Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding

Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, Dan Pei

清華大學
Tsinghua University

Virtual Conference
KDD2021
August 14th – 18th

Outline

Background

Algorithm

Evaluation

Conclusion

Outline

Background

Algorithm

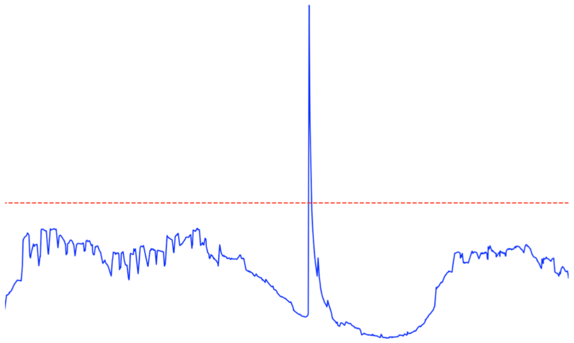
Evaluation

Conclusion

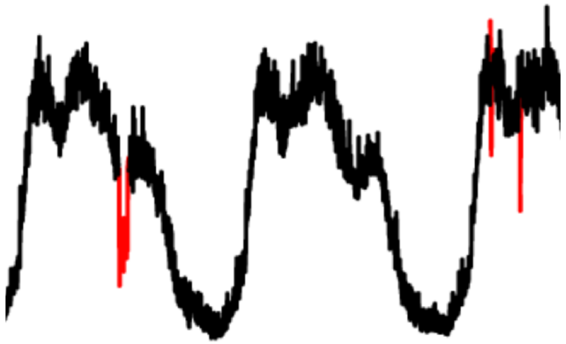
Time Series Anomaly Detection

- Anomaly Detection: images, graphs, texts, time series, etc.

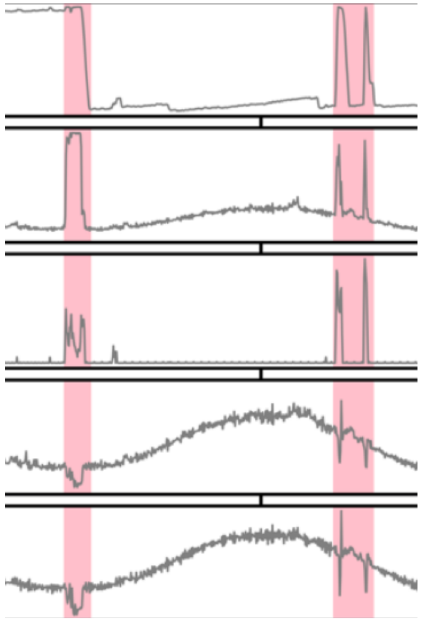
- Monitor the status of entities (e.g., systems, services) in the domain of manufacturing industry and Information Technology (IT) systems.



Static Threshold
(Domain Expert)

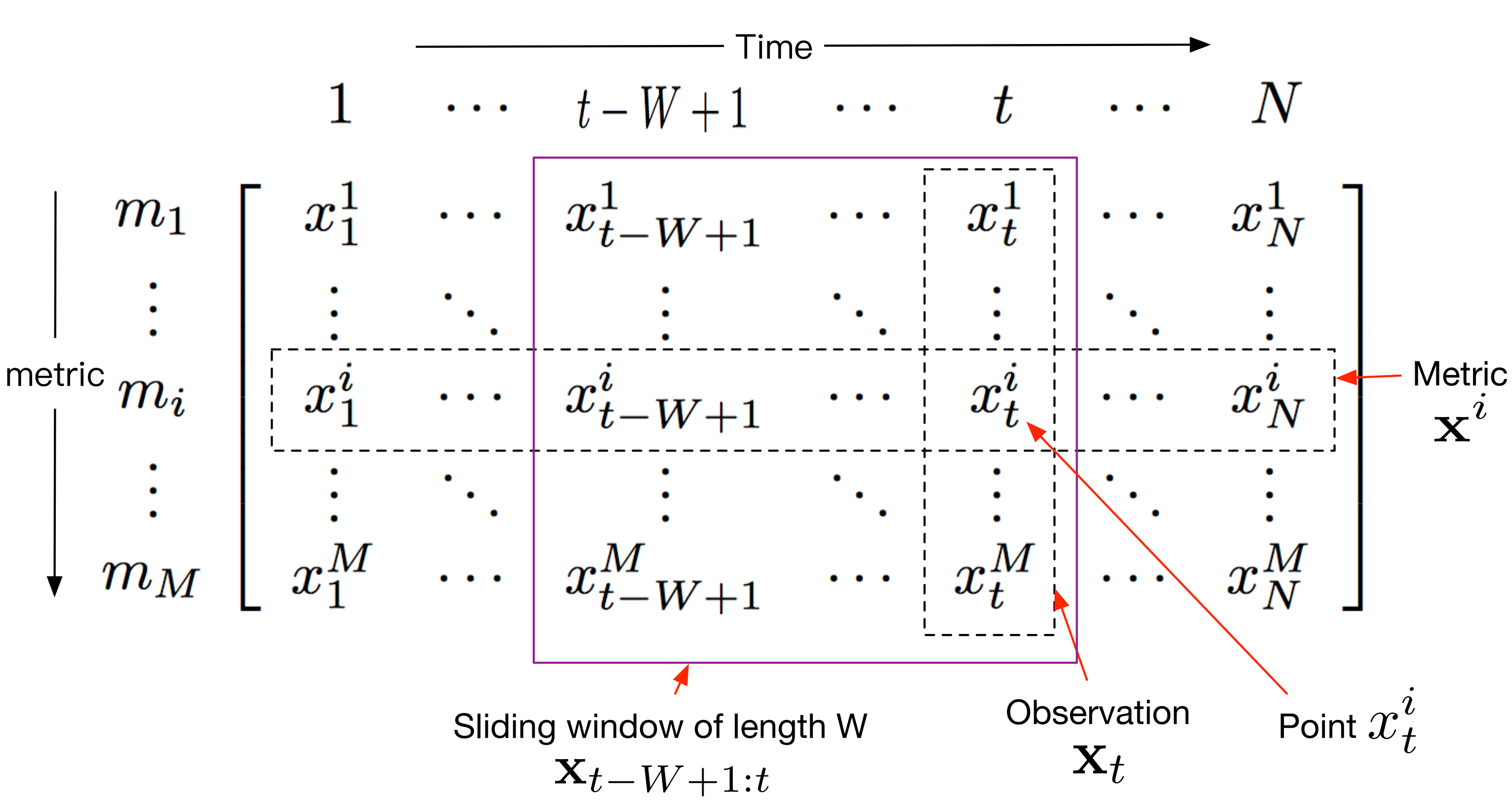


Univariate Time Series
(Single System Metric)
[WWW'18]



Multivariate Time Series
(Entity Metrics)
[KDD'19]

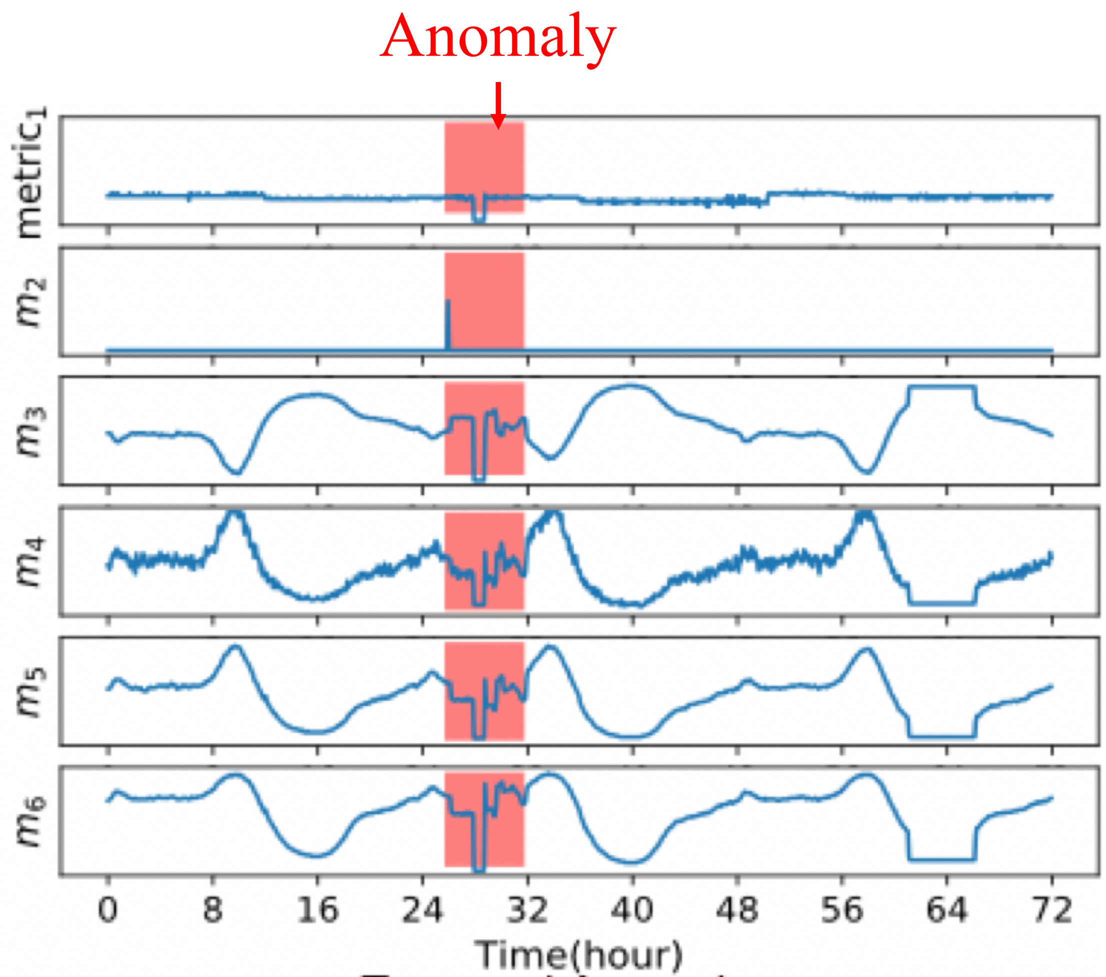
Multivariate Time Series (MTS)



Each metric describes a different part or attribute of a complex entity (i.e., MTS).

e.g., CPU utilization, TCP active opens, memory utilization, packets transmitted per second, …, in a Web Application Server.

