# Unsupervised Anomaly Detection for Intricate KPIs via Adversarial Training of VAE

第二个窗口

**Wenxiao Chen**, Haowen Xu, Zeyan Li, Dan Pei,
Jie Chen, Honglin Qiao, Yang Feng, Zhaogang Wang

清華大學
Tsinghua University

Alibaba Group
阿里巴巴集团

Background

Challenges

Ideas

Experiments

Background

Challenges

Ideas

Experiments

# Key Performance Indicators

Fig1: Web Applications

# Key Performance Indicators


Request

Fig2: From user to server

Monitor

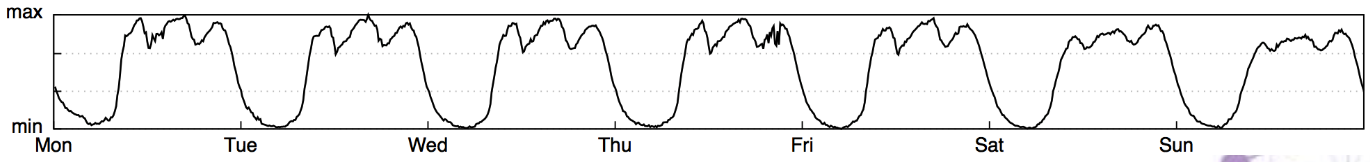



Fig3: Page view

It is stable?

