SAITS : SELF-ATTENTION-BASED IMPUTATION FOR TIME SERIES

고려대학교 산업경영공학과
데이터 애널리틱스 및 헬스케어 시스템 연구실
202202144 석사과정
김홍범

1. Introduction

1. Traditional Methods

- 결측치 삭제(Deletion)
  - sample 혹은 feature를 제거 -> 파라미터 추정에 bias가 생김

- 결측치 대체(Imputation)
  - unbiased하고 주변 값의 정보를 이용해 대체 -> 그렇다면 어떤 값을 채워야 할까?

- 결측치 종류
  - MCAR(Missing completely at random)
  - MAR(Missing at random)
  - MNAR(missing not at random)

2. Related work

- **RNN-based**
  - M-RNN, BRITS등의 모델은 bi-rnn 의 hidden state에 따라 결측치 처리를 진행
  - 이처럼 RNN을 통한 Time decay를 학습할 수 있었음
- **GAN-based**
  - GAN 역시 근간은 RNN이나, 생성자와, 판별자를 기반으로 학습을 진행함
  - 위의 두 모델의 근간은 RNN이고, Bi-RNN을 활용한다 할지라도 두 방향의 평균으로 대치한 결과로서 완전한 양방향 모델은 아님
- **VAE(Variation auto-encoder)-based**
  - Latent space에서의 Gaussian process를 기반으로 하여 결측치 처리를 진행함
  - 데이터의 구체적인 구조 또는 분포와 일치하지 않을 때가 존재함
  - GAN, VAE 모두 훈련이 어렵다는 단점을 가지고 있음
- **Self-attention-based**
  - Self-attention 기반으로 진행됨, 대표적인 모델은 NRTSI
  - NRTSI는 두 개의 중첩된 루프로 구성되어 병렬로 계산되는 Self-attention의 이점을 약화시킴

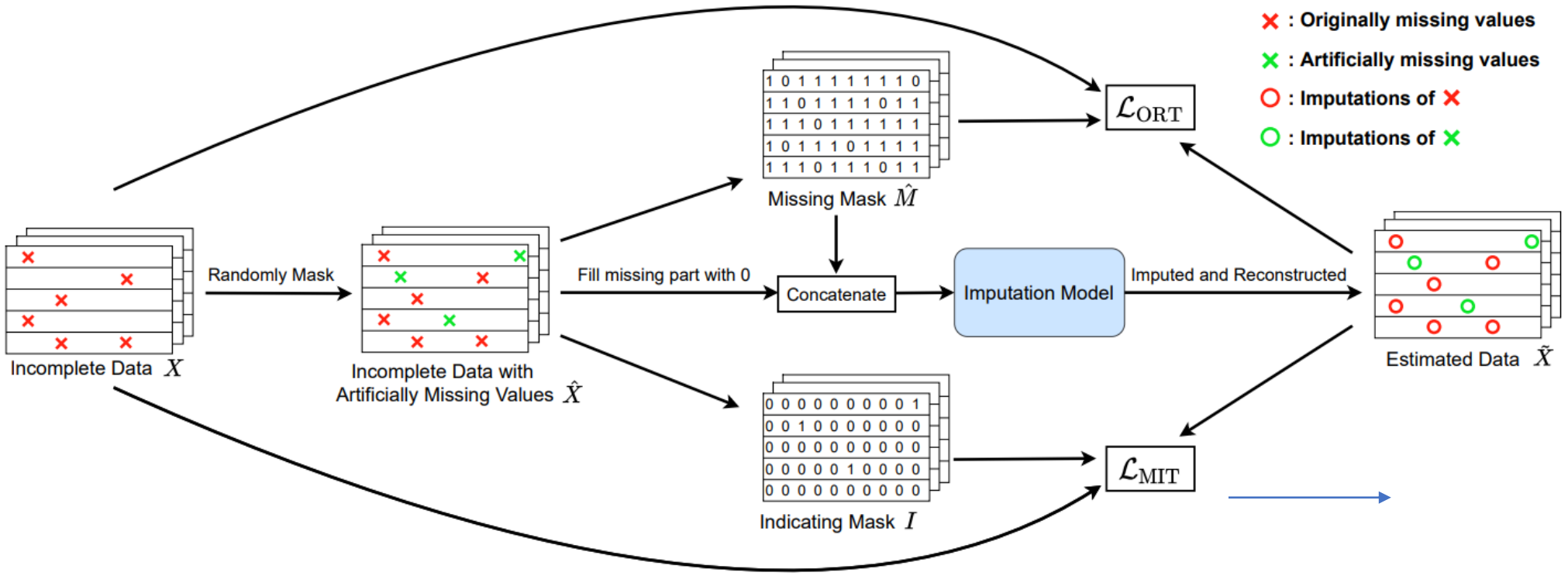
3. Methodology

- Joint-optimization training approach
- Two learning tasks
  1. MIT(Masked Imputation task) -> 인위적인 결측치 생성 후 예측 진행
  2. ORT(Observed Reconstruction task) -> 실제 결측치 예측 진행

- SAITS model(Weighted combination of two DMSA blocks)
  - Diagonally-masked self-attention
  - Positional encoding and feed-forward network
  - The second DMSA block
  - The weighted combination block
  - Loss function of learning objectives

3. Methodology

1. Joint-optimization training approach



3. Methodology

2. Two learning tasks

$\hat{x}$: Actual input time series $\quad\quad$ $\widehat{M}$: Missing mask vector

$\tilde{x}$: Estimated time series(reconstructions)

$$\hat{M}_t^d = \begin{cases} 1 & \text{if } X_t^d \text{ is observed} \\ 0 & \text{if } \hat{X}_t^d \text{ is missing} \end{cases}, \quad I_t^d = \begin{cases} 1 & \text{if } \hat{X}_t^d \text{ is artificially masked} \\ 0 & \text{otherwise} \end{cases}$$

