Chen et al. (2018), represents a continuation of the research initiated in (Weinan, 2017; Lu et al., 2018) on the integration of machine learning and differential equations. An ODE

$$\frac{dx(t)}{dt} = f(x(t), t) \tag{3}$$

where $t \in \mathbb{R}^+$ represents the time, $x(t)$ is the state variables, and the function f models the dynamics. The aforementioned approach is predicated on the assumption that the function f is already known. Despite the significant challenge posed by complex systems, which require considerable time and knowledge investment, the lack of understanding of their underlying dynamics adds an additional layer of difficulty (Li et al., 2024). To address these challenges, a neural network f_{NN} with the parameters θ_{NN} can be trained to replace f in order to ensure that no hidden dynamics are overlooked.

$$\frac{dx(t)}{dt} = f_{\text{NN}}(x(t), t; \theta_{\text{NN}}) \tag{4}$$

When the data pair $\left(x(t), \frac{dx(t)}{dt}\right)$ is unavailable and only $x(t)$ is accessible, a NODE based approach can be used by training the neural network (NN) f_{ODE} with integrated ODEs, where θ_{ODE} are the parameters:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f_{\text{ODE}}(x(t))dt = \text{ODESolve}\left(x(t_0), f_{\text{ODE}}, t_0, t_1; \theta_{\text{ODE}}\right) \tag{5}$$

In Equation (5) using θ_{ODE} parameters of f_{ODE} and (t_0, t_1) are the limits of the integration. The initial condition of integration is set as $x(t_0)$. Therefore, Chen et al. (2018) integrated ODE solvers (such as adaptive Runge-Kutta (Butcher, 1996)) into PyTorch (Paszke et al., 2019), thus enabling solving the ODE in Equation (5). To address the challenge of sbstantial memory costs, Chen et al. (2018) proposed the use of the adjoint method, which improves memory efficiency while maintaining gradient accuracy.

The aforementioned approaches illustrate that a number of artificial intelligence based techniques have already been developed for the analysis of weir structures. In contrast to existing approaches our approach based on a NODE model for the analysis not only of the efficiency of a weir structure but also of the flow behavior, including velocity and total energy, associated with different weir structures represents a significant advancement in the field.

3. Methods

3.1. Generation of FlowProp: A 2D Flow Properties Dataset

In contrast to Equation (5), where t is used to describe a system temporal behavior, we employ the variable $s \in \{0, \ldots, S\}$, with $S \in \mathbb{N}$ denoting the total number of steps, to elucidate the flow properties of a considered step. To generate our 2D dataset of hydraulic structures with flow properties, called FlowProp, two simulations with different geometries serving as weir structure were performed. For simulation the commercially available software FLOW-3D HYDRO Version 2023R1 by Flow Science (2023) was used. The volume-of-fluid method was used and RNG (Renormalized Group) was selected as the turbulence model as suggested by (Safarzadeh and Noroozi, 2017). With a weir height of $P = 0.3$ m, the mesh size was set to 0.005 m. The simulations were performed for $Q = 0.05$ m^3/s. Since our selected initial geometries triangular ge_{tri} and rectangular ge_{rec} represent linear structures, the simulation was reduced to 2D. As a result of the simulation, h_s as well as H_s and v_s were extracted. The evolution of h_s, H_s and v_s can be seen in Figure 2 for ge_{tri}. The ge_{rec} can be found in Appendix A.2.

Additional simulations of linearly interpolated geometries (ge_n) between ge_{tri} and ge_{rec} were performed to prove a linear correlation between the resulting upstream h_s. Details are provided in Appendix A.3. By understanding the linear correlation of h_s between ge_n, the amount of n geometries $G = \{ge_0, ge_1, \ldots, ge_n\}$ saved as 2D images with matching waterlines $W = \{w_0, w_1, \ldots, w_n\}$ can be defined when generating FlowProp. This step is possible without additional simulations using

$$ge_n = (1 - l) \cdot ge_{\text{tri}} + l \cdot ge_{\text{rec}} \tag{6}$$

where l is the interpolation step. This step is used for both, the interpolation of ge_n and w_n. The mean interpolated simulation and further information of saving can be seen in Appendix A.4.

