Multi-Scale Recurrent Transformer model for Predicting KSTAR PF Superconducting Coil Temperature.

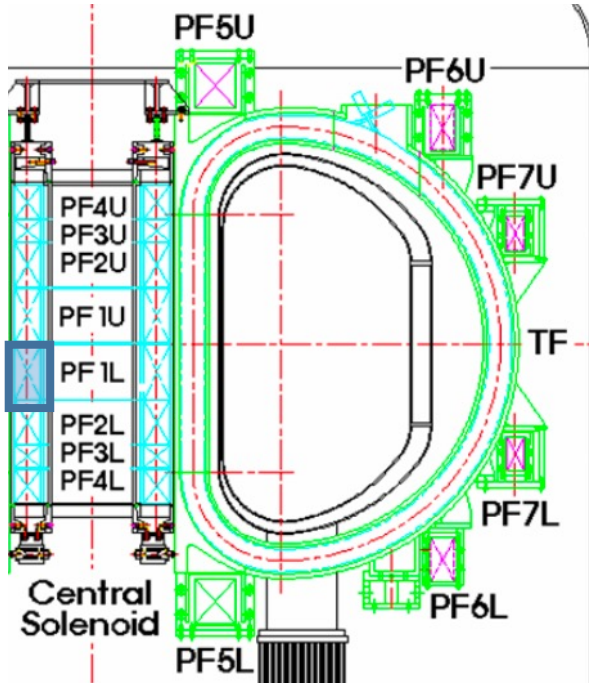
Giil Kwon, Hyunjung Lee

giilkwon@kfe.re.kr

Introduction

❖Korea Superconducting Tokamak Advanced Research (KSTAR) superconducting coil system

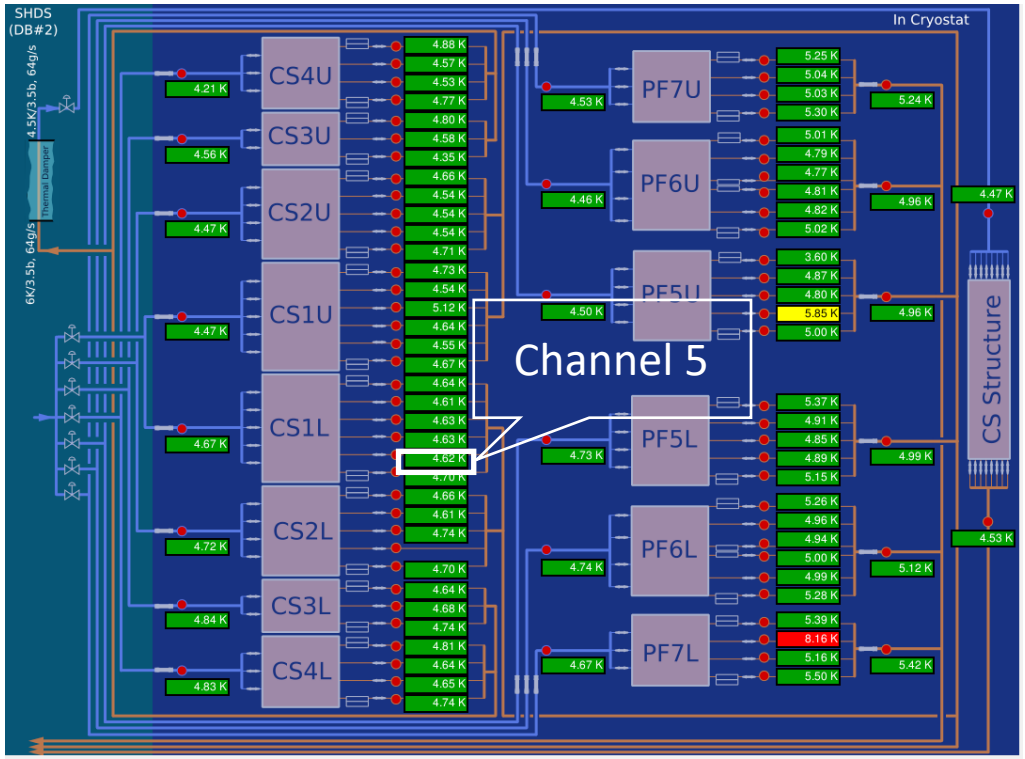
- KSTAR is the Superconducting Tokamak in south Korea.
- The superconducting coil system is one of the most important components in KSTAR.
- Superconducting coils are cooled by forced-flow supercritical helium about 4.5 K.
- To protect the superconducting coil system, we need to predict next superconducting coil temperature.
  - This is because excessive heat not only degrade the performance of the coil but also can break the superconducting coil system.
    - ✓ The rising of poloidal field (PF) coil's temperature is mostly by AC losses according to the variations of current and magnetic field



KSTAR Tokamak Superconducting coils

Introduction

- The measurement of the coil temperature
  - Coil's temperature are measured on Tokamak Monitoring System (TMS).
    - TMS measured the temperature data and publish EPICS PV every 1 second.
  - Thermometer measure the temperature of coolant(Helium).
    - Thermometers are located at inlet and outlet of each cooling channels.
    - In this work, we use data from outlet channel 5 thermometer of the PF1 coil.



Tokamak Monitoring System(TMS) GUI

Introduction

❖Problem formulation

- Suppose we have a collection of N univariate time series data, $\{X_{n,t_0:t_j}\}_{n=1}^{N}$, where $X_{n,t_0:t_j} = \left[X_{n,t_0}, X_{n,t_1}, \cdots, X_{n,t_j}\right], X_{n,t} \in \mathbb{R}$, denotes $n'th$ time series data value at time t. ${X^{\sigma}}_{n,t_0:t_j} = \left[{X^{\sigma}}_{n,t_0}, {X^{\sigma}}_{n,t_1}, \cdots, {X^{\sigma}}_{n,t_j}\right], {X^{\sigma}}_{n,t} \in \mathbb{R}$, denotes the subsampled time series data in σ-scale. we will predict the next τ step time series values, $\{X_{n,t_{j+1}:t_{j+\tau}}\}_{n=1}^{N}$.
  - In this work, N=1,λ=10, we will do ten step ahead forecast of univariate time series data.
  - We will forecast the temperature of PF1L(channel 5) data.
- We are going to formulate the multi step prediction into ten-step ahead prediction problem, where Φ is learnable parameter while training.
  - Δ indicates input data length. In this work, Δ= 10~100

$$P\left(X_{n,t_i+\Delta+1:t_i+\Delta+\tau+1} \middle| X_{n,t_i:t_i+\Delta}, {X^{\sigma}}_{n,t_i:t_i+\Delta}; \Phi\right)$$

$$= \prod_{t=t_i}^{t_i+\tau/\lambda} P(X_{n,t+\Delta+1:t+\Delta+\lambda+1} | X_{n,t:t+\Delta}, {X^{\sigma}}_{n,t:t+\Delta}; \Phi)$$

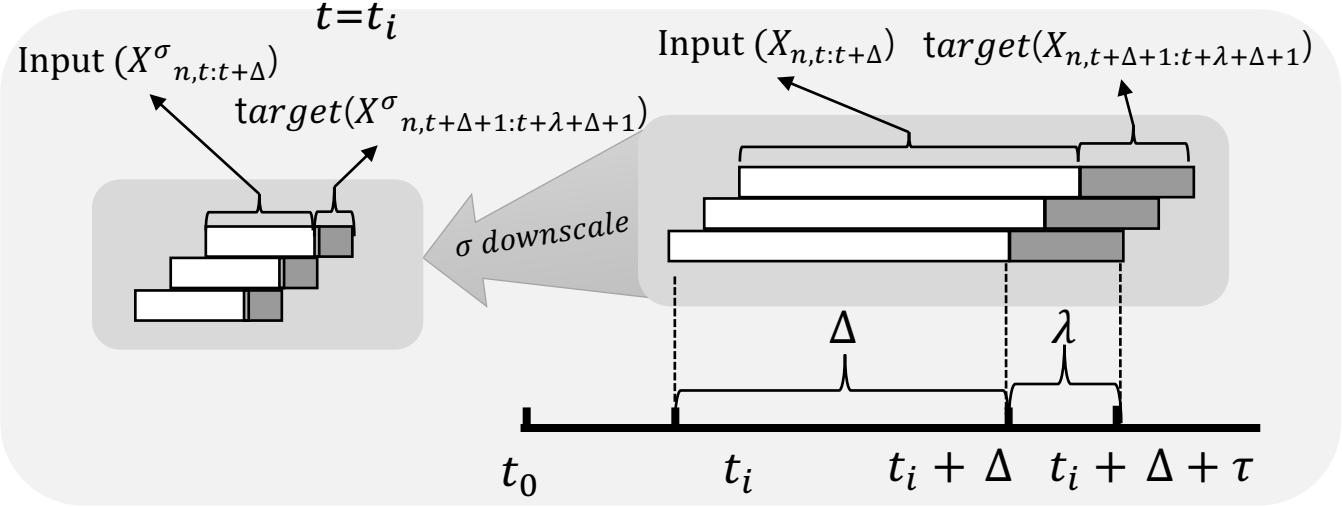
Introduction

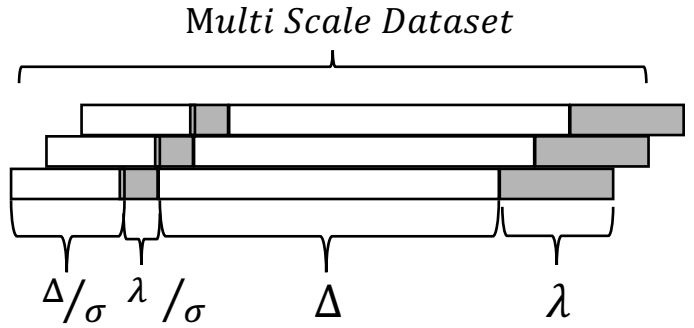
❖Problem formulation

- We change the model to recurrent model by adding latent variables Z_{n,t_i}, ${Z^{\sigma}}_{n,t_i}$.
  - Z_{n,t_i} denotes the latent variable of n'th time series data at time t_i.
  - ${Z^{\sigma}}_{n,t_i}$ denotes the latent variable of n'th time series data in σ-scale at time t_i.