# Key Performance Indicators

Smooth KPIs: e.g, The respond time of server, and page view

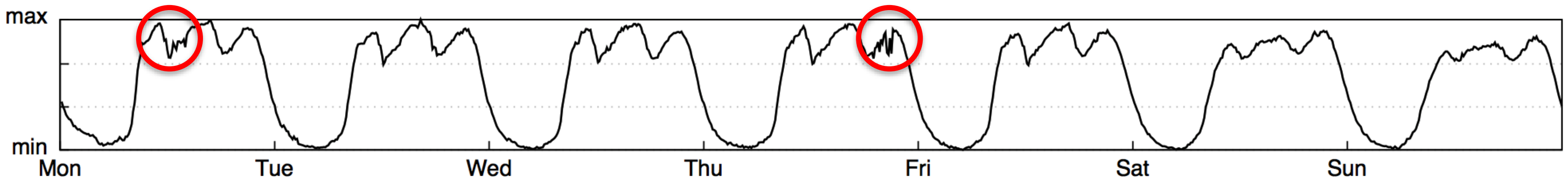



Fig4: An example of Anomalies in page view

Intricate KPIs: e.g, The query per second and transaction per second

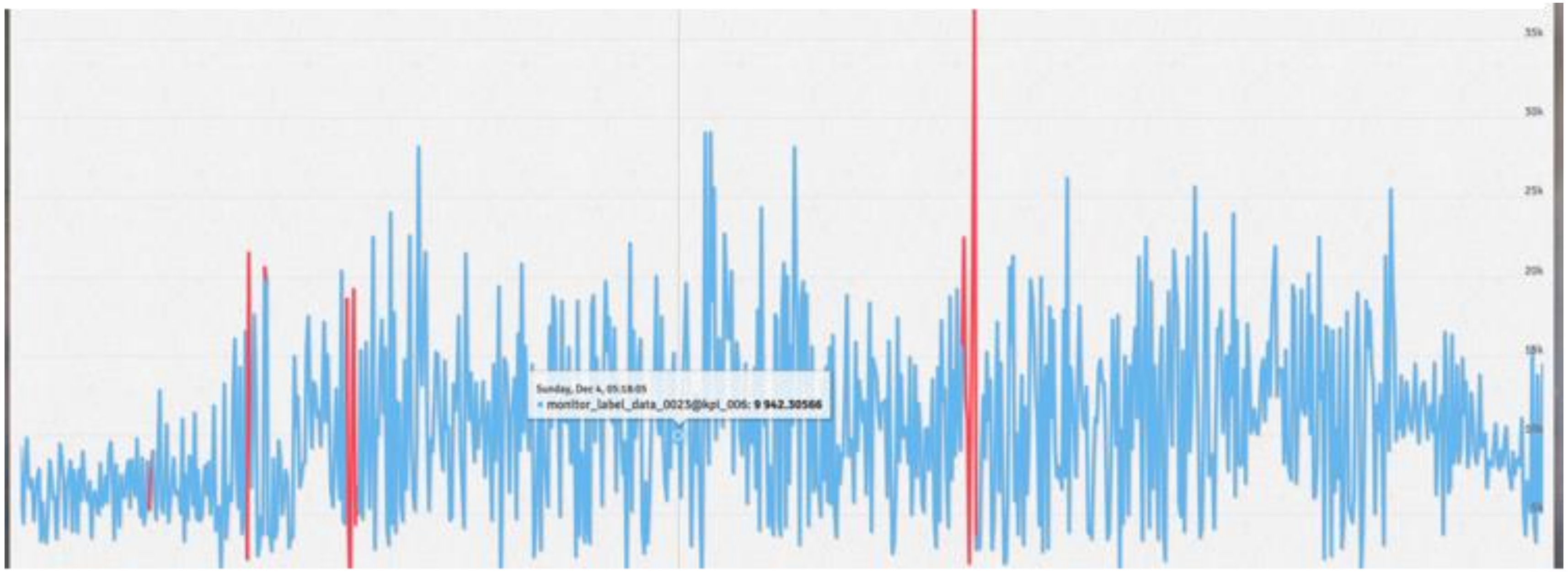


Fig5: An example of Anomalies in database

