Transformer-Based Modular Modeling Scheme for Digital Twin

Anonymous Authors[1]

Abstract

Our article presents a novel data-driven master Digital Twin (DT) modeling scheme for large hydraulic networks in a static regime. We bring two main contributions in this paper: (i) an input embedding pipeline for 0-D physical systems with a finely customized multidimensional encoding of physical values; (ii) a Stochastic Padding Augmentation (SPA) training method that addresses the challenge of modular inference for serial physical systems. In one representative test case of free serial pump concatenation, our innovative approach reduces the normalized prediction MAE (Mean Absolute Error) of variable pressure from 13.95% to 0.24%. This work paves the way for a real master DT modeling scheme that allows parallel concatenation of physical units and long-range prediction.

Notations

Given $n \in \mathbb{N}$, we denote $[n] = \{1, \dots, n\}$. Given $x \in \mathbb{R}$, we denote $\lfloor x \rfloor$ its integer part and $\lceil x \rceil = \lfloor x \rfloor + 1$. When performing computation in $\mathbb{R}^n$, given $i \in [n]$, the notation e_i designates the one-hot vector of $\mathbb{R}^n$, full of zeros and with only 1 in the i^{th} entry. The ℓ_1 norm on $\mathbb{R}^n$ is denoted by $\| \cdot \|_1$ ($\forall x \in \mathbb{R}^n$: $\|x\|_1 = \sum_{i=1}^{n} |x_i|$.

1. Introduction

The term Digital Twin (DT) was first introduced by Michael Grieves and John Vickers of NASA in the context of product management (2019), although early implementations of a similar concept date back to the 1960s, particularly in NASA's Apollo 13 program (Zhang et al., 2021). A Digital Twin can be interpreted as a virtual counterpart to a physical entity (Semeraro et al., 2021; Tao et al., 2022; Jones et al., 2020; Tao et al., 2018). It allows for *ultra-high fidelity simulation* of the status and behavior of the physical entity, providing significant value in product management, maintenance, safety, and reliability (Glaessgen & Stargel, 2012). Over the past two decades, DTs have found their primary applications in manufacturing and are scaling to various domains such as energy, aerospace and healthcare (Tao et al., 2022; Yu et al., 2022; Jones et al., 2020; Tao et al., 2018). Recently, as environmental challenges grow, digitalization has become an important green approach in the energy sector (Yu et al., 2022). Energy Digital Twins (EDTs) enable effective system management, better asset performance, and optimal decision-making, which can help reduce energy consumption and environmental impact (Yu et al., 2022; Do Amaral et al., 2023; Ghenai et al., 2022).

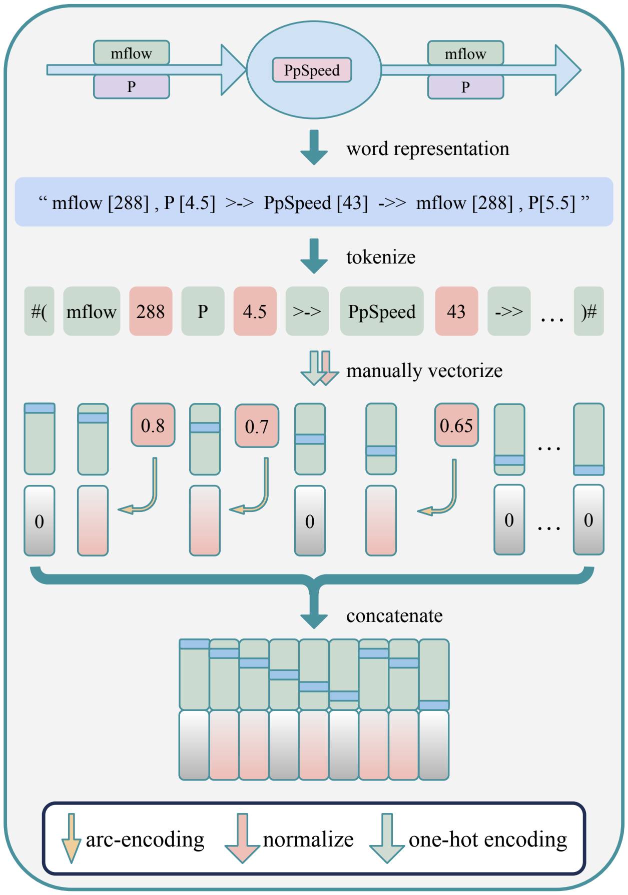

Figure 1. Proposed input representation pipeline to transform static system status into Transformer input. Arc-encoding transforms scalar values into a 1D vector.

We focus on the virtual counterpart modeling and calibration

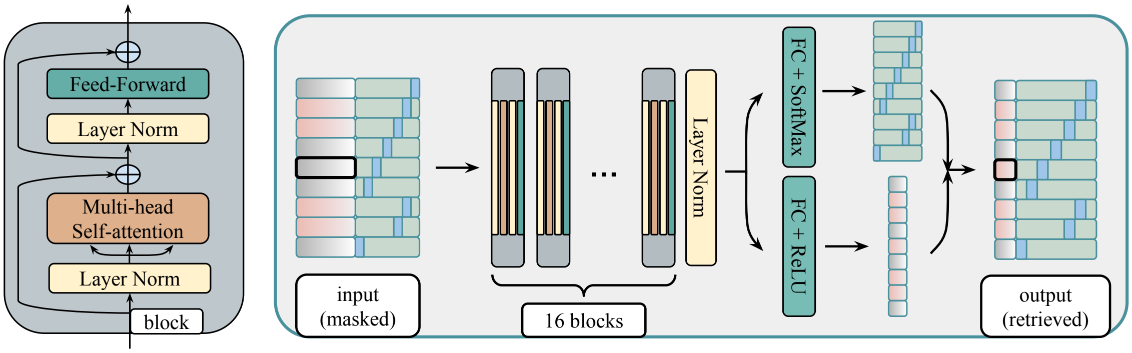

Figure 2. Model structure and input flow. The model receives Arc-encoded inputs and outputs float-encoded system status. For training and tests, several physical values will be masked and retrieved by the model. The one-hot encoded part is expected to be retained.

