UC San Diego

PrimeNet: Pre-Training for Irregular Multivariate Time Series

Ranak Roy Chowdhury*, Jiacheng Li, Xiyuan Zhang, Dezhi Hong, Rajesh K. Gupta, Jingbo Shang

University of California San Diego, La Jolla, CA, USA

In Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI '23)

February 7 - 14, 2023, Washington, DC, USA

*Primary Author Contact: Ranak Roy Chowdhury (https://ranakroychowdhury.github.io/)

Code is publicly available at https://github.com/ranakroychowdhury/PrimeNet

Outline

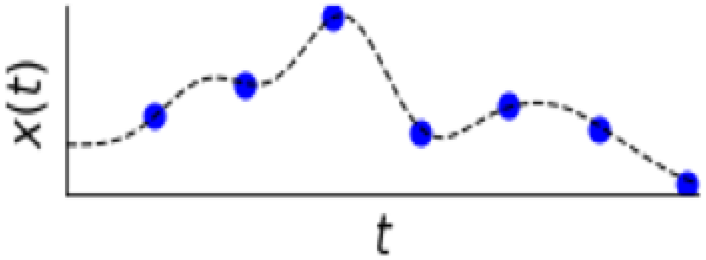
- Motivation
- Current Approaches
- Methodology
- Experiments
- Conclusion

Outline

- Motivation
- Current Approaches
- Methodology
- Experiments
- Conclusion

Motivation

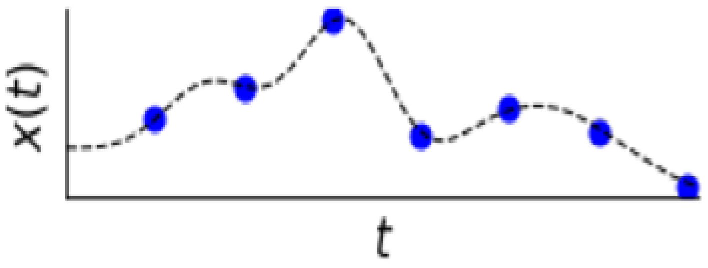
Traditional sequential modeling assumes 1) *regular*



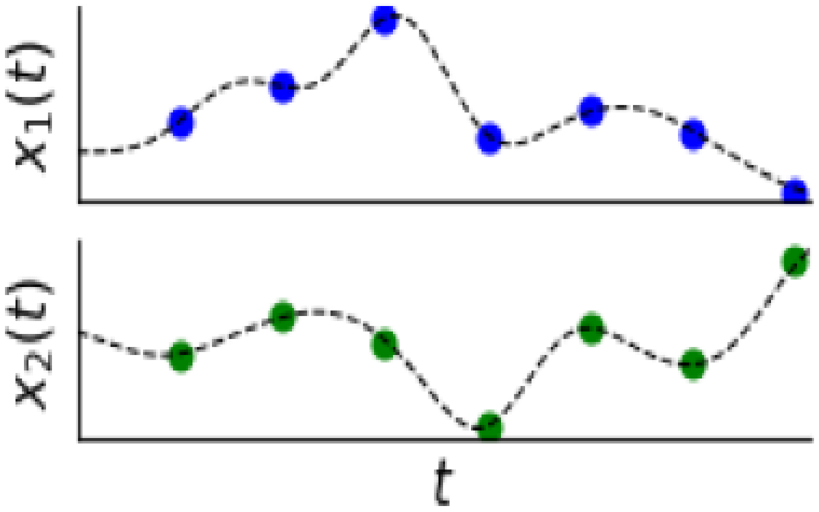
Univariate Regularly Sampled

Motivation

Traditional sequential modeling assumes 1) *regular* and 2) *synchronous* data



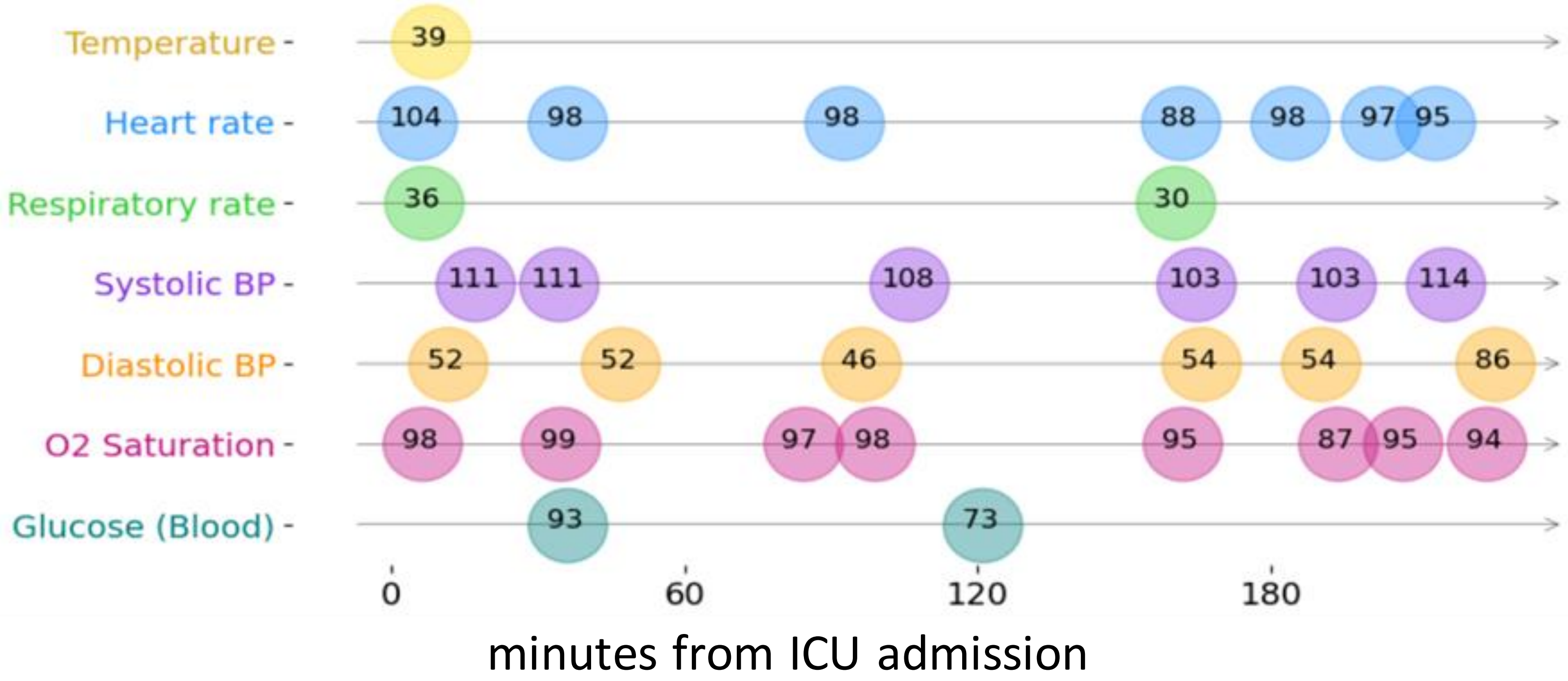
Univariate Regularly Sampled



Multivariate Regularly Sampled

Motivation

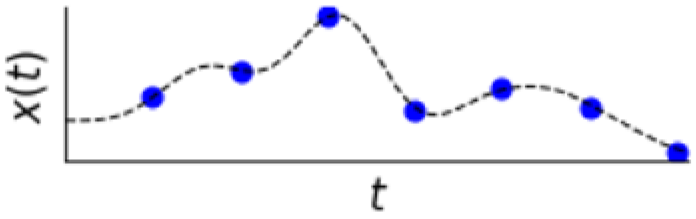
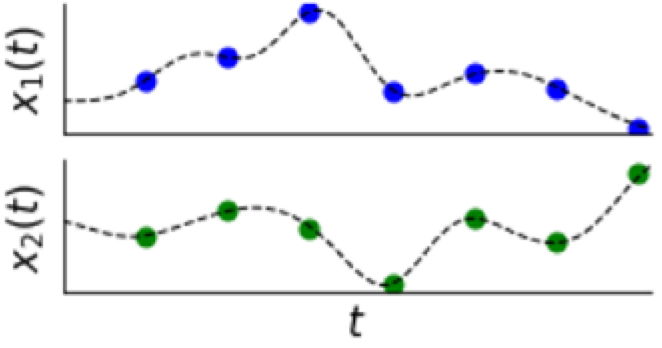
Both assumptions are violated in real-world applications, like Healthcare (Tipirneni ‘21), IoT (Lechner ‘20)