$$P\left(X_{n,t_i+\Delta+1:t_i+\Delta+\tau+1} \middle| X_{n,t_i:t_i+\Delta}, {X^{\sigma}}_{n,t:t+\Delta}; \Phi'\right)$$

$$= \prod_{t=t_i}^{t_i+\tau/\lambda} P(X_{n,t+\Delta+1:t+\Delta+\lambda+1}, Z_{n,t_i+1} | X_{n,t:t+\Delta}, {X^{\sigma}}_{n,t:t+\Delta}, Z_{n,t_i}, {Z^{\sigma}}_{n,t_i}; \Phi')$$





Δ : input sequence length.
λ : output sequence length.
σ : sub sampling scale

Model Architecture

❖ Long-Short-Term Memory(LSTM)

- LSTM is one kind of Recurrent neural network.
- LSTM is well suited to predict, classify the time series data.

❖ Transformer

- Transformer is the self-attention based deep learning model.
- It mostly used to solve Natural Language Problem(NLP).(such as GPT-3, BERT).
- Nowadays, it also used to forecast time series data.(Informer, Perceiver, Reformer)

❖ TransformerXL

- TransformerXL model is used for modeling the language model.
- TransformerXL is the transformer based algorithms that have recurrence.
- TransformerXL concatenates the input from the previous layer at each layer and uses this concatenated data as the valu e and the key input for the attention module.
- Additionally, the current input is used as a query to the attention module, allowing it to attend to both the concatenated i nput data and current input data

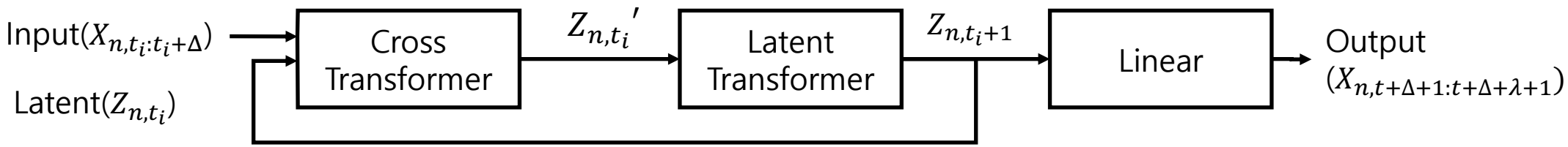
❖ Perceiver

- The model build upon the Transformer that can learn multimodal data.
- The model iteratively attend to the input data by alternating cross-attention and latent transformer.
- The model unrolled in depth to the same input rather then in time to different inputs

Model Architecture

❖Recurrent Transformer(RT)

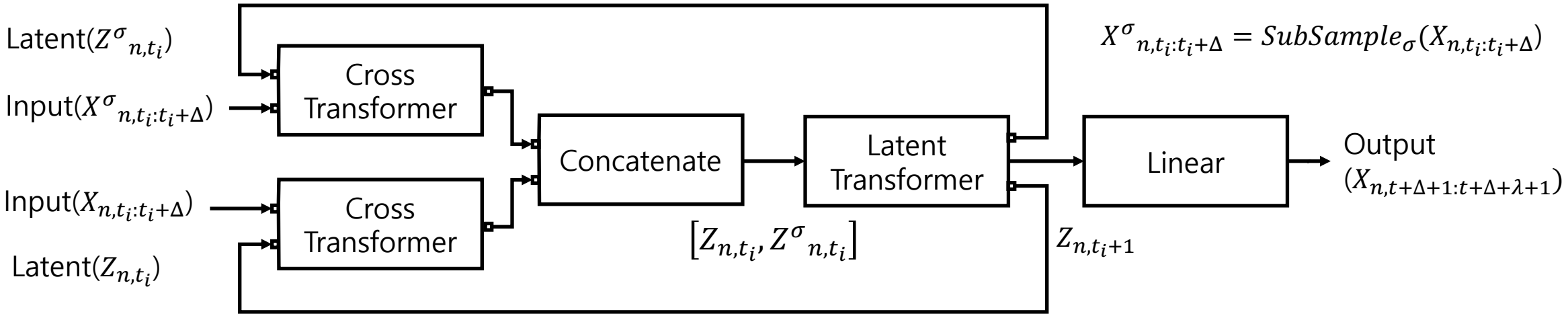
- We build recurrent transformer upon the perceiver by iteratively giving time series input and attend input data to latent variable by using cross-transformer module.
- As the perceiver did, our model also alternating cross-attend module and latent transformer.
- As RNN did, we give latent variable as input to the model.
- Unlike Perceiver, our model unrolled in time to different inputs.



Model Architecture

❖Multi-Scale Recurrent Transformer(MSRT)

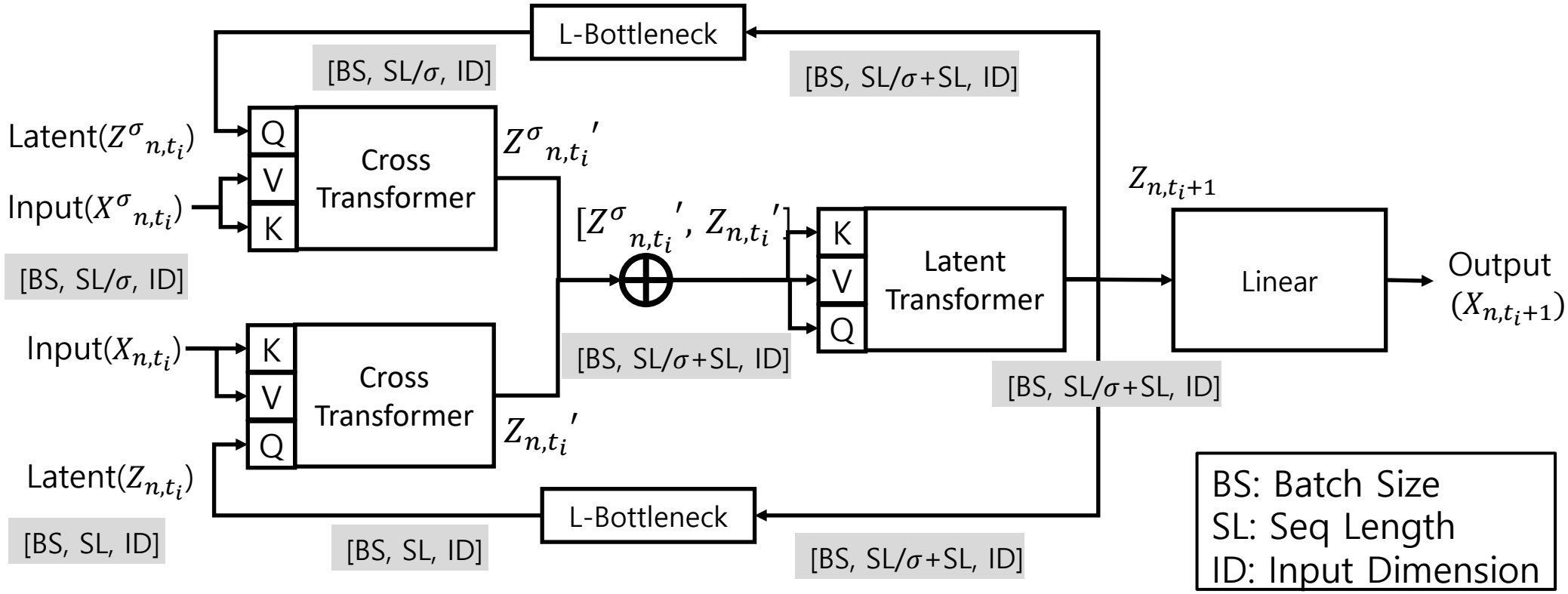
- We build multi-scale recurrent transformer upon the recurrent transformer by giving subsampled time series input($X^{\sigma}{}_{n,t_i:t_i+\Delta}$) and input ($X_{n,t_i:t_i+\Delta}$) and attend to latent variables($Z^{\sigma}{}_{n,t_i}$ or Z_{n,t_i}) by using cross-transformer modules.
- Downsampled input can be used to calculate low frequency features. And input can be used to get high frequency features from input.