aspect of large hydraulic networks, such as those used in cooling and heating systems. Our primary goal is a master model that predicts internal thermodynamic variables (such as pressure, temperature, volumic flow) between different pieces of equipment (referred to as *units*), such as pumps, exchangers, chillers, etc. As for current progress, we consider static systems where no time variable is involved: the system's solution is that at the infinity of time.

Currently, conventional DT construction of such systems is delivered first by manual decomposition of the system. Then, modeling is done through interface-assisted software. Each physical unit presents a set of first-principles equations which describe its behaviors. Several parameters are left out and only confirmed during calibration, upon which the model really becomes a *twin* of the unit. With some numerical techniques, the software is then able to solve the system of differential equations together and produce a solution.

An intuitive way to turn to neural networks (NN) approach is to model the finite number of minimal physical units with multi-layer perceptrons (MLPs), then concatenate accordingly to cases. However, this solution is also heavily tailor-made. Besides, the input/output correspondences of each MLP limit the flexibility of the entire model. In particular for calibration and optimization tasks, one might need to treat some input variables as outputs and vice-versa, but that is not allowed by such method.

Transformer (Vaswani, 2017) as a NN architecture, differs from many of the more traditional ones in a way that it does not presume a fixed input length, making it applicable to systems of virtually any size. In this work, we treat sequences of physical variables and values in the same manner as words are tokenized in large language models (LLMs). We propose a Transformer-based approach to create a master twin model. We apply BERT (Devlin, 2018)-like training, which enables a bidirectional Transformer to learn the distribution of valid system statuses. The model is trained to map any incomplete system status to a valid static solution, while enough information is given for the solution to be unique. This approach also allows fast calibration, which is interpreted as a special case of predictive inference. Once established, this method can be used for efficient predictive maintenance (Fault Detection) and energy cost estimation.

It is probably trivial to learn such distribution for any single unit. However, the unlimited free assembly between physical units can be a major challenge. We specifically desire *modular inference*, that is, concatenation of new tokens to the sequence of a trained scenario should only bring little effect on the outcome of the trained scenario. Staying with a small training scheme, we intuitively interpret Attention mechanism as a convex combination operation and propose a data-augmentation method to enable such modular inference.

In this paper, we propose 2 techniques for Transformer-based modular modeling of physical systems: (i) Arc-encoding (ARC), which encodes scalar physical values in a multidimensional manner; (ii) Stochastic Padding Augmentation (SPA), which samples tokens from the input manifold for padding and forgoes mask-filling with large negative values. The contributions of our work are as below:

1. A noval tokenization-embedding pipeline to encode physical system status for Transformer modeling. This input representation differs from casual LLM tokenization convention, which faces the challenge of properly tokenizing numbers.
2. A bidirectional Transformer model that accurately performs system status predictions within a 0-D and static regime. The calculation speed is higher than traditional software solvers.
3. A data-augmentation method based on the intuition

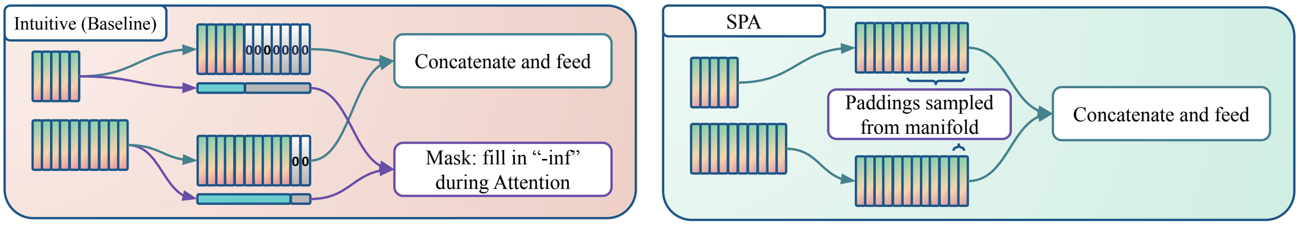

Figure 3. Our proposed Stochastic Padding Augmentation (SPA). In SPA, we pad with representations sampled from the input manifold and forgo padding mask-filling with large negative value.

that Attention mechanism is a convex combination among projections of visible tokens. We show valid performance improvements brought by this training scheme in terms of modular inference in a short-range prediction task.

2. Related Work

BERT (Devlin, 2018) proposed two pretraining tasks for language models of Transformer structure exploiting unlabeled text data. One of the tasks is to fill in masked words in a sentence. This permits the model to learn the distribution of "valid sentences". Our work applies this idea on physics systems. The model is trained to learn the distribution of "valid static solutions of systems", but in a more deterministic manner.