Motivation

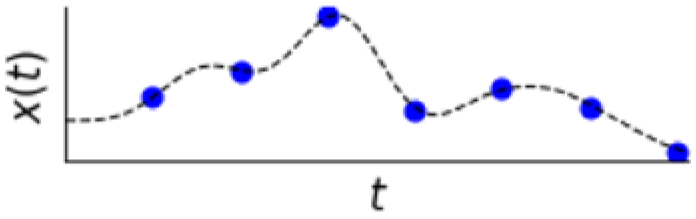
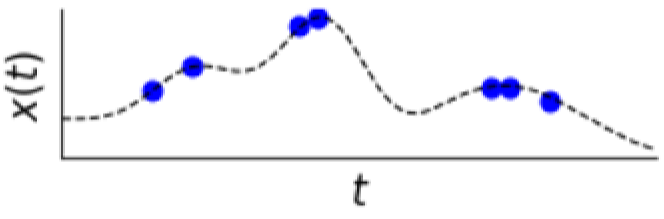
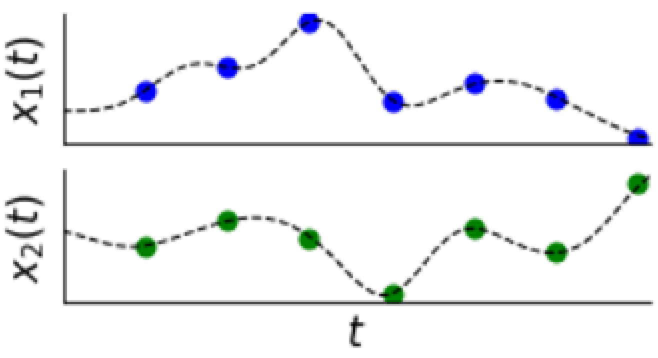
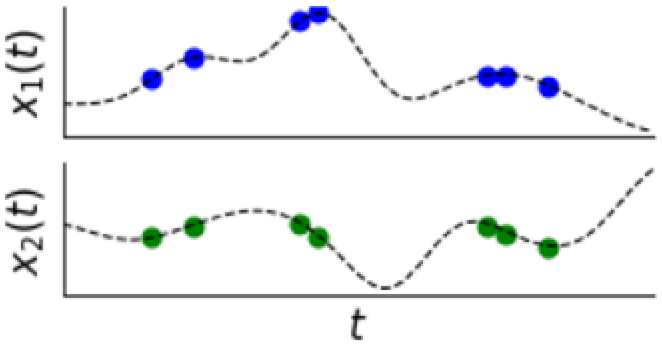
- Other Application include IoT, Finance
- IoT: power outage, network drop, different sampling frequency among different sensors in a system
- Finance: Consumer spending pattern, stock trading frequency

Motivation

	Regular
Uni-variate	
Multi-variate	

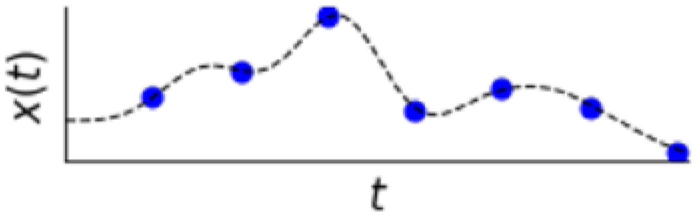
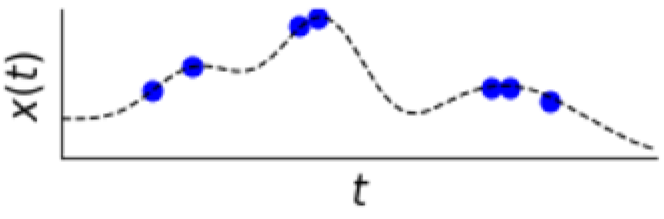
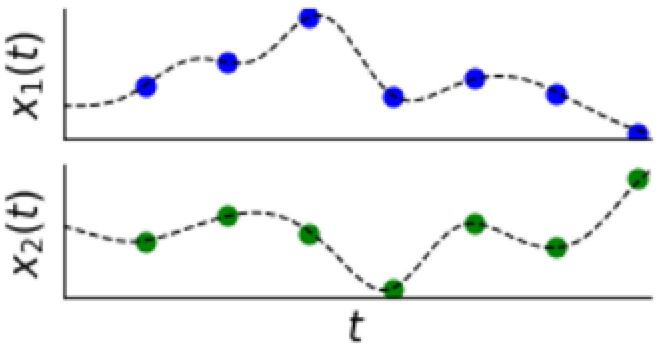
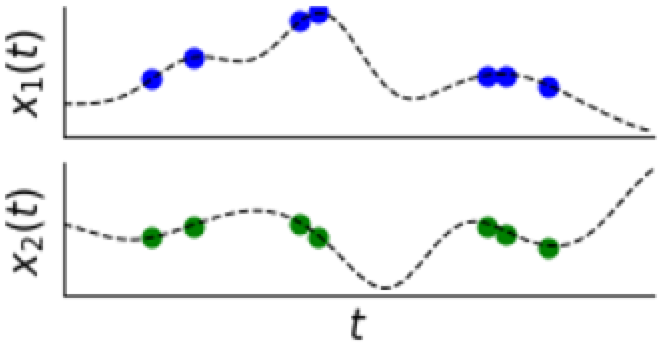
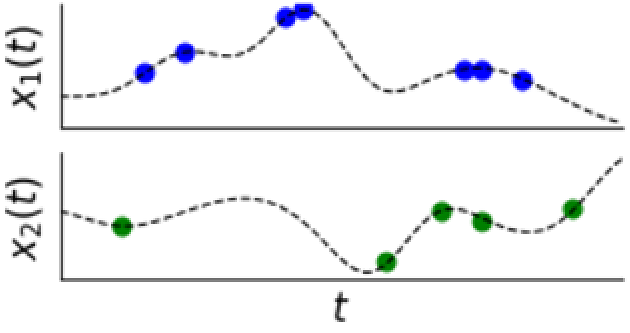
Reference: Shukla, Satya Narayan, and Benjamin M. Marlin. "A survey on principles, models and methods for learning from irregularly sampled time series." *arXiv preprint arXiv:2012.00168* (2020).

Motivation

	Regular	Irregular and Synchronous
Uni-variate		
Multi-variate		

Reference: Shukla, Satya Narayan, and Benjamin M. Marlin. "A survey on principles, models and methods for learning from irregularly sampled time series." *arXiv preprint arXiv:2012.00168* (2020).

Motivation

	Regular	Irregular and Synchronous	Irregular and Asynchronous
Uni-variate			
Multi-variate			

Reference: Shukla, Satya Narayan, and Benjamin M. Marlin. "A survey on principles, models and methods for learning from irregularly sampled time series." *arXiv preprint arXiv:2012.00168* (2020).

Outline

- Motivation
- Current Approaches
- Methodology
- Experiments
- Conclusion

Current Approaches

Two approaches to learn from irregular time series:

1. Transform irregular to regular:
    a. Discretization followed by interpolation

Current Approaches

Two approaches to learn from irregular time series:

1. Transform irregular to regular:
   a. Discretization followed by interpolation
2. Directly model irregularity:
   a. ODE
   b. Set
   c. Attention

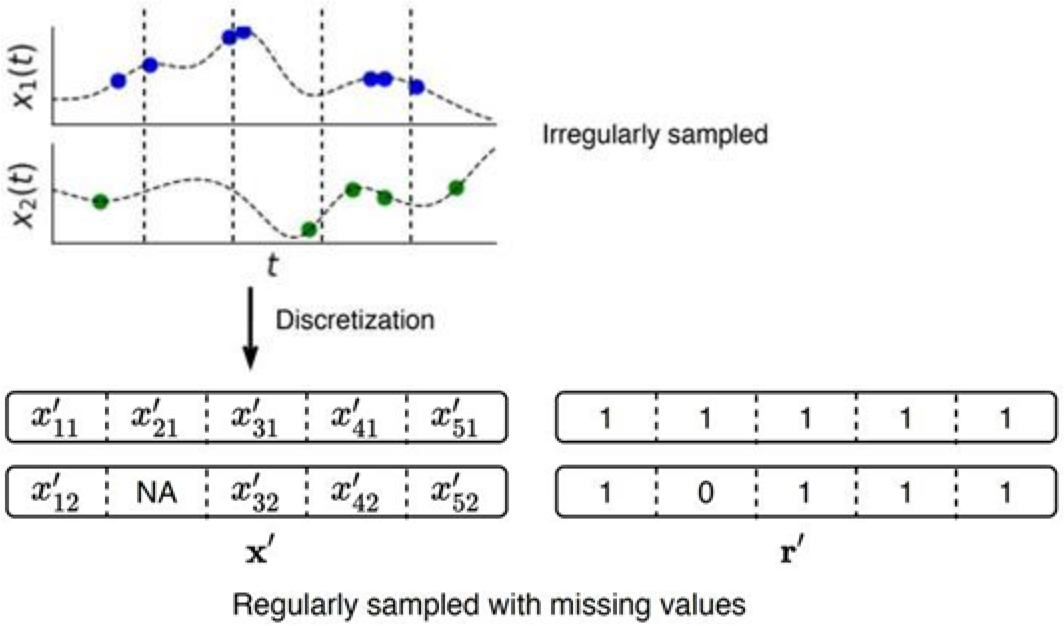
Current Approaches

Two approaches to learn from irregular time series:

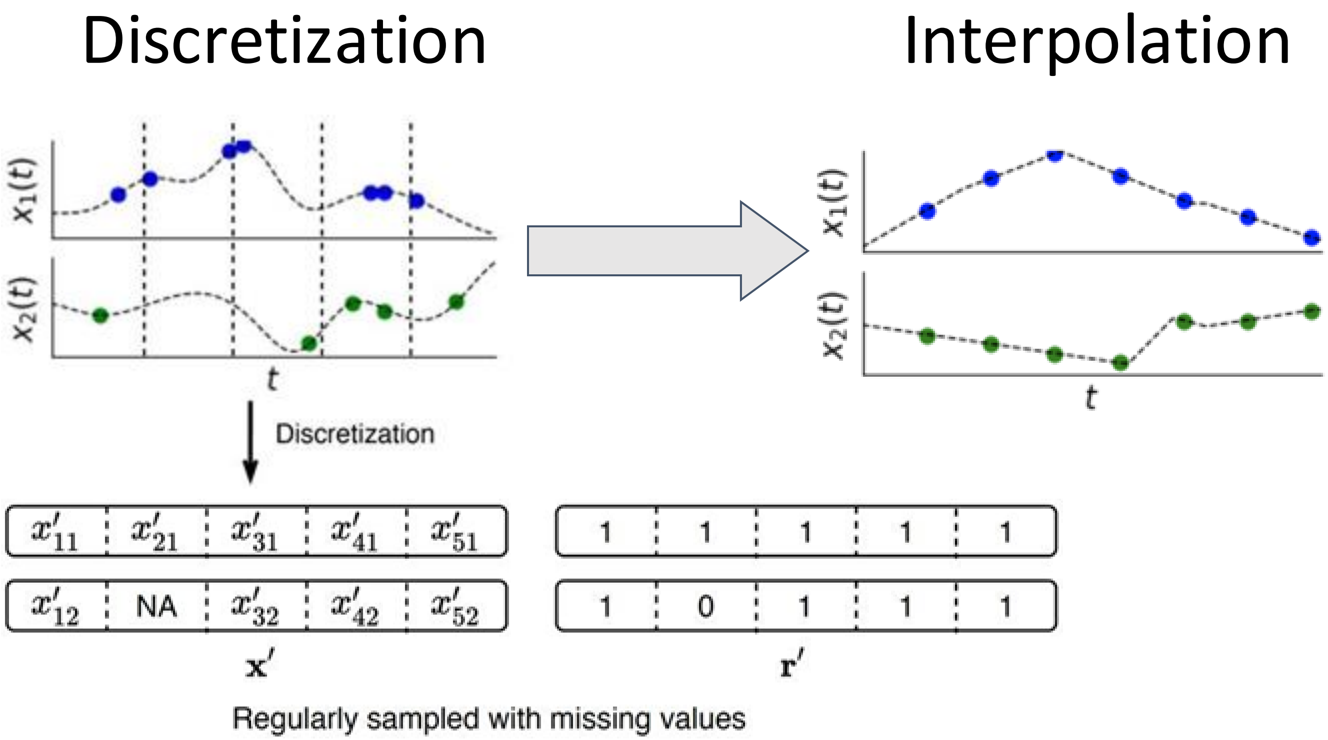
1. Transform irregular to regular:
    a. Discretization followed by interpolation
2. Directly model irregularity:
    a. ODE
    b. Set
    c. Attention

Approach 1

Discretization



Approach 1



- Narrow bins will create:
  - high missing %
  - exploding sequence
  - difficult imputation
- Wide bins will:
  - aggregate data
  - lose fine-grained details

Approach 1

- Irregularity may be useful to end task
- Consider what frequency of doctor visit may reveal about patient health
- Discretization followed by interpolation method, abstract away the irregularity, harming performance

Current Approaches

Two approaches to learn from irregular time series:

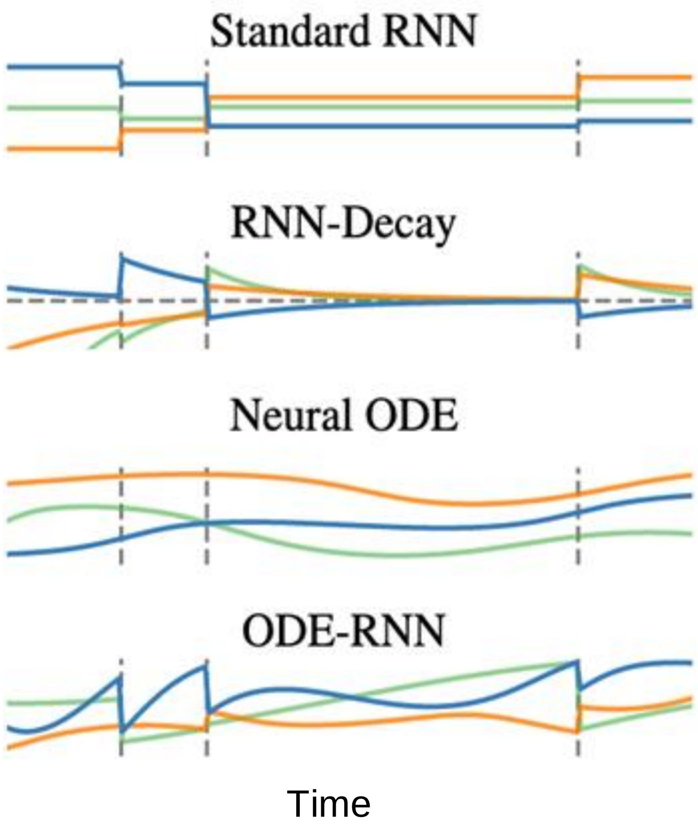
1. Transform irregular to regular:
    a. Discretization followed by interpolation
2. Directly model irregularity:
    a. ODE
    b. Set
    c. Attention

Current Approaches

Two approaches to learn from irregular time series:

1. Transform irregular to regular:
   a. Discretization followed by interpolation
2. Directly model irregularity:
   a. ODE
   b. Set
   c. Attention

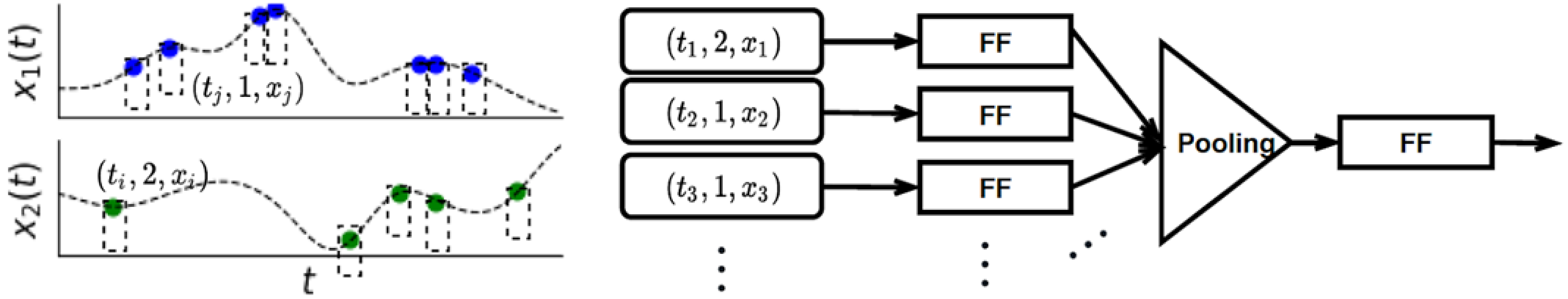
Approach 2: ODE



[1]

[1] Rubanova, Yulia, Ricky TQ Chen, and David K. Duvenaud. "Latent ordinary differential equations for irregularly-sampled time series." *Advances in neural information processing systems* 32 (2019).

Approach 2: Set



$$\mathbf{h} = g_\phi(\mathrm{pool}(\{f_\theta(t_{in}, d_{in}, x_{in}) | 1 \leq i \leq L_n\})$$ [1]

- Uses set-based representation

[1] Horn, Max, et al. "Set functions for time series." *International Conference on Machine Learning*. PMLR, 2020.

Approach 2: Attention

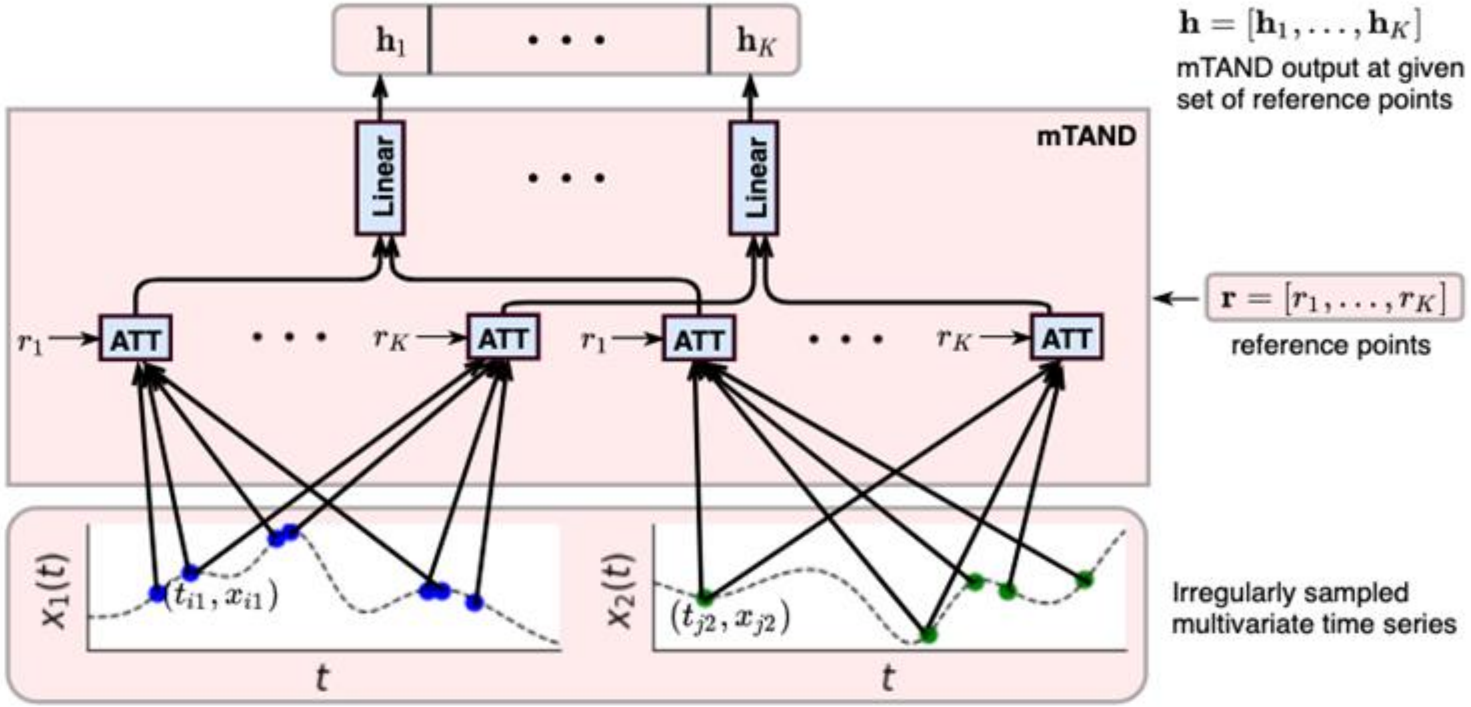


Fig: Time-Attention Module

- Reference time points, *r* -> queries
- Observed time points, *t* -> keys

[1] Shukla, Satya Narayan, and Benjamin M. Marlin. "Multi-time attention networks for irregularly sampled time series." *arXiv preprint arXiv:2101.10318* (2021).