3.2. Learning of Waterlines with a NODE-Approach

For learning flow properties with a NODE approach we incorporate a convolutional encoder enc_{G} to extract features from ge_n and a NODE model to predict waterline dynamics in

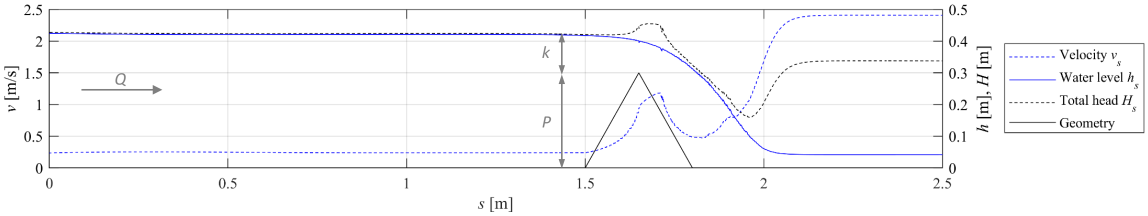

Figure 2: Flow properties (water level h_s, flow velocities v_s, total energy H_s) of the triangular geometry with flow direction from left to right with P describing the height of the weir and k the overfall height over the weir structure.

the headwater, overfall region and tailwater. Each w_n is represented by 2D coordinates $\{(x_{\mathrm{c},0}, y_{\mathrm{c},0}), ..., (x_{\mathrm{c},s}, y_{\mathrm{c},s})\}$, where each $y_{\mathrm{c},s}$ denotes the h_s.
As mentioned in Section 2.3 we focus on its spatial evolution in x_s. The $\mathrm{enc}_{\mathrm{G}}$ compresses the 2D image so $S = 502$ and our initial condition $x(s_0)$ is defined as $(x_{\mathrm{c},0}, y_{\mathrm{c},0})$. In our case $x(s_1)$ is described:

$$x\left(s_1\right) = \mathrm{ODESolve}\left(x\left(s_0\right), f_{\mathrm{ODE}}, s_0, s_1, \mathrm{enc}_{\mathrm{G}}; \theta_{\mathrm{WODE}}\right), \tag{7}$$

which captures the dynamics of the w_n under the parameters θ_{WODE}. We define the training objective to minimize the Mean Squared Error (MSE) between the predicted $\hat{y}_{\mathrm{c,s}}$ and actual water levels $y_{\mathrm{c,\ s}}$ as in previous works before (Norcliffe et al., 2021; Lee and Parish, 2021). Next to the waterline, we determine c_D from Equation (2) to evaluate the weir efficiency influenced by its shape and flow properties. For this study, the c_D is calculated from the flow depth h_s at a location of $3 \times P$ upstream of the weir.

4. Experiments

In this section, we perform three main experiments to validate the proposed method on our dataset FlowProp. First, we analyze the reconstruction capability of our approach. We compare these results to different network architectures of our NODE approach. Next, we evaluate the method by calculating c_D, to compare these in terms of effectiveness. The third and final experiment evaluates the NODE's interpolation and extrapolation capabilities.
Settings: To implement our approach, we use PyTorch (Paszke et al., 2019) and the torchdiffeq library (Chen, 2018) as the ODE solver, employing the Runge-Kutta of order 5 of Dormand-Prince-Shampine (Dormand and Prince, 1980) with adaptive step size for numerical integration. We implement the Adam optimizer (Kingma and Ba, 2017) with a learning rate of $1e-3$ to train the model, where after $1,000$ iterations the learning rate gets reduced by a factor of 0.5. The complete architecture of our NODE approach can be found in Appendix E.2.
Datasets: In order to evaluate the proposed methodology, we utilize the aforementioned dataset, FlowProp. Furthermore, we employ a chronological train-test split to create two

Table 1: Comparison of test set reconstruction results considering different NODE network architectures. According to the lowest MSE and AD, NODE4 performs best on the FlowProp dataset. AD is multiplied by 10^4.