Energy Digital Twin Numerous numerical tools are available to produce EDTs. A distinction can be made between two main categories: (i) physical equation-based methods (*first-principles* methods) and (ii) data-driven methods (Yu et al., 2022). The first category is widely developed by software providers and engineering simulation companies. In our application domain, two significant drawbacks of the conventional approach are the tailor-made modeling process and the substantial time required to solve differential equations. The time complexity grows exponentially with the size of the system. We hope to cope with these problems with data-driven approach.

Transformer for Twin We found several adaptations of Transformer to Digital Twin modeling. However, most of these works demonstrated Transformer's performance as a tailored twin (Zhao, 2024; Rosyadi et al., 2024; Sun et al., 2023; Sha et al., 2023; Hou et al., 2023). Some leaned more on the aspect of product management for DT: (Sun et al., 2023; Praharaj et al., 2024) focused on model lightweight and (Wang et al., 2023) proposed a lifelong learning scheme. Additionally, (Lin et al., 2024) transformed physics data into word tokens and applied trained LLMs for zero-shot inference. We notice that it is a common method to apply 1-D convolutional layers (CNN) to handle spatial inter-token information (Zhao, 2024; Rosyadi et al., 2024; Praharaj et al., 2024).

Relative Positioning Encoding (Dufter et al., 2022) surveyed position information handling in Transformers and classified common approaches into 2 main categories: Adding Position Embeddings (APE) and Modifying Attention Matrix (MAM). MAM allows a position-invariant injection of relative position information, which is crucial for modular inference, therefore is adopted in this work.

3. Method

3.1. Baseline

Model Our baseline model is a multi-layer bidirectional Transformer encoder (Devlin, 2018) which much respects the original Transformer structure (Vaswani, 2017) except for the adoption of pre-layer normalization (Pre-LN) (Xiong et al., 2020). As shown in Figure 2, the model contains 16 layers of multi-head self Attention and feed-forward layer of factor 4. The model size d_{model} is the size of input representations.

Input representation The process is shown in Figure 1. We first write the status of a physical system into a word sequence, then one-hot encode the sematic elements according to a manual dictionary. This is feasible for large hydraulic networks because the number of physical units is finite and very limited. The numeric elements are normalized, Arc-encoded (Section 3.2), then concatenated to its corresponding semantic part. Each column of the input representations has 2 segments: one-hot segment and numeric segment. For those tokens without a scalar value attached (e.g. BOS, EOS, arrows indicating flow direction), the numeric segment is set to 0 vector.

After turning a system status into matrix representation, the input is padded with SPA (Section 3.3). Our input is 3-dimensional: $[B, T, d_{model}]$ where B is batch size, T is the number of tokens (*SPA max-length*), $d_{model} = d_{one\_hot} + d_{arc}$ is the total size of the 2 segments.

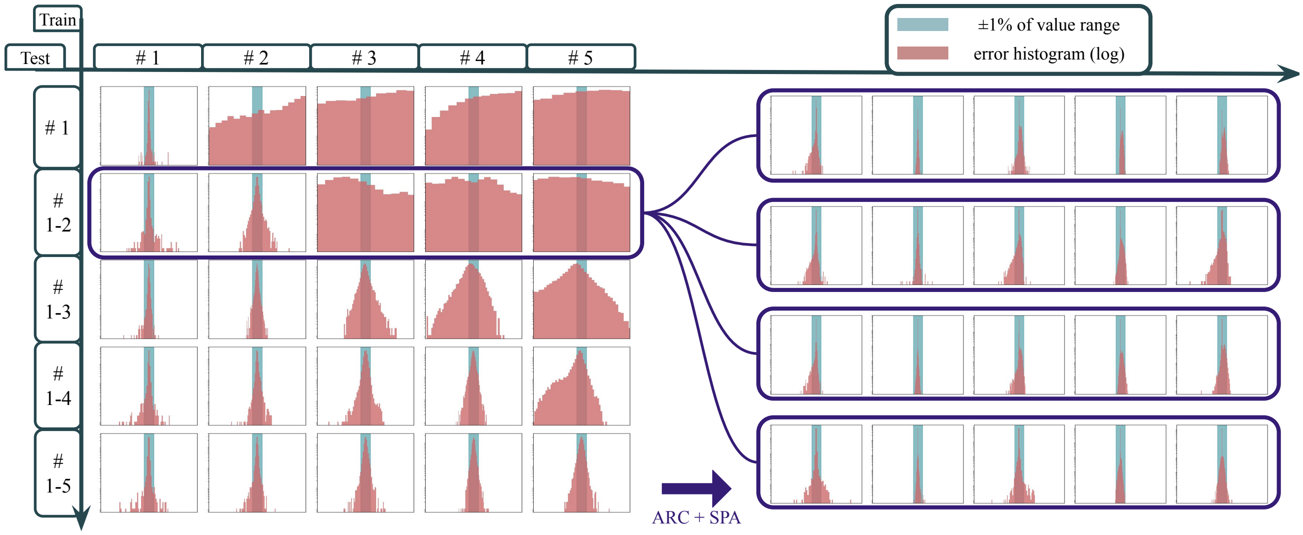

Figure 4. The error histogram of randomly dropped *pressures* under different train-test configurations. On the left is baseline method, different rows signify different training data and different columns signify different test data. By descending on the figure, the training scenarios get enriched in an accumulative way. On the right is same structure histogram with proposed method under the configuration of training on #1-2. The 4 rows represent 4 different SPA max-lengths: 30, 50, 80, 120. Our proposed method allows the model to infer the behaviors of each physical unit in a modular manner. The demonstrated error distribution gives an impression as if such test scenario had been "seen" during training.