Research Question

How to learn *self-supervised* representation from *unlabeled irregular* multivariate time series?

Research Question

How to learn *self-supervised* representation from *unlabeled* *irregular* multivariate time series?

Outline

- Motivation
- Current Approaches
- Methodology
- Experiments
- Conclusion

Methodology

Designed 2 self-supervised tasks for irregular time-series:

- Contrastive Learning
- Reconstruction

Methodology

Designed 2 self-supervised tasks for irregular time-series:

- Contrastive Learning
- Reconstruction

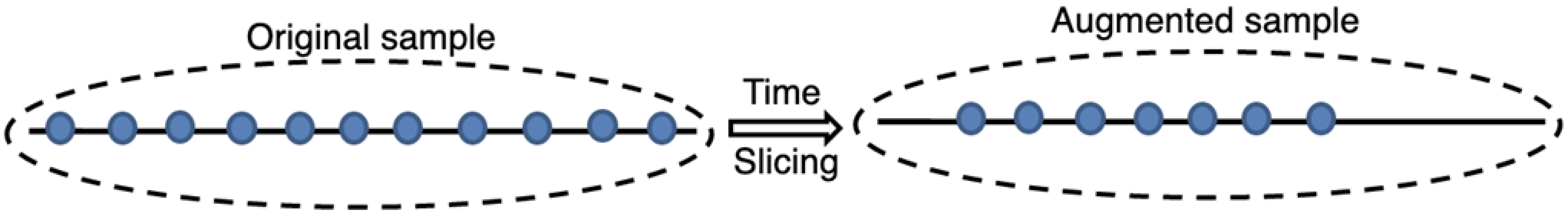
What is Contrastive Learning?

- Requires an anchor and positive which are *similar* and
- An in-batch negative *different* from the anchor and positive
- Pulls anchor and positive closer, while pushing away the negative

$$\mathcal{L}_{CL} = -\log \frac{\exp(\tilde{\mathrm{X}}_i \tilde{\mathrm{X}}_j / \tau)}{\sum_{k=1}^{2B} \exp(\tilde{\mathrm{X}}_i \tilde{\mathrm{X}}_k / \tau)}$$

Contrastive Learning for Time Series

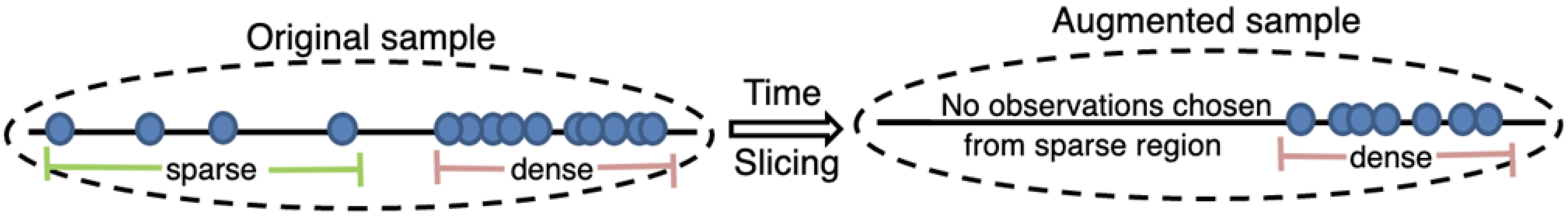
- Time Slicing for *regular* time series



  - A continuous subsequence forms the anchor
  - A succeeding or preceeding subsequence forms the positive
  - Original sample has constant sampling density
  - Anchor and positive has same density to that of the original sample
  - Resulting in representative anchor and positive

Contrastive Learning for Time Series

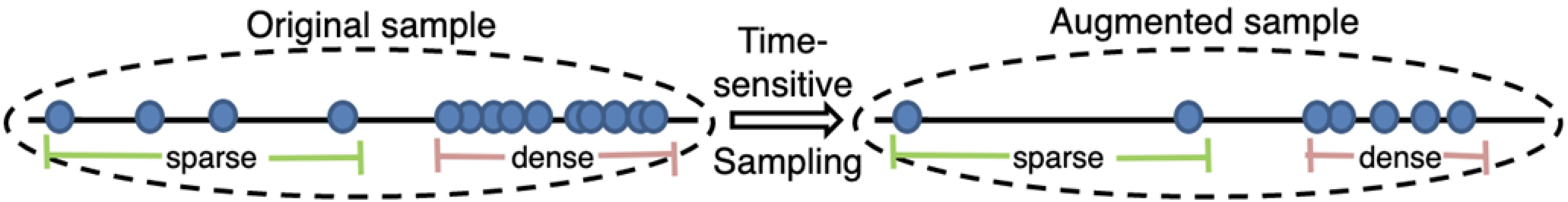
- Time Slicing for *irregular* time series



  - Original sample may consist of both dense *and* sparse regions
  - Anchor and positive may have densities different to the original sample
  - Resulting in unrepresentative samples

Time Contrastive Learning (TimeCL)

- Instead, we form a subsequence by *randomly* sampling *separately from dense and sparse regions* -> stratified sampling



  - Sample randomly instead of a continuous subsequence
  - Sample separately from both dense *and* sparse regions
  - Resulting in more representative samples

Methodology

Designed 2 self-supervised tasks for irregular time-series:

- Contrastive Learning -> Time Contrastive Learning (TimeCL)
- Reconstruction

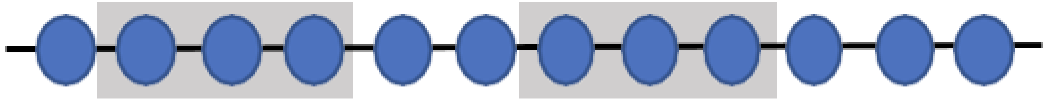
What is Reconstruction?

- Randomly mask data and reconstruct them from unmasked data
- Masked out data used as supervision/target

$$\mathcal{L}_{Reco} = \left\| \mathrm{M}_V \odot (\tilde{\mathrm{X}}_U - \mathrm{X}_V) \right\|_2^2$$

Reconstruction for Time Series

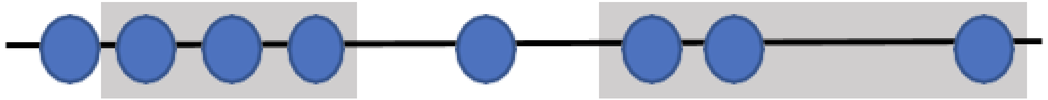
- Constant length masking for *regular* time series



  - will mask constant *duration* of points throughout sample
  - because sampling density is constant.
  - Similar difficulty of reconstruction throughout sample.

Reconstruction for Time Series

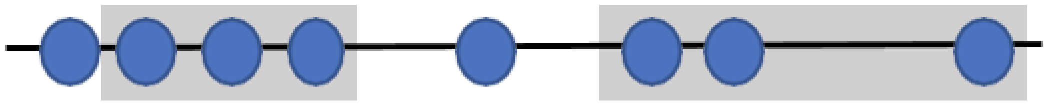
- Constant length masking for ir*regular* time series



  - will mask over *different* duration
  - because sampling density varies throughout sample.
  - Reconstruction is easy for dense regions but hard for sparse regions.

Reconstruction for Time Series

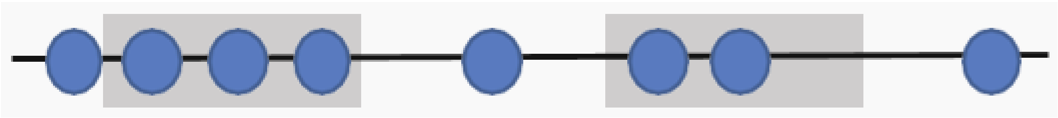
- Constant length masking for ir*regular* time series



  - will mask over *different* duration
  - because sampling density varies throughout sample.
  - Reconstruction is easy for dense regions but hard for sparse regions.
  - Need to adjust mask length based on signal’s local density
  - Mask more points from dense regions than from sparse regions
  - Reconstruction is tractable across regions of different sampling density.