- Masked Imputation Task (MIT)

$$\ell_{\text{MAE}}\left(estimation, target, mask\right) = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T}|(estimation - target) \odot mask|_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T} mask_t^d}$$

$$\mathcal{L}_{\text{MIT}} = \ell_{\text{MAE}}\left(\tilde{X}, X, I\right)$$

- Observed Reconstruction Task(ORT)

$$\mathcal{L}_{\text{ORT}} = \ell_{\text{MAE}}\left(\tilde{X}, X, \hat{M}\right)$$

3. Methodology

3. SAITS(Self – attention-based Imputation for Time Series)

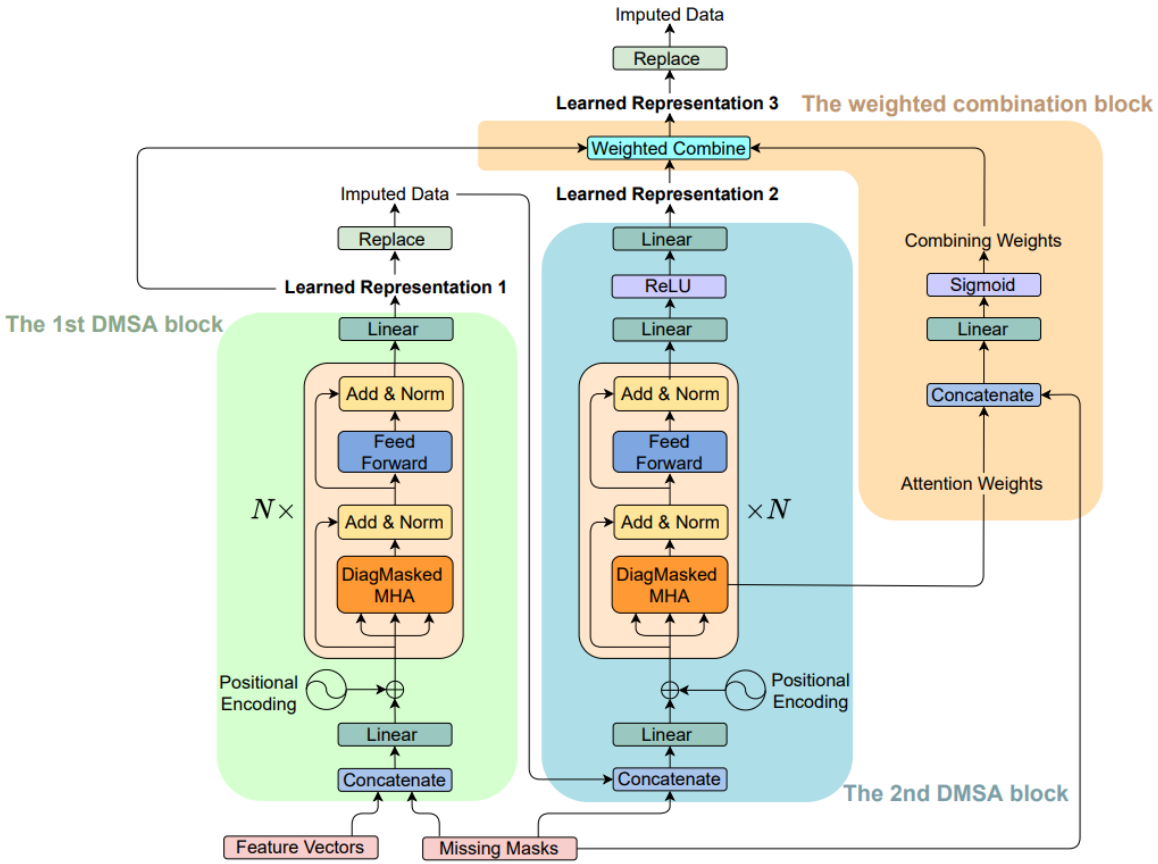
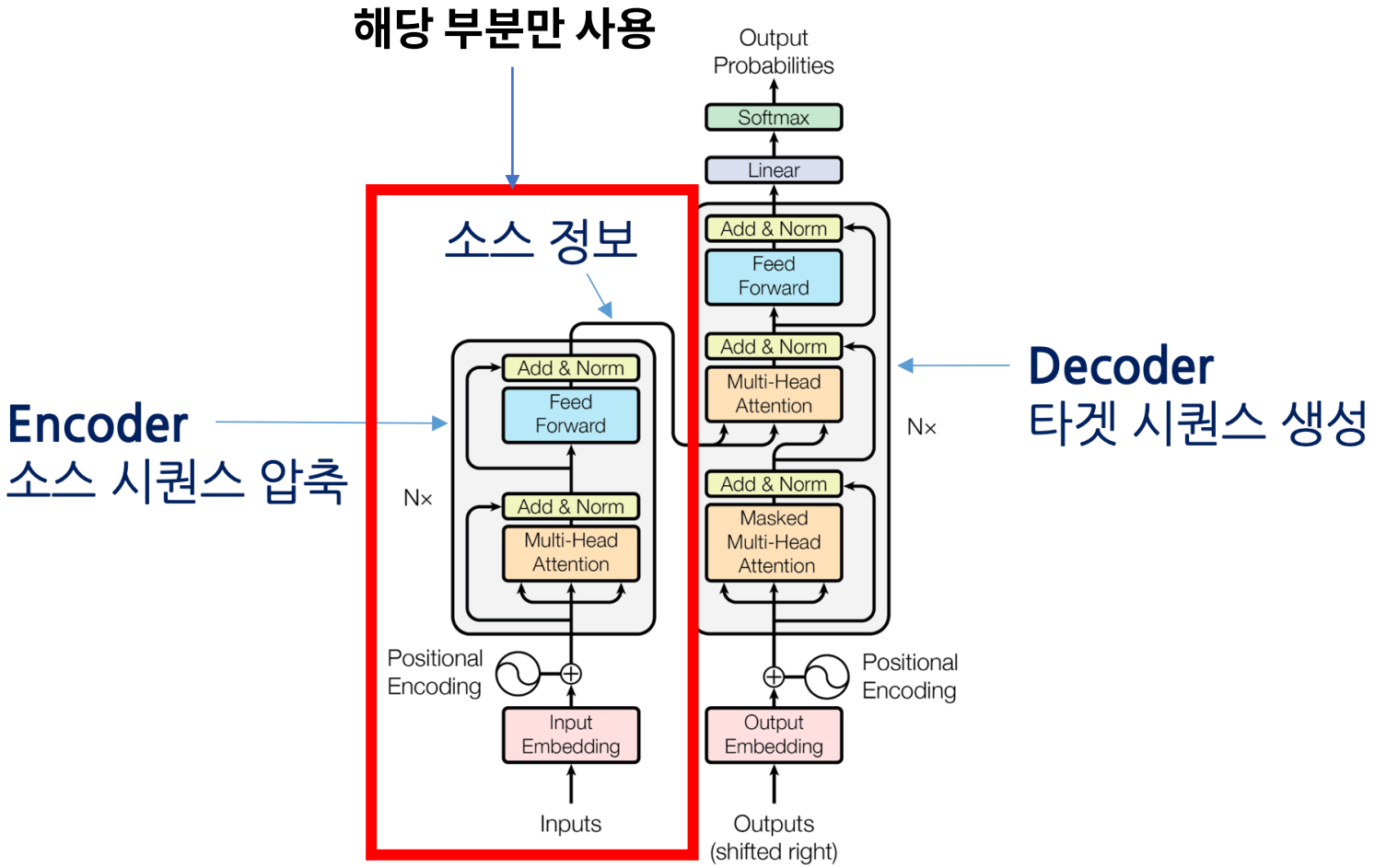