Positioning handling We apply Modifying Attention Matrix (MAM) (Dufter et al., 2022) with an invariant multiplicative effect on the weight matrix right ahead of softmax. This MAM enforces the far-apart tokens to much less participate in the computation of each other, unless their conveyed information gradually fuses throughout multiple layers. In this work, our multiplicative MAM factor quickly decays to near-0 values starting from the fifth token distance. This MAM adapts especially to short-range learning.

Training Our training borrows the idea of word masking in BERT. Random physical variable values are masked with NA representation in input and retrieved by the model. We apply Cross-Entropy loss for the one-hot segments and L1 (MAE) loss for numeric segments.

3.2. Arc-Encoding

Our method proposes an innovative way to encode physical variables more efficient than simple 0-1 normalization. It presents three interesting aspects.

1. It embeds 0-1 values in a multi-dimensional space $\mathbb{R}^{d_{\mathrm{Arc}}}$ (and not just in a two dimensional subspace of $\mathbb{R}^{d_{\mathrm{Arc}}}$ which would be the naive solution) such that projections performed during the Attention mechanism will possibly produce d_{Arc} incoherent features to encode the whole range of values.

2. The norm of the embedding is constant equal to 1. Interestingly enough, we found out that for this task, the performances were optimal when conditioning on the ℓ_1 norm $\|\cdot\|_1$ and not on the ℓ_2 norm.

3. It provides a clever way to encode NA values (for unknowns values, arrows, BOS, EOS) as $0 \in \mathbb{R}^{d_{\mathrm{Arc}}}$, positioning them equidistantly from all other values in $\mathrm{Arc}([0,1])$ (in ℓ_1 norm).

4. Distances are preserved such that the embedding of two close (resp. far) values will stay close in $\mathbb{R}^{d_{\mathrm{Arc}}}$ (resp. far in $\mathbb{R}^{d_{\mathrm{Arc}}}$)

The definition of the embedding Arc is defined followingly:

$$\begin{aligned} \mathrm{Arc} : [0,1] &\longrightarrow \quad \mathbb{R}^{d_{\mathrm{Arc}}} \\ t \quad &\longmapsto \quad \Phi(e^{\log(d_{\mathrm{Arc}})t}) \end{aligned}$$

where

$$\begin{aligned} \Phi : [1, d_{\mathrm{Arc}}] &\longrightarrow \quad \mathbb{R}^{d_{\mathrm{Arc}}} \\ x \quad &\longmapsto \quad \frac{1}{x}\left(\sum_{i=1}^{\lfloor x \rfloor} e_i + (x - \lfloor x \rfloor) e_{\lceil x \rceil}\right). \end{aligned}$$

Note in particular that for all $j \in [d_{\mathrm{Arc}}]$, $\Phi(j) = \frac{1}{j}\sum_{i=1}^{j} e_i$. The first two aspects mentioned above are clearly satisfied by construction of Arc since we see that the family of vectors

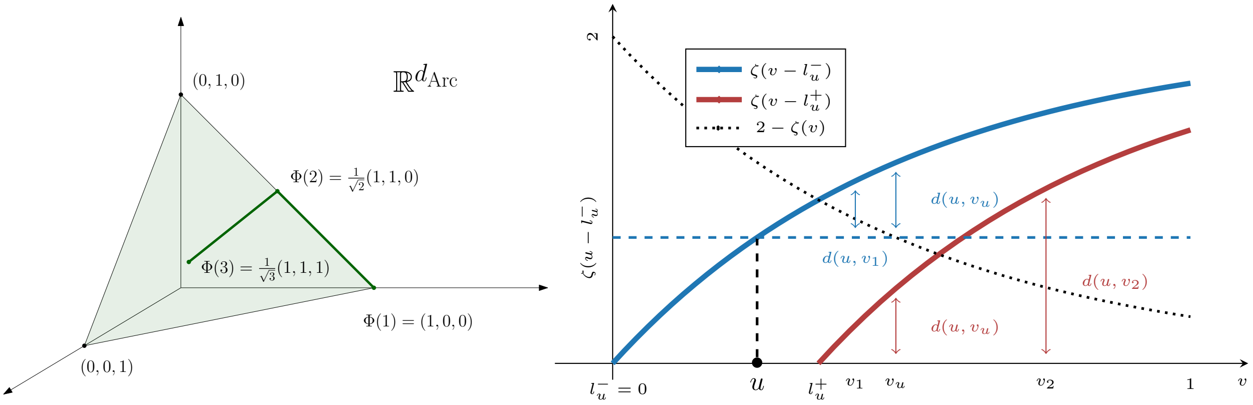

Figure 5. **(Left)** Representation of the image projected by the mapping Φ when $d_{\mathrm{Arc}} = 3$. **(Right)** Graphical computation of $d(u,v)$ for $d_{\mathrm{Arc}} = 7$, $u \in [0,1]$ and different values of $v \in [u,1]$: $v_1 < v_u = \zeta^{-1}(2 - \zeta(u - l_u^-)) < v_2$.

$(\mathrm{Arc}(0), \ldots, \mathrm{Arc}(1)) = (\Phi(1), \ldots, \Phi(d_{\mathrm{Arc}}))$ are linearly independent and $\forall x \in [1, d_{\mathrm{Arc}}]$, $\|\Phi(x)\|_1 = \frac{\lfloor x \rfloor}{x} + \frac{x - \lfloor x \rfloor}{x} = 1$.