Time Reconstruction (TimeReco)

- We propose constant *time* masking



  - masks constant *duration*, not *length*, of data throughout sample
  - Masks few points in sparse regions but more points in dense regions.
  - Adjusts the mask length based on signal’s local density.
  - Balances difficulty of reconstruction among regions of different sampling density

Methodology

Designed 2 self-supervised tasks for irregular time-series:

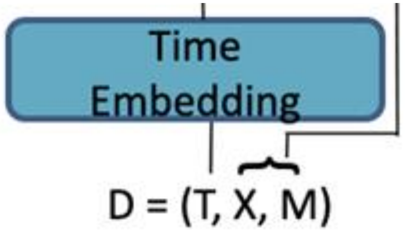
- Contrastive Learning -> Time Contrastive Learning (TimeCL)
- Reconstruction -> Time Reconstruction (TimeReco)

$$\mathcal{L} = \eta\mathcal{L}_{CL} + (1-\eta)\mathcal{L}_{Reco}$$

Model

- Time-Embedding Layer (Shukla ‘21)

$$\phi h(\mathrm{T})[i] = \begin{cases} \omega_{0h}.\mathrm{T} + \alpha_{0h}, \text{if } i = 0 \\ sin(\omega_{ih}.\mathrm{T} + \alpha_{ih}), \text{if } 0 < i < d_r \end{cases}$$



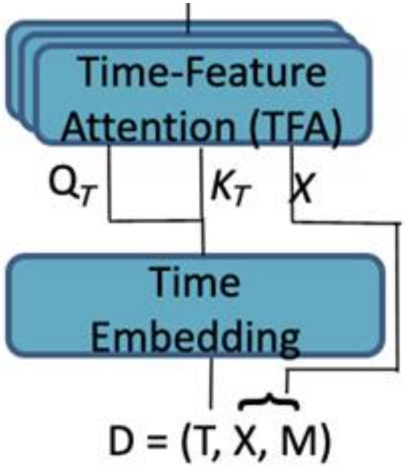
Model

- Time-Embedding Layer (Shukla ‘21)

$$\phi h(\mathrm{T})[i] = \begin{cases} \omega_{0h}.\mathrm{T} + \alpha_{0h}, \text{if } i = 0 \\ sin(\omega_{ih}.\mathrm{T} + \alpha_{ih}), \text{if } 0 < i < d_r \end{cases}$$

- Time-Feature Attention (TFA) (Shukla ‘21)

$$\mathrm{TFA}(\mathrm{Q}_T, \mathrm{K}_T, \mathrm{M}, \mathrm{X}) = (\mathrm{M} \odot \mathrm{A}_T)\mathrm{X},$$
$$\mathrm{A}_T = \mathrm{softmax}(\mathrm{Q}_T \mathrm{K}_T / d_r)$$



Model

- Time-Embedding Layer (Shukla ‘21)

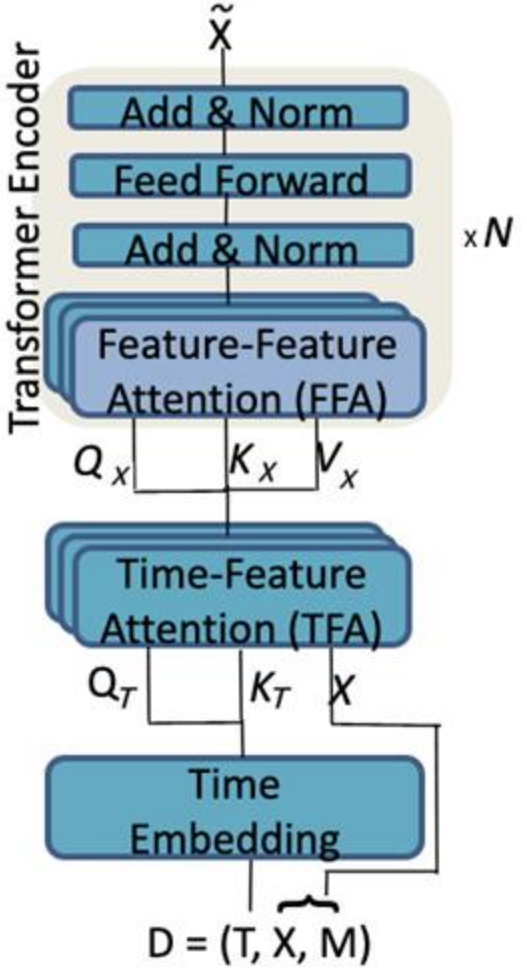
$$\phi h(\mathrm{T})[i] = \begin{cases} \omega_{0h}.\mathrm{T} + \alpha_{0h}, \text{if } i = 0 \\ sin(\omega_{ih}.\mathrm{T} + \alpha_{ih}), \text{if } 0 < i < d_r \end{cases}$$

- Time-Feature Attention (TFA) (Shukla ‘21)

$$\mathrm{TFA}(\mathrm{Q}_T, \mathrm{K}_T, \mathrm{M}, \mathrm{X}) = (\mathrm{M} \odot \mathrm{A}_T)\mathrm{X},$$
$$\mathrm{A}_T = \mathrm{softmax}(\mathrm{Q}_T \mathrm{K}_T / d_r)$$

- Feature-Feature Attention (FFA) (Vaswani

$$\mathrm{FFA}(\mathrm{Q}_X, \mathrm{K}_X, \mathrm{V}_X, \mathrm{M}) = (\mathrm{M} \odot \mathrm{A})\mathrm{V}_X,$$
$$\mathrm{A} = \mathrm{softmax}(\mathrm{Q}_X \mathrm{K}_X / d_r)$$



PrimeNet

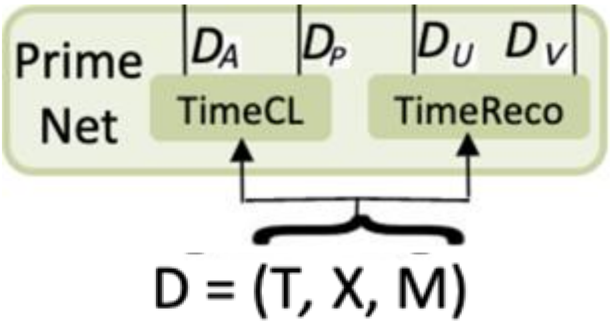
D = (T, X, M)