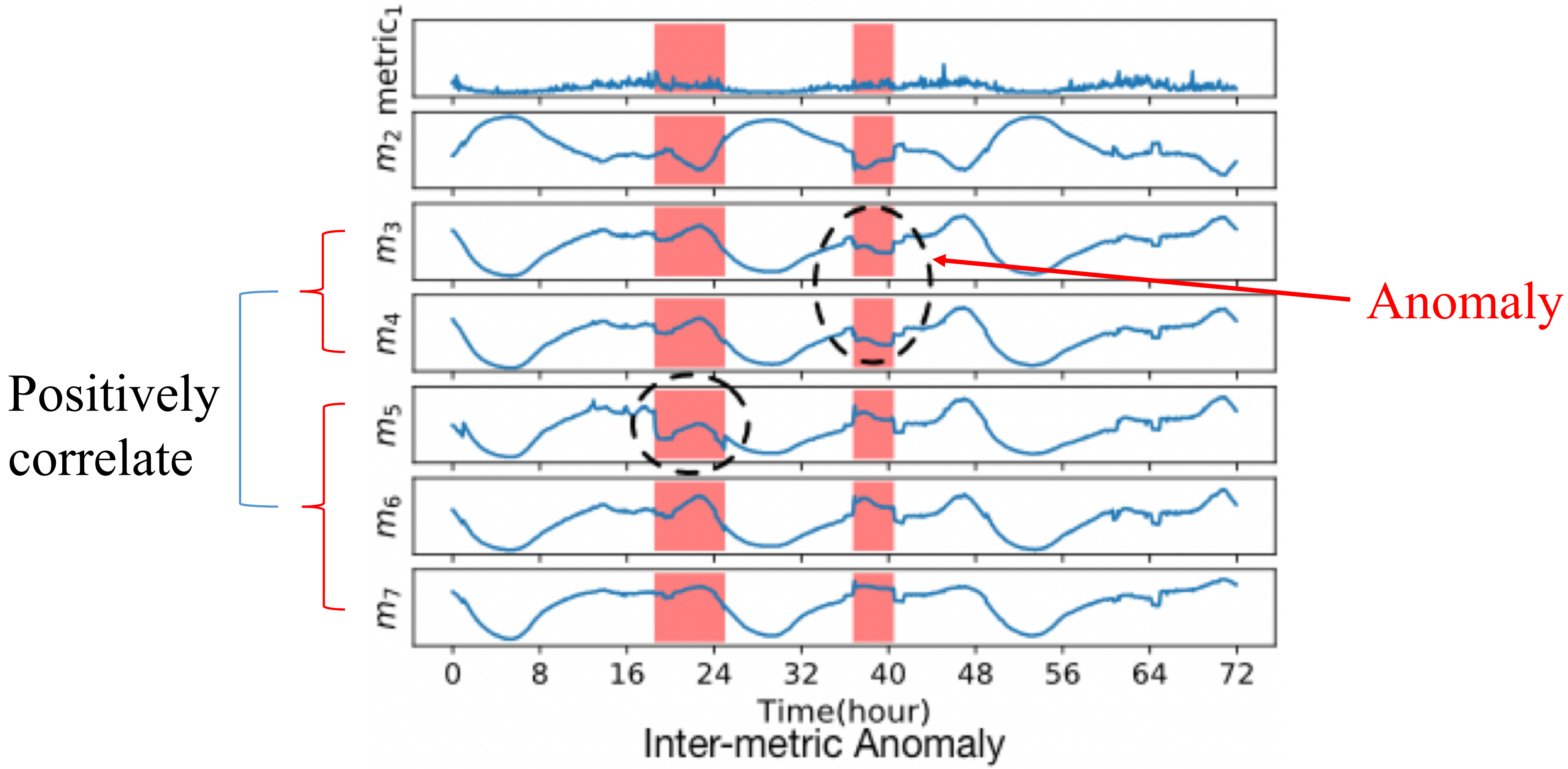
Anomaly Types in MTS



Temporal Anomaly

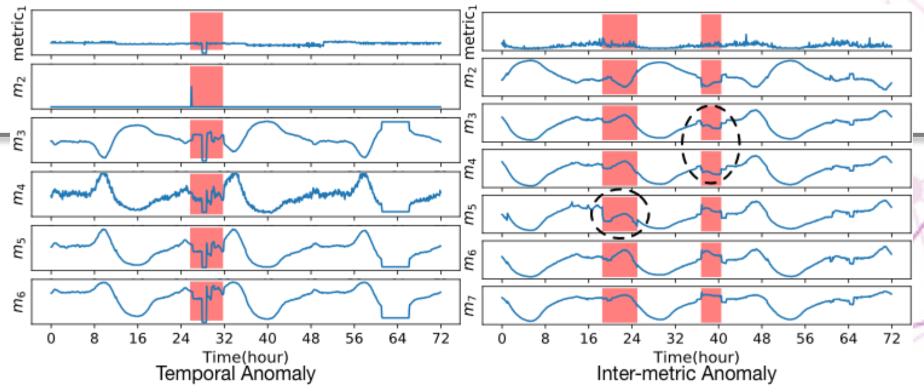
Inherent patterns like periodicity within each metric.
e.g., system-level failure, rebooting

Anomaly Types in MTS



Relationships among all metrics of an entity at a time period.
e.g., local fluctuations in parts of the system

Challenges



- How to precisely detect different kinds of anomalies in multivariate time series?
  - How to *learn and fuse inter-metric and temporal representations* to capture the normal patterns of MTS?
  - How to prevent the model from *overfitting to potential anomalies* in real-world data?

- How to interpret each detected anomaly (i.e., find a group of most anomalous metrics for each entity anomaly)?
  - How to find the *normal patterns that each anomalous metric should have followed*?

