



MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG



COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Transformer Networks Accurately Predict Outputs of Parametric Dynamical Systems with Time-Varying External Inputs

Shuwen Sun, Lihong Feng, Peter Benner

Model Reduction and Surrogate Modeling — MORE 2024
La Jolla, California, USA
09–13 September 2024

Supported by:

IMPRS
ProEng
Magdeburg





Motivation

Natural Language Processing and Large Language Models

- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**



Motivation

Natural Language Processing and Large Language Models

- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**

- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).



Motivation

Natural Language Processing and Large Language Models

- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**

- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).
- Needs knowledge not only of the last word, but the whole sentence (memory).

Even more true in German than English. . .



- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**

- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).
- Needs knowledge not only of the last word, but the whole sentence (memory).
Even more true in German than English. . .
- In MLP/FFNN, vanishing gradient problem leads to loss of memory — no **attention** is paid to the past or context!



Motivation

Natural Language Processing and Large Language Models

- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**

- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).
- Needs knowledge not only of the last word, but the whole sentence (memory).
Even more true in German than English. . .
- In MLP/FFNN, vanishing gradient problem leads to loss of memory — no **attention** is paid to the past or context!
- Cure: **Recurrent Neural Networks (RNN)**, **Long Short-Term Memory (LSTM)**.



Motivation

Natural Language Processing and Large Language Models

- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**

- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).
- Needs knowledge not only of the last word, but the whole sentence (memory).
Even more true in German than English. . .
- In MLP/FFNN, vanishing gradient problem leads to loss of memory — no **attention** is paid to the past or context!
- Cure: **Recurrent Neural Networks (RNN)**, **Long Short-Term Memory (LSTM)**.
- These are not new. Why the revolution during the last four years?



Motivation

Natural Language Processing and Large Language Models

- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



open**GPT-X**

- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).
- Needs knowledge not only of the last word, but the whole sentence (memory).
Even more true in German than English. . .
- In MLP/FFNN, vanishing gradient problem leads to loss of memory — no **attention** is paid to the past or context!
- Cure: **Recurrent Neural Networks (RNN)**, **Long Short-Term Memory (LSTM)**.
- These are not new. Why the revolution during the last four years?
↪ GPT = Generative Pretrained **Transformer**



- Undoubtedly, large-language models have changed our life in many areas, including academia and science.



- In one task of **Natural Language Processing (NLP)**, words from a dictionary are ordered by their probability to be next in a sentence (query).
- Needs knowledge not only of the last word, but the whole sentence (memory).
Even more true in German than English. . .
- In MLP/FFNN, vanishing gradient problem leads to loss of memory — no **attention** is paid to the past or context!
- Cure: **Recurrent Neural Networks (RNN)**, **Long Short-Term Memory (LSTM)**.
- These are not new. Why the revolution during the last four years?
 - ↪ GPT = Generative Pretrained **Transformer**
 - ↪ introduction of the transformer architecture in Deep Learning was the breakthrough!



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* †
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a **new simple network architecture, the Transformer**, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to

Presented at 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA.

Cited 132,331 times so far (September 6, 2024, 12:11pm).



Motivation

Attention and Surrogate Modeling

- **Our question:** is attention/are transformer networks any good for surrogate modeling?



Motivation

Attention and Surrogate Modeling

- **Our question:** is attention/are transformer networks any good for surrogate modeling?
- A sentence is a *sequence* of words, pretrained transformer network suggests the next word.



Motivation

Attention and Surrogate Modeling

- **Our question:** is attention/are transformer networks any good for surrogate modeling?
- A sentence is a *sequence* of words, pretrained transformer network suggests the next word.
- Words are encoded as vectors for input into the transformer.



- **Our question:** is attention/are transformer networks any good for surrogate modeling?
- A sentence is a *sequence* of words, pretrained transformer network suggests the next word.
- Words are encoded as vectors for input into the transformer.
- **Analogy:** a trajectory of a time-dependent problem is a *sequence* of vectors $x(t_0), \dots, x(t_k)$. Can a pretrained transformer network suggest $x(t_{k+1})$?