Model	**Layers**	**Neurons**	↓ **AD [m^2]**	↓ **MSE**	**Params**
NODE1	5	100	168.1254	1.3611×10^{-4}	195,977
NODE2	10	100	8.3524	2.6265×10^{-5}	246,477
NODE3	15	100	6.9407	2.0175×10^{-5}	296.977
NODE4	20	100	**1.7985**	$\mathbf{9.9456 \times 10^{-6}}$	347,477
NODE5	25	100	17.7146	4.2681×10^{-5}	397,977
NODE6	30	100	16.9965	9.3494×10^{-4}	448,477

distinct subsets. Additionally, we generate a dataset comprising simulations of predominantly non-polygonal weirs. In the first subset FlowProp-First, the first 20% of samples, comprising ge_{rec} and trapezoidal shapes, are removed and used exclusively as the test set. The second subset is formed by removing the final 20% of samples from the initial dataset, which are used exclusively as the test set. This subset predominantly contains trapezoidal shapes and ge_{tri} and is named FlowProp-Last. The third dataset, designated as FlowProp-Diverse, comprises ten predominantly non-polygonal weirs, with selected shapes detailed in Appendix D. For all datasets, we define the sample size as $n = 100$.

4.1. Reconstruction of Waterlines

We have tested different NODE configurations of our presented approach (see Section 3.2) to reconstruct the course of the waterline in the headwater, overfall region and tailwater. The tests were performed with different hidden layer sizes and a fixed number of 100 neurons per layer. We used our FlowProp dataset. The main network architectures tested are presented in Table 1. Figure 3 presents the reconstruction quality with our best model, NODE4. The red area between the predicted and simulated waterlines represents the areal deviation, calculated using Simpson's Rule (Virtanen et al., 2020). It can be observed that the waterline can be accurately reconstructed in the headwater, overfall region and tailwater. However, with all other tested NODE configurations (1-3, 5) the waterline in the tailwater cannot be precisely reconstructed or the reconstruction task fails in total. After the overfall, the waterline tends to rise in the tailwater region, as shown in Appendix B. As the number of hidden layers increases, the waterline adapts more to the actual course of the tailwater, but after 25 hidden layers the waterline cannot be reconstructed (NODE6). In addition to the qualitative evaluation, Table 1 shows the areal deviation (AD) in m^2 and the MSE of the various NODE architectures. Despite the relative small MSE and AD of those other configurations, NODE4 performs best with the lowest evaluation metrics, indicating that NODE4 effectively learned the underlying physical properties by solving Equation (7).

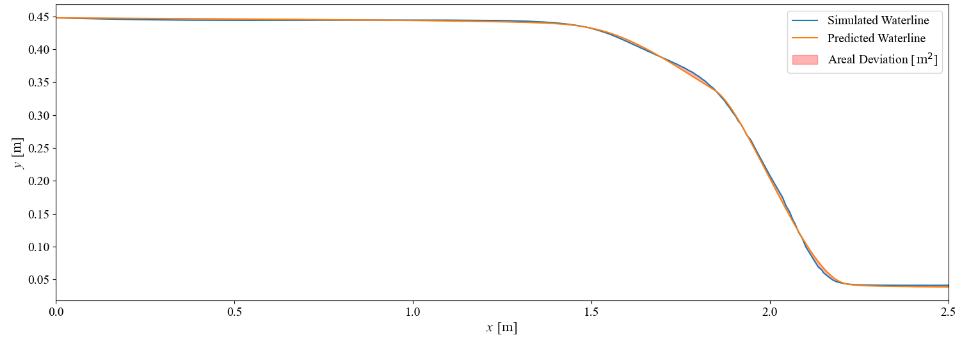

Figure 3: The areal deviation between reconstructed and simulated waterline is highlighted in red for model NODE4. It shows strong reconstruction capability in the headwater, overfall region and tailwater.

4.2. Prediction of Discharge Coefficients

In the second experiment, we utilize our best model from the experiment before NODE4 to predict values of h_s to calculate c_D for a comparison of different weirs based on their efficiency using FlowProp. We compare NODE4 to a FC-NN to predict h_s. The FC-NN uses enc_G similar to our NODE approach to process the weir shapes as described in Appendix E. We compare the MSE of the predicted c_D values with those obtained from the simulations, demonstrating the accuracy of the NODE4 predictions. The mean c_D of the FlowProp test set is 0.6712. The MSE achieved using the FC-NN model is 2.8944×10^{-6} and with NODE4 2.6464×10^{-5}. With only a small deviation, NODE4 can also be used to predict the efficiency of a weir in terms of its c_D. Further, our results are represent state-of-the-art results in comparison to (Fatahi-Alkouhi et al., 2024; Iqbal and Ghani, 2024).