Pre-Training

PrimeNet

Task, T

Data Augmentation
for Task, T

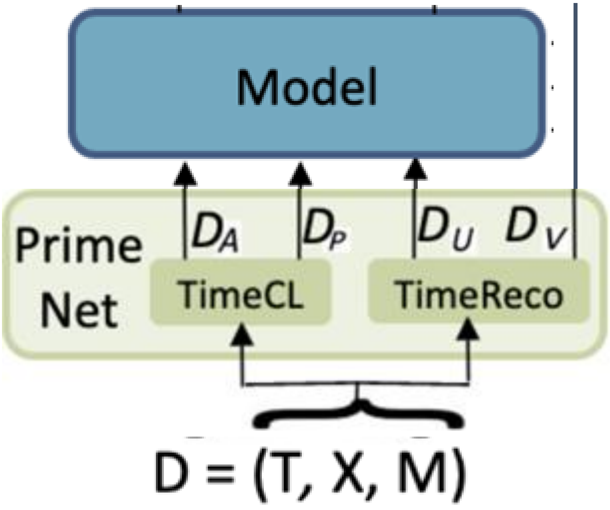


Pre-Training

PrimeNet

Task, T

Data Augmentation
for Task, T

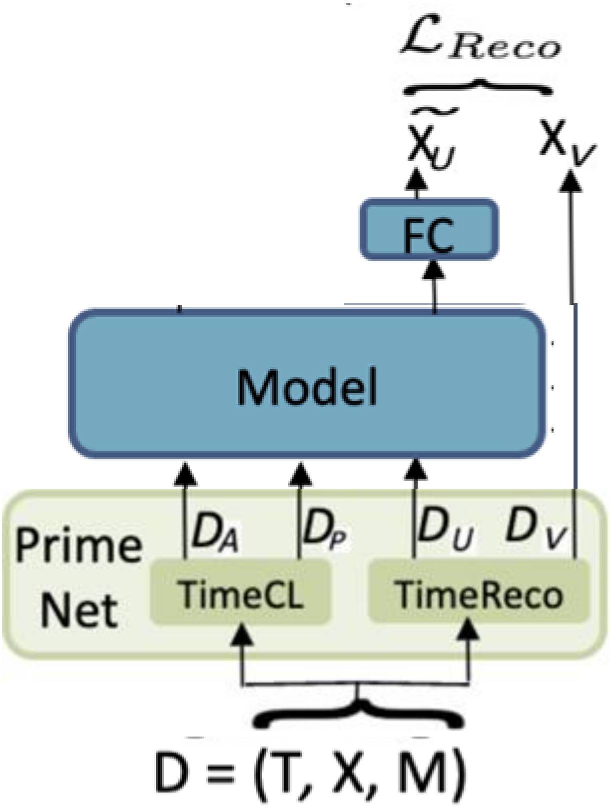


Pre-Training

PrimeNet

Task, T

Data Augmentation
for Task, T

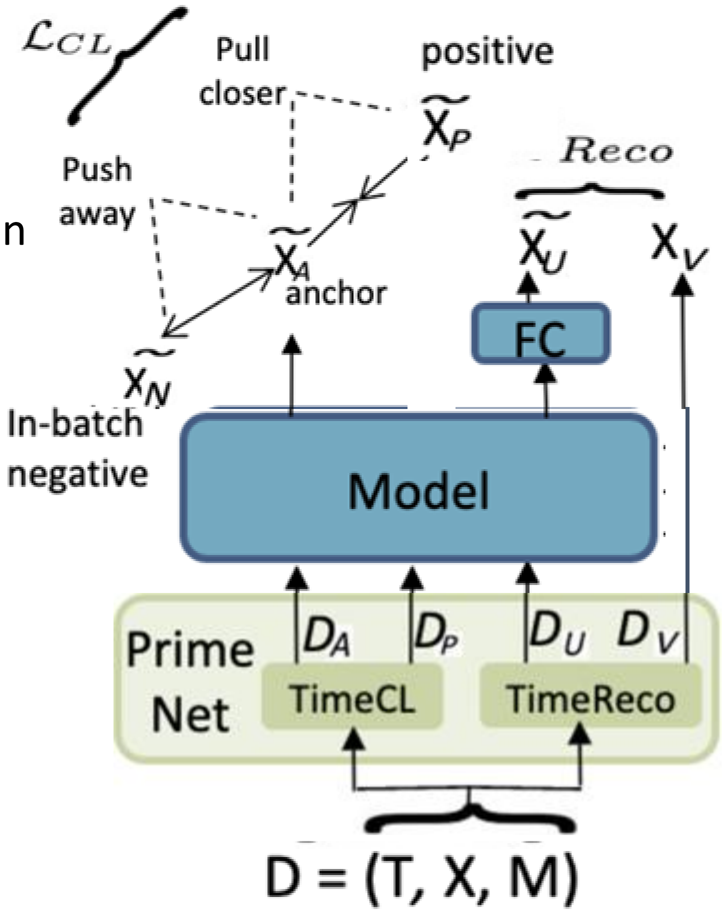


Pre-Training

PrimeNet

Task, T

Data Augmentation for Task, T

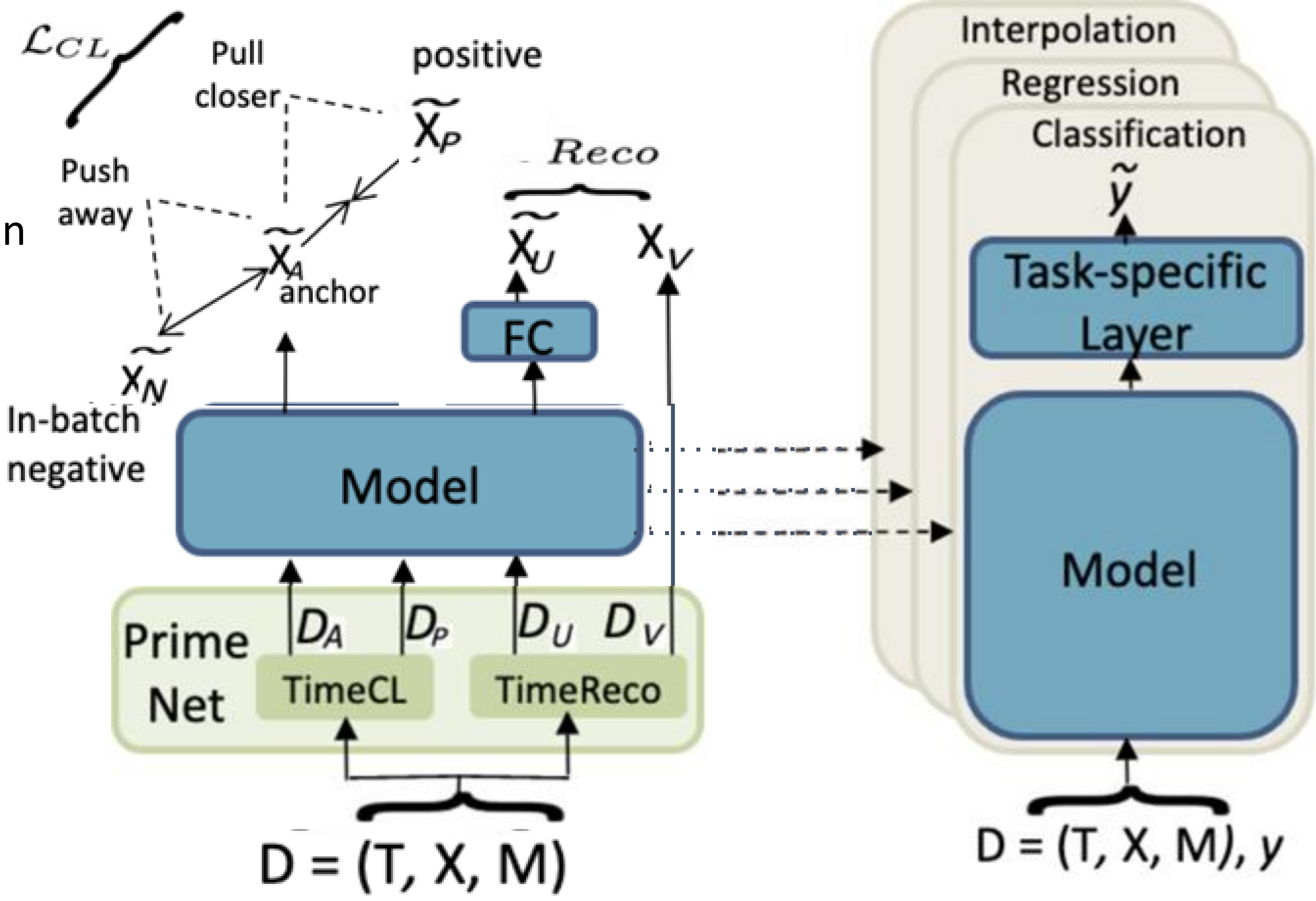


Pre-Training

PrimeNet

Task, T

Data Augmentation for Task, T



Pre-Training Finetuning

Outline

- Motivation
- Current Approaches
- Methodology
- **Experiments**
- Conclusion

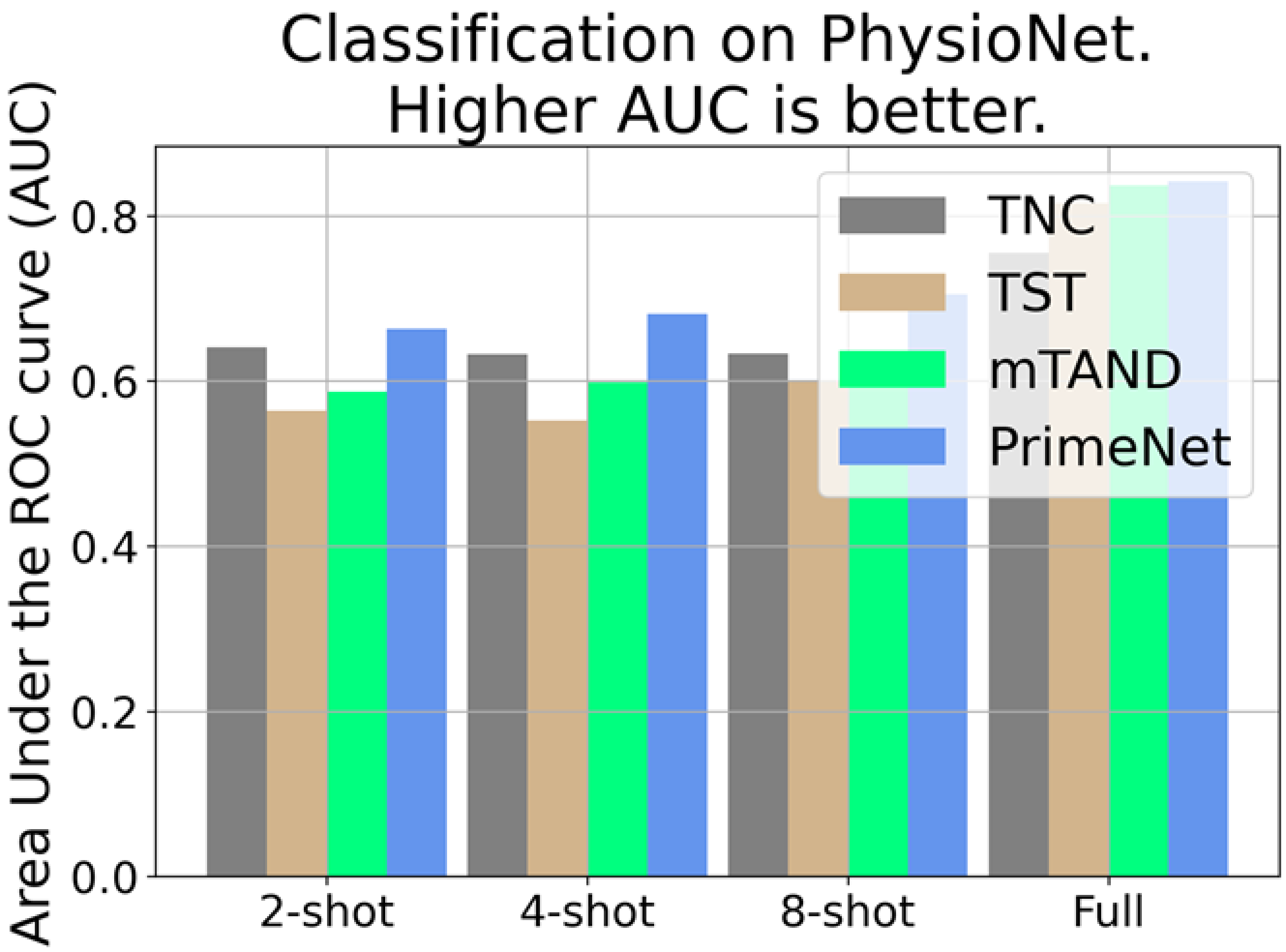
Experiments - Datasets

- Used 4 naturally occurring irregular data from Healthcare and IoT for classification, interpolation, and regression tasks
- 48 hours of physiological data after patient admission to ICU in *PhysioNet* and *MIMIC-III* datasets
- 3D position of human body from *Activity* dataset
- Weather data from *Appliances Energy* dataset