- **Our question:** is attention/are transformer networks any good for surrogate modeling?
- A sentence is a *sequence* of words, pretrained transformer network suggests the next word.
- Words are encoded as vectors for input into the transformer.
- **Analogy:** a trajectory of a time-dependent problem is a *sequence* of vectors $x(t_0), \dots, x(t_k)$. Can a pretrained transformer network suggest $x(t_{k+1})$?
- Such a pretrained transformer network could serve as a surrogate of a dynamical system for forecasting/extrapolation.



- **Our question:** is attention/are transformer networks any good for surrogate modeling?
- A sentence is a *sequence* of words, pretrained transformer network suggests the next word.
- Words are encoded as vectors for input into the transformer.
- **Analogy:** a trajectory of a time-dependent problem is a *sequence* of vectors $x(t_0), \dots, x(t_k)$. Can a pretrained transformer network suggest $x(t_{k+1})$?
- Such a pretrained transformer network could serve as a surrogate of a dynamical system for forecasting/extrapolation.
- Attempts to use RNNs or LSTMs in surrogate models have been attempted with certain success, e.g., [OTTO/ROWLEY 2017, FRESKA/MANZONI/DEDÉ 2021, FENG 2023, ...].



- **Our question:** is attention/are transformer networks any good for surrogate modeling?
- A sentence is a *sequence* of words, pretrained transformer network suggests the next word.
- Words are encoded as vectors for input into the transformer.
- **Analogy:** a trajectory of a time-dependent problem is a *sequence* of vectors $x(t_0), \dots, x(t_k)$. Can a pretrained transformer network suggest $x(t_{k+1})$?
- Such a pretrained transformer network could serve as a surrogate of a dynamical system for forecasting/extrapolation.
- Attempts to use RNNs or LSTMs in surrogate models have been attempted with certain success, e.g., [OTTO/ROWLEY 2017, FRESKA/MANZONI/DEDÉ 2021, FENG 2023, ...].
- **Remark:** the attention mechanism should in particular be useful for non-Markovian time series data.



Problem setting

$$\begin{aligned}\frac{dx(\boldsymbol{\mu}, t)}{dt} &= \mathbf{f}(x(\boldsymbol{\mu}, t), \boldsymbol{\mu}) + \mathbf{B}u(\boldsymbol{\mu}, t), \\ \mathbf{y}(\boldsymbol{\mu}, t) &= \mathbf{g}(x(\boldsymbol{\mu}, t)).\end{aligned}$$

where

- $\mathbf{x}(\boldsymbol{\mu}, t) \in \mathbb{R}^n$ is the **state-space variable**,
- $\boldsymbol{\mu} = (\mu_1, \dots, \mu_m) \in \mathbb{R}^m$ is the vector of **parameters**,
- $\mathbf{u}(\boldsymbol{\mu}, t)$ is an external, potentially parameter-dependent, **input signal**, and
- $\mathbf{y}(\boldsymbol{\mu}, t) = (y_1(\boldsymbol{\mu}, t), \dots, y_q(\boldsymbol{\mu}, t))^T \in \mathbb{R}^q$ denotes the **quantities-of-interest (Qols)** or **system output**.




Problem setting

$$\begin{aligned}\frac{dx(\boldsymbol{\mu}, t)}{dt} &= \mathbf{f}(x(\boldsymbol{\mu}, t), \boldsymbol{\mu}) + \mathbf{B}u(\boldsymbol{\mu}, t), \\ \mathbf{y}(\boldsymbol{\mu}, t) &= \mathbf{g}(x(\boldsymbol{\mu}, t)).\end{aligned}$$

where

- $\mathbf{x}(\boldsymbol{\mu}, t) \in \mathbb{R}^n$ is the **state-space variable**,
- $\boldsymbol{\mu} = (\mu_1, \dots, \mu_m) \in \mathbb{R}^m$ is the vector of **parameters**,
- $\mathbf{u}(\boldsymbol{\mu}, t)$ is an external, potentially parameter-dependent, **input signal**, and
- $\mathbf{y}(\boldsymbol{\mu}, t) = (y_1(\boldsymbol{\mu}, t), \dots, y_q(\boldsymbol{\mu}, t))^T \in \mathbb{R}^q$ denotes the **quantities-of-interest (QoIs)** or **system output**.