Figure 3: The SAITS model architecture.

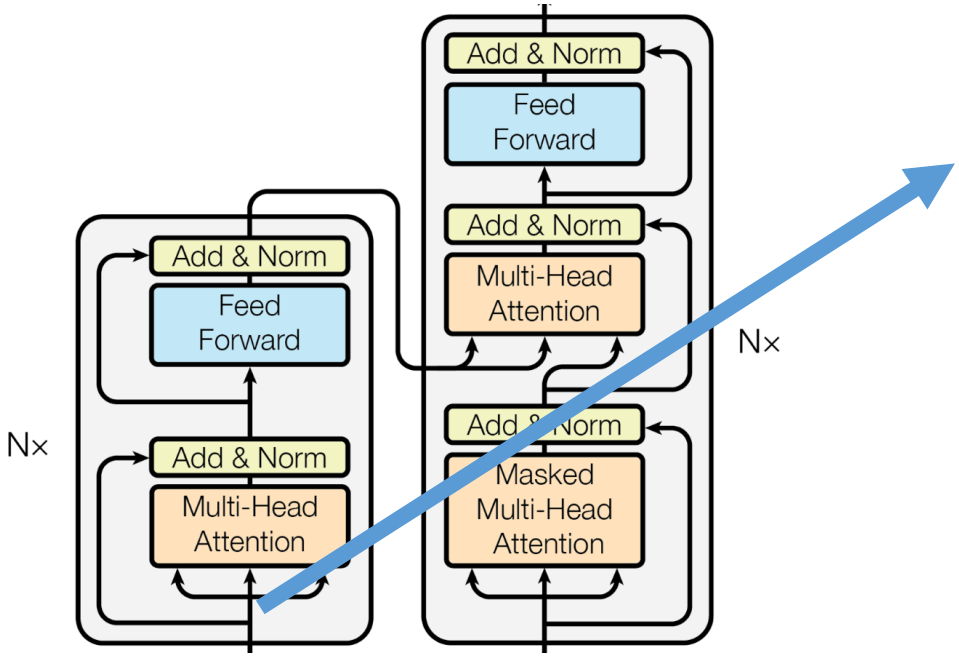
3. Methodology

4. Simple Transformer 구조



3. Methodology

5. Self-attention(Q,K,V) – Encoder 구조 파악



X : 입력 벡터 시퀀스, W : 가중치 행렬

Q=X×W_Q

K=X× W_K

V=X× W_V

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\text{K}}})\mathbf{V}$$

Self-attention

- 위의 3가지 요소사이들의 관계성을 추출하는 과정(어떤 Attention을 가지고 있는지 파악)

Multi-Head Attention

- Self attention을 여러 번 수행함
  - ◆ 단순 한번 셀프 어텐션을 진행하는 것보다 효율적임
- 방법은 셀프 어텐션 결과값을 concat 후에 그에 맞는 가중치 벡터를 곱한 형태로서 주어짐

3. Methodology

5. Self-attention(Q,K,V) – Encoder 구조 파악

- **For-example**
- $X = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $W_Q = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ $W_K = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ $W_V = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$
- **Query Vector** **Key Vector** **Value Vector**

$X \times W_Q = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 2 & 2 \\ 2 & 1 & 3 \end{bmatrix}$ $X \times W_K = \begin{bmatrix} 0 & 1 & 1 \\ 4 & 4 & 0 \\ 2 & 3 & 1 \end{bmatrix}$ $X \times W_V = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 0 \\ 2 & 6 & 3 \end{bmatrix}$