# Existing Method

- Statistical
  – Anomaly detectors based on traditional statistical models [INFOCOM2012]
- Supervised
  – Supervised ensemble learning with above detectors – Opprentice[IMC2015]
- Unsupervised
  – Unsupervised anomaly detection based on VAE – Donut [WWW2018]

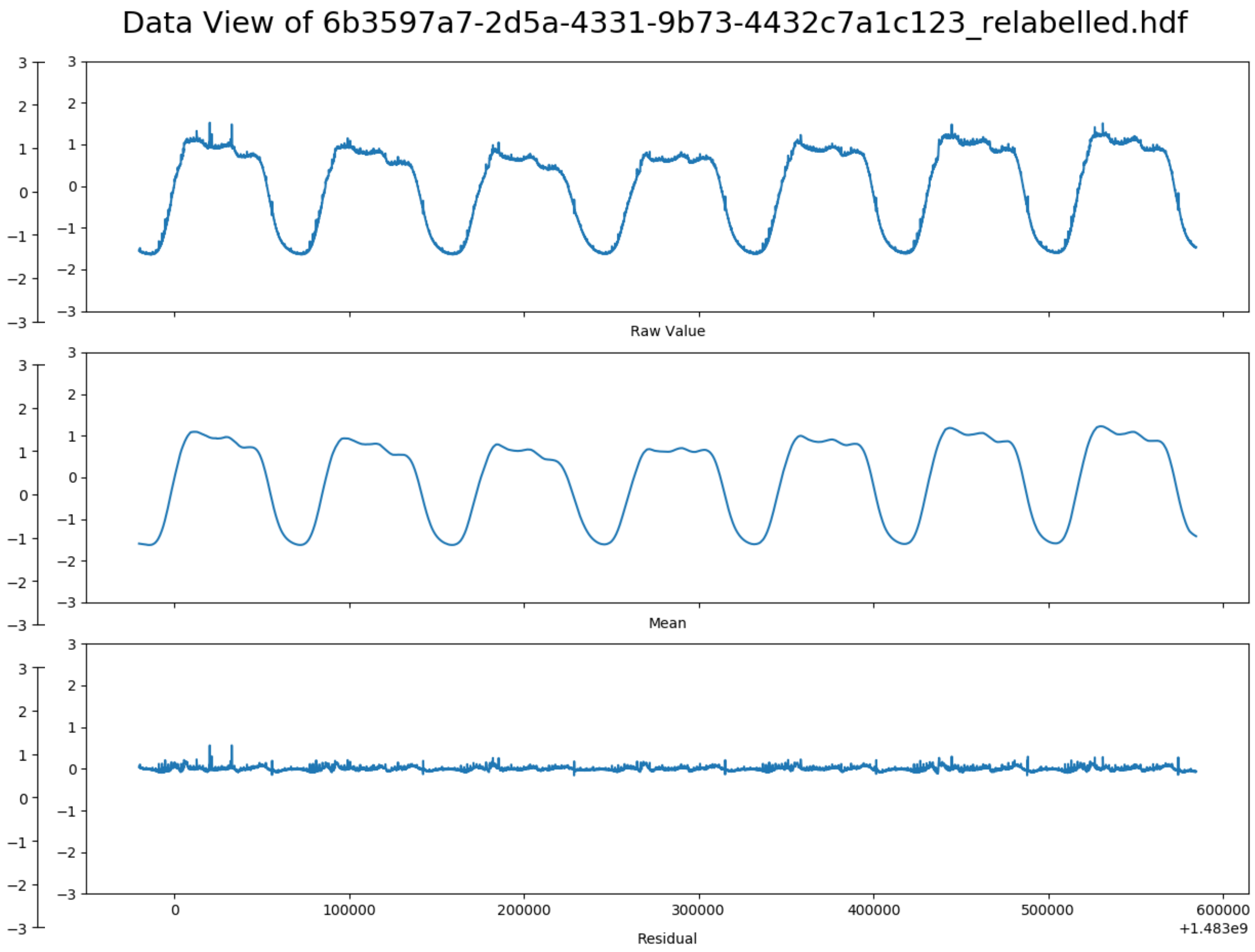
They can only work on smooth KPIs.