- Aim to predict long-term evolution of $\mathbf{y}(\boldsymbol{\mu}, t)$ under variation of both, $\boldsymbol{\mu}$ and $\mathbf{u}(\boldsymbol{\mu}, t)$.
- We try and adapt the **Temporal Fusion Transformer (TFT)** model suggested for forecasting in

 B. Lim, S. Ö. Arik, N. Loeff, T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting* 37:1748–1764, 2021.



Spatial-temporal $x(\mu, t)$ prediction using DNNs

- Intrusive MOR [J. BARNETT ET AL., 2023], [Y. KIM ET AL., 2022], [K. LEE & K. CARLBERG, 2020].
- Combine data compression (POD, autoencoder) with latent space dynamics identification via e.g., feedforward NN, LSTM, CNN, RBF interpolation, DMD, SINDy, neural operator, etc.
[J. DUAN & J. S. HESTHAVEN, 2024], [S. FRESCA ET AL., 2021], [C. BONNEVILLE ET AL., 2024],
[K. KONTOLATI ET AL., 2024], [P. GOYAL & P. B., 2021/24],
- Predicting the dynamics via neural operator learning [Z. LI ET AL., 2021], [L. LU ET AL., 2021],

Spatial-temporal $x(\mu, t)$ prediction using transformers

- Combine autoencoder with latent space dynamics identification via transformers
[N. GENEVA & N. ZABARAS, 2022], [A. SOLERA-RICO, ET AL., 2024].
- Neural operator learning using transformers
[Z. HAO ET AL., 2023], [E. CALVELLO ET AL., 2024], [O. OVADIA ET AL., 2024],



Interpretability

- μ and $u(\mu, t)$ are separately considered as two different classes of inputs to the transformer. In other existing transformer related works, they are blended.
- An interpretable multi-head attention is proposed and used in TFT; vanilla attention is used in other existing works.

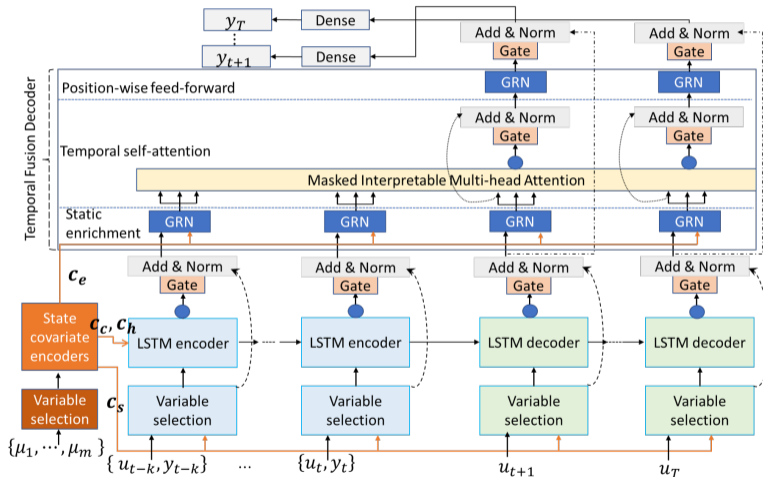
Automatic hyperparameter determination

The TFT hyperparameters, such as learning rates, mini-batch size, dropout rate, number of heads, and so on, can be automatically tuned during the optimization process. There is no need for manual hyperparameter fine-tuning.



Temporal $y(\mu, t)$ prediction using transformer

- No need of data compression.
- We apply TFT [B. LIM ET. AL., 2021], a transformer model that gives *improved interpretability* of the learning process and the attention mechanism.
- TFT was originally used for predicting the *quantiles* of a *scalar-valued* output.
- Main modifications of the TFT structure are: modified loss function and modified data format.
- We extended TFT to multiple-output TFT for system dynamics prediction. The multiple-output TFT is able to predict the *actual* values of *vector-valued* system outputs.
- Multiple-output TFT adds new dimension to its interpretability.