- **Self-attention example**

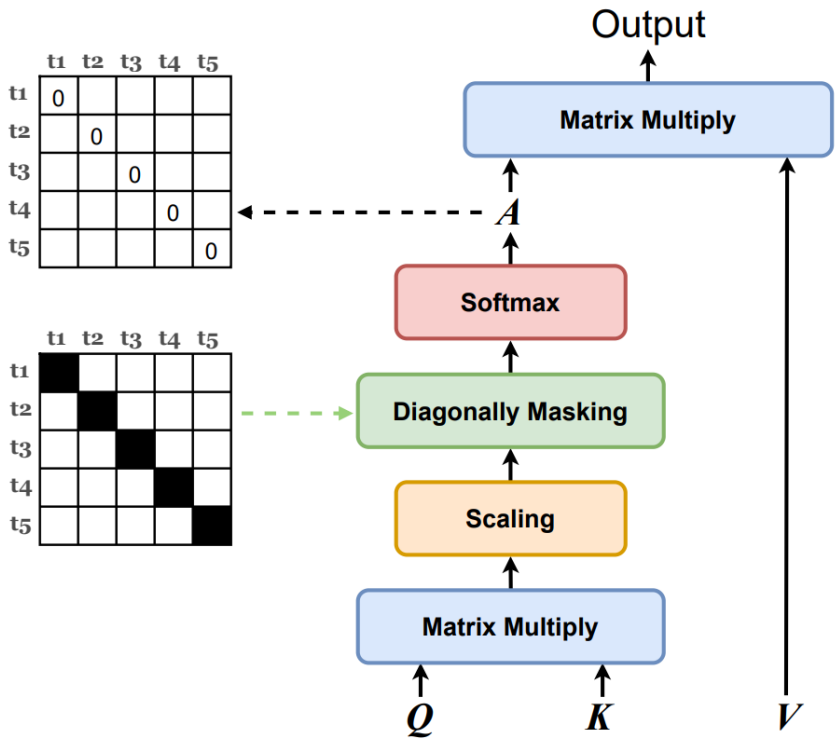
$[1 \quad 0 \quad 2] \times \begin{bmatrix} 0 & 4 & 2 \\ 1 & 4 & 3 \\ 1 & 0 & 1 \end{bmatrix} = [2 \quad 4 \quad 4]$, $\text{softmax}([\frac{2}{\sqrt{3}}, \frac{4}{\sqrt{3}}, \frac{4}{\sqrt{3}}]) = [0.13613, 0.43194, 0.43194]$

$\text{self\_attention(first query)} = [0.13613 \quad 0.43194 \quad 0.43194] \times \begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 0 \\ 2 & 6 & 3 \end{bmatrix} = [1.8639 \quad 6.3194 \quad 1.7042]$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\mathrm{K}}})\mathbf{V}$$

3. Methodology

6. Diagonally-masked self-attention



Self-attention과의 차이점

Diagonally-Masked Self Attention(DMSA)

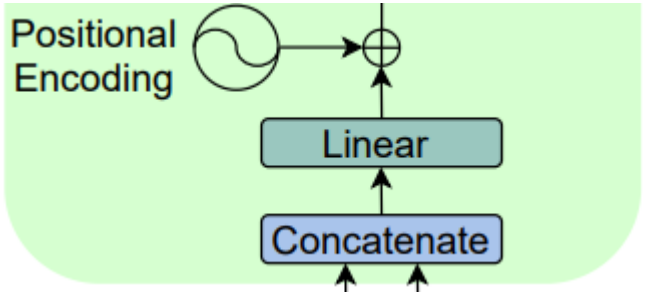
- 대각 성분은 Time-series에서 추정의 왜곡을 불러일으킴
  - ◆ Diagonally masking 진행을 통해 해결

장점

- Temporal dependency와 feature간의 correlation을 학습할 수 있음

3. Methodology

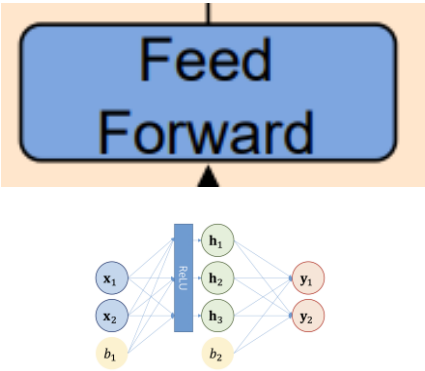
7. Positional Encoding and Feed-Forward Network



$$\text{PosEnc}(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \quad \text{PosEnc}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