Model Architecture

❖Multi-Scale Recurrent Transformer(MSRT)

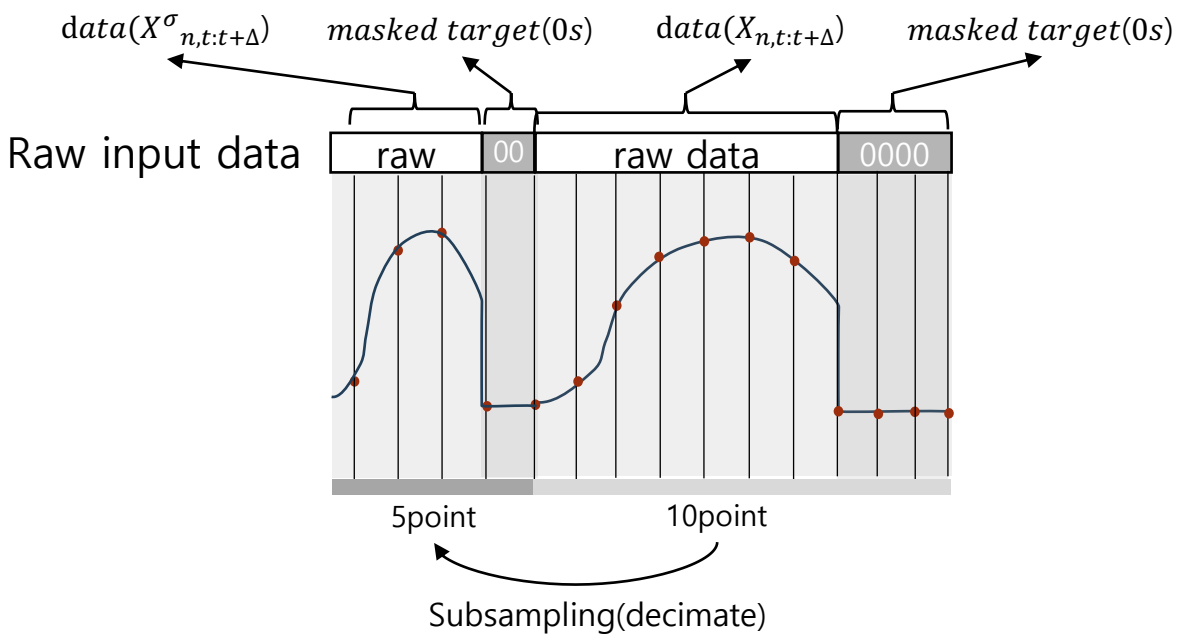
- L-Bottleneck module computes features of shorter length than the original input that are useful for learning the data.(SL/σ+SL => SL/σ or SL)

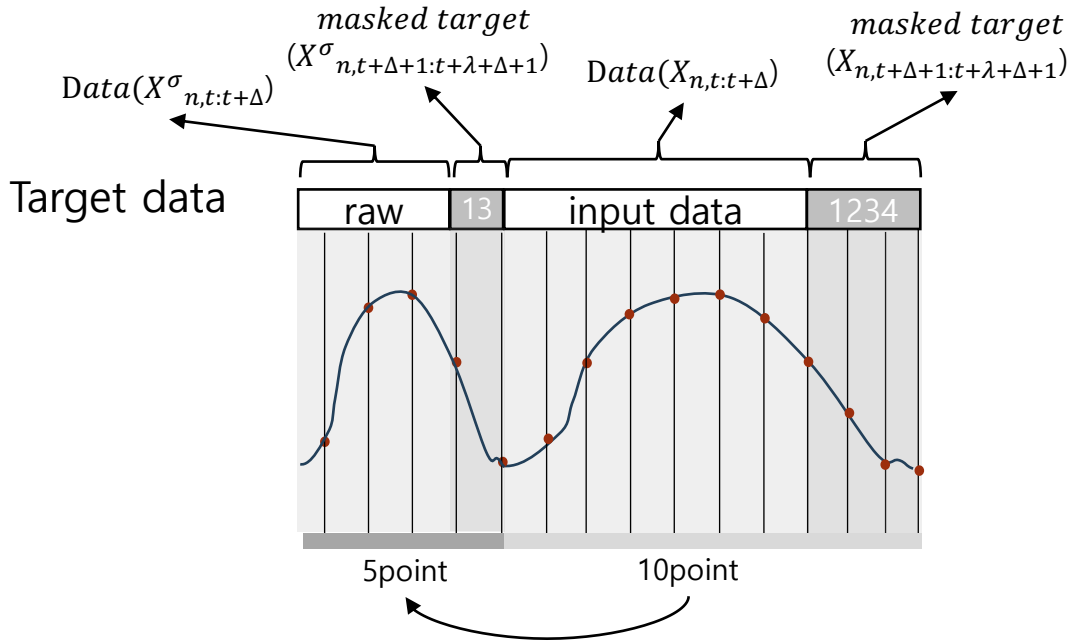


Model Architecture

❖Multi-Scale Dataset

- We use input train data that masked out target data part in this case λ length prediction target.
- We appended down sampled data to the original data, so that when we used the data, we didn't have to down sample it.
  - Ex) In σ =2 case, Raw data(10pt) subsampled to have half sampling rate(5pt)
  - From Higher sampling rate signal we can calculate high frequency features and lower sampling rate signal we can calculate low frequency features.

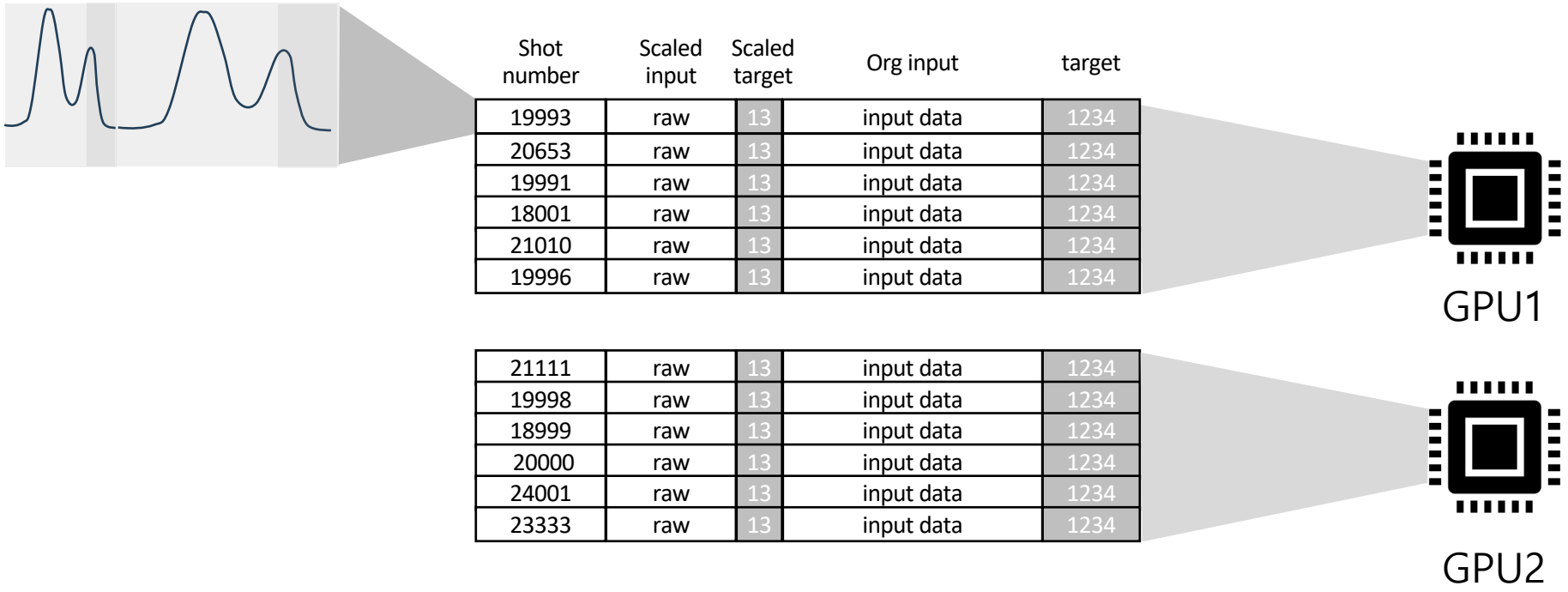




Model Architecture

❖Multi-Scale Dataset

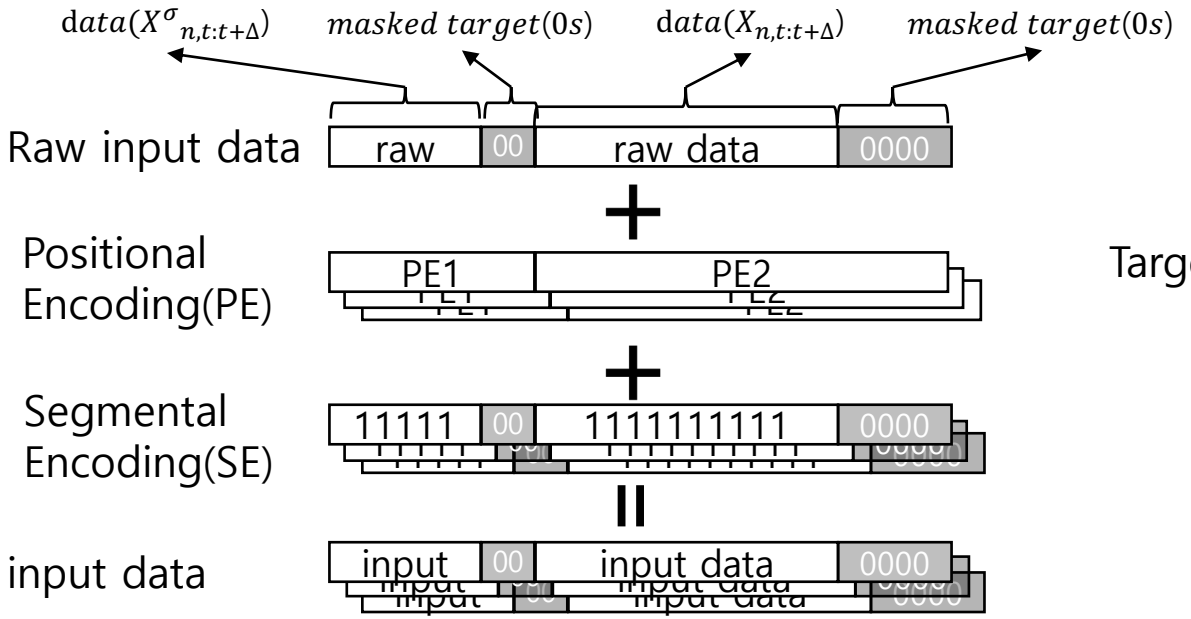
- The dataset consists of multiple shot data from the KSTAR campaign.
- We divided the dataset into 2 sets and process in parallel on 2 GPUs
- At every epoch, Each shot data in the dataset is shuffled in a randomized order.