Building Blocks

- **Gating:** Gated Residual Network (GRN).
- **Variable selection:** Selecting the most relevant input variables.
- **Static covariate encoders:** Integrating parameters μ_1, \dots, μ_m into the network.
- **Temporal processing:** LSTM encoder, decoder & interpretable multi-head attention.

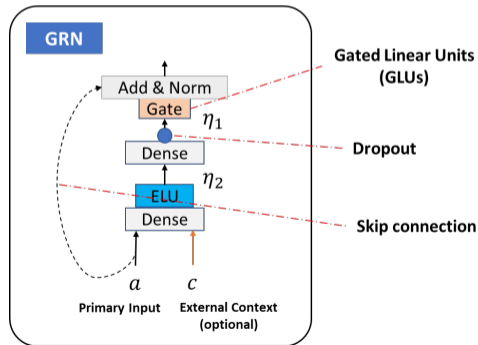


Gated Residual Network (GRN)

- GRN is an important building block in different parts of the transformer, which provides a non-linear process step: GLU, only where needed.

$$\text{GRN}(a, c) = \text{LayerNorm}(a + \text{GLU}(\eta_1)).$$

- Gated Linear Unit (GLU):** in GRN, GLU is skipped when its output is close to 0.





Gated Residual Network (GRN)

- GRN is an important building block in different parts of the transformer, which provides a non-linear process step: GLU, only where needed.

Variable Selection Network (VSN)

- VSN, selecting relevant input variables, greatly improves the performance and the interpretability of the TFT model via putting larger weights on the most salient ones.

Static Covariate Encoder

- The static covariate encoder integrates parameters μ_1, \dots, μ_m into the network.



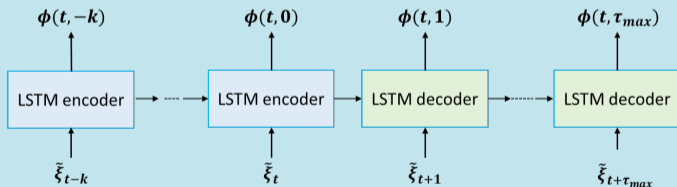
Temporal processing

- LSTM encoder-decoder is responsible for local temporal processing. Multi-headed attention is responsible for global temporal processing.

Temporal processing

- LSTM encoder-decoder is responsible for local temporal processing. Multi-headed attention is responsible for global temporal processing.

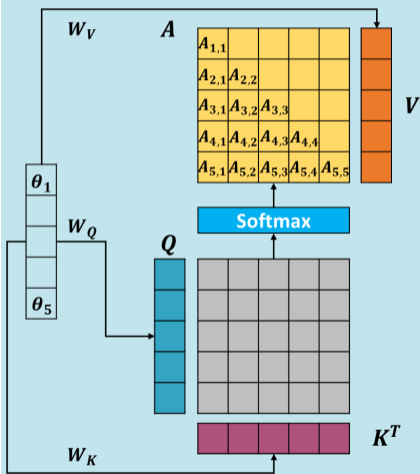
LSTM encoder-decoder



- $n = -k, \dots, 0, 1, \dots, \tau_{\max}$ are the position indices.
- The inputs $\tilde{\xi}_{t+n}, n = -k, \dots, 0$ include the information of both $u(t - t_n)$ and $y(t - t_n)$. $\tilde{\xi}_{t+n}, n = 1, \dots, \tau_{\max}$ include the information of $u(t + t_n)$ only.
- The outputs of the LSTM encoder-decoder, $\phi(t, n)$, are fed into the multi-head attention.