where pos is the time-step position, i is the dimension

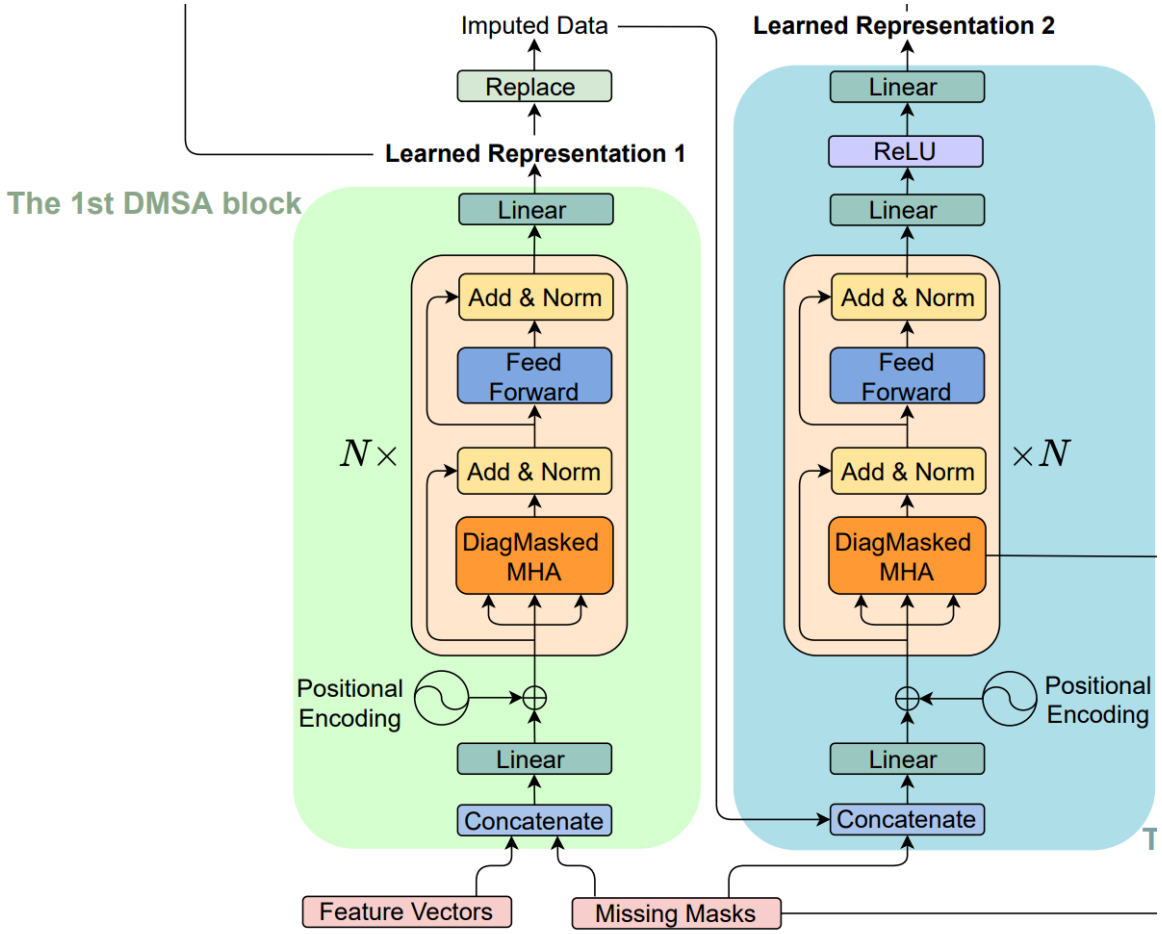
- Transformer 구조는 기존의 RNN 모델과 다르게 병렬 처리 진행
  - ◆ 데이터의 입력 순서를 잃어버림
- 이를 해결한 방법이 Positional encoding임
  - ◆ 기존의 Naïve한 방법들은 한계가 명확함(Memory loss)



- 멀티 헤드 어텐션 이후의 진행 과정
- 신경망의 한 종류로서 Input layer, hidden layer, output layer로 구성된 네트워크임

3. Methodology

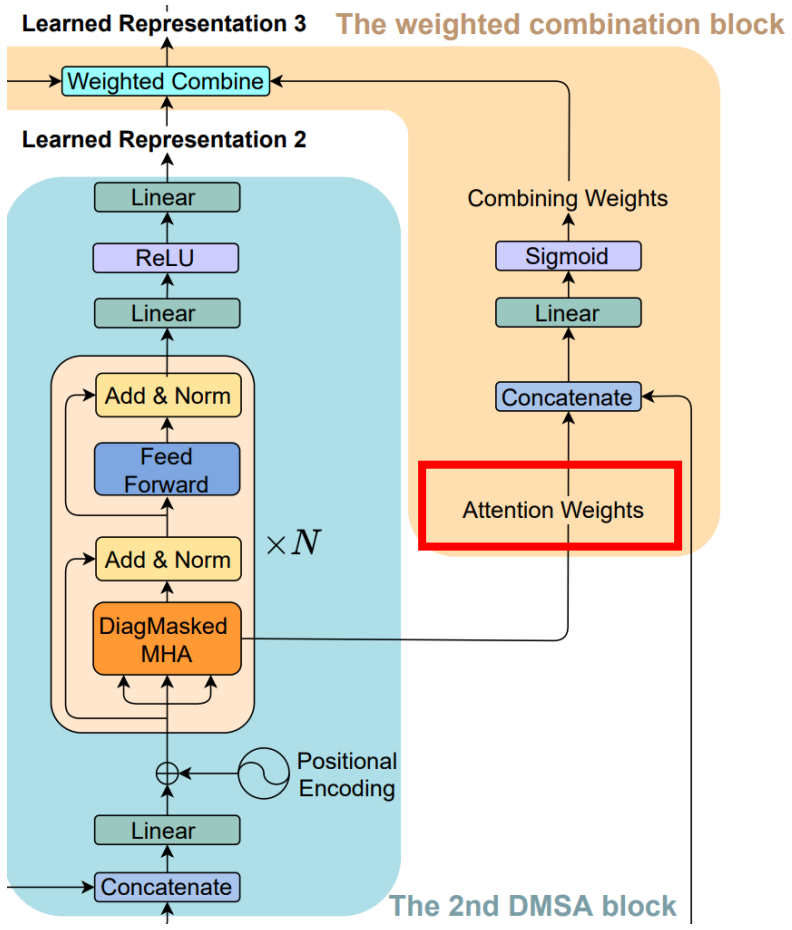
8. DMSA Block



- 총 2개의 블록 으로 구성되어 있음
- 2번째 DMSA 블록에서는 1번째 DMSA 블록에서 학습된 값을 가져와 사용함
- 각각의 블록을 거치면서 Missing value($\hat{X}$)는 $\tilde{X}_1, \tilde{X}_2$로 치환됨