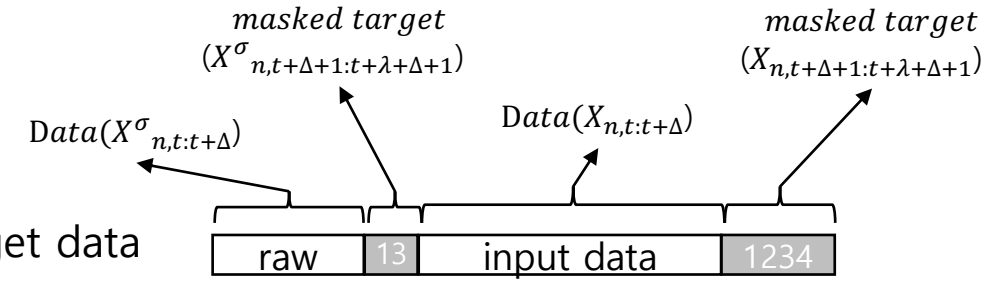


Model Architecture

❖MSRT Embedded Encodings

- To encode the position and mask information of the data, we create a position encoding(PE) and a segment encoding(SE) and add them to the input data.
  - We use Positional Encoding(PE) to encode position in input data.
  - We use Segmental Encoding(SE) to encode masked area in input data.

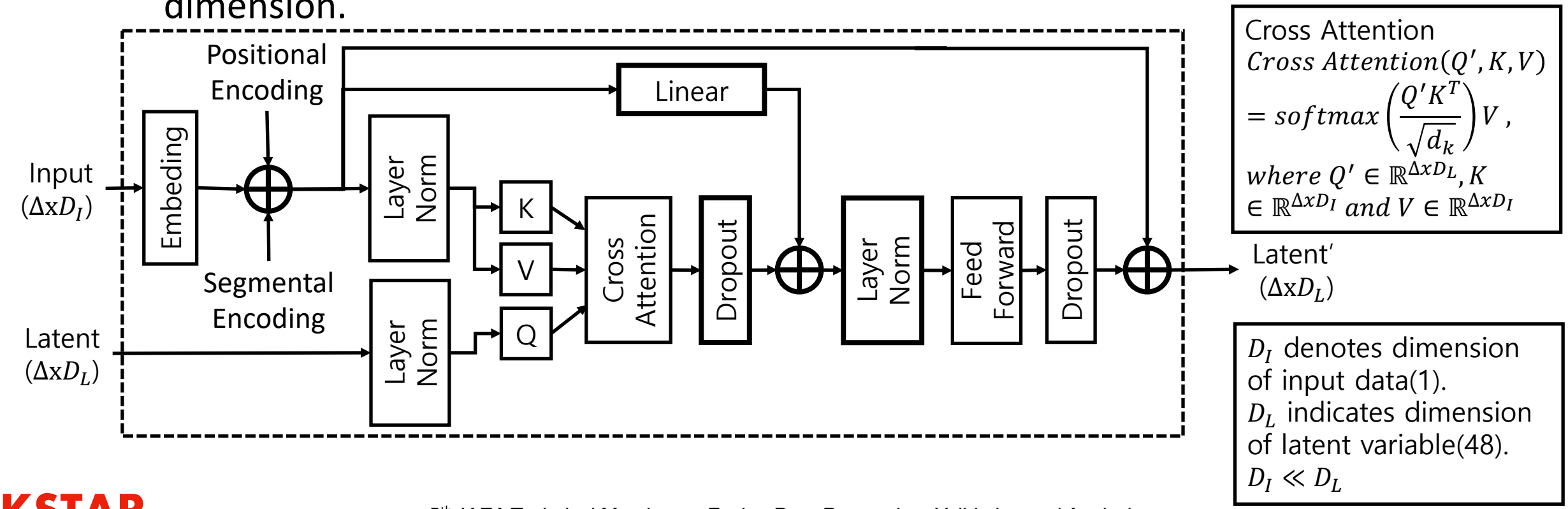




Model Architecture

❖Cross Transformer(CT)

- CT attend current input (X_{n,t_i}) to Latent(Z_{n,t_i}).
- CT uses Cross Attention module to attend input to latent which has different dimension.



Cross Attention

$Cross\ Attention(Q', K, V)$

$= softmax\left(\frac{Q'K^T}{\sqrt{d_k}}\right)V$,

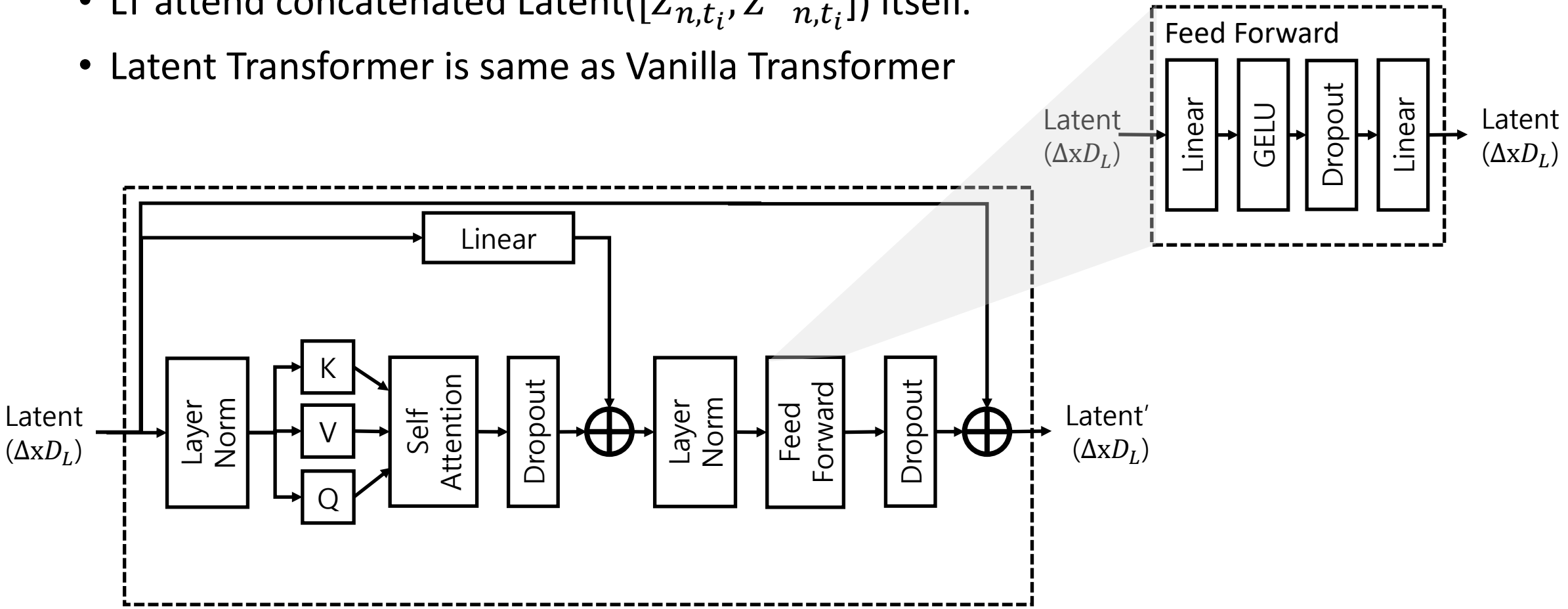
$where\ Q' \in \mathbb{R}^{\Delta x D_L}, K \in \mathbb{R}^{\Delta x D_I}\ and\ V \in \mathbb{R}^{\Delta x D_I}$

D_I denotes dimension of input data(1).
D_L indicates dimension of latent variable(48).
$D_I \ll D_L$

Model Architecture

❖Latent Transformer

- LT attend concatenated Latent($[Z_{n,t_i}, Z^{\sigma}{}_{n,t_i}]$) itself.
- Latent Transformer is same as Vanilla Transformer

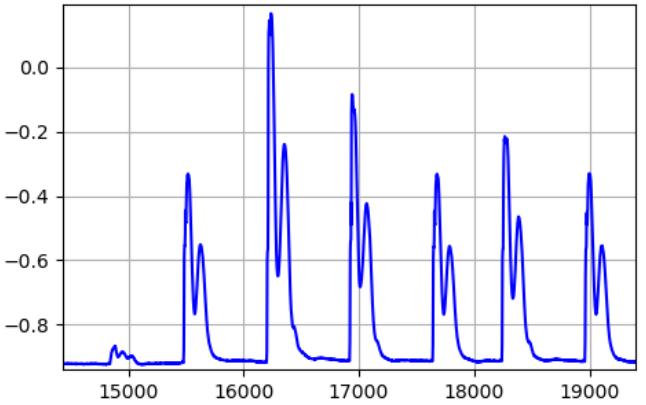


Experiment

❖KSTAR Dataset

- Data acquired from Magnet Power Supply(MPS), Helium Distribution System(HDS), Tokamak Monitoring System(TMS).
  - MPS generate PF current related PV, TMS generate temperature related PV, HDS generate helium pressure and flow rate related PV.
- We have interpolated the data to have each data every 1 seconds.
- The dataset consists of data collected while 2018 KSTAR campaign (1085 shots,24days)
  - train: 870 shot, test: 162 shot, validation : 55 shot.