Self-attention

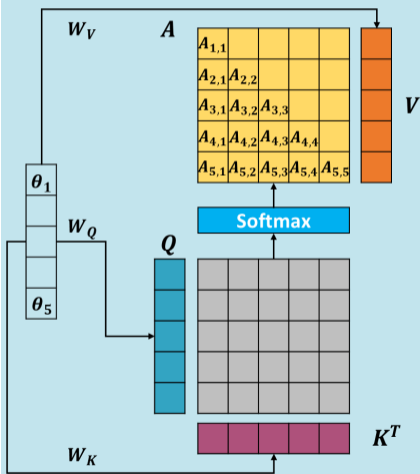


- In TFT, inputs $\theta = [\theta_1, \dots, \theta_{N_t}] \in \mathbb{R}^{N_t \times d}$ to the self-attention integrates μ , $u(t, \mu)$ and $y(t, \mu)$. N_t is the number of time steps.
- θ is linearly transformed into **Queries** $Q = \theta W_Q$, **Keys** $K = \theta W_K$ and **Values** $V = \theta W_V$.
- The attention weight matrix A is derived from the scaled outer product QK^T followed by a softmax function.

$$\underbrace{\text{softmax} \left(\frac{Q \times K^T}{\sqrt{d_k}} \right)}_{\text{Attention weight matrix } A} \times V$$



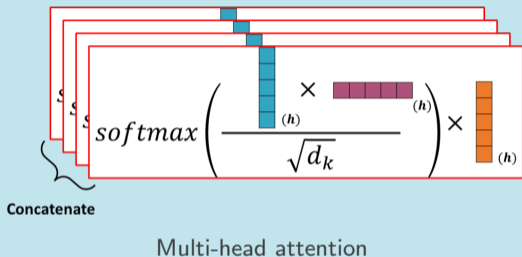
Self-attention



- In TFT, inputs $\theta = [\theta_1, \dots, \theta_{N_t}] \in \mathbb{R}^{N_t \times d}$ to the self-attention integrates μ , $u(t, \mu)$ and $y(t, \mu)$. N_t is the number of the time steps.
- θ is linearly transformed into Queries $Q = \theta W_Q$, Keys $K = \theta W_K$ and Values $V = \theta W_V$.
- The attention weight matrix A is derived from the scaled outer product QK^T followed by a softmax function.
- The magnitude of the entry $A_{i,j}$ in the attention weight matrix A interprets the correlation between the feature at t_i and the feature at t_j in the time series.



Standard multi-head attention



- Attention is employed n_h times in parallel resulting in n_h heads with n_h attention weight matrices $\mathbf{A}_h, h = 1, \dots, n_h$.
- However, various \mathbf{A}_h might not be sufficiently informative to describe the correlation between the features.



Interpretable multi-head attention

Averaged attention weight matrix $\bar{\mathbf{A}}$

$$\frac{1}{n_h} \sum_h \underbrace{\text{softmax} \left(\frac{\begin{matrix} \text{blue vector} \\ (h) \end{matrix} \times \begin{matrix} \text{pink vector} \\ (h) \end{matrix}}{\sqrt{d_h}} \right)}_{\text{Attention weight matrix } \mathbf{A}_h} \times \begin{matrix} \text{orange vector} \end{matrix}$$

Attention weight matrix \mathbf{A}_h

Interpretable multi-head attention

- *Interpretable* multi-head attention averages the attention weight matrices $\mathbf{A}_h, h = 1, \dots, n_h$, leading to a single attention weight matrix $\bar{\mathbf{A}}$.
- *Interpretable* multi-head attention resembles the formulation of self-attention, allowing simple interpretability studies by analyzing a single attention weight matrix, like in self-attention.



Actual-valued output prediction

Original TFT utilizes quantile loss to predict the quantile values of traffic volume, electricity consumption, etc. We modified the loss function to MSE to enable TFT to predict the actual values of the system outputs.