Let us then see how the distance is treated. If $\lfloor x \rfloor < \lfloor y \rfloor$, one can compute:

$$\begin{aligned}
&\|\Phi(z) - \Phi(x)\| \\
&\quad = \lfloor x \rfloor \left(\frac{1}{x} - \frac{1}{y}\right) + \left|\frac{1}{y} - \frac{x - \lfloor x \rfloor}{x}\right| + \frac{\lfloor y \rfloor - \lfloor x \rfloor - 1}{y} + \frac{y - \lfloor y \rfloor}{y} \\
&\quad = \begin{cases} \dfrac{2\lfloor x \rfloor}{x}\left(1 - \dfrac{x}{y}\right) & \text{if } \dfrac{x - \lfloor x \rfloor}{x} \leq \dfrac{1}{y} \\ 2\left(1 - \dfrac{\lfloor x \rfloor + 1}{y}\right) & \text{if } \dfrac{x - \lfloor x \rfloor}{x} \geq \dfrac{1}{y} \end{cases} \qquad (1)
\end{aligned}$$

Note that both quantities are equal when $\frac{x - \lfloor x \rfloor}{x} = \frac{1}{y}$. Besides, if $\lfloor x \rfloor = \lfloor y \rfloor$, then $\|\Phi(z) - \Phi(x)\| = \frac{2\lfloor x \rfloor}{x}\left(1 - \frac{x}{y}\right)$ (it is not surprising since simple functional analysis allows to show that $\frac{x - \lfloor x \rfloor}{x} \leq \frac{1}{\lfloor x \rfloor + 1} \leq \frac{1}{y}$, the first condition is thus satisfied). For simplicity, for $u \in [0,1]$, and $v \in [u,1]$, we denote:

$$l_u^- \equiv \frac{\log(\lfloor e^{u \log d_{\mathrm{Arc}}} \rfloor)}{\log d_{\mathrm{Arc}}} \leq u \leq l_u^+ \equiv \frac{\log(\lfloor e^{u \log d_{\mathrm{Arc}}} \rfloor + 1)}{\log d_{\mathrm{Arc}}}$$
$$\text{and: } d(u,v) \equiv \|\mathrm{Arc}(u) - \mathrm{Arc}(v)\|.$$

As a direct consequence of (1), one can show:

Lemma 3.1. *The mapping* Arc *satisfies for all* $u \in [0,1]$ *and and any* $v \in [u,1]$*:*

$$d(u,v) = \max\left(\zeta(v - l_u^-) - \zeta(u - l_u^-), \zeta(v - l_u^+)\right)$$
$$\textit{where: } \zeta : t \mapsto 2(1 - e^{-t \log(d_{\mathrm{Arc}})})$$

In particular, if $l_u^- = u$, $d(u,v) = \zeta(v - u)$.

Figure 5 represents graphically the computation of the distance $d(u,v)$ that depends on the position of v towards $v_u \equiv \zeta^{-1}(2 - \zeta(u - l_u^-))$. One can check that the proximity relation is preserved, meaning that $|u - v| \leq |u' - v'| \iff d(u,v) \leq d(u',v')$.

3.3. Obstacle for Modular Inference

Recall the original attention equation in (Vaswani, 2017): $\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_K}})V$, where Q, K, V are projections by 3 matrices of the input X at current layer. We notice that the final stage of attention is matrix multiplication between a row-normalized[1] weight with V, which means each output row of attention is a convex combination of rows of V. Denote N the number of visible tokens in an inference case, denote $V = [v_1, \cdots, v_N]$ where v_i^T is i^{th} row of V, we have the j^{th} token's output row as

$$[\mathrm{Attention}(Q,K,V)]_j = \sum_{i=1}^{N} \sigma_{ji} v_i, \quad \sum_i \sigma_{ji} = 1; \quad (2)$$
$$\sigma_{ji} \geq 0, \quad \forall i \in [N]$$

where σ_{ji} is given by j^{th} row of Q and all rows of K.

Our goal is to enforce the model to output invariant outcome for "mastered" sequence of tokens while more tokens are concatenated at input. Suppose the model has already perfect performance on the above N-token sequence. Now we concatenate M new tokens to the N-token sequence, the j^{th} token's outcome in (2) becomes:

$$[\mathrm{Attention}(Q,K,V)]'_j = \sum_{i=1}^{N} \sigma'_{ji} v'_i + \sum_{m=1}^{M} \sigma_{jm} v_m,$$
$$\sum_i \sigma'_{ji} + \sum_m \sigma_{jm} = 1;$$
$$\sigma'_{ji} \geq 0, \quad \forall i \in [N]; \quad \sigma_{jm} \geq 0, \quad \forall m \in [M]$$

[1] "Normalized" in a way that each row has a 1-norm of 1.

where $\{\sigma_{jm}\}_m$ are by construction non-negative, due to which every element of $\{\sigma'_{ji}\}_i$ suffered a decay. It is intuitive that the new outcome of the N-token sequence will be affected. Although we apply a decaying multiplicative factor by MAM for far-apart tokens, we still do not even loosely have the promise of robustness.

To enable this robustness, we propose a data augmentation method: Stochastic Padding Augmentation (SPA) as shown in Figure 3. With SPA, we forgo padding masking (mask-filling with large negative values) during attention. We preserve a large enough padded input length (we refer to as *SPA max-length*) and sample token presentations from the valid input manifold to pad. More precisely, every sampled padding vector is the concatenation of a random one-hot vector and the Arc-encoding of a normalized random scalar value.