Example Data



Experiment

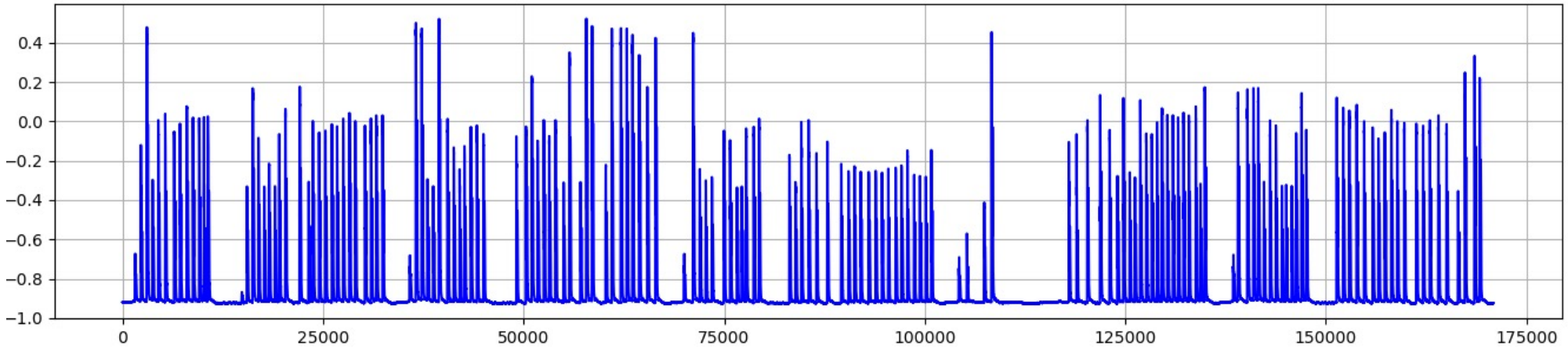
❖Environment

- We use Pytorch library to build our program.
- We use 2 GPU card when we training the dataset.

Hardware specification

	Name	# of HW
CPU	AMD Ryzen Threadripper 1920x	1
GPU	Nvidia 2070 super	2

Dataset



Experiment

❖Training

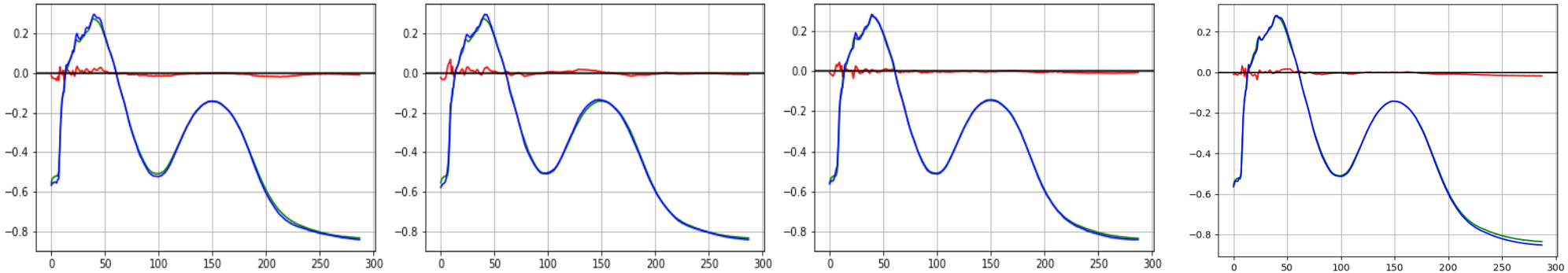
- We used Adam W optimizer with $\beta_1 = 0.9, \beta_2 = 0.99\ and\ \varepsilon = 10^{-8}$. And StepLR scheduler with step size = 1.0 and $\gamma = 0.9$.
- Model dimension = 512 and latent dimension = 64
- Training Time : it takes 72 hour to train the MSRT model that uses 100 length sequence as input(32 batch).

❖Model Parameter Setting

- To compare the prediction result, each model were setup to have the similar parameters.
- For the consideration of real time case, we use small model.
  - LSTM : hidden size = 64, # of layer = 1, Epoch = 200, batch size = 32, input window = 100, input dimension = 1
  - Transformer : # of head = 8, # of layer = 2, Epoch = 200, batch size = 32, input window = 100, input dimension = 1
  - RT, MSRT: # of head = 8, # of layer = 1, Epoch = 200, batch size = 32, input window = 100, input dimension = 1

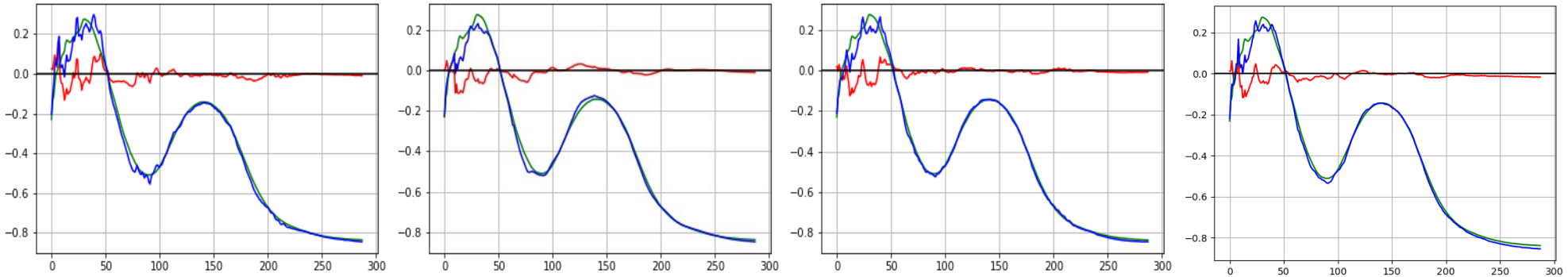
Experiment Result: 19993 shot

1 time step ahead prediction



(a) Transformer (b) TransformerXL (c) RT model (g) MSRT model

10 time step ahead prediction

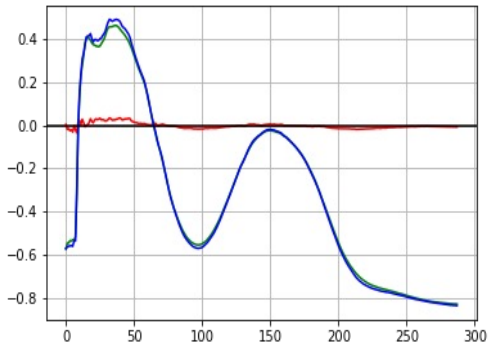


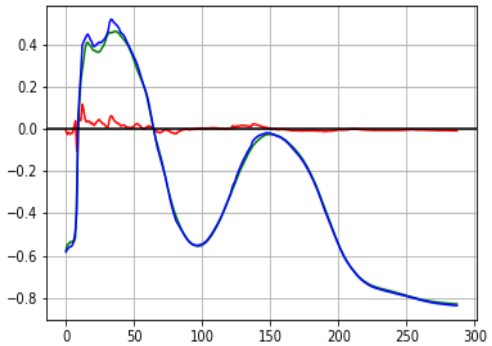
(a) Transformer (b) TransformerXL (c) RT model (g) MSRT model

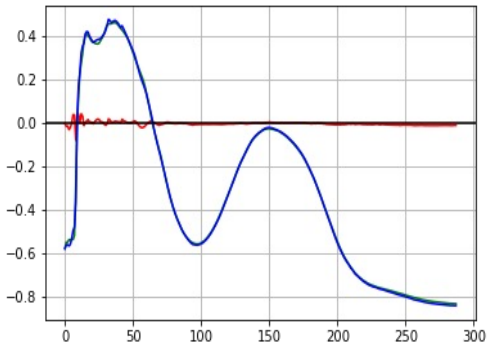
Green: ground truth Blue: predicted, Red: error