Outline

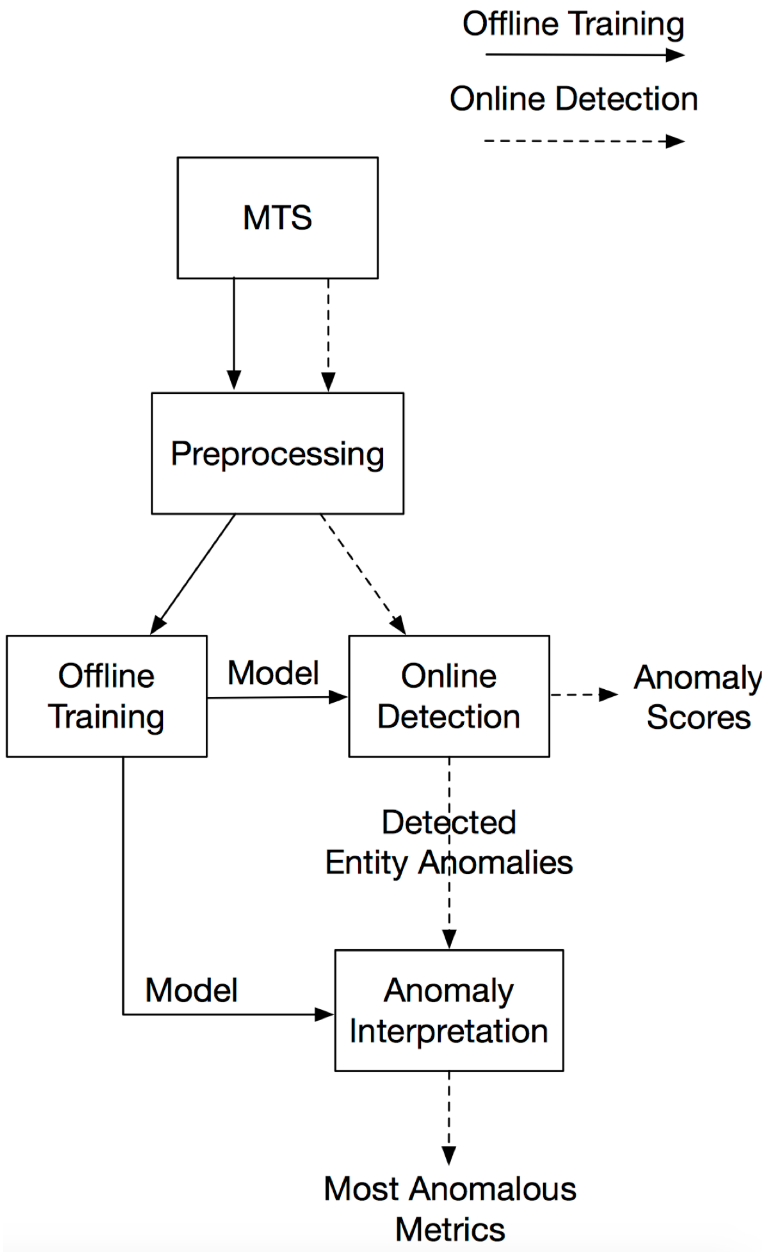
Background

Algorithm

Evaluation

Conclusion

Overview of InterFusion

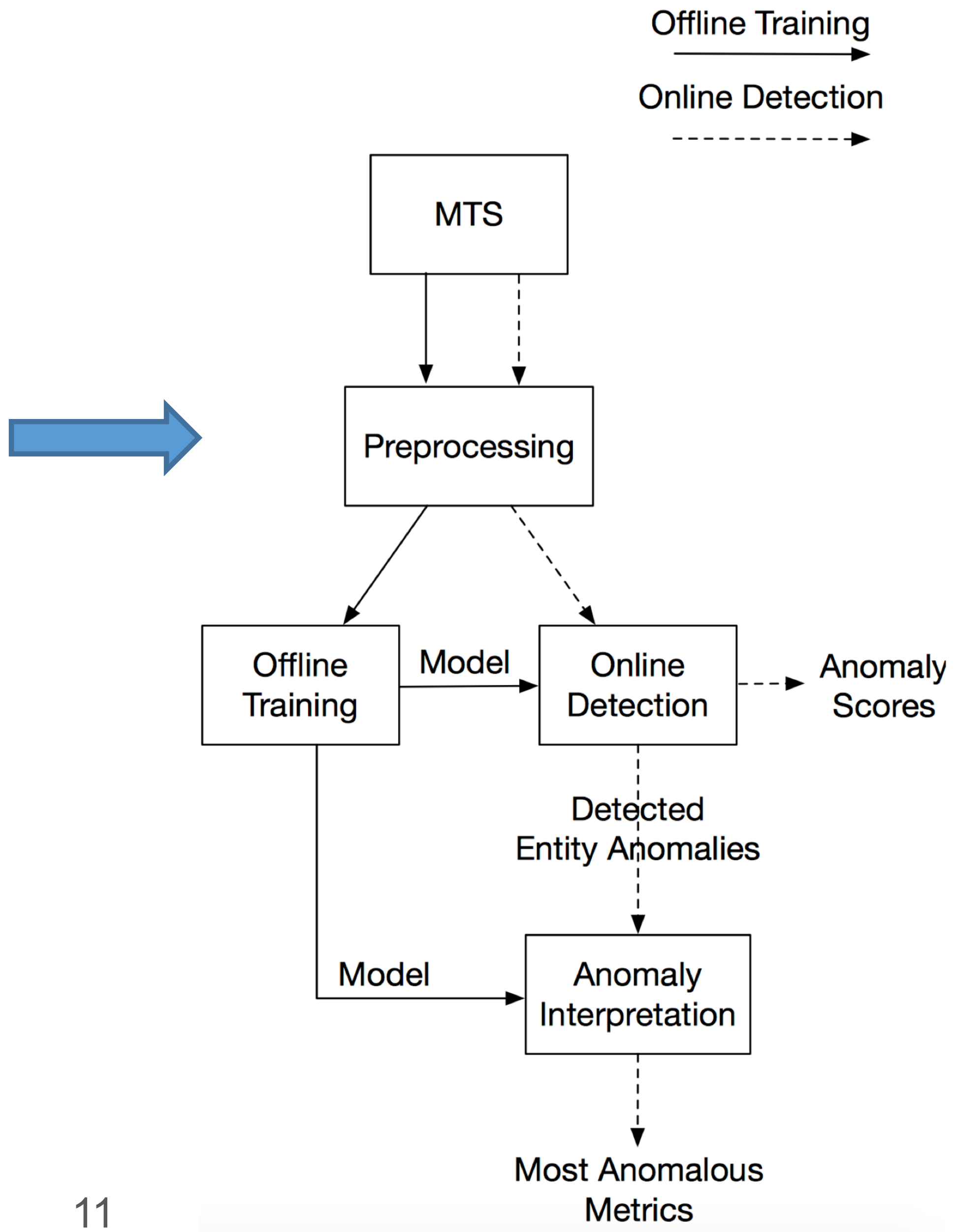


Preprocessing: data normalization & split MTS data into sliding windows

Training & Detection: Unsupervised model training, detect MTS anomalies

Interpretation: Find most anomalous metrics (to accelerate troubleshooting, to explain detection results to users)

Overview of InterFusion

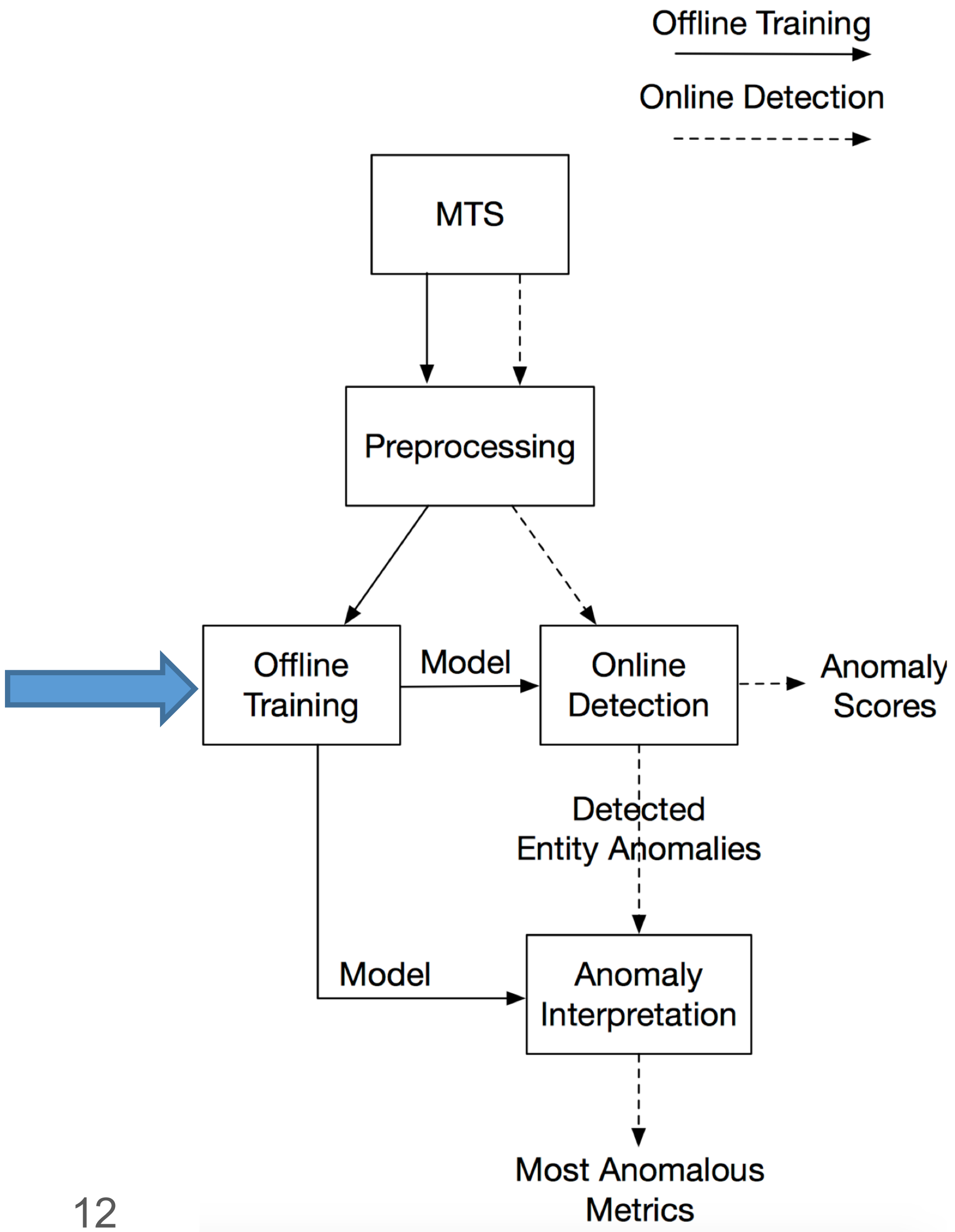


Preprocessing: data normalization & split MTS data into sliding windows

Training & Detection: Unsupervised model training, detect MTS anomalies

Interpretation: Find most anomalous metrics (to accelerate troubleshooting, to explain detection results to users)

Overview of InterFusion

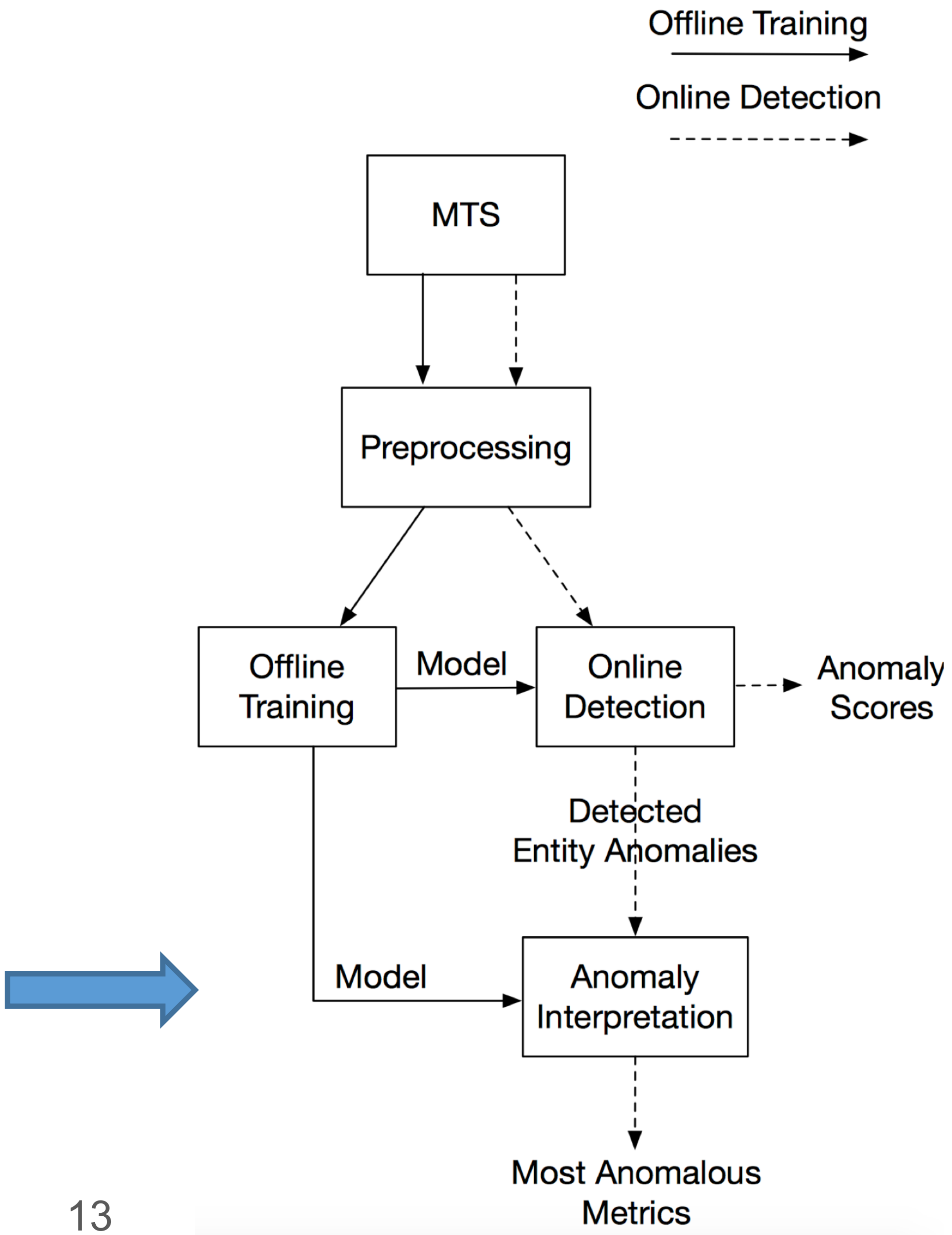


Preprocessing: data normalization & split MTS data into sliding windows

Training & Detection: Unsupervised model training, detect MTS anomalies

Interpretation: Find most anomalous metrics (to accelerate troubleshooting, to explain detection results to users)