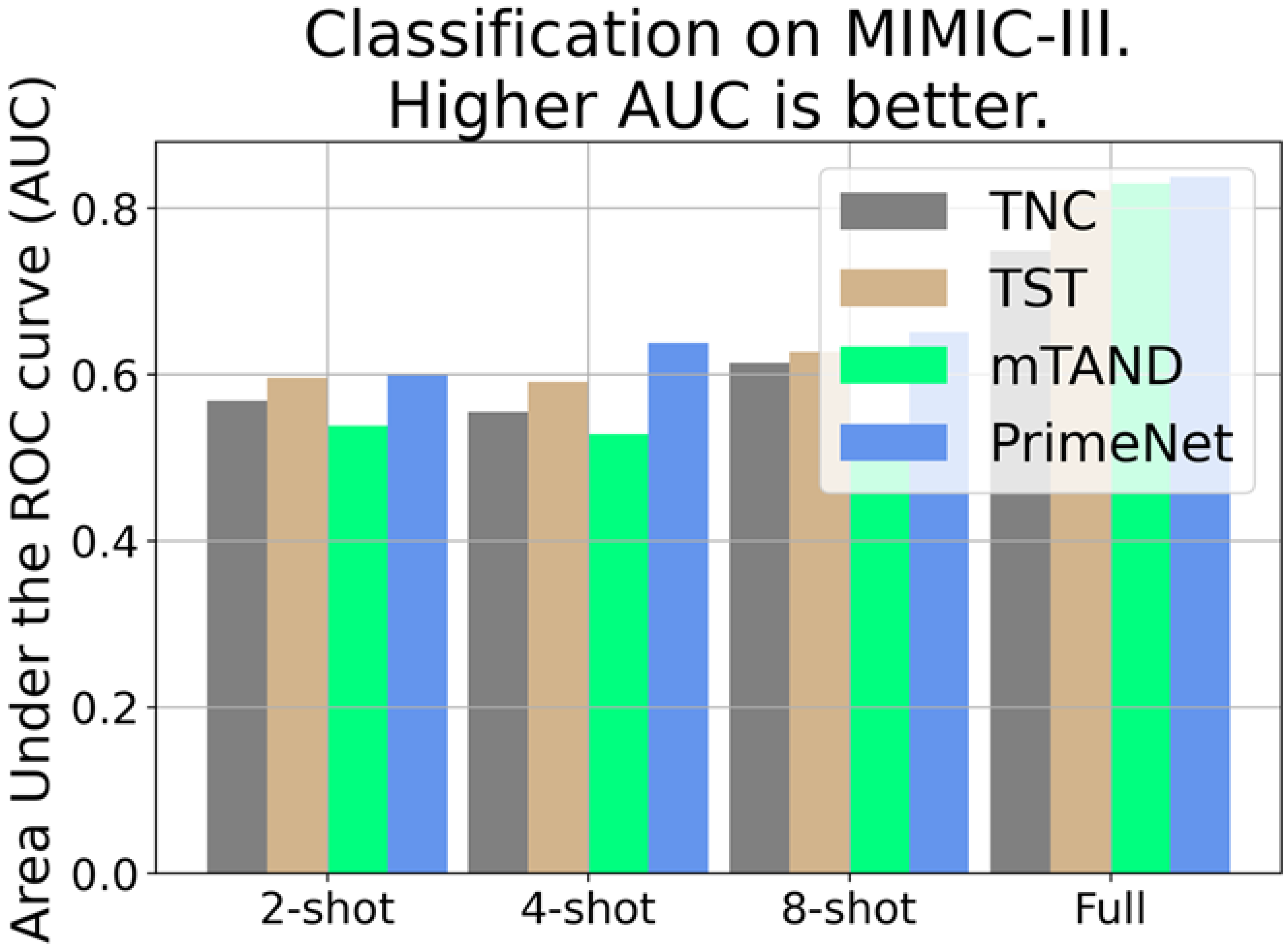
Experiments - Baselines

- Self-supervised regular time series methods
  - TS2Vec (Yue ‘21), TNC (Tonekaboni ‘21), TST (Zerveas ‘21)
- Fully- or semi- supervised irregular time series methods
  - GRU-Mean (Che ‘18), P-LSTM (Neil ‘16), RNN-VAE (Chen ‘18), ODE-RNN (Rubanova ‘19), L-ODE (Rubanova ‘19), mTAND (Shukla ‘21)