# Smooth KPIs

Original data

Smooth Curve

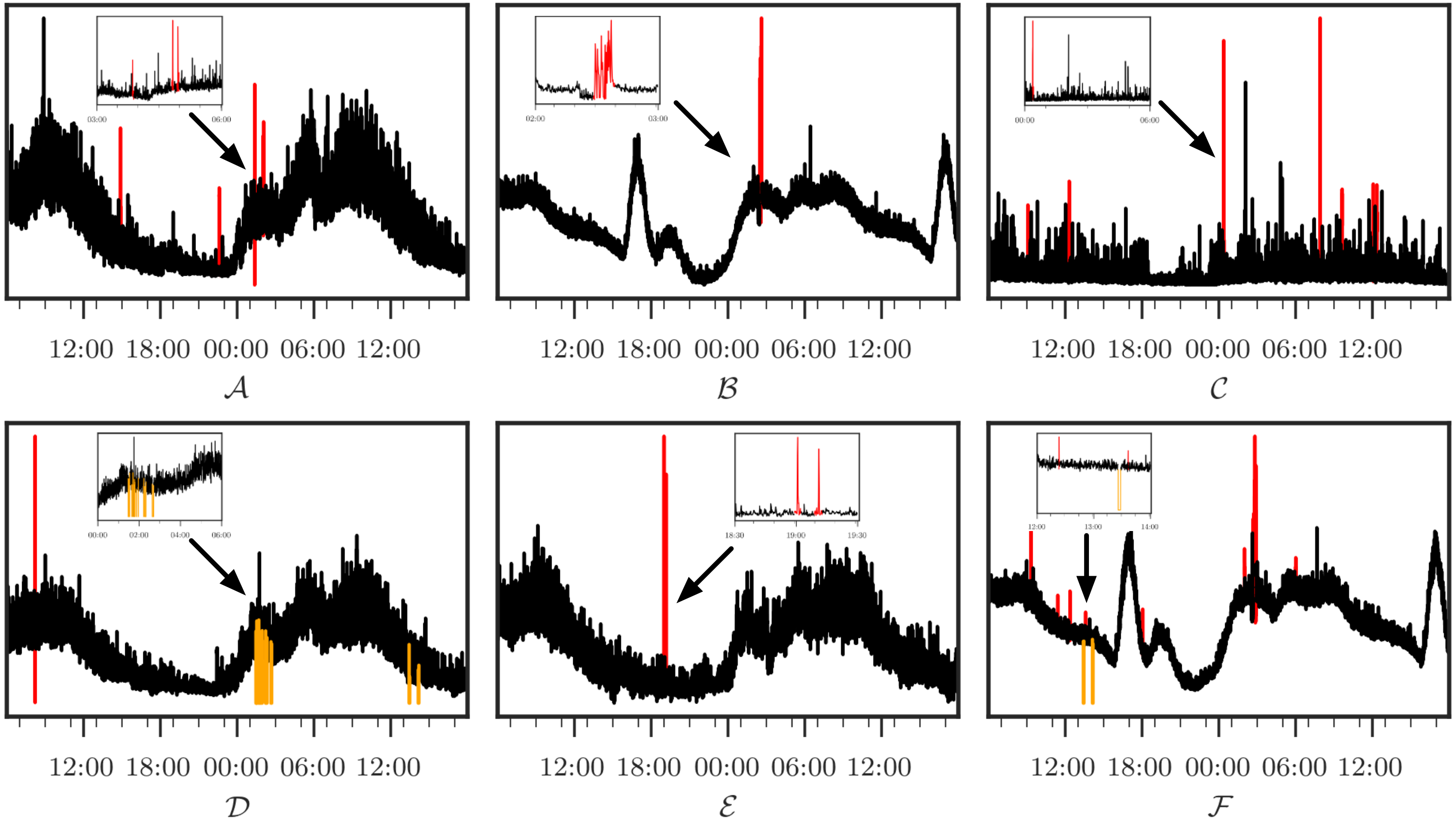
Gaussian Noise




Time of 2016-12-29 10:51:00 to 2017-01-05 10:50:00
Time of 2017-04-02 00:00:00 to 2017-04-08 23:59:50

Fig6: Smooth KPI

# Intricate KPIs




- micro-congestion
- fine granularity
- prevalent and important (e.g, database, server)
- little studied