Overview of InterFusion

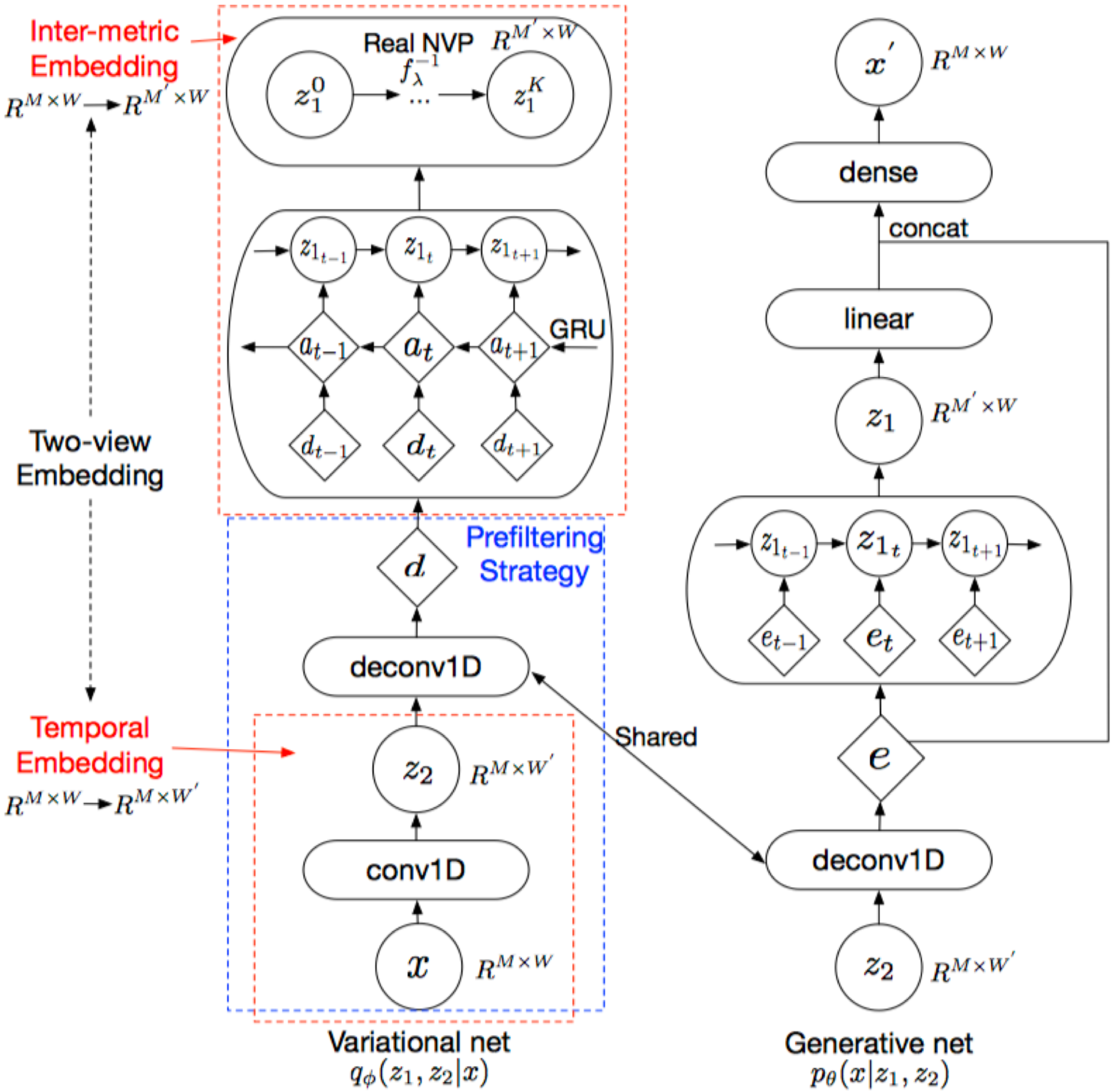


Preprocessing: data normalization & split MTS data into sliding windows

Training & Detection: Unsupervised model training, detect MTS anomalies

Interpretation: Find most anomalous metrics (to accelerate troubleshooting, to explain detection results to users)

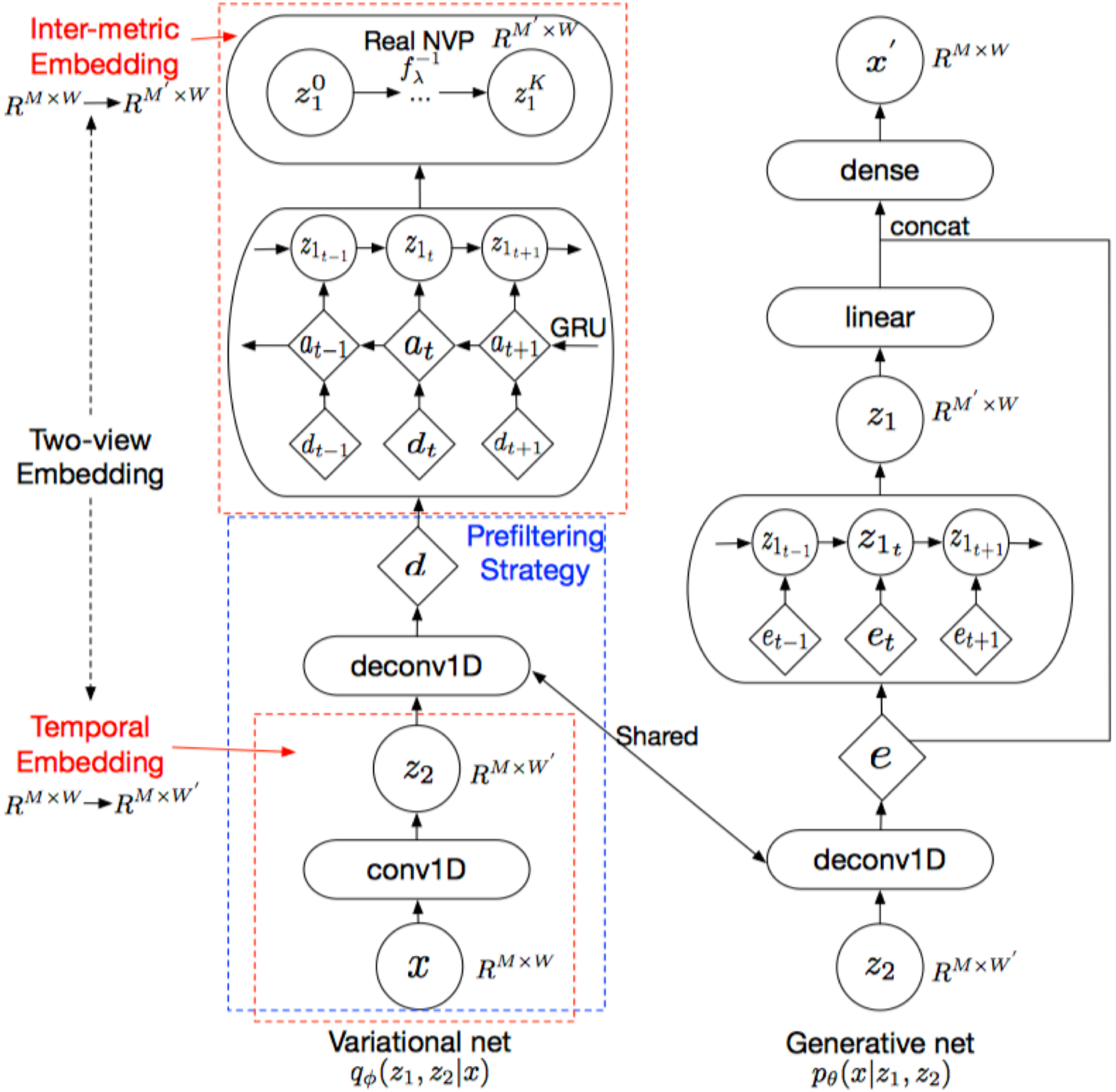
Network Architecture of InterFusion



Core idea: model the MTS using HVAE with jointly trained *hierarchical stochastic* latent variables, each of which explicitly learns low-dimensional *inter-metric or temporal embeddings*.

Challenges	**Designs**
Learn and fuse inter-metric and temporal embeddings	Hierarchical Structure
	Two-View Embedding
Prevent overfitting to anomalies	Prefiltering Strategy