3. Methodology

9. The weighted combination block



- 그림을 보면 Attention Weights를 가져오는 것을 확인할 수 있음
- 해당 가중치(η)를 통하여 $\tilde{X}_3$를 추정함

$\hat{X}_C$: Imputed data

$$\hat{A} = \frac{1}{h}\sum_{i=1}^{h} A_i$$

$$\eta = \text{Sigmoid}\left(\text{Concat}\left(\hat{A}, \hat{M}\right) W_\eta + b_\eta\right)$$

$$\tilde{X}_3 = (1-\eta) \odot \tilde{X}_1 + \eta \odot \tilde{X}_2$$

$$\hat{X}_c = \hat{M} \odot \hat{X} + \left(1-\hat{M}\right) \odot \tilde{X}_3$$

3. Methodology

10. Loss Function of Learning Objectives

$$\mathcal{L}_{\text{ORT}} = \frac{1}{3}\left(\ell_{\text{MAE}}\left(\tilde{X}_1, X, \hat{M}\right) + \ell_{\text{MAE}}\left(\tilde{X}_2, X, \hat{M}\right) + \ell_{\text{MAE}}\left(\tilde{X}_3, X, \hat{M}\right)\right)$$

$$\mathcal{L}_{\text{MIT}} = \ell_{\text{MAE}}\left(\hat{X}_c, X, I\right)$$

$$\mathcal{L} = \mathcal{L}_{\text{ORT}} + \lambda\,\mathcal{L}_{\text{MIT}}$$

- Two learning tasks들에 대하여 각각 loss를 계산함
- 최종 loss는 둘의 합으로 계산됨

4. Experiments

1. Datasets

Table 1: General information of four datasets used in this work.

	PhysioNet-2012	Air-Quality	Electricity	ETT
Number of total samples	11,988	1,461	1,400	5,803
Number of features	37	132	370	7
Sequence length	48	24	100	24
Original missing rate	80.0%	1.6%	0%	0%

- 서로 다른 도메인의 3가지 데이터셋 + 검증을 위한 데이터셋(ETT)
- PhysioNet-2012 : ICU에 입원한 환자의 48시간 상태 데이터
- Air-Quality : 베이징의 12개 모니터링 사이트에서 얻은 시간별 대기 오염물질 데이터
- Electricity : 15분마다 370명의 고객으로부터 수집된 전력 소비 데이터(결측치 X -> 인공 결측치 생성)
- ETT : 2년동안 15분마다 수집된 오일 온도와 6가지 유형의 외부 전력 부하 기능을 포함한 데이터(결측치 X -> 인공 결측치 생성)

4. Experiments

2. Baseline methods, Experimental setup

Baseline methods

- 2가지의 Naïve한 방법과 5가지의 SOTA 딥러닝 모델들과의 성능 비교를 진행함
  - Naïve : Median, Last
  - SOTA Deep Learning : GRUI-GAN, E^2GAN, M-RNN, GP-VAE, BRITS

Experimental setup

- Evaluation Metric : MAE, RMSE, MRE
- Batch size : 128
- Early stopping : 30 epoch(without decrease of MAE)
- Optimizer : Adam
- GPU : Nvidia Quadro RTX 5000 - PyTorch

5. Results

1. Imputation performance comparison

Method	PhysioNet-2012	Air-Quality	Electricity	ETT
Median	0.726 / 0.988 / 103.5%	0.763 / 1.175 / 107.4%	2.056 / 2.732 / 110.1%	1.145 / 1.847 / 139.1%
Last	0.862 / 1.207 / 123.0%	0.967 / 1.408 / 136.3%	1.006 / 1.533 / 53.9%	1.007 / 1.365 / 96.4%
GRUI-GAN	0.765 / 1.040 / 109.1%	0.788 / 1.179 / 111.0%	/	0.612 / 0.729 / 95.1%
E^2GAN	0.702 / 0.964 / 100.1%	0.750 / 1.126 / 105.6%	/	0.584 / 0.703 / 89.0%
M-RNN	0.533 / 0.776 / 76.0%	0.294 / 0.643 / 41.4%	1.244 / 1.867 / 66.6%	0.376 / 0.428 / 31.6%
GP-VAE	0.398 / 0.630 / 56.7%	0.268 / 0.614 / 37.7%	1.094 / 1.565 / 58.6%	0.274 / 0.307 / 15.5%
BRITS	0.256 / 0.767 / 36.5%	0.153 / 0.525 / 21.6%	0.847 / 1.322 / 45.3%	0.130 / 0.259 / 12.5%
Transformer	0.190 / 0.445 / 26.9%	0.158 / 0.521 / 22.3%	0.823 / 1.301 / 44.0%	0.114 / 0.173 / 10.9%
SAITS-base	0.192 / 0.439 / 27.3%	0.146 / 0.521 / 20.6%	0.822 / 1.221 / 44.0%	0.121 / 0.197 / 11.6%
SAITS	**0.186 / 0.431 / 26.6%**	**0.137 / 0.518 / 19.3%**	**0.735 / 1.162 / 39.4%**	**0.092 / 0.139 / 8.8%**

- MAE/RMSE/MRE 순서대로 평가지표를 나타냄
- Electricity의 GRUI-GAN, E^2GAN은 훈련에 실패하여 결과 없음(Loss explosion)