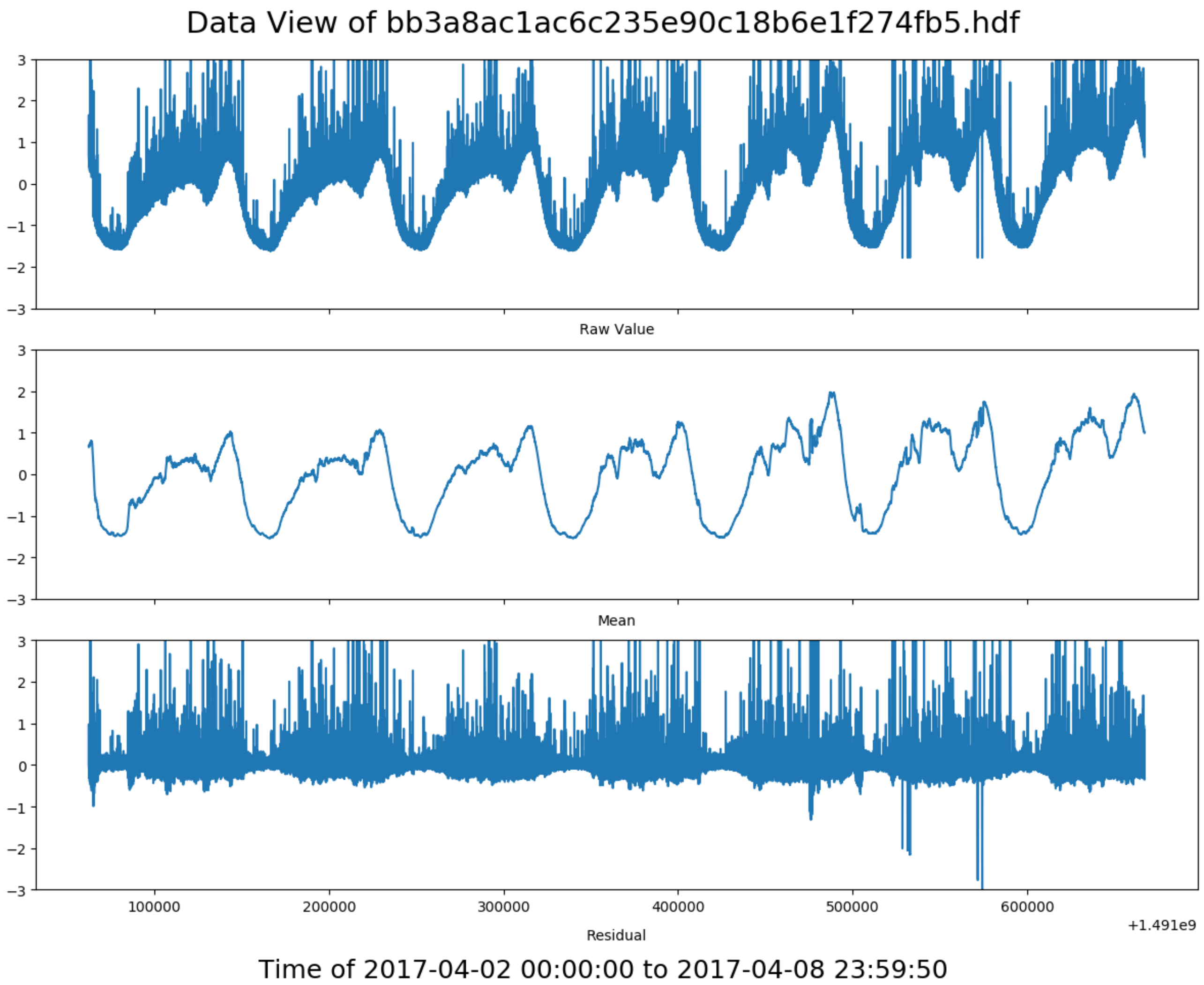
Fig7: Intricate KPIs

# Intricate KPIs

Original data

Smooth Curve

Noise




- non-Gaussian noises
- hard to model

Fig8: KPI A

# Intricate KPIs

Original data

Smooth Curve

Noise

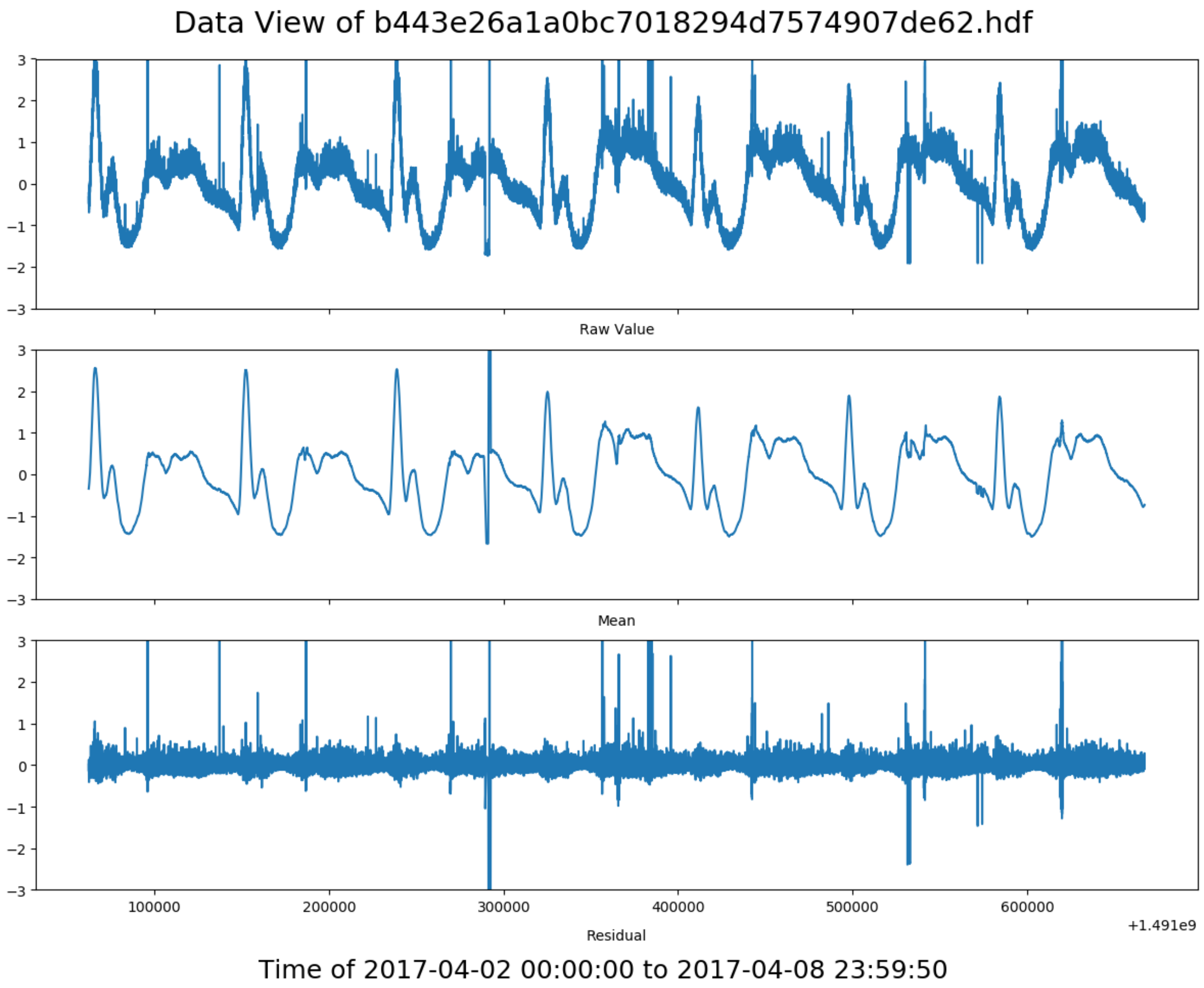



Fig9: KPI B

- non-Gaussian noises
- hard to model

# Intricate KPIs

Original data

Smooth Curve

Noise

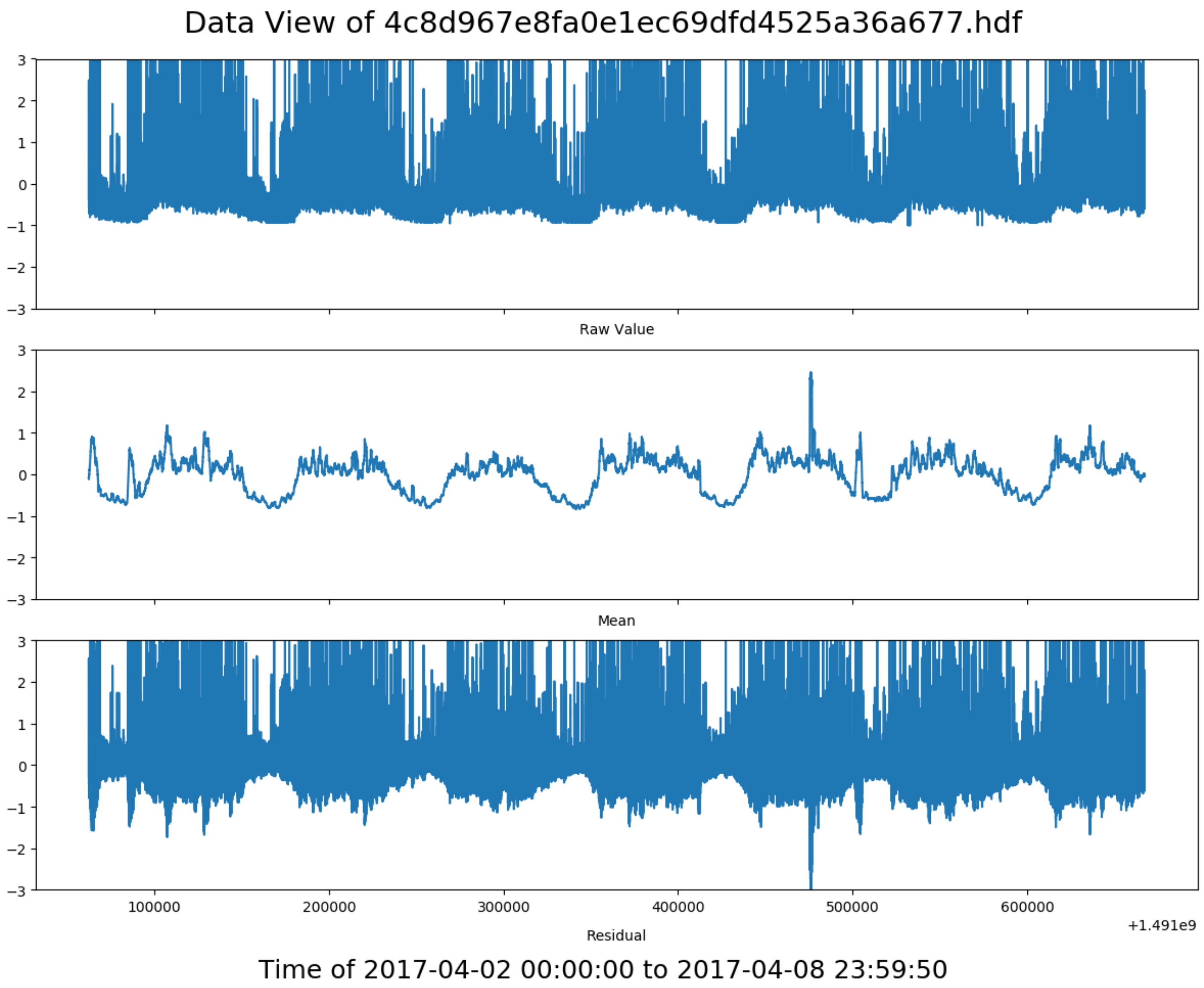



Fig10: KPI C

- non-Gaussian noises
- hard to model

# Donut

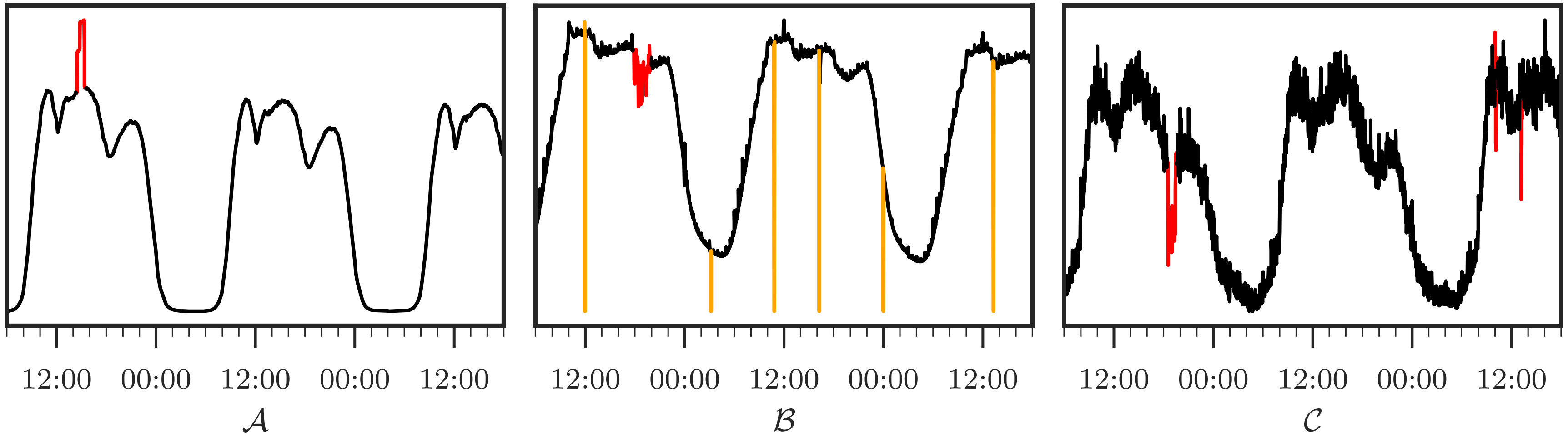



Fig11: The Dataset of Donut