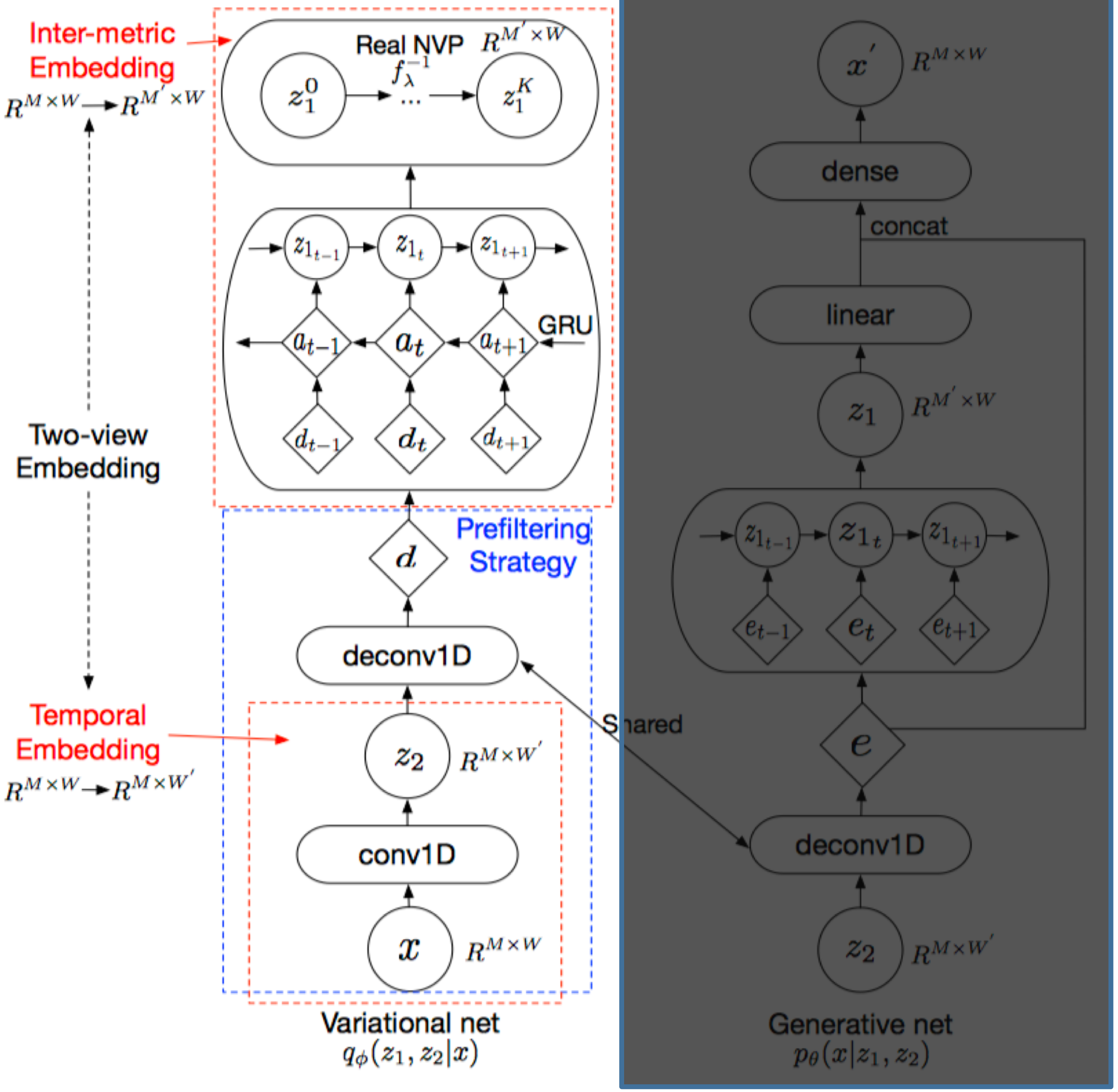
Hierarchical Structure



$$p_\theta(x, z_1, z_2) = p_\theta(x|z_1, z_2)p_\theta(z_1|z_2)p_\theta(z_2)$$

Hierarchical latent variables,
rather than learning independently

Two-view Embedding

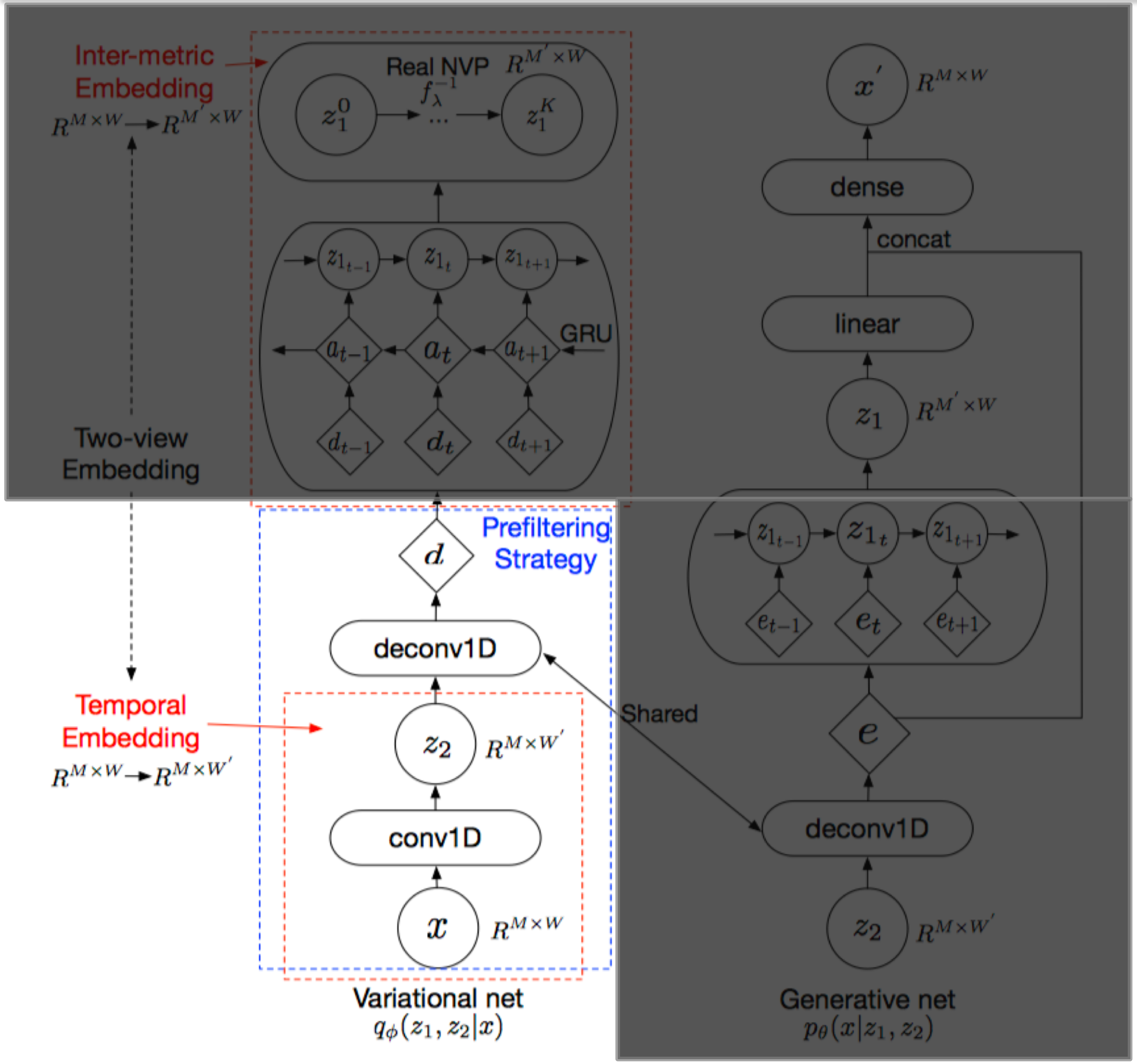


Two-view Embedding with auxiliary "*reconstructed input*" $\mathbf{d} \in R^{M\times W}$

Compress the MTS along **time** and **metric** dimensions to obtain temporal and inter-metric embeddings.

help *InterFusion* learn better inter-metric embeddings that are *aware of the learned temporal information*, while preserving the *time consistency* inside the inter-metric embeddings..

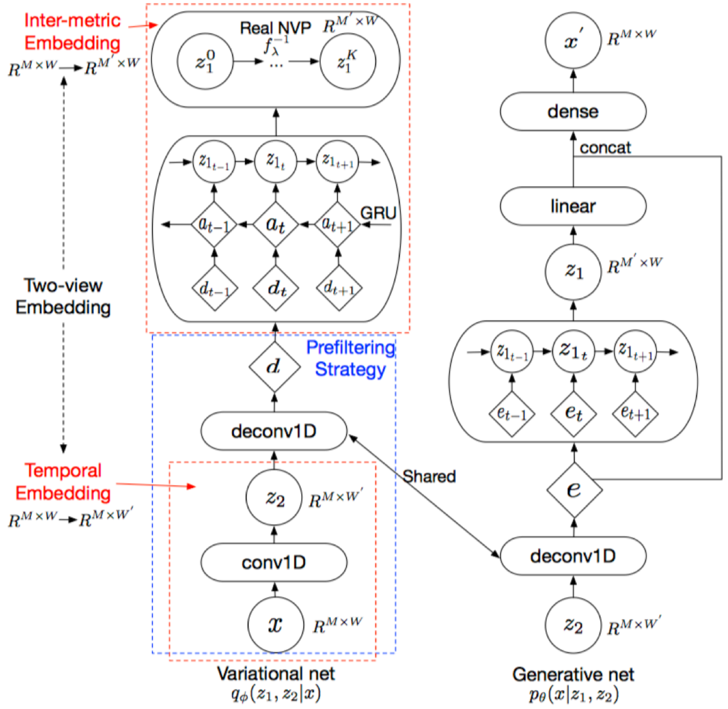
Prefiltering Strategy



Prefiltering strategy reduces the **risk of overfitting** to potential anomalies, while preserving the **flexibility** of the learned inter-metric embeddings

Derive z1 from reconstructed input **d** rather than raw input **x**.

Offline Training



Deduce training objective with auxiliary deterministic variables **d**, **e**:

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \theta, \phi) &= \mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2, \mathbf{d}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{e})\right] \\ &\quad - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}_1, \mathbf{z}_2, \mathbf{d}|\mathbf{x}) \| p_\theta(\mathbf{z}_1, \mathbf{z}_2, \mathbf{e})\right) \\ &= \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, \mathbf{e}) + \underline{\log p_\theta(\mathbf{z}_1, \mathbf{e}|\mathbf{z}_2)}\right. \\ &\quad \left. + \log p_\theta(\mathbf{z}_2) \underline{- \log q_\phi(\mathbf{z}_1, \mathbf{d}|\mathbf{z}_2, \mathbf{x})} - \log q_\phi(\mathbf{z}_2|\mathbf{x})\right]\end{aligned}$$

Share parameters of DeconvNets g:

$$\mathbf{d}_{1:W} \sim q_\phi(\mathbf{d}_{1:W}|\mathbf{z}_2, \mathbf{x}) = q_\phi(\mathbf{d}_{1:W}|\mathbf{z}_2) = \delta(\mathbf{d}_{1:W} - g(\mathbf{z}_2))$$

$$p_\theta(\mathbf{e}_{1:W}|\mathbf{z}_2) = \delta(\mathbf{e}_{1:W} - g(\mathbf{z}_2))$$

$$q(\mathbf{d}_{1:W}|\mathbf{z}_2) = p(\mathbf{e}_{1:W}|\mathbf{z}_2)$$

Online Inference

Use sliding window $(\mathbf{x}_{t-W+1}, \ldots, \mathbf{x}_t)$ for detecting anomaly at time t, use the **negative reconstruction probability** of last data $\mathbf{x}_t$ as the anomaly score.

Assume the last data $\mathbf{x}_t$ as an anomaly beforehand, and apply **MCMC imputation** [ICML'14] on x to get a more reasonable reconstruction.