4.3. Model Accuracy on Novel Geometries

In the third experiment, we evaluate the NODE's generalization ability with the datasets FlowProp-First, FlowProp-Last and FlowProp-Diverse. This approach allows us to examine NODE's interpolation and extrapolation capabilities in predicting the waterlines and c_D. The results of our best model NODE4 and second best model NODE3 are summarized in Table 2. The small MSE and AD indicate strong generalization and interpolation capabilities, despite the NODE was not being exposed to these specific shapes during training (an example visualization is given in Appendix B). Table 2 lists the MSE of the predicted c_D. The results show that NODE outperforms FC-NN on test sets containing out-of-distribution shapes relative to the training set, suggesting NODE's superior generalization ability by effectively learning the waterline as an ODE.

Table 2: Comparison of the MSE of the predicted and simulated c_D with our best model NODE4 and a FC-NN. NODE4 predicts the c_D more accurately considering the MSE when using untrained geometries for testing. The results of Experiment 2 are also shown for comparison.

	NODE4	**FC-NN**
Test Set	↓ MSE	↓ MSE
FlowProp	9.9456×10^{-6}	2.8944×10^{-6}
FlowProp-First	3.3652×10^{-4}	4.6653×10^{-4}
FlowProp-Last	4.3783×10^{-4}	4.7144×10^{-4}
FlowProp-Diverse	2.2431×10^{-2}	2.2198

Table 3: Results of the waterline generation based on the third experiment. We compare our best model NODE4 of the first experiment with the second best model NODE3. AD is multiplied by 10^4.

Model	**Test Set**	**↓ AD [m²]**	**↓ MSE**
NODE3	FlowProp-First	29.5337	9.9260×10^{-5}
NODE3	FlowProp-Last	56.8599	1.1873×10^{-4}
NODE3	FlowProp-Diverse	173.1117	1.9759×10^{-3}
NODE4	FlowProp-First	6.3794	5.2104×10^{-5}
NODE4	FlowProp-Last	33.4424	5.7798×10^{-5}
NODE4	FlowProp-Diverse	200.6530	2.5601×10^{-3}

5. Conclusion and Future Work

In this study, we evaluated the efficiency and flow properties of varying weir geometries using NODE. We proposed FlowProp, a novel dataset that represents the flow behavior based on different geometric shapes of weirs. Our experiments show that NODE model can accurately describe the water level in the headwater, the overfall height and tailwater with just small deviations. The NODE model accurately predicts the discharge coefficient, closely matching simulated values for both polygonal and non-polygonal weirs. Moreover, the NODE can generalize beyond the training data, although its performance is less accurate in the overfall region and tailwater, indicating room for improvement in handling diverse weir shapes. Furthermore, the transfer to three-dimensional geometries as for example non-linear weirs can be evaluated in a following study. In future work, we compare the performance of the NODE with baseline models such as RNN, LSTM, and GRU to evaluate its effectiveness.

Acknowledgments

This research as part of the project LaiLa is funded by dtec.bw - Digitalization and Technology Reasearch Center of the Bundeswehr which we gratefully acknowledge. dtec.bw is funded by the European Union - NextGenerationEU. In addition, this research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 520460697.

References

D. Aigner and G. Bollrich. *Handbuch der Hydraulik: für Wasserbau und Wasserwirtschaft*. Beuth Verlag, 2015.

H. Bashiri. Towards a new design equation for piano key weirs discharge capacity. *Proceedings of the International Symposium on Hydraulic Structures*, 2016.

L. Besser and M. Oertel. Tailwater influence on downstream flow conditions of piano key weirs. *Proceedings of the 10th International Symposium on Hydraulic Structures*, 2024.

J. C. Butcher. A history of runge-kutta methods. *Applied numerical mathematics*, 20(3): 247–260, 1996.

R. Chen. torchdiffeq, 2018. URL https://github.com/rtqichen/torchdiffeq.

R. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

R. Fatahi-Alkouhi, E. Afaridegan, and N. Amanian. Discharge coefficient estimation of modified semi-cylindrical weirs using machine learning approaches. *Stochastic Environmental Research and Risk Assessment*, pages 1–22, 2024.