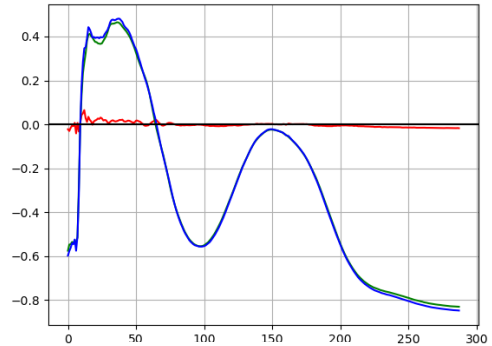
Experiment Result : 20158 shot

1 time step ahead prediction

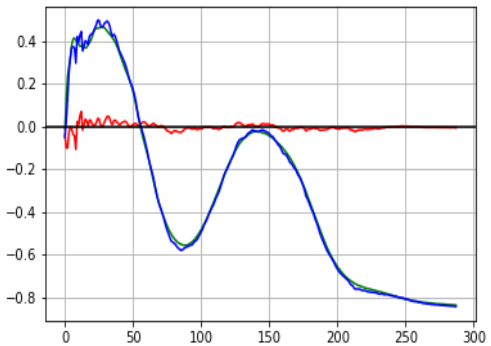

(a) Transformer

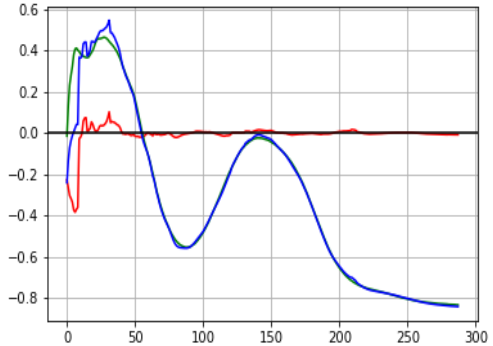

(b) TransformerXL

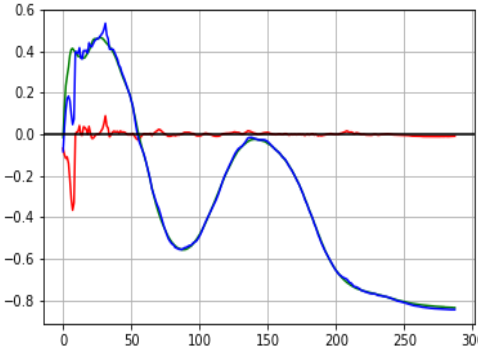

(c) RT model

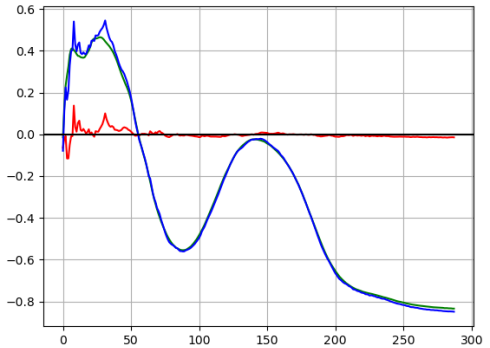

(g) MSRT model

10 time step ahead prediction


(a) Transformer


(b) TransformerXL

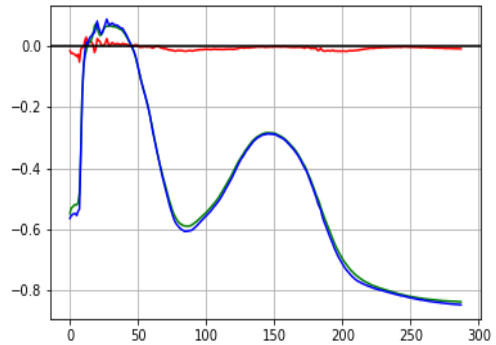

(c) RT model

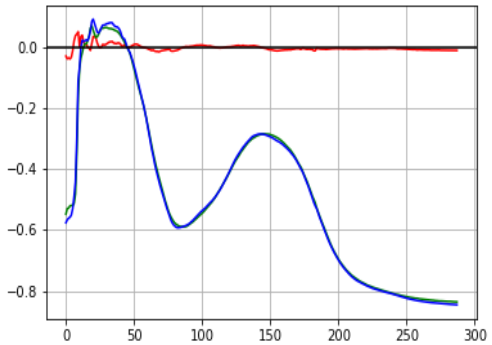

(g) MSRT model

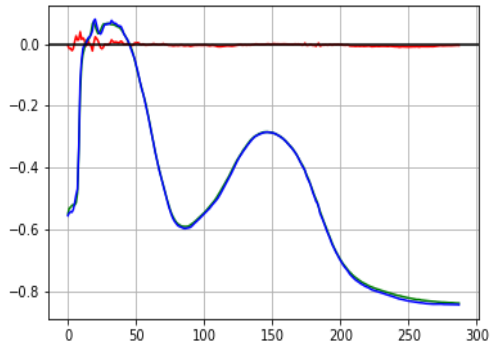
Green: ground truth Blue: predicted, Red: error