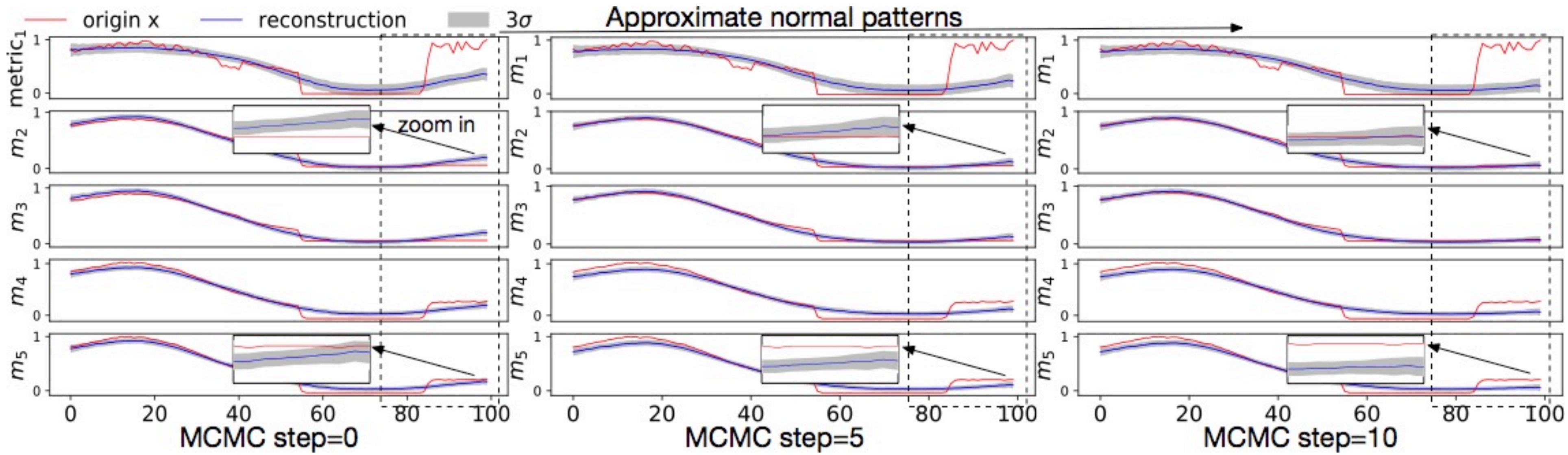
$$\mathbf{x} = (\mathbf{x}_o, \mathbf{x}_t) \xrightarrow[\text{imputation}]{\text{MCMC}} \mathbf{x}^0 = (\mathbf{x}_o, {\mathbf{x}_t}^0) \xrightarrow[\text{S times}]{\text{repeat}} \bar{\mathbf{x}} = (\mathbf{x}_o, {\mathbf{x}_t}')$$

Anomaly score:

$$-\mathbb{E}_{q_\phi(\mathbf{z}_1, \mathbf{z}_2 | \bar{\mathbf{x}})}[\log p_\theta(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2)] = -\frac{1}{L}\sum_{l=1}^{L}[\log p_\theta(\mathbf{x}|\mathbf{z}_1^{(l)}, \mathbf{z}_2^{(l)})]$$

An observation with a higher score is more likely to be an anomaly.

Anomaly Interpretation

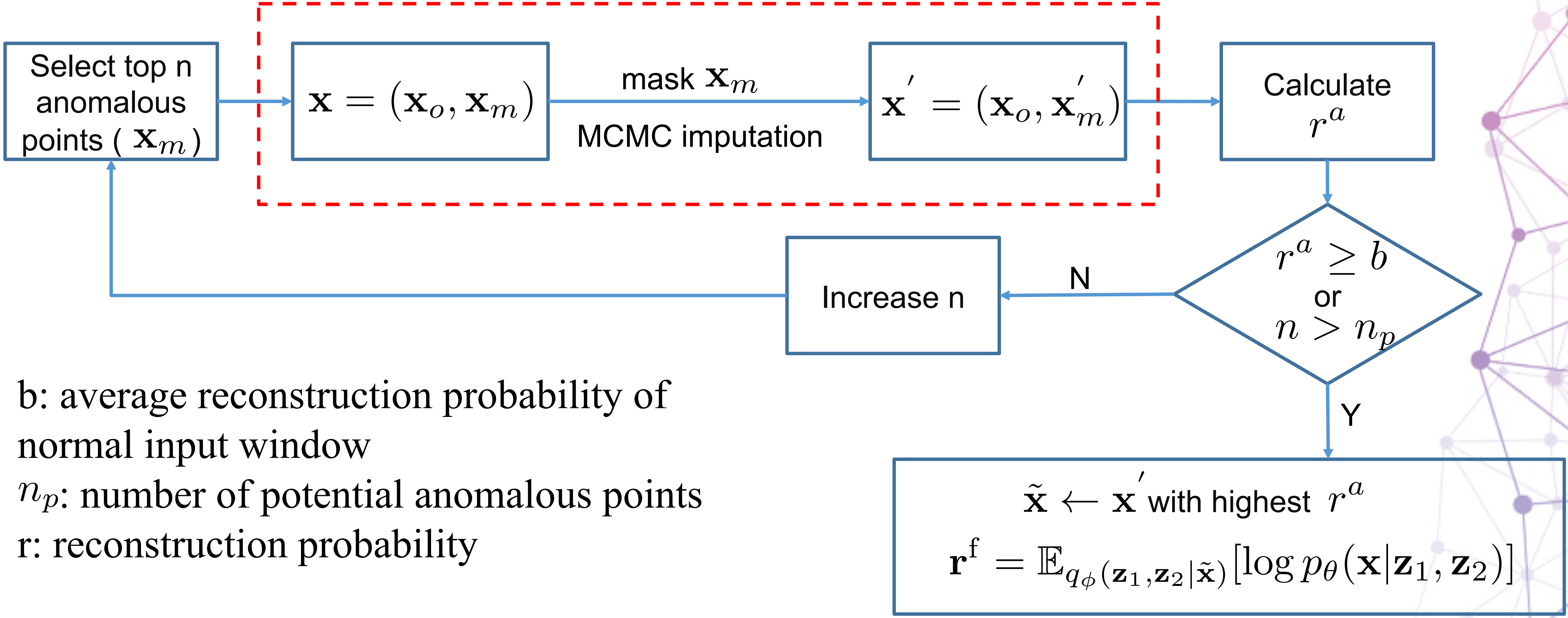


Goal: find a group of most anomalous metrics for each detected entity anomaly.
Non-Goal: directly find the root cause of anomaly

Challenges: The anomalies can affect the estimation of reconstructions at **all** dimensions (anomalous or not).

Idea: Use an MCMC-based method to **approximate the normal patterns**, and then interpret the anomalies based on the revised reconstruction probability.

Anomaly Interpretation



b: average reconstruction probability of normal input window

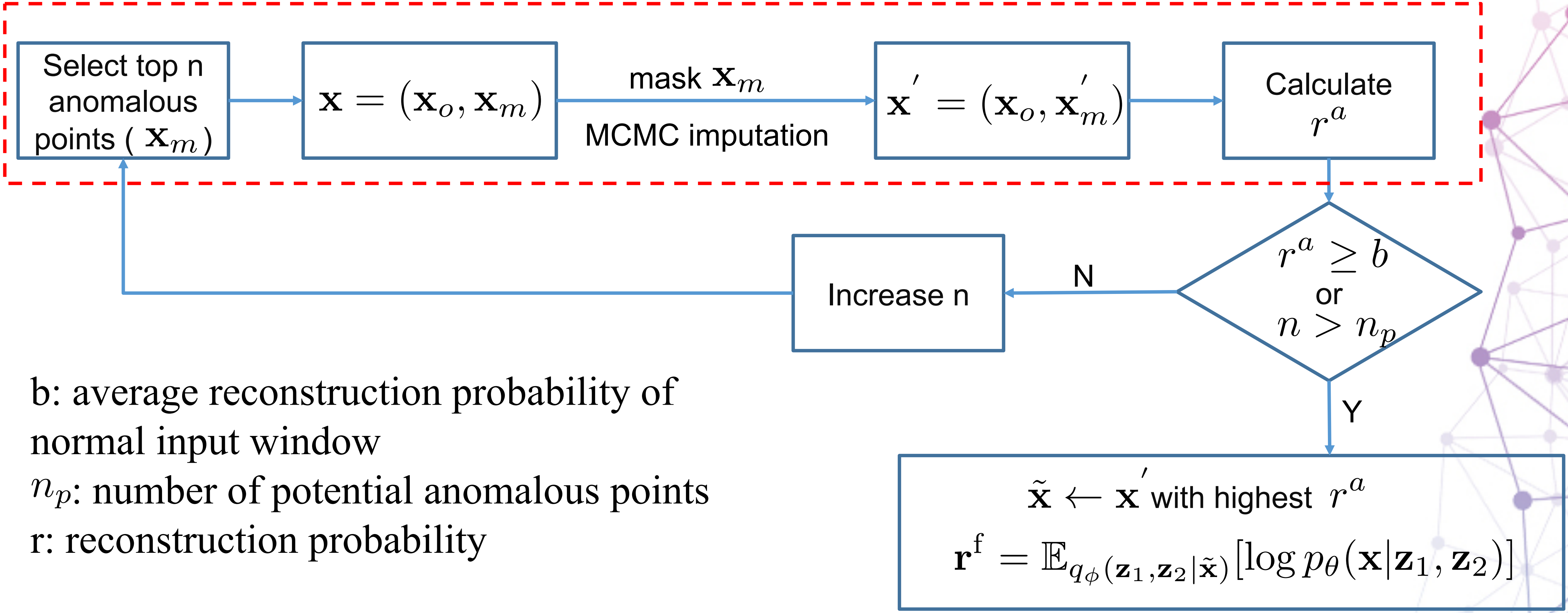
n_p: number of potential anomalous points

r: reconstruction probability

Anomaly Score = $-\mathbf{r}^{\mathrm{f}}$

Metrics with higher anomaly score are more likely to be anomalous metrics.

Anomaly Interpretation



b: average reconstruction probability of normal input window

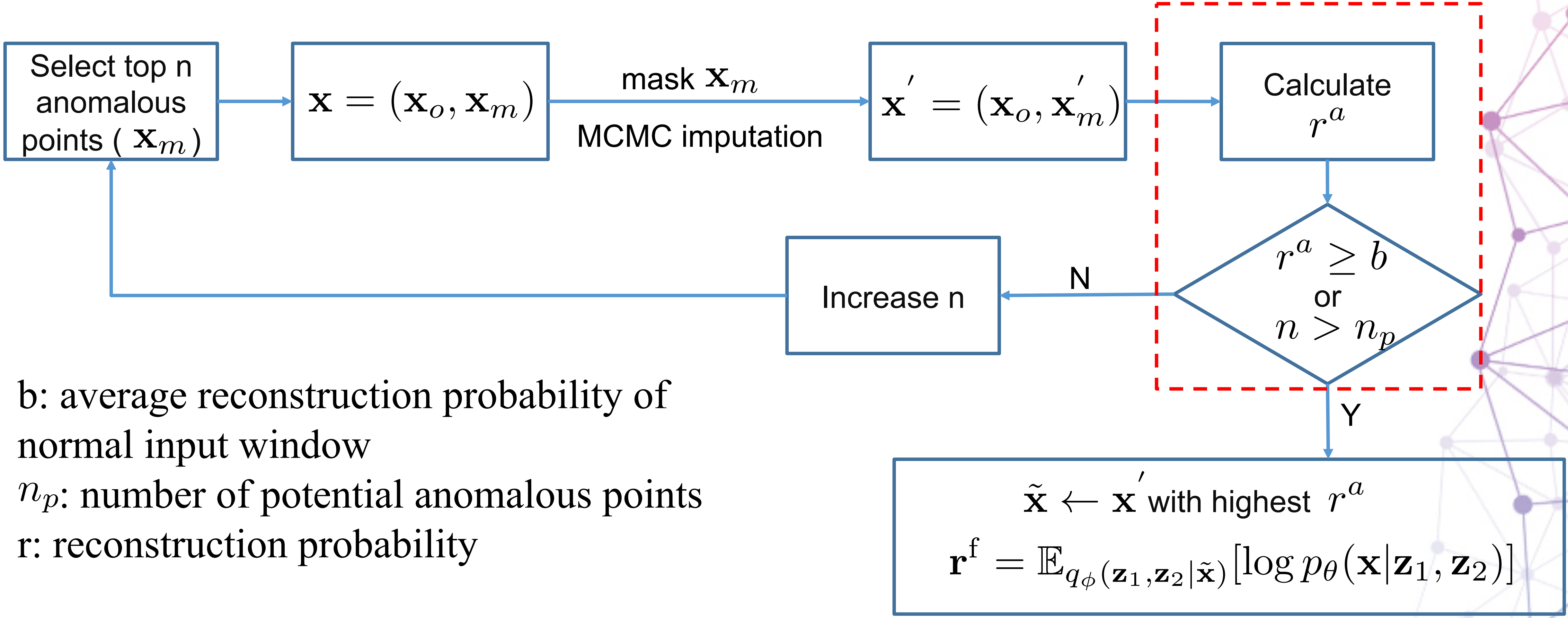
n_p: number of potential anomalous points

r: reconstruction probability

Anomaly Score = $-\mathbf{r}^{\mathrm{f}}$

Metrics with higher anomaly score are more likely to be anomalous metrics.