R. Ferguson. Hydraulics and hydraulic geometry. *Progress in Physical Geography*, 10(1): 1–31, 1986.

Inc. Flow Science. *FLOW-3D, Version 2023R1*. Santa Fe, NM, 2023. URL https://www.flow3d.com/.

A. Guven. A multi-output descriptive neural network for estimation of scour geometry downstream from hydraulic structures. *Advances in Engineering Software*, 42, 2011.

M. Iqbal and U. Ghani. Prediction of the discharge capacity of piano key weirs using artificial neural networks. *Journal of Hydroinformatics*, 26(5):1167–1188, 2024.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

K. Lee and E. Parish. Parameterized neural ordinary differential equations: Applications to computational physics problems. *Proceedings of the Royal Society A*, 477(2253):20210162, 2021.

X. Li, Q. Zhu, C. Zhao, X. Duan, B. Zhao, X. Zhang H. Ma, J. Sun, and W. Lin. Higher-order granger reservoir computing: simultaneously achieving scalable complex structures inference and accurate dynamics prediction. *Nature Communications*, 15(1):2506, 2024.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, pages 3276–3285. PMLR, 2018.

A. Norcliffe, C. Bodnar, B. Day, J. Moss, and P. Liò. Neural ode processes, 2021. URL https://arxiv.org/abs/2103.12413.

M. Oertel and D. Bung. Numerische strömungssimulation von fließgewässern – wo stehen wir und was werden wir im laufe der nächsten jahre erreichen? *Wasserbauliche Mitteilungen Institut für Wasserbau und Technische Strömgsmechanik, TU Dresden*, 50, 2014.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang J., Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

A. Safarzadeh and B. Noroozi. 3d hydrodynamics of trapezoidal piano key spillways. *International Journal on Civil Engineering*, 2017.

S. Salih, M. Habib, I. Aljarah, H. Faris, and Z. Yaseen. An evolutionary optimized artificial intelligence model for modeling scouring depth of submerged weir. *Engineering Applications of Artificial Intelligence*, 96, 2020.

X. Shen and M. Oertel. Influence of piano key weir crest shapes on flow characteristics, scale effects, and energy dissipation for in-channel application. *Journal of Hydraulic Engineering*, 149(6), 2023.

T. Strobl and F. Zunic. *Wasserbau Aktuelle Grundlagen - Neue Entwicklungen*. Springer, 2006.

C. Thorenz. Boundary conditions for hydraulic structures modelling with openfoam. *Proceedings of the 10th International Symposium of Hydraulic Structures*, 2024.

D. Valero and D. Bung. Interfacial velocity estimation in highly aerated stepped spillway flows with a single tip fibre optical probe and artificial neural networks. *Proceedings of the 6th IAHR IJREWHS 2016*, 2016.

P. Virtanen, R. Gommers, T. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J.

Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Ee Weinan. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 1(5):1–11, 2017.

Z. Yuhong and H. Wenxin. Application of artificial neural network to predict the friction factor of open channel flow. *Commun Nonlinear Sci Numer Simulat*, 14, 2009.

Appendix A. Dataset Properties

A.1. Simulation Parameters

Table 4 summarizes the parameters and boundary conditions set and retrieved from the numerical simulations in FLOW 3D. All simulations hat the same setting accept for the weir geometry. In total the numerical models contained of 51,000 cells. To optimize the simulation an initial water body was included in the model.

Table 4: Simulation parameters and boundary conditions.

Sim	**Inlet BC**	**Oulet BC**	**Cell size**	**Sim time**	**Geometry**	**Comp time**
			[m]	[s]		[hh:mm:ss]
RECT	Flow rate	Continuative	0.005	28	Rectangle	01:08:53
TRIA	Flow rate	Continuative	0.005	28	Triangle	01:07:35
INT1	Flow rate	Continuative	0.005	28	Trapezoid 1	01:08:56
INT2	Flow rate	Continuative	0.005	28	Trapezoid 2	01:09:17
INT3	Flow rate	Continuative	0.005	28	Trapezoid 3	01:16:08
INT4	Flow rate	Continuative	0.005	28	Trapezoid 4	01:09:22
INT5	Flow rate	Continuative	0.005	28	Trapezoid 5	01:08:33
INT6	Flow rate	Continuative	0.005	28	Trapezoid 6	01:10:22
INT7	Flow rate	Continuative	0.005	28	Trapezoid 7	01:09:54

A.2. Initial Rectangle Simulation

Figure 4 shows the simulation results of initial rectangular geometry (ge_{rec}) with development of free surface elevation (h_s), flow velocities (v_s) and total energy H_s.