Multiple-output prediction

Weight matrix \bar{A} in original TFT Weight matrix \tilde{A} in multiple-outputs TFT

$\bar{A}_{1,1}$				
$\bar{A}_{2,1}$	$\bar{A}_{2,2}$			
$\bar{A}_{3,1}$	$\bar{A}_{3,2}$	$\bar{A}_{3,3}$		
$\bar{A}_{4,1}$	$\bar{A}_{4,2}$	$\bar{A}_{4,3}$	$\bar{A}_{4,4}$	
$\bar{A}_{5,1}$	$\bar{A}_{5,2}$	$\bar{A}_{5,3}$	$\bar{A}_{5,4}$	$\bar{A}_{5,5}$

$a_{1,1}$	$a_{1,2}$	$a_{1,3}$			
$a_{2,1}$	$a_{2,2}$	$a_{3,2}$			
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$			
	$\tilde{A}_{2,1}$			$\tilde{A}_{2,2}$	

- Original TFT: scalar-valued output prediction.
- The proposed *multiple-output TFT* framework: multiple outputs with extended interpretability.
- In the weight matrix \bar{A} of the original TFT, each scalar element $\bar{A}_{i,j}$ describes the correlation between the single output at t_i and itself at t_j .
- In *multiple-output TFT*, the element $a_{k,\ell}$ in the i, j -th block $\tilde{A}_{i,j}$ provides the correlation between the k -th output at t_i and the ℓ -th output at t_j .



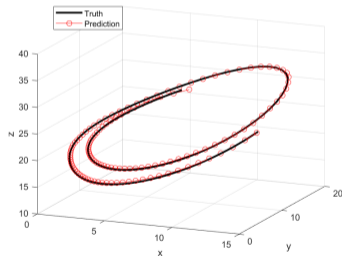
Lorenz-63 model

Governing equations of chaotic dynamics of the Lorenz-63 model:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z.$$

General setting:

- $\rho = 28$, $\sigma = 10$ and $\beta = 8/3$.
- **Parameters:** random initial states $x_0 \sim \mathcal{U}(-20, 20)$, $y_0 \sim \mathcal{U}(-20, 20)$ and $z_0 \sim \mathcal{U}(10, 40)$.
- **Training data:** time series with 2048 groups of random initial states, training time: 2.7 hours.
- **Testing:** time series with 256 groups of random initial states. Given any x_0, y_0, z_0 , multiple-output TFT is used to predict $x(t), y(t), z(t)$ at the subsequent 127 time steps. Prediction time for each testing case is 0.4ms on average.



One of the four testing cases.



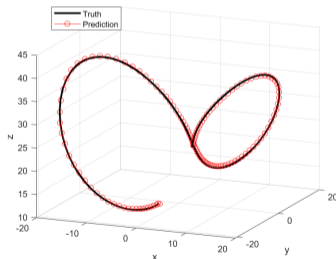
Lorenz-63 model

Governing equations of chaotic dynamics of the Lorenz-63 model:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z.$$

General setting:

- $\rho = 28$, $\sigma = 10$ and $\beta = 8/3$.
- **Parameters:** random initial states $x_0 \sim \mathcal{U}(-20, 20)$, $y_0 \sim \mathcal{U}(-20, 20)$ and $z_0 \sim \mathcal{U}(10, 40)$.
- **Training data:** time series with 2048 groups of random initial states, training time: 2.7 hours.
- **Testing:** time series with 256 groups of random initial states. Given any x_0, y_0, z_0 , multiple-output TFT is used to predict $x(t), y(t), z(t)$ at the subsequent 127 time steps. Prediction time for each testing case is 0.4ms on average.



One of the four testing cases.



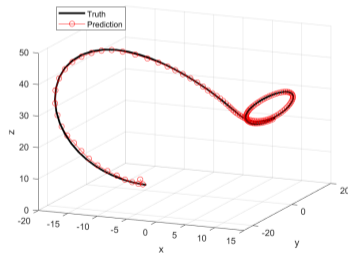
Lorenz-63 model

Governing equations of chaotic dynamics of the Lorenz-63 model:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z.$$

General setting:

- $\rho = 28$, $\sigma = 10$ and $\beta = 8/3$.
- **Parameters:** random initial states $x_0 \sim \mathcal{U}(-20, 20)$, $y_0 \sim \mathcal{U}(-20, 20)$ and $z_0 \sim \mathcal{U}(10, 40)$.
- **Training data:** time series with 2048 groups of random initial states, training time: 2.7 hours.
- **Testing:** time series with 256 groups of random initial states. Given any x_0, y_0, z_0 , multiple-output TFT is used to predict $x(t), y(t), z(t)$ at the subsequent 127 time steps. Prediction time for each testing case is 0.4ms on average.