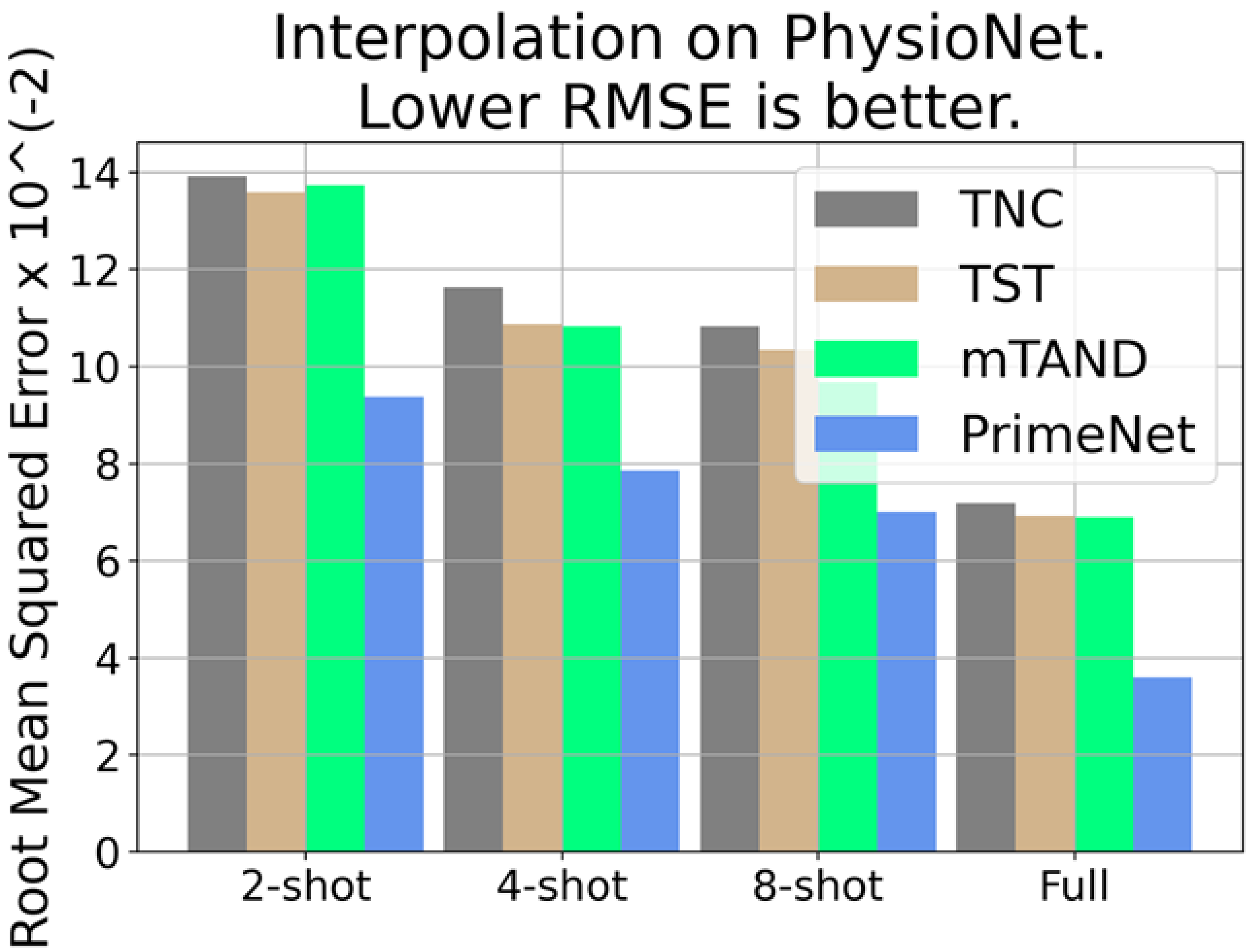
Experiments - Results



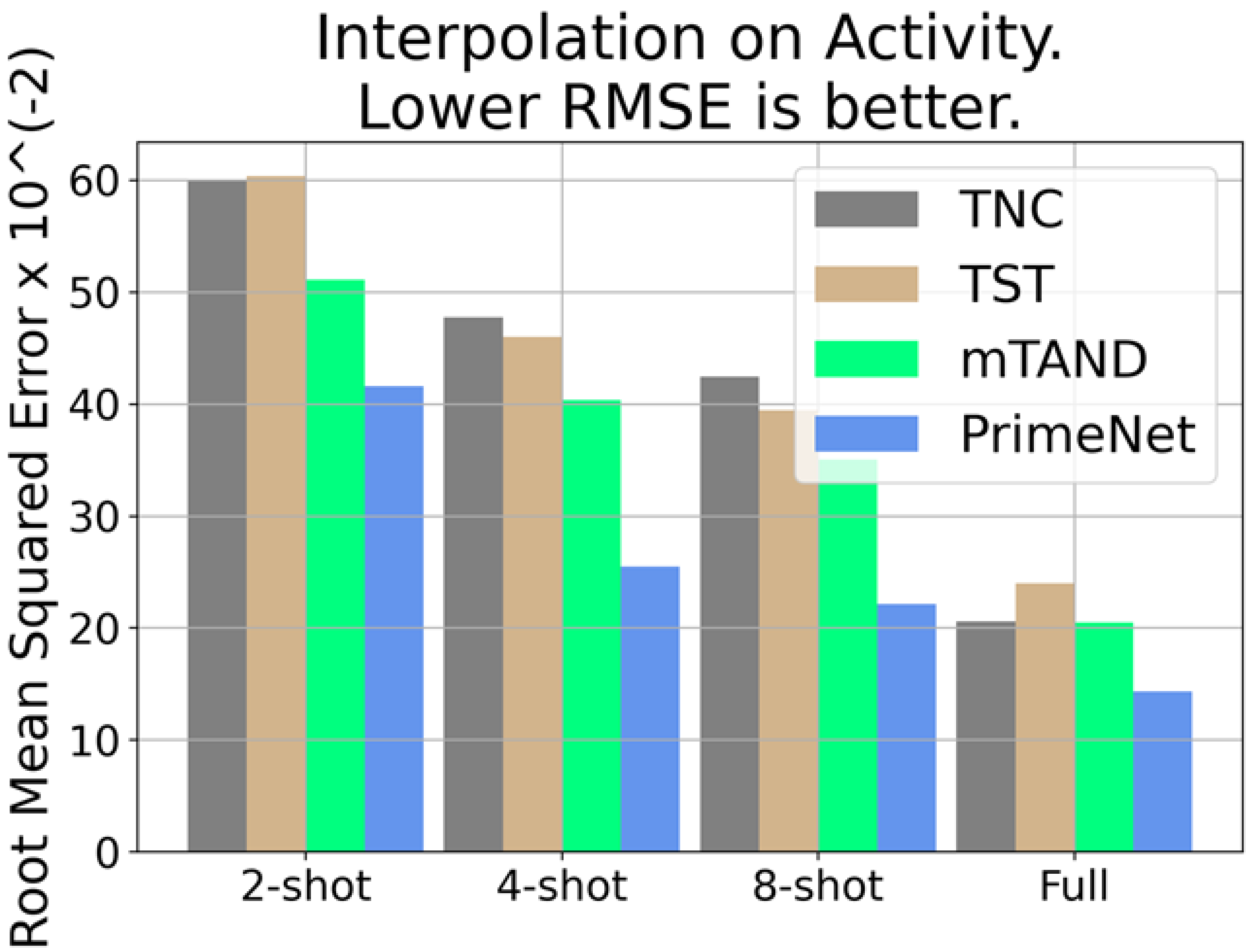
Experiments - Results



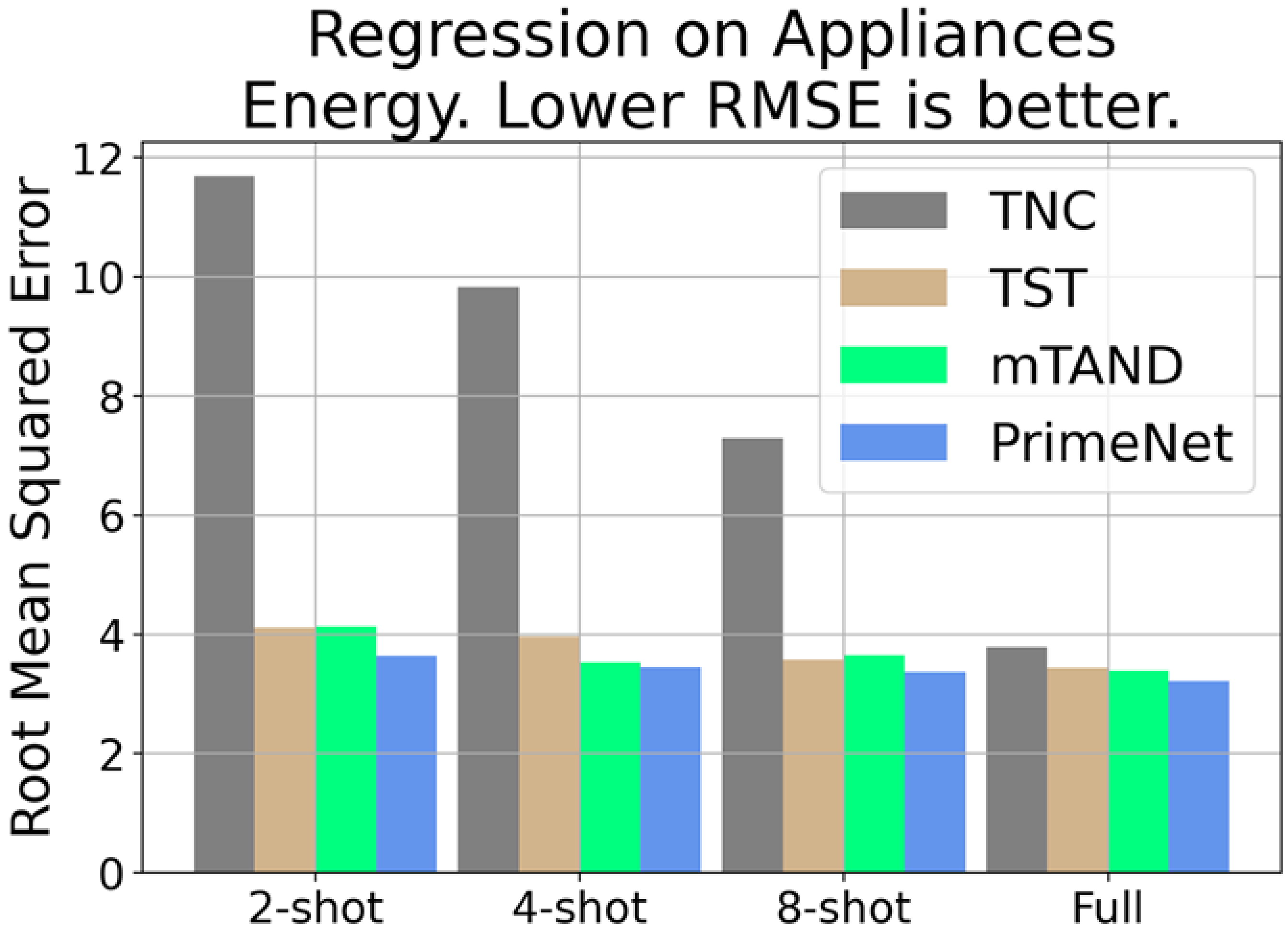
Experiments - Results



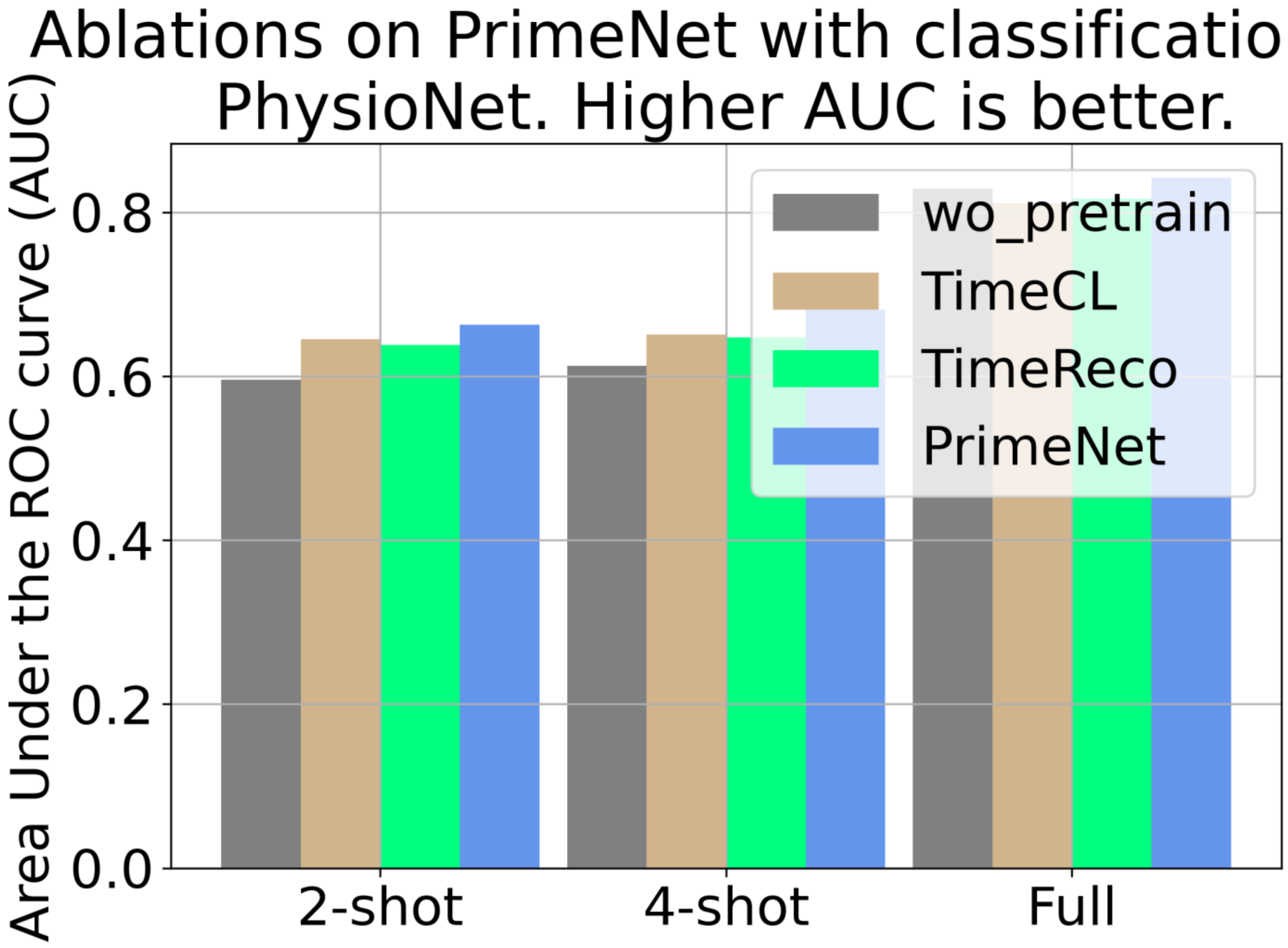
Experiments - Results



Experiments - Results



Experiments - Ablations



Outline

- Motivation
- Current Approaches
- Methodology
- Experiments
- Conclusion

Conclusion

- First to propose a pre-trained model for irregular time series
- Designed 2 time-sensitive self-supervision tasks:
  - TimeCL: preserve sampling density variation in anchor and positive
  - TimeReco: adjusts mask length to make reconstruction tractable
- Pre-trained model fine-tuned on labeled data
- Outperformed baselines on Healthcare and IoT data for classification, interpolation and regression, under full and few-shot data
- Ablation highlights the importance of both tasks and their design choices

Questions?

- Contact: Ranak Roy Chowdhury (https://ranakroychowdhury.github.io/)
- Implementation: https://github.com/ranakroychowdhury/PrimeNet