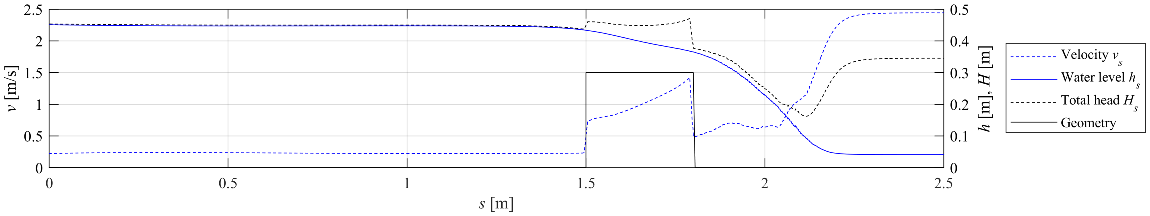

Figure 4: Flow properties (free surface elevation h_s, flow velocities v_s, total energy H_s) of the rectangular geometry ge_{rec} with flow direction from left to right.

A.3. Linear Correlation

We examine the linear correlation between the initial geometries (ge_{tri}), ge_{rec} and additional simulations of linear interpolated geometries (ge_n) between ge_{tri}, ge_{rec} and the resulting h_s. With a linear fitting equation

$$h_s = 0.1107 \cdot C + 0.4192 \tag{8}$$

the linear correlation can be proven with an uncertainty of less than 2 % as shown in Figure 5. The factor C represents the crest length in flow direction.

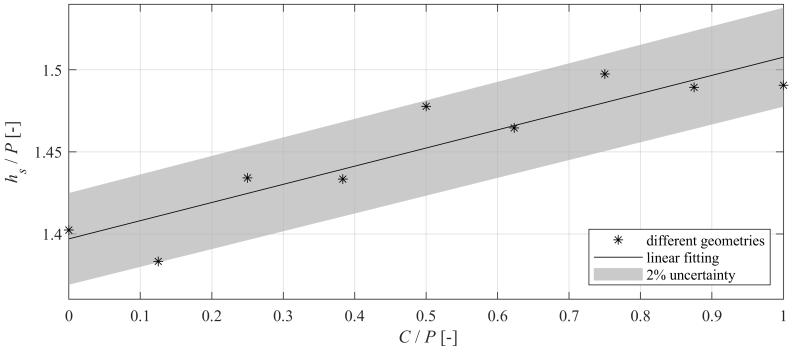

Figure 5: Linear fitting equation for interpolated geometries ge_n between the rectangular geometry ge_{rec} and triangular geometry ge_{tri} and the upstream flow depths h_s describing an approximated linear correlation.

A.4. Interpolation Result

Each simulation can be decomposed into the weir geometry with the same position and the course of h_s. The initial geometries ge_{tri}, ge_{rec} are described by four vertices, whereby the upper vertices of ge_{tri} share the same position. The waterline from the simulation is described by a large number of data points with their respective coordinates. The interpolated geometries ge_n of the simulation result in symmetric trapezoids with different base sizes, which can be merged with the interpolated waterlines to create new simulations. Figure 6 shows the mean interpolated geometry between ge_{rec} and ge_{tri}.

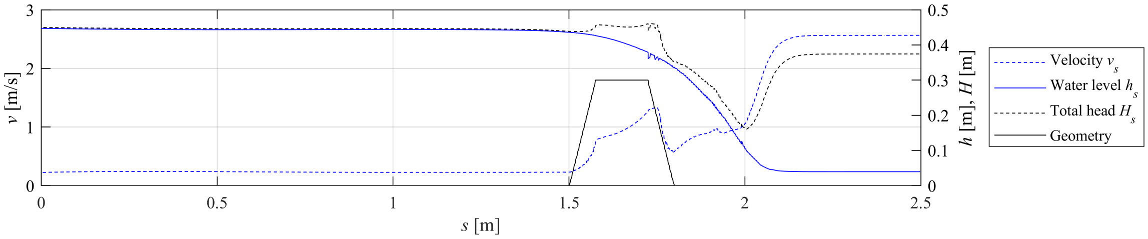

Figure 6: The mean interpolated trapezoidal geometry between the rectangular geometry ge_{rec} and triangular geometry ge_{tri}.