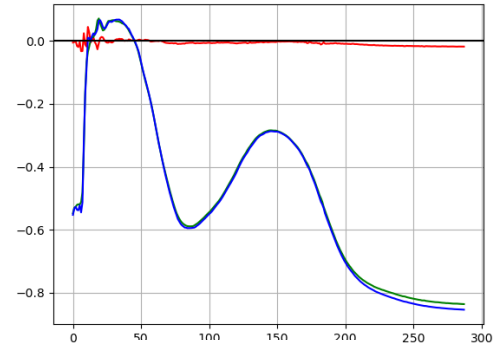
Experiment Result : 20169 shot

1 time step ahead prediction

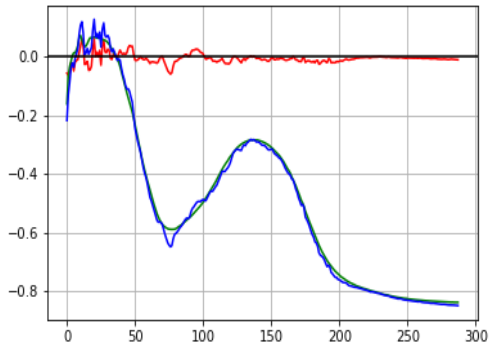

(a) Transformer

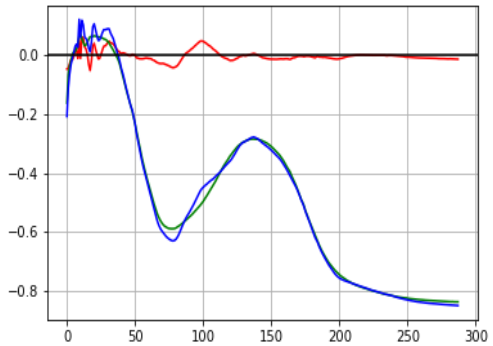

(b) TransformerXL

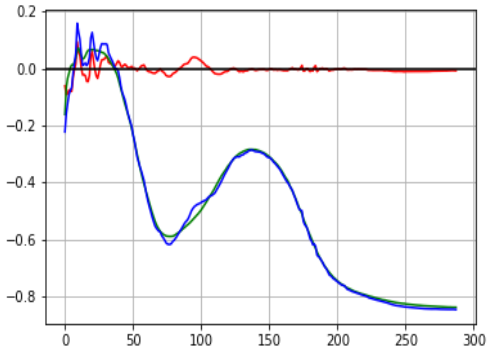

(c) RT model

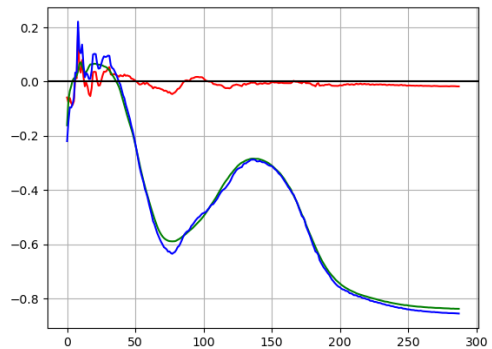

(g) MSRT model

10 time step ahead prediction


(a) Transformer


(b) TransformerXL

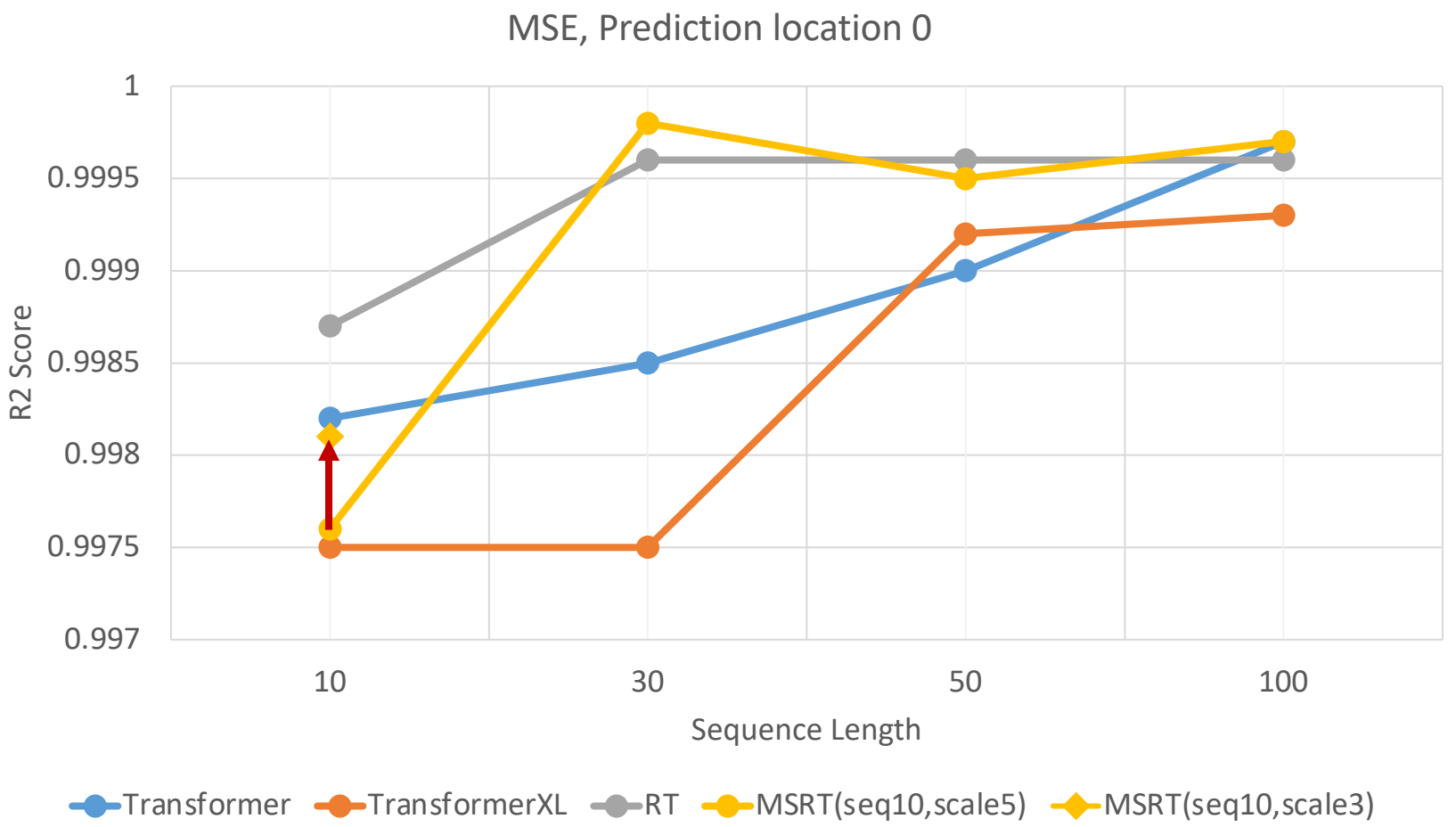

(c) RT model


(g) MSRT model

Green: ground truth Blue: predicted, Red: error

Experiment Result

- 1 step ahead forecast test result

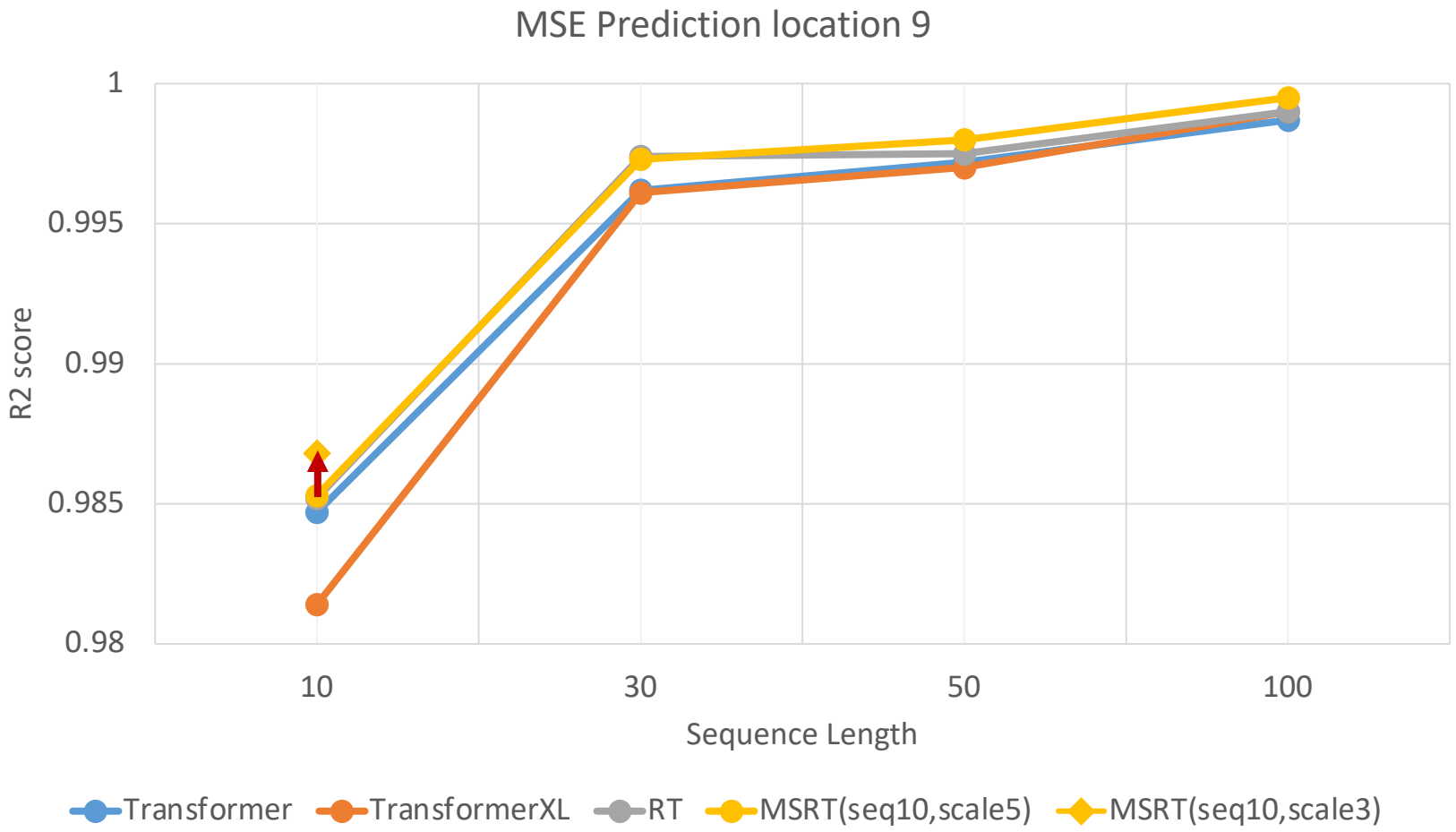


❖R2-score of 1 step ahead prediction.

- Our model (MSRT) has a higher score than the other models for all sequence lengths except for the sequence length of 10.
  - In the MSRT model, an input sequence length of 10 is scaled to 2 because it has a scale factor 5.
  - But 2 is too short to contain information and hinders the learning of the input data.
  - To validate this idea, we also change the scale factor 5 to 3. and as a result the r2 score increase.

Experiment Result

- 10 step ahead forecast test result

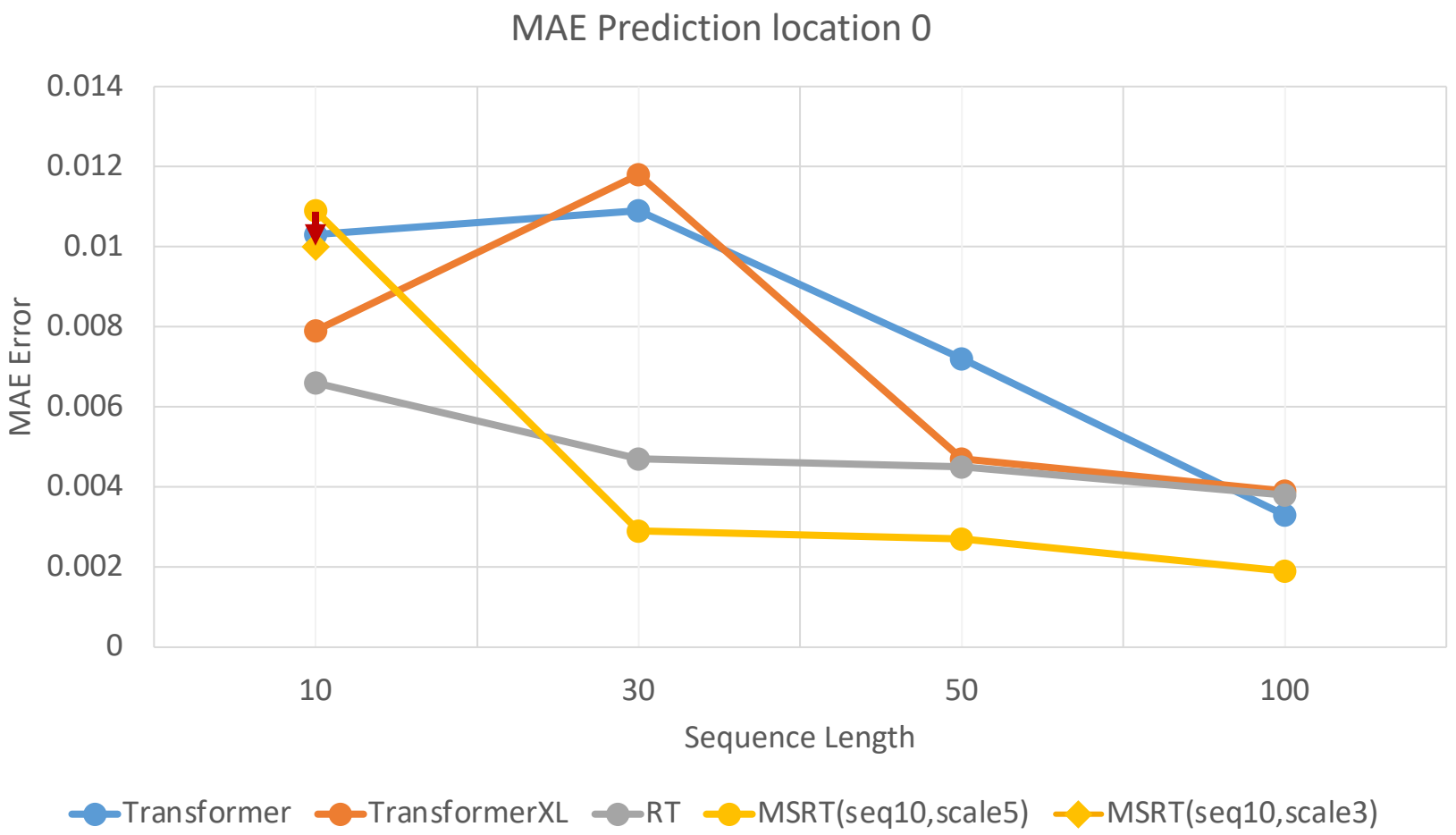


❖R2-score of 10 step ahead prediction.