By fixing a same SPA max-length during training and tests, denoted as $N+M$, the j^{th} token's outcome in (2) is now

$$[\text{Attention}(Q,K,V)]_j = \sum_{i=1}^{N} \sigma_{ji} v_i + \sum_{m=1}^{M} \sigma_{jm} v_m$$

during both training and inference. Without SPA, the part $\sum_{m=1}^{M} \sigma_{jm} v_m$ would be zero projections induced by zero padding, which brings bias to this combination. Our heuristic relies on the idea that the stochastic similarity between SPA tokens and actual tokens will provide robustness to the Attention mechanism's centroid estimation through padding.

4. Experiments

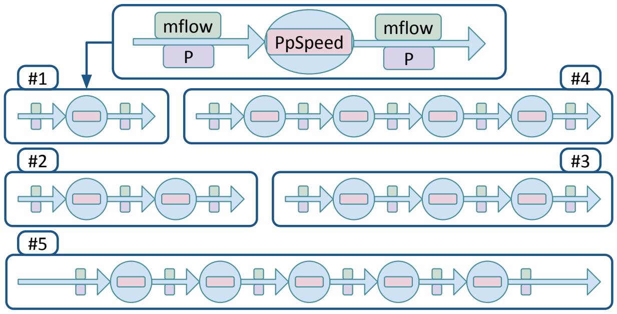

Figure 6. The serial unit concatenation scenarios in our toy example. The most exploited case is to train on scenario #1-2 and test on scenario #5.

In this section, we use a toy example to demonstrate our proposed methods: serial concatenation of pumps. This toy example naturally extends to larger problems because the number of physical units is finite in hydraulic networks. Therefore, the number of involved physical variables and targeted physical relations is also finite. Adapting to all physical units simply involves introducing additional variable names with corresponding data, and potentially utilizing a larger Transformer.

Our main focus is to train on simpler scenarios and test on more complicated ones. As shown in Figure 6. We refer to unit concatenation of 1, 2, $\cdots$, 5 pumps as scenario #1, #2, $\cdots$, #5. For each scenario, the number of valid input tokens is different. For example, scenario #1 presents 9 valid tokens and #2 presents 14. Under human logic, if a model has mastered the functionality of #1, then it is reasonable that it knows to solve #2 (under the condition that each minimal truncation of a pump has unique solution, which corresponds to the complexity that #1 dataset can offer), since #2 is the twice modular concatenation of #1. However, the baseline model would fail since it is used to dealing with only 9 visible tokens. In this section, we will demonstrate the improvement of our approach in a short-range prediction manner (with respect to the complexity that #1 dataset offers).

4.1. Model Optimization

The Transformer model parameters are initialized with normal distribution $\mathcal{N}(0, 0.02)$. We adapted AdamW optimizer. Each configuration is trained with 3 trials and for each trial 240 epochs with decaying learning rates. The trial with best MAE for masked numeric values is chosen for final representation.

4.2. Data

There are three physical variables to be masked and retrieved in this problem (Table 1).

Table 1. Three types of involved variables and their value range. The unit "rpm" stands for "rotation per minute".

Abbr.	Variable Name	Value Range	Unit
P	pressure	1-6	bar
mflow	volumic flow rate	5-360	L/s
PpSpeed	rotational speed of pump	30-50	rpm

The 5 concatenation situations in Figure 6 involve a same pump, whose functioning nature can be described as :

$$\mathrm{P_{out}} - \mathrm{P_{in}} = f(\mathrm{mflow}, \mathrm{PpSpeed}, \mathrm{params})$$

where P_{in} and P_{out} respectively signify pressure before and after the pump. The intrinsic parameters are omitted in experiments since they remain identical. We generate 500k synthetic data of up to 5 concatenations of this pump with a physics equation-based model, then filter with valid value ranges and obtain 5 datasets. For training and test data, one of the three types of variables is randomly dropped at various positions, respecting the rule that the minimal truncations of each pump all present enough information for a unique solution.

4.3. Comprehensive Test Case Example

For a comprehensive view of our predictive inference, we demonstrate a test case of our proposed method in Figure 7. In this test case, we manually dropped 3 types of values at 4 positions at once. We specify that the model is trained on data of scenario #1 and #2, input structures of which at best represent less than half of that of the test input.

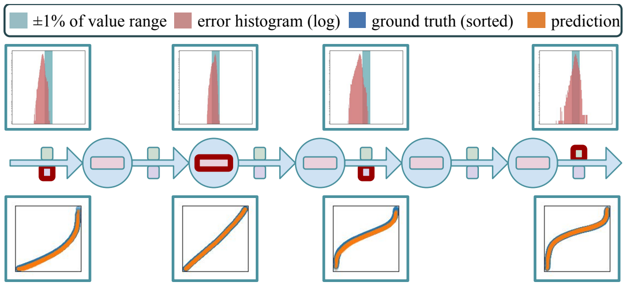

Figure 7. A test on scenario #5 while the model is only trained on scenario #1 and #2: on 20K test data, P^0, $\mathrm{PpSpeed}^1$, P^3, and mflow^5 are simultaneously masked for the model to retrieve. On the top line, each error histogram (pink) is computed with 20K error values. The green band signifies ± 1% of the value range. As for each figure on the bottom line, the blue points are 20K ground truth values, sorted in ascending order; the orange points are the corresponding predictions, which obey the re-ordering of the ground truths. In the ideal case, orange points should overlap exactly with the blue ones.

4.4. Comparison between Baseline and Proposed

One intuitive train-test configuration is to train on scenario #1 and test on scenario #5. In order to better observe the performance of baseline method, we accumulatively enrich training scenarios and test on #1 to #5 separately. Some qualitative results are shown in Figure 4.