5. Results

2. Performance comparison between missing rates

Method	20%	30%	40%	50%
Median	2.053 / 2.726 / 109.9%	2.055 / 2.732 / 110.0%	2.058 / 2.734 / 110.2%	2.053 / 2.728 / 109.9%
Last	1.012 / 1.547 / 54.2%	1.018 / 1.559 / 54.5%	1.025 / 1.578 / 54.9%	1.032 / 1.595 / 55.2%
M-RNN	1.242 / 1.854 / 66.5%	1.258 / 1.876 / 67.3%	1.269 / 1.884 / 68.0%	1.283 / 1.902 / 68.7%
GP-VAE	1.124 / 1.502 / 60.2%	1.057 / 1.571 / 56.6%	1.090 / 1.578 / 58.4%	1.097 / 1.572 / 58.8%
BRITS	0.928 / 1.395 / 49.7%	0.943 / 1.435 / 50.4%	0.996 / 1.504 / 53.4%	1.037 / 1.538 / 55.5%
Transformer	0.843 / 1.318 / 45.1%	0.846 / 1.321 / 45.3%	0.876 / 1.387 / 46.9%	0.895 / 1.410 / 47.9%
SAITS-base	0.838 / 1.264 / 44.9%	0.845 / 1.247 / 45.2%	0.873 / 1.325 / **46.7%**	0.939 / 1.537 / 50.3%
SAITS	**0.763 / 1.187 / 40.8%**	**0.790 / 1.223 / 42.3%**	**0.869 / 1.314 / 46.7%**	**0.876 / 1.377 / 46.9%**
Method	60%	70%	80%	90%
Median	2.057 / 2.734 / 110.2%	2.050 / 2.726 / 109.8%	2.059 / 2.734 / 110.2%	2.056 / 2.723 / 110.1%
Last	1.040 / 1.615 / 55.7%	1.049 / 1.640 / 56.2%	1.059 / 1.663 / 56.7%	1.070 / 1.690 / 57.3%
M-RNN	1.298 / 1.912 / 69.4%	1.305 / 1.928 / 69.9%	1.318 / 1.951 / 70.5%	1.331 / 1.961 / 71.3%
GP-VAE	1.101 / 1.616 / 59.0%	1.037 / 1.598 / 55.6%	1.062 / 1.621 / 56.8%	1.004 / 1.622 / 53.7%
BRITS	1.101 / 1.602 / 59.0%	1.090 / 1.617 / 58.4%	1.138 / 1.665 / 61.0%	1.163 / 1.702 / 62.3%
Transformer	**0.891** / 1.404 / **47.7%**	0.920 / 1.437 / 49.3%	0.924 / 1.472 / 49.5%	0.934 / 1.491 / **49.8%**
SAITS-base	0.969 / 1.565 / 51.9%	0.972 / 1.601 / 52.0%	1.012 / 1.608 / 54.2%	1.001 / 1.630 / 53.6%
SAITS	0.892 / **1.328** / 47.9%	**0.898 / 1.273 / 48.1%**	**0.908 / 1.327 / 48.6%**	**0.933 / 1.354** / 49.9%

- 결측치가 없는 Electricity의 인위적인 결측치 비율을 조정하며 성능 비교를 진행함(20% ~ 90%)

5. Results

3. Downstream classification task

Method	ROC-AUC	PR-AUC	F1-score
Median	83.4% ± 0.4%	46.0% ± 0.6%	38.5% ± 3.1%
Last	82.8% ± 0.3%	46.9% ± 0.4%	39.5% ± 2.4%
GRUI-GAN	83.0% ± 0.2%	45.1% ± 0.7%	38.8% ± 2.0%
E^2GAN	83.0% ± 0.2%	45.5% ± 0.5%	35.6% ± 2.0%
M-RNN	82.2% ± 0.2%	45.4% ± 0.6%	38.8% ± 3.5%
GP-VAE	83.4% ± 0.2%	48.1% ± 0.7%	40.9% ± 3.3%
BRITS	83.5% ± 0.1%	49.1% ± 0.4%	41.3% ± 1.8%
Transformer	84.3% ± 0.5%	49.2% ± 1.4%	41.2% ± 1.9%
SAITS-base	84.6% ± 0.2%	49.8% ± 0.4%	41.5% ± 2.0%
SAITS	**84.8% ± 0.2%**	**51.0% ± 0.5%**	**42.7% ± 2.8%**

- 결측치 처리를 각 모델로 진행 후 -> 환자의 사망여부를 레이블로 하여 예측을 진행함
- 데이터 간의 Imbalance가 존재하므로 AUROC, AUPRC,F1-SCORE등의 지표로 성능 비교를 진행함