- Our model (MSRT) has a higher score than the other models for all sequence lengths

Experiment Result

- 1 step ahead forecast test result

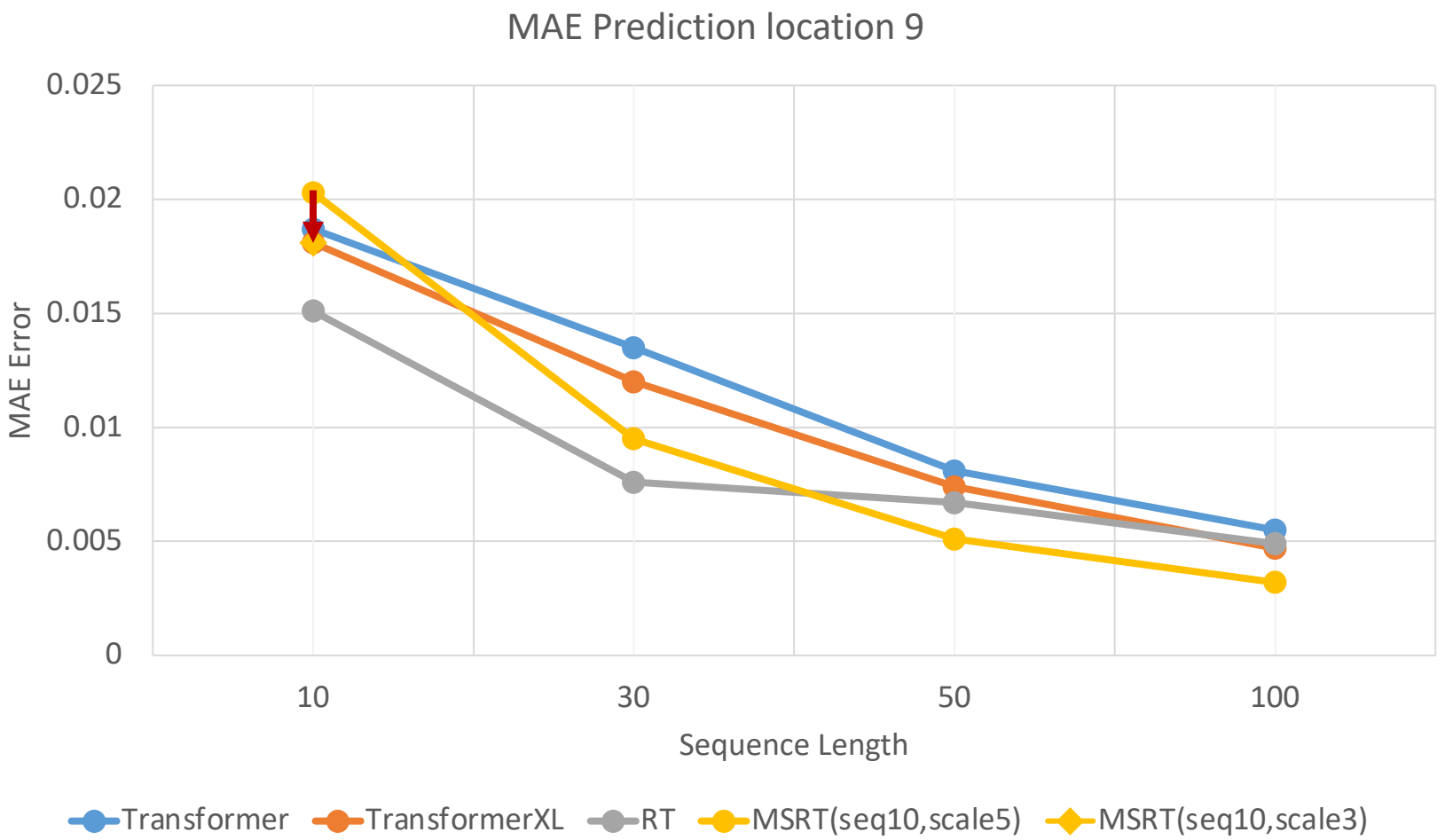


❖MAE loss of 1 step ahead prediction.

- Our model (MSRT) has a lower error than the other models for all sequence lengths except for the sequence length of 10.

Experiment Result

- 10 step ahead forecast test result



❖MAE Loss of 10 step ahead prediction.

- Our model (MSRT) has a lower error than the other models for sequence 50 and 100 lengths.

Experiment Result

- Table of prediction test result.
- We highlight the top-3 performance result with color.
-For most metrics, MSRT model using sequence 100 length outperformed the other models.

Total length :315 Output 10	R2 score ↑		MAE ↓		RMSE ↓		MAPE ↓	
	Pt 0	Pt 9	Pt 0	Pt 9	Pt 0	Pt 9	Pt 0	Pt 9
LSTM(seq i10)	0.7915	0.9874	0.0856	0.0141	0.1303	0.0360	0.6140	0.5157
Transformer(seq i10)	0.9982	0.9847	0.0103	0.0187	0.0136	0.0404	0.0770	0.5273
Transformer(seq i30)	0.9985	0.9962	0.0109	0.0135	0.0119	0.0185	0.0559	0.0894
Transformer(seq i50)	0.9990	0.9972	0.0072	0.0081	0.0086	0.0133	0.0278	0.0470
Transformer(seq i100)	0.9997	0.9987	0.0033	0.0055	0.0042	0.0083	0.0144	0.0221
TransformerXL (seq i10)	0.9975	0.9814	0.0079	0.0181	0.0160	0.0443	0.1238	1.2441
TransformerXL (seq i30)	0.9975	0.9961	0.0118	0.0120	0.0152	0.0183	0.1109	0.1615
TransformerXL (seq i50)	0.9992	0.9970	0.0047	0.0074	0.0076	0.0135	0.0592	0.1043
TransformerXL (seq i100)	0.9993	0.9990	0.0039	0.0047	0.0061	**0.0072**	0.0342	**0.0200**
RT(seq i10)	0.9987	0.9852	0.0066	0.0151	0.0116	0.0393	0.0562	0.5552
RT(seq i30)	0.9996	0.9974	0.0047	0.0076	0.0061	0.0150	0.0442	0.1077
RT(seq i50)	0.9996	0.9975	0.0045	0.0067	0.0052	0.0122	0.0345	0.0775
RT(seq i100)	0.9996	0.9990	0.0038	0.0049	0.0046	**0.0072**	0.0160	0.0271
Msrt(seq i10 $\sigma = 5$)	0.9976	0.9853	0.0109	0.0203	0.0158	0.4428	0.0797	0.4428
Msrt(seq i10 $\sigma = 3$)	0.9981	0.9868	0.0100	0.0181	0.0141	0.0380	0.0847	0.3822
Msrt(seq i30 $\sigma = 5$)	**0.9998**	0.9973	0.0029	0.0095	0.0045	0.0745	0.0406	0.0745
Msrt(seq i50, $\sigma = 5$)	0.9995	0.9980	0.0027	0.0051	0.0061	0.0503	0.1733	0.0503
Msrt(seq i100, $\sigma = 5$)	0.9997	**0.9995**	**0.0019**	**0.0032**	**0.0039**	0.0265	**0.0127**	0.0265

Experiment

❖1/10 step ahead forecast test result

- We compare forecast accuracy of the Multi Scale Recurrent Transformer(MSRT) with other algorithms(LSTM, Transformer, TransformerXL, RT) which has similar parameters(batch size=32, model dimension = 64, # of epoch = 100).
  - MSRT, with sequence length 10(seq10) is less accurate than RT(seq10) .
    - ✓ This is because the added scaled data is too short (length 2) to have any additional information about the signal and hinder data learning.
  - MSRT, with sequence length 100(seq100) is most accurate than others
  - MSRT with sequence length 30(seq30) also has similar forecast accuracy to the conventional Transformer with sequence length 100.

Conclusion

- ❖We have presented the Multi-Scale Recurrent Transformer, a Recurrent Transformer based model that can 10 step ahead forecast of univariate time series data.
  - Our model successfully learn, and 10 step ahead forecast of KSTAR PF1 temperature data.
  - Our model predicted time series data more accurately than other existing models(LSTM, Transformer, TransformerXL).
- ❖In the future, we would like to forecast multiple channel data using multi cross transformer model.