One can observe for baseline that the lower triangular zone of the train-test cases has nice constrained error distributions, which is natural since the test sequences were all "seen" during training in these configurations. However, the model demonstrates catastrophic behaviors when asked to extrapolate to "unseen" cases, while structurally, the "unseen" cases are only longer unit repetitions of trained cases.

In the following part of this section, we will focus on the configuration of training on #1-2 to more precisely evaluate our proposed method, since it is the minimal training scenario which covers all 3 possible positions of variable pressure in serial systems: before a pump, after a pump, and between 2 pumps.

Table 2. Mean and std of normalized absolute errors of three types of randomly masked variables. Comparison is listed between baseline and proposed method (Arc-encoding + SPA). The models are trained on scenario #1-2. The 5 rows indicate 5 test scenarios.

Test		**Baseline (%)**		**SPA+ARC (%)**	
		mean	std	mean	std
#1	P	0.07	0.12	**0.03**	0.12
	mfl	0.04	0.04	**0.03**	0.07
	PpSpd	0.12	0.25	**0.05**	0.07
#2	P	0.25	0.29	**0.02**	0.06
	mfl	0.08	0.07	**0.02**	0.06
	PpSpd	0.29	0.41	**0.03**	0.05
#3	P	8.30	9.30	**0.14**	0.58
	mfl	0.51	0.46	**0.07**	0.19
	PpSpd	4.22	3.13	**0.06**	0.11
#4	P	9.26	7.73	**0.13**	0.41
	mfl	0.68	0.67	**0.11**	0.29
	PpSpd	7.04	6.23	**0.09**	0.16
#5	P	13.95	10.82	**0.24**	0.80
	mfl	0.57	0.59	**0.15**	0.35
	PpSpd	10.88	9.23	**0.13**	0.24

In Table 2, we quantitatively list the comparison between baseline and proposed method, while training on scenario #1-2 and testing on #1 - #5 separately. The five test datasets all respectively contain 20K test data. For each data in tests, one of the three variable types will be randomly chosen to be masked and retrieved by the model. The positions of the dropped values are randomly chosen. For instance, in some cases, all the PpSpeeds in the sequence will be dropped simultaneously, since the truncated minimal systems still present unique solutions in this case.

One can observe that the error of baseline increments drastically when the test scenario contains more unit concatenations than in training data. On the other hand, the prediction error of proposed method only augmented slightly. The #3, #4 and #5 test of proposed method share similar mean and std range with the baseline #2 test. This suggests that the proposed model is able to infer in a modular manner, under which learning on minimal scenario permits accurate inference on longer unit concatenations.

4.5. Arc-Encoding vs without

In this subsection, we compare the test prediction accuracy between natural float encoding and Arc-encoding. As for the value masking for float encoding, we use an "impossible" value: -1. We retain the train-test configurations in Section 4.4. In Table 4, one can observe that Arc-encoding has some slight advantages in terms of bias in this use case.

Table 3. Normalized mean and 90 percentile of absolute error of three types of variables in % with different SPA max-lengths for training and tests. The error is overall minimal if we use same SPA max-length for training and test. There is a bigger risk in training with large max-length and testing with small ones. We specify that the test sequence of scenario #5 is of length 29 and the train sequences are of length 9 and 14 for scenario #1 and #2.

Train		Test - P (%)				Test - mflow (%)				Test - PpSpeed (%)			
SPA max-length		30	50	80	120	30	50	80	120	30	50	80	120
mean	30	0.03	0.07	0.29	0.62	0.21	0.19	0.22	0.23	0.41	0.36	0.35	0.37
	50	0.12	0.06	0.09	0.16	0.17	0.03	0.05	0.28	1.46	0.10	1.54	3.12
	80	0.27	0.14	0.03	0.11	1.05	0.20	0.08	0.21	0.82	0.21	0.08	0.18
	120	0.46	0.29	0.15	0.03	0.54	0.37	0.19	0.08	6.03	4.49	2.31	0.09
90 percentile	30	0.03	0.02	0.02	0.03	0.61	0.44	0.80	0.97	0.69	0.51	0.58	0.82
	50	0.13	0.04	0.02	0.03	0.58	0.11	0.17	1.02	6.43	0.34	6.63	13.65
	80	0.22	0.05	0.02	0.03	3.46	0.71	0.25	0.73	2.26	0.68	0.23	0.66
	120	0.50	0.22	0.10	0.03	2.12	1.58	0.87	0.36	30.36	22.11	10.98	0.33

Table 4. Mean and std of normalized absolute errors of three types of randomly masked variables. Comparison is listed between 2 input encoding methods: without and with Arc-encoding. The models are trained on scenario #1-2. The 5 rows indicate 5 test scenarios.

Test		**SPA (%)**		**ARC+SPA (%)**	
		mean	std	mean	std
#1	P	0.23	0.39	**0.03**	0.12
	mfl	0.11	0.14	**0.03**	0.07
	PpSpd	0.15	0.20	**0.05**	0.07
#2	P	0.16	0.19	**0.02**	0.06
	mfl	0.11	0.12	**0.02**	0.06
	PpSpd	0.14	0.18	**0.03**	0.05
#3	P	0.34	0.30	**0.14**	0.58
	mfl	0.11	0.12	**0.07**	0.19
	PpSpd	0.15	0.20	**0.06**	0.11
#4	P	0.36	0.27	**0.13**	0.41
	mfl	0.12	0.14	**0.11**	0.29
	PpSpd	0.19	0.23	**0.09**	0.16
#5	P	0.57	0.43	**0.24**	0.80
	mfl	**0.13**	0.15	0.15	0.35
	PpSpd	0.24	0.27	**0.13**	0.24

4.6. Robustness across SPA max-lengths

Our SPA max-length is primarily fixed throughout training and tests. As explained in Section 3.3, utilizing different max-lengths might affect the balancing in the convex combination between V rows in Equation (2).