One of the four testing cases.



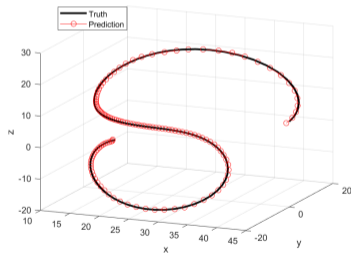
Lorenz-63 model

Governing equations of chaotic dynamics of the Lorenz-63 model:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z.$$

General setting:

- $\rho = 28$, $\sigma = 10$ and $\beta = 8/3$.
- **Parameters:** random initial states $x_0 \sim \mathcal{U}(-20, 20)$, $y_0 \sim \mathcal{U}(-20, 20)$ and $z_0 \sim \mathcal{U}(10, 40)$.
- **Training data:** time series with 2048 groups of random initial states, training time: 2.7 hours.
- **Testing:** time series with 256 groups of random initial states. Given any x_0, y_0, z_0 , multiple-output TFT is used to predict $x(t), y(t), z(t)$ at the subsequent 127 time steps. Prediction time for each testing case is 0.4ms on average.



One of the four testing cases.



FitzHugh-Nagumo model

Governing equations of the FitzHugh-Nagumo model:

$$\begin{aligned}\varepsilon \frac{\partial v(x,t,\varepsilon,c)}{\partial t} &= \varepsilon \frac{\partial^2 v(x,t,\varepsilon,c)}{\partial x^2} + f(v(x,t,\varepsilon,c)) - w(x,t,\varepsilon,c) + c, \\ \frac{\partial w(x,t,\varepsilon,c)}{\partial t} &= bv(x,t,\varepsilon,c) - \gamma w(x,t,\varepsilon,c) + c, \quad x \in [0, L], \quad t \in [0, 5]s, \\ f(v) &= v(v - 0.1)(1 - v), \quad L = 1, \quad b = 0.5, \quad \gamma = 2.\end{aligned}$$

Boundary conditions:

$$\begin{aligned}v(x, 0, \varepsilon, c) &= 0, \quad w(x, 0, \varepsilon, c) = 0, \quad x \in [0, L], \\ v_x(0, t, \varepsilon, c) &= -u(t), \quad v_x(L, t, \varepsilon, c) = 0, \quad t \in [0, 5]sec.\end{aligned}$$

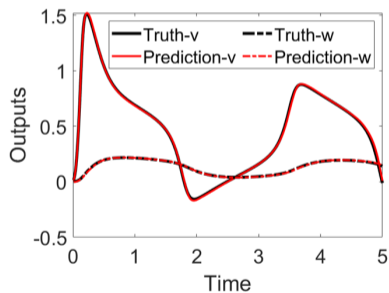
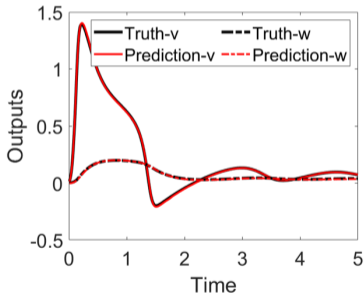
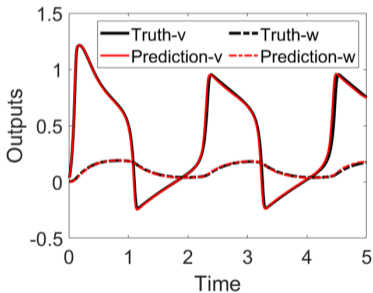
General setting:

- **Parameters:** $\varepsilon \in [0.01, 0.04]$ and $c \in [0.03, 0.07]$, so that $\boldsymbol{\mu} = (\varepsilon, c)^T$.
- **Time-dependent external input:** $u(t) = 5 \times 10^4 t^3 e^{-15t}$.
- Given initial value of outputs $\boldsymbol{y}(0, 0, \boldsymbol{\mu}) = (v(0, 0, \boldsymbol{\mu}), w(0, 0, \boldsymbol{\mu}))^T$ at any parameter value, multiple-output TFT is used to predict the subsequent 499 time steps.
- **Training data:** Time series corresponding to 120 training parameter sets, training time is 33min.