Appendix B. Examples

Figure 7 shows the failed reconstruction of NODE1 with 5 hidden layers and 100 neurons in each hidden layer. NODE1 cannot accurately reconstruct the overfall region and the tailwater of the waterline. The course of the waterline in the tailwater region rises upward.

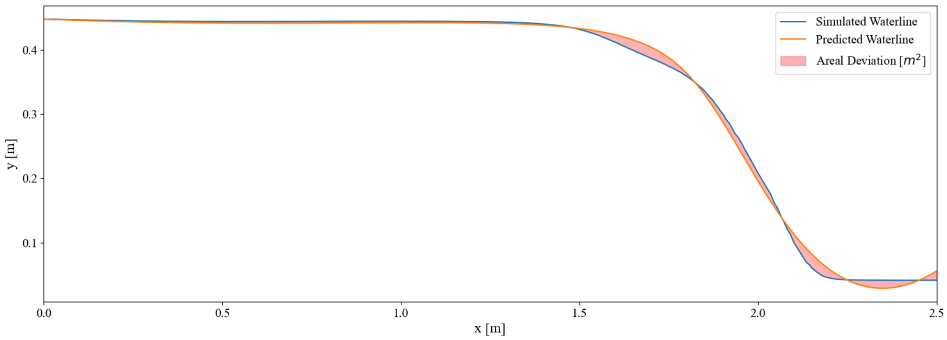

Figure 7: Failed reconstruction of NODE1. The waterline in the tailwater moves upwards.

Figure 7 shows the failed reconstruction of NODE6 with 30 hidden layers and 100 neurons in each hidden layer. NODE6 fails to reconstruct the overfall region and the tailwater of the waterline. After the overfall the waterline follows an almost linear downward trend.

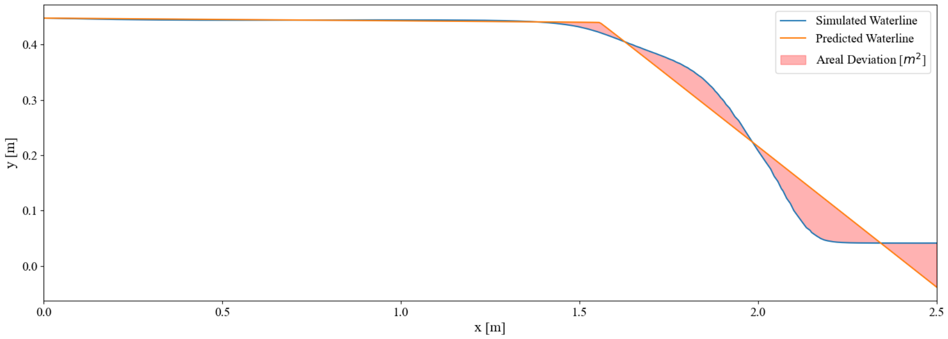

Figure 8: Failed reconstruction of NODE6. The waterline after the overfall moves downwards without reconstructing the tailwater region.

In Figure 9 the waterline shows failed accuracy in the overfall region when doing the third experiment.

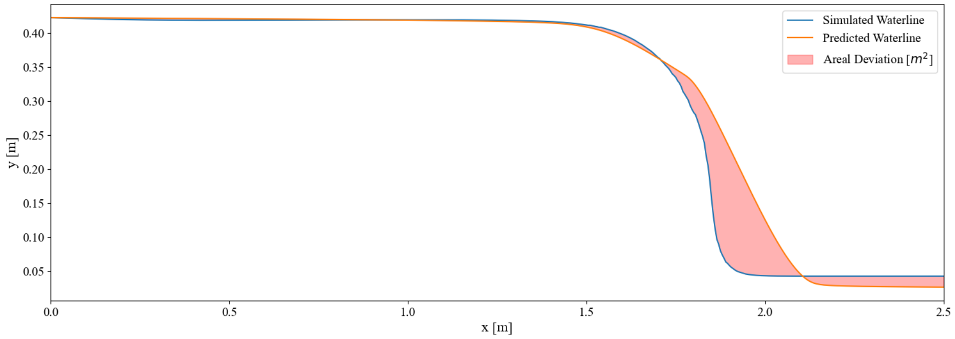

Figure 9: Prediction of the waterline of a circular weir geometry of FlowProp-Diverse. The overfall region cannot be predicted accurately.