We investigate in Table 3 the risky cases where the training and test SPA max-lengths do not match. The 4 models were trained on scenarios #1-2 and tested on #5 under same configurations in Section 4.4. One can observe a solid promise of trustworthy performances while using invariant max-length. As for the cases where training and test max-lengths do not match, the model is still robust in terms of P and mflow predictions.

5. Conclusion

In this work, we initiated trials to utilize data-driven approach in the modeling of a master Digital Twin (DT) of large hydraulic networks. We introduced a tokenization-embedding pipeline to represent physical system statuses for BERT-like training. Starting from the intuition that Attention mechanism is a convex combination operation, we proposed Stochastic Padding Augmentation (SPA) to enable modular inference. We demonstrated our proposed Arc-encoding and SPA for a model to be trained on limited unit concatenations of a pump, and tested on longer unseen concatenations. Our method shows promising performances in the context of short-range prediction.

This work paves the way for a real Transformer-based master Digital Twin for large hydraulic networks which is capable of treating parallel unit concatenations and long-range prediction.

Impact Statement

This paper presents work whose goal is to advance the application of data-driven approach in the field of Digital Twin. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Do Amaral, J., Dos Santos, C., Montevechi, J., and De Queiroz, A. Energy digital twin applications: a review. *Renewable and Sustainable Energy Reviews*, 188:113891, 2023.

Dufter, P., Schmitt, M., and Schütze, H. Position information in transformers: An overview. *Computational Linguistics*, 48(3):733–763, 2022.

Ghenai, C., Husein, L. A., Al Nahlawi, M., Hamid, A. K., and Bettayeb, M. Recent trends of digital twin technologies in the energy sector: A comprehensive review. *Sustainable Energy Technologies and Assessments*, 54: 102837, 2022.

Glaessgen, E. and Stargel, D. The digital twin paradigm for future nasa and us air force vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*, pp. 1818, 2012.

Grieves, M. *Virtually Intelligent Product Systems: Digital and Physical Twins*, pp. 175–200. 07 2019. ISBN 978-1624105647. doi: 10.2514/5.9781624105654.0175.0200.

Hou, G., Zhang, T., Guo, Z., Huang, T., and Li, Q. Accurate modeling of chp plant by digital twin and transformer neural network. In *2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pp. 1–4. IEEE, 2023.

Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. Characterising the digital twin: A systematic literature review. *CIRP journal of manufacturing science and technology*, 29:36–52, 2020.

Lin, Y.-Z., Shi, Q., Yang, Z., Latibari, B. S., Shao, S., Salehi, S., and Satam, P. Ddd-gendt: Dynamic data-driven generative digital twin framework. *arXiv preprint arXiv:2501.00051*, 2024.

Praharaj, L., Gupta, D., and Gupta, M. A lightweight edge-cnn-transformer model for detecting coordinated cyber and digital twin attacks in cooperative smart farming. *arXiv preprint arXiv:2411.14729*, 2024.

Rosyadi, I., Nazaruddin, Y. Y., and Siregar, P. I. Enhancing fault diagnosis accuracy in electric motors: A digital twin approach with transformer model. In *2024 14th Asian Control Conference (ASCC)*, pp. 1591–1596. IEEE, 2024.

Semeraro, C., Lezoche, M., Panetto, H., and Dassisti, M. Digital twin paradigm: A systematic literature review. *Computers in Industry*, 130:103469, 2021.

Sha, Z., Sun, C., Wei, S., Huo, R., and Ren, H. A transformer based classified traffic prediction scheme for energy digital twin network. In *2023 IEEE International Conference on Energy Internet (ICEI)*, pp. 218–223. IEEE, 2023.

Sun, Y., Shi, Y., Hu, Q., Xie, C., and Su, T. Dtformer: An efficient digital twin model for loss measurement in uhvdc transmission systems. *IEEE Transactions on Power Systems*, 39(2):3548–3559, 2023.

Tao, F., Zhang, H., Liu, A., and Nee, A. Y. Digital twin in industry: State-of-the-art. *IEEE Transactions on industrial informatics*, 15(4):2405–2415, 2018.

Tao, F., Xiao, B., Qi, Q., Cheng, J., and Ji, P. Digital twin modeling. *Journal of Manufacturing Systems*, 64:372–389, 2022.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Wang, K., Zhang, L., Cheng, H., Lu, H., and Chen, Z. A lifelong learning method based on event-triggered online frozen-ewc transformer encoder for equipment digital twin dynamic evolution. *IEEE Internet of Things Journal*, 2023.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Yu, W., Patros, P., Young, B., Klinac, E., and Walmsley, T. G. Energy digital twin technology for industrial energy management: Classification, challenges and future. *Renewable and Sustainable Energy Reviews*, 161:112407, 2022.

Zhang, L., Zhou, L., and Horn, B. K. Building a right digital twin with model engineering. *Journal of Manufacturing Systems*, 59:151–164, 2021.

Zhao, X. A novel digital-twin approach based on transformer for photovoltaic power prediction. *Scientific Reports*, 14(1):26661, 2024.