FitzHugh-Nagumo model

- **Testing:** Time series corresponding to 6 testing parameter sets $\{\mu_1^*, \dots, \mu_6^*\}$.
- **Prediction time** for all 6 testing cases: 0.5 seconds.

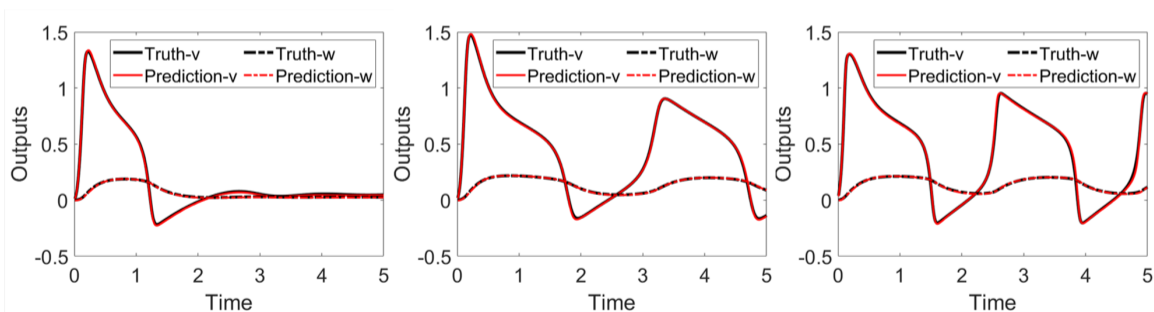


Six testing cases for 499 time steps prediction.



FitzHugh-Nagumo model

- **Testing:** Time series corresponding to 6 testing parameter sets $\{\mu_1^*, \dots, \mu_6^*\}$.
- **Prediction time** for all 6 testing cases: 0.5 seconds.

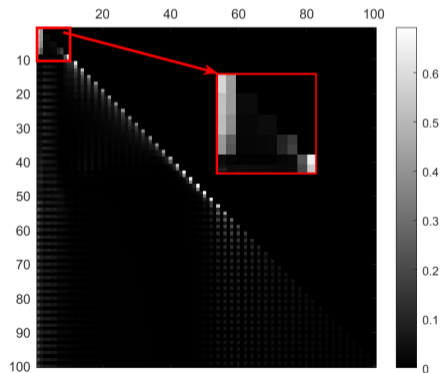


Six testing cases for 499 time steps prediction.



FitzHugh-Nagumo model: numerical illustration of the attention weight matrix

The attention weight matrix of the multiple-output TFT: in contrast to the diagonal (scalar) elements of the attention weight matrix of the original TFT, this matrix has 2×2 diagonal blocks, as this example has 2 outputs. This illustrates the correlations between the different outputs at individual time instants.





Conclusions and Outlook

- We have adapted the transformer model TFT in order to predict the outputs of parametric dynamical systems with time-varying external inputs.
- We extend the original framework of TFT to multiple-output TFT which is capable of predicting multiple outputs.
- Multiple-output TFT enriches the interpretable information of the original TFT by adding the correlation between different outputs within two individual time instants.



Conclusions and Outlook

- We have adapted the transformer model TFT in order to predict the outputs of parametric dynamical systems with time-varying external inputs.
- We extend the original framework of TFT to multiple-output TFT which is capable of predicting multiple outputs.
- Multiple-output TFT enriches the interpretable information of the original TFT by adding the correlation between different outputs within two individual time instants.
- **Future work** could be combining TFT with data compression to realize spatial-temporal prediction in the parameter domain.
- **Future work must** be the better understanding of TFT/transformer models and their mathematical analysis.