Appendix C. Derivation of the Discharge Coefficient

Given the classical discharge Equation (1) the total energy (H_s) is calculated using the Bernoulli equation with k as the overfall height over the weir structure:

$$H_s = \frac{v_s^2}{2g} + k \tag{9}$$

The discharge (Q) can be described by the continuity equation (Ferguson, 1986) with v as the upstream averaged flow velocity and A the flow cross section as:

$$Q = v_s \cdot A \tag{10}$$

The flow cross section A is calculated from h_s as the water level and b the width of the cross sections:

$$A = b \cdot h_s \tag{11}$$

Equation (10) can be combined with Equation (11) and solved for v_s:

$$v_s = \frac{Q}{b \cdot h_s} \tag{12}$$

To finally determine the discharge coefficient c_D, Equation (9) is substituted into Equation (1) and solved for c_D. Equation (12) replaces the need to directly measure v_s. This results in the Equation (2).

To determine c_D of a weir, we measure a reference free surface elevation (h_0) at $3 \times P$ as described in Section 1, resulting in a reference point at 0.6 m. c_D can then determined by inserting h_0 for h_s in Equation (2).

Appendix D. Additional Dataset

Figure 10 shows six non-polygonal weirs that are part of the FlowProp-Diverse dataset. We used this dataset to show that our approach can be applied to shapes that are outside the distribution of the initial polygonal training samples.

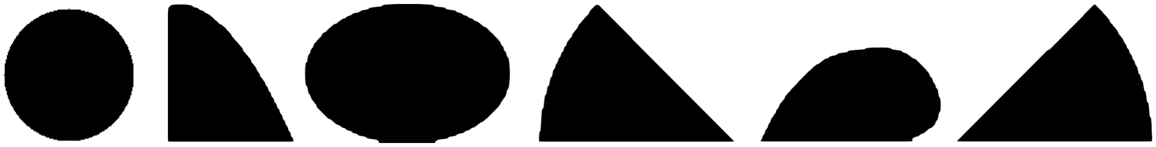

Figure 10: Six additional weirs of the dataset FlowProp-Diverse. From left to right: Circle, Ogee, Ellipsoid, Tilting Left, Hose and Tilting Right.

Appendix E. Model Architecture

E.1. Encoder enc_G

The encoder, denoted as enc_G, processes input images with 128×128 pixels. It employs three convolutional layers with increasing filter sizes of 1, 16, 32, and 64, respectively, each with a kernel size of 3×3. ReLU activation is applied after each convolutional layer to introduce non-linearity. To downsample the feature map and emphasize important features such as edges, we apply max pooling after each convolutional layer. Max pooling is particularly effective in capturing the outer shape of weir geometries, which is crucial for this task. The final layer is fully connected to compresses the extracted features into a representation of 8 neurons.

E.2. Neural Ordinary Differential Equations NODE

Our NODE model utilizes $\mathrm{enc}_{\mathrm{G}}$ to process the images of the weir geometrics. The NODE architecture includes an ODE component. This component is a neural network consisting of an input layer, hidden layers, and an output layer. It has 20 hidden fully connected layers, each with 100 neurons and an ELU activation function. It uses as ODE solver the Runge-Kutta of order 5 of Dormand-Prince-Shampine. The NODE is trained with the Adam optimizer and a learning rate of $1e-3$, which get reduced by a factor of 0.5 after $1,000$ epochs. The total trainable parameters are $347,477$.

E.3. Fully Connected Neural Network FC-NN

The FC-NN model employs $\mathrm{enc}_{\mathrm{G}}$ for processing geometric shape images. Therefor, the input dimension is 8. It consists of 20 hidden fully connected layers, each with 100 neurons, followed by an ELU activation function with an total of $347,277$ trainable parameters. The FC-NN is trained using the Adam optimizer with the default learning rate of $1e-3$ until convergence.