Anomaly Interpretation



b: average reconstruction probability of normal input window

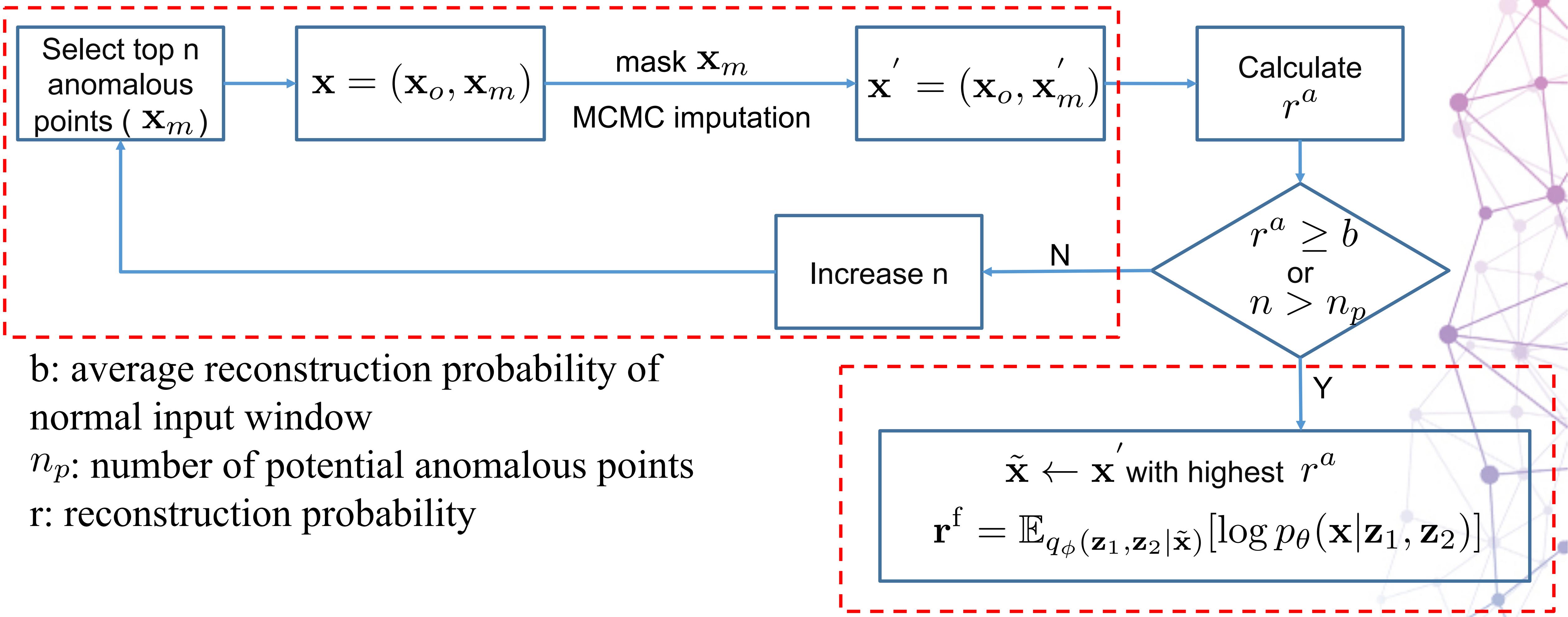
n_p: number of potential anomalous points

r: reconstruction probability

Anomaly Score = $-\mathbf{r}^{\mathrm{f}}$

Metrics with higher anomaly score are more likely to be anomalous metrics.

Anomaly Interpretation



b: average reconstruction probability of normal input window

n_p: number of potential anomalous points

r: reconstruction probability

Anomaly Score = $-\mathbf{r}^{\mathrm{f}}$

Metrics with higher anomaly score are more likely to be anomalous metrics.

Challenges & Designs

Challenges	**Designs**
Learn and fuse inter-metric and temporal embeddings	Hierarchical Structure Two-View Embedding
Prevent overfitting to anomalies	Prefiltering Strategy
Approximate the normal patterns of anomalous metrics	MCMC-based Interpretation Method

Outline

Background

Algorithm

Evaluation

Conclusion

Datasets

Dataset	# entities	# metrics	Train	Test	Anomaly (%)
SWaT	1	51	475200	449919	12.13
WADI	1	118	789371	172801	5.85
SMD	12	38	304168	304174	5.84
ASD	12	19	102331	51840	4.61

Segment-level Evaluation Metrics

For anomaly detection: point-adjust approach [WWW'18, KDD'19, KDD'20]
Metrics: F1-score, AUROC, AP

For anomaly interpretation: we propose IPS metric for segment-level evaluation.

$$\mathrm{IPS} = \sum_{a=1}^{A} \frac{w_a \boxed{|G_{\Phi_a} \cap I_{\Phi_a}|}}{\boxed{|G_{\Phi_a}|}}, \qquad w_a = \boxed{\frac{N_{\phi_a}}{\sum_{a=1}^{A} N_{\phi_a}}}$$

Interpretation accuracy for each anomaly segment

Segment importance according to anomaly duration

Research Questions

- RQ1: How does *InterFusion* perform on MTS anomaly detection and interpretation, in comparison with the state-of-the-art methods?
- RQ2: How effective is each design choice in *InterFusion*?
- RQ3: Is *InterFusion* feasible to be deployed in production?

Research Questions

- RQ1: How does *InterFusion* perform on MTS anomaly detection and interpretation, in comparison with the state-of-the-art methods?

- RQ2: How effective is each design choice in *InterFusion*?

- RQ3: Is *InterFusion* feasible to be deployed in production?

RQ1. Anomaly Detection

Average best-F1 for InterFusion and baselines

Methods	SWaT	WADI	SMD	ASD	Avg.
LSTM-NDT	0.8133	0.5067	0.7687	0.4061	0.6237
MSCRED	0.8346	0.5469	0.8252	0.5948	0.7004
MAD-GAN	0.8431	0.7085	0.8966	0.6325	0.7702
OmniAnomaly	0.7344	0.7927	0.9628	0.8344	0.8311
DSANet	0.8924	0.8739	0.9630	0.8740	0.9008
USAD	0.8227	0.4275	0.9024	0.7987	0.7378
VAEpro	0.8369	0.8200	0.8693	0.8522	0.8446
InterFusion	**0.9280**	**0.9103**	**0.9817**	**0.9531**	**0.9433**

RQ1. Anomaly Interpretation

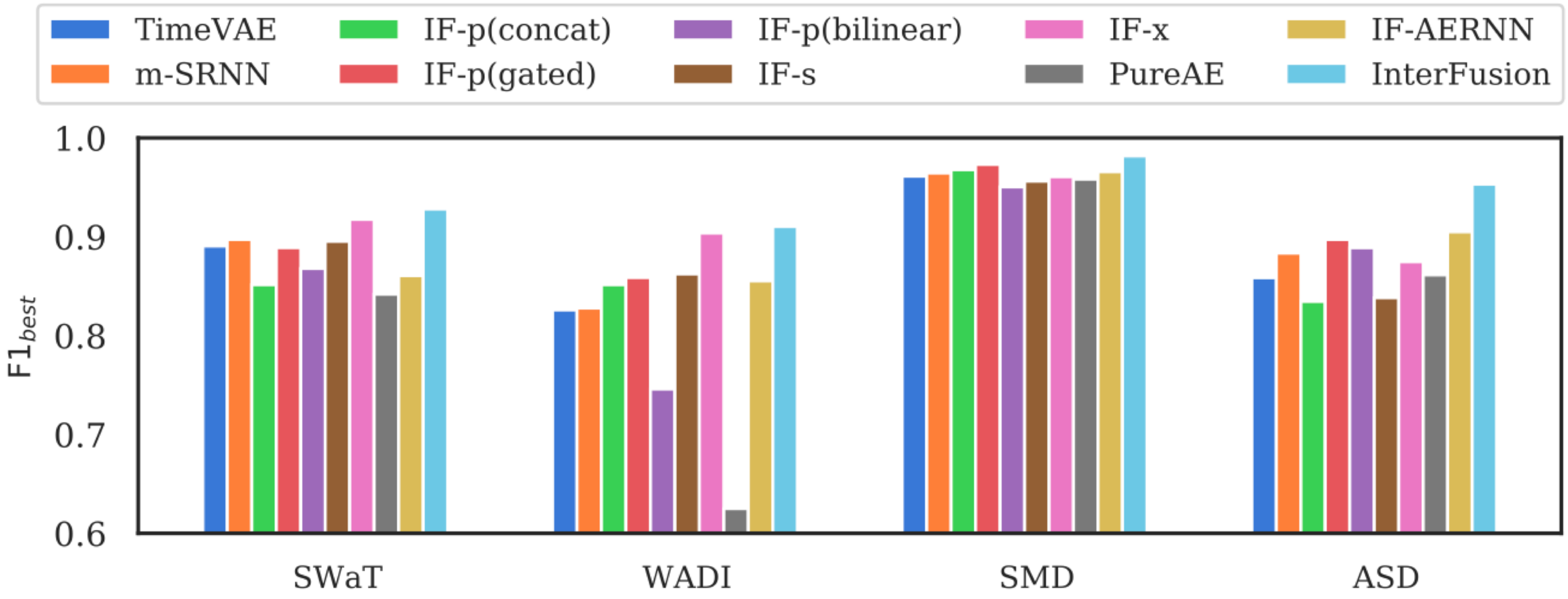
Interpretation IPS for InterFusion and baselines

Methods	SMD	ASD	Avg.
LSTM-NDT	0.5751	0.8619	0.7185
MSCRED	0.6421	0.7652	0.7037
OmniAnomaly	0.8008	0.8029	0.8019
DSANet	0.6713	0.8123	0.7418
VAEpro	0.5681	0.8236	0.6959
VAEpro*	0.7433	0.8916	0.8175
InterFusion-nI	0.7752	0.8881	0.8317
InterFusion	**0.8340**	**0.9107**	**0.8724**

Research Questions

- RQ1: How does *InterFusion* perform on MTS anomaly detection and interpretation, in comparison with the state-of-the-art methods?
- RQ2: How effective is each design choice in *InterFusion*?
- RQ3: Is *InterFusion* feasible to be deployed in production?