5. Results

4. Comparing with NRTSI

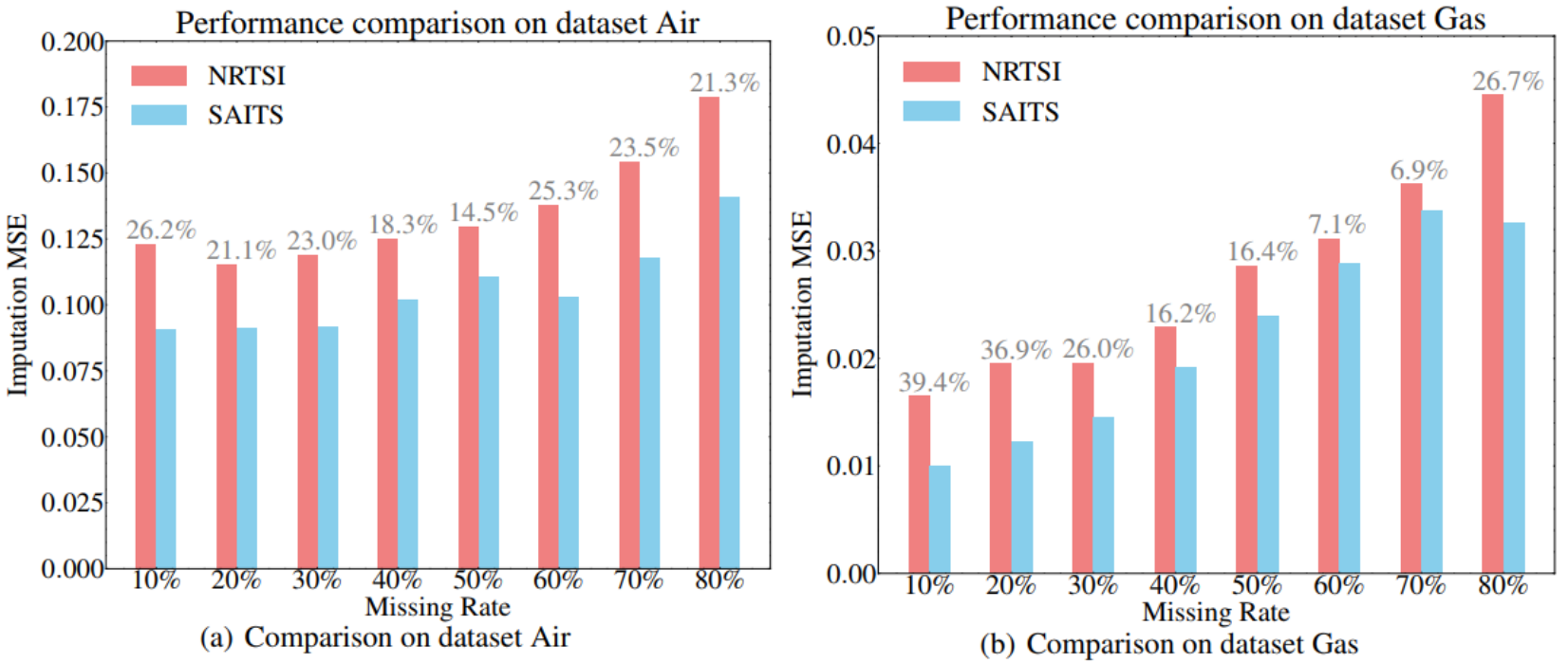
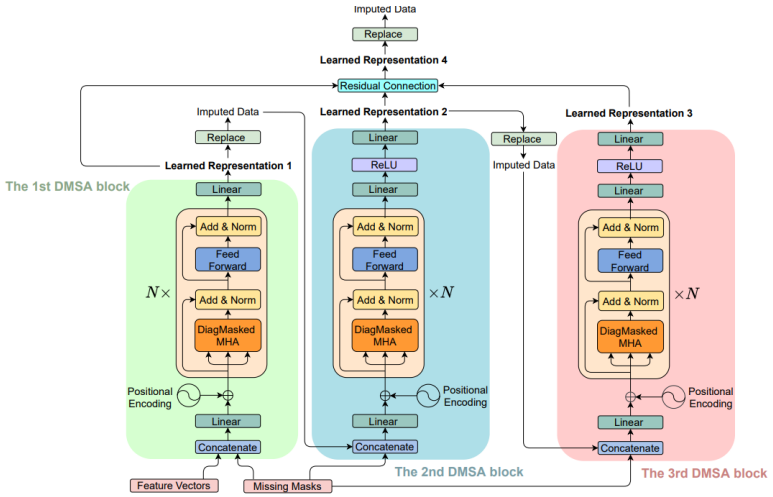


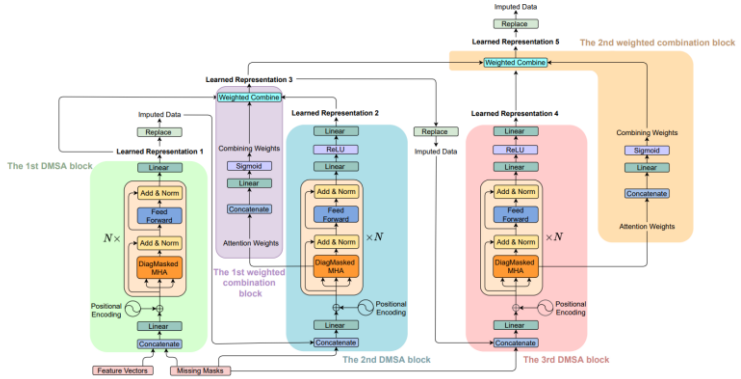
Figure 5: The visualized comparison with NRTSI on datasets Air and Gas. The percentage numbers above the bars indicate, compared with NRTSI, the amount of imputation MSE reduced by SAITS.

- Self-attention base 모델인 NRTSI 대비 높은 성능을 보임
- 전체적인 모델의 코드는 공개가 되어 있지 않고, 전처리만 공개되어 이를 이용하여 구현 후 비교 진행

6. Ablation studies

Q : 2개 이상의 DMSA 블록을 적용한다면 성능이 증가하지 않을까?





Model	PhysioNet-2012	Air-Quality	Electricity	ETT
SAITS-3residual	0.189 / 0.620 / 27.0%	0.158 / **0.509** / 22.2%	0.740 / **1.020** / 39.6%	0.103 / 0.145 / 9.6%
SAITS-3cascade	**0.185** / **0.418** / **26.4%**	0.146 / 0.512 / 20.5%	0.800 / 1.147 / 42.8%	0.096 / 0.141 / **8.8%**
SAITS	0.186 / 0.431 / 26.6%	**0.137** / 0.518 / **19.3%**	**0.735** / 1.162 / **39.4%**	**0.092** / **0.139** / **8.8%**

- 전체적인 성능에서 기존의 SAITS가 더 뛰어남
- 또한 블록의 증가에 따른 파라미터수 증가와 Compuatational resource의 낭비가 심함다는 결론을 내림

7. Conclusions

Conclusion

- Joint-optimization training approach을 통하여 SOTA 보다 높은 성능을 가지는 모델을 생성함
- 두개의 DMSA 블록의 가중 조합으로 구성된 SAITS라는 모델을 생성하였고, 해당 모델은 RNN을 사용하지 않더라도 시간 종속성 및 변수 간의 상관관계를 학습할 수 있었음
- 3가지 Real world dataset을 통해 검증을 진행함
- 또한 기존의 BRITS 대비 SAITS는 MAE를 12~38% 가량 줄일 수 있었고, 2~2.6배 빠른 훈련 속도를 달성함
- 특히 기존의 Transformer 백본 모델인 NRTSI등 보다 확실히 높은 정확도를 보였음

Future work

- 현재 MCAR로 Missing pattern을 가정하고 진행하는데, 향후 만약 missing value가 패턴을 가지는 경우 이를 적용하여 고려해볼 예정이다.(MAR 등)
- 다양한 도메인 영역에서 해당 모델을 적용하고 검증해 볼 예정

Thank you