RQ2. Ablation Studies

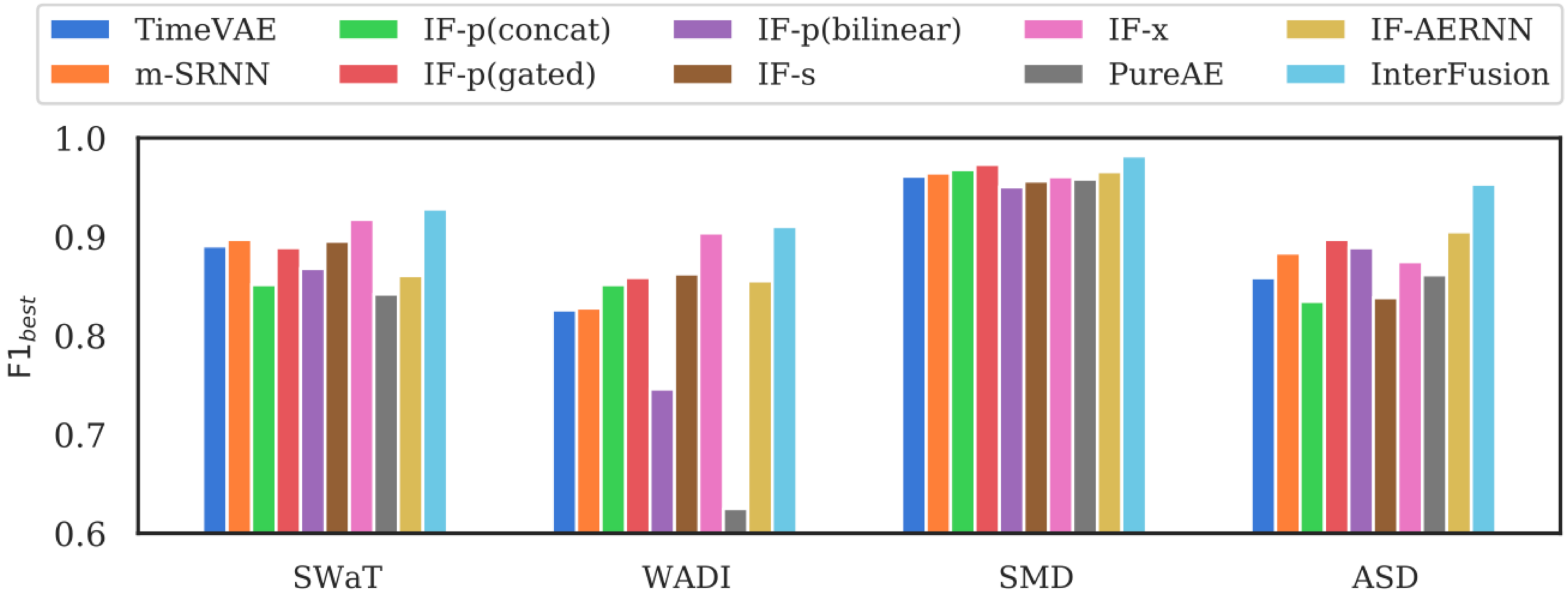


Inter-metric and temporal embeddings: m-SRNN, TimeVAE

InterFusion-p (concat, gated, bilinear): different feature fusion methods

InterFusion-s: without two-view embedding

RQ2. Ablation Studies



InterFusion-x: without prefiltering strategy

PureAE and InterFusion-AERNN: generalizability of InterFusion’s designs

Research Questions

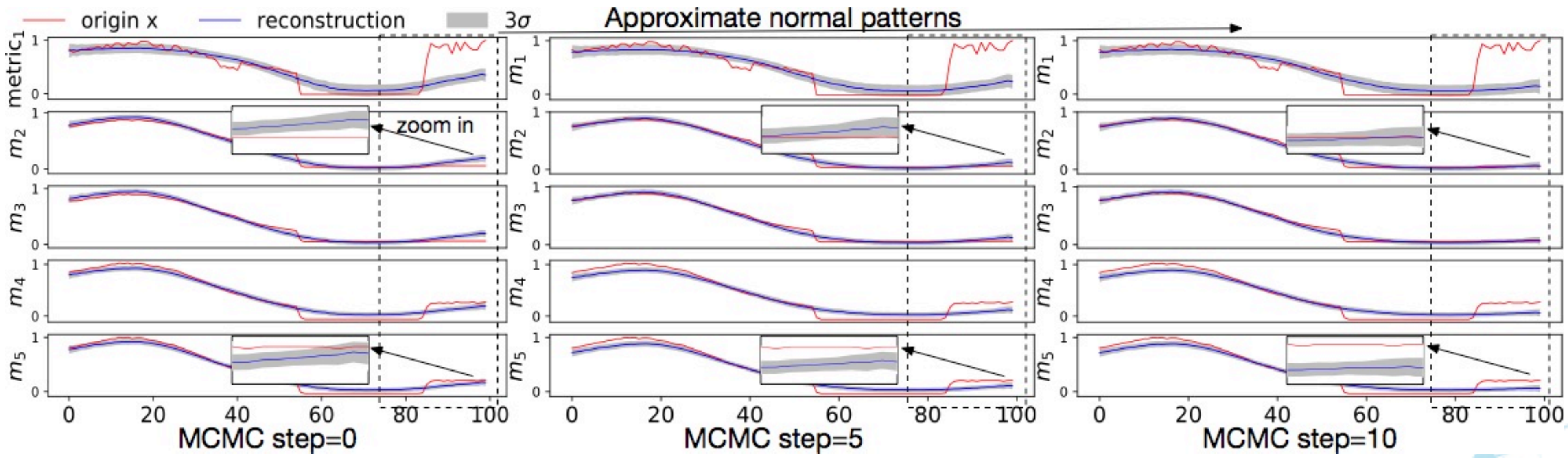
- RQ1: How does *InterFusion* perform on MTS anomaly detection and interpretation, in comparison with the state-of-the-art methods?
- RQ2: How effective is each design choice in *InterFusion*?
- RQ3: Is *InterFusion* feasible to be deployed in production?

RQ3. Feasibility Study

On ASD (Application Server Dataset) from a large Internet company.

Detection: overall precision of 0.93, overall recall of 0.99. Successfully detect severe and subtle anomalies.

Interpretation: tell the operators about the most affected metrics and the extent to which the metrics deviate from their normal patterns.



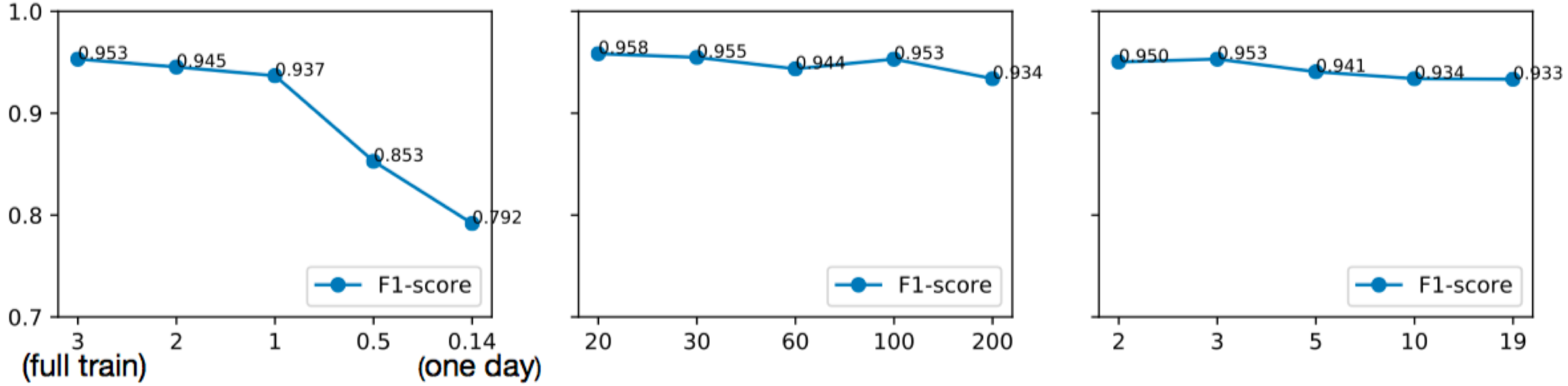
RQ3. Feasibility Study

Computation time:

- Offline training: about 6 minutes for 3-week training data (5-minute interval).
- Online detection: less than 1 second per point.
- Interpretation: about 2~15 seconds for each entity anomaly.

Parameter sensitivity:

- Robust to potential anomalies in training data. (prefiltering strategy)
- Low demand for training data. (achieve high performance even with one-week data)
- NOT sensitive to sliding window length and dimensions of latent variables.



(a) weeks of training data (b) window length (c) latent dimension M’

Outline

Background

Algorithm

Evaluation

Conclusion

Contribution

InterFusion

- The first MTS anomaly detection algorithm that employs HVAE with explicit low-dimensional inter-metric and temporal embeddings to jointly learn robust MTS representations

- Proposes a novel anomaly interpretation method based on MCMC imputation, and a new segment-wise evaluation metric consistent with the system operators' preferences.

- Achieves overall best F1-Score higher than 0.94 and overall interpretation accuracy of 0.87 on four real-world datasets, outperforming the state-of-the-art methods.

Conclusion

- Simultaneously learning low-dimensional inter-metric and temporal embeddings improves the anomaly detection performance for each type of anomalies than just learning a single type representation for MTS.

- A two-view embedding with hierarchical stochastic latent variables has been shown as an effective way to jointly learn robust MTS representations.

- The MCMC-based imputation approach can help obtain reasonable latent embeddings and reconstructions at detected entity anomalies in MTS, which can help better interpret the detected MTS anomalies.

Code and Data: https://github.com/zhhlee/InterFusion

THANK YOU!

Q&A?

lizhihan17@mails.tsinghua.edu.cn

清華大學 Tsinghua University