# Donut

- A recent future of W data points at time t is called a window at time t. Donut tries to model the distribution of normal windows by VAE(Variational Auto Encoder) and find anomalies by likelihood.

- The training objective of VAE, is the evidence lower bound of likelihood(ELBO).

$$\mathcal{L}_{vae} = \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z})\right]\right]$$

- In Donut, $p_\theta(\mathrm{x}|\mathrm{z})$ is diagonal multivariate gaussian distribution and it works well on seasonal smooth KPIs.

# Donut

- Element-wise posterior:
  - $\ln p_\theta(\mathrm{x}|\mathrm{z}) = \sum_i \ln p_\theta(\mathrm{x}_i|\mathrm{z})$

- It is useful for smooth KPIs but not for Intricate KPIs.

# Out of Expectation

- Donut assumes that the data is seasonal smooth with diagonal gaussian noise but the intricate KPIs are not.
- VAE will only learn the mean and variance locally.

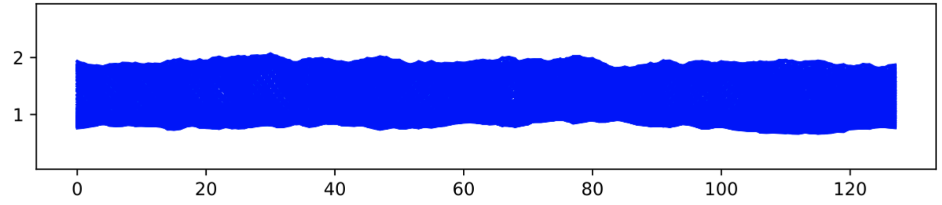



Fig12: Reconstructed element-wise gaussian distribution

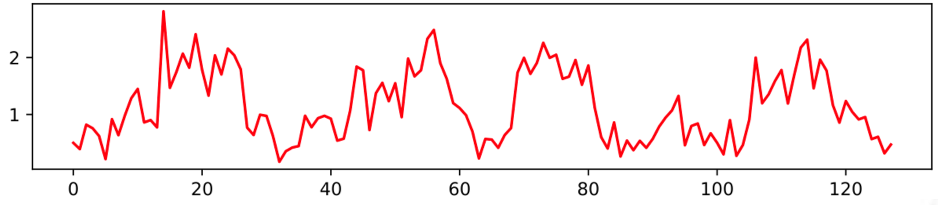



Fig13: Original curve

Background

Challenges

Ideas

Experiments

# Challenges

Dataset is too intricate for VAE to learn

- Element-wise gaussian posterior is not appropriate
- Reconstruction loss is too hard to learn
- The limit of training method

# Challenges

Dataset is too intricate for VAE to learn

Element-wise gaussian posterior is not appropriate

Reconstruction loss is too hard to learn

The limit of training method

# Challenges

Dataset is too intricate for VAE to learn

Element-wise gaussian posterior is not appropriate

Reconstruction loss is too hard to learn

The limit of training method

# Challenges

$$\mathcal{L}_{vae} = \mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p_\theta(\mathbf{z})\right]\right]$$

- $\mathbb{E}_{p(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]\right]$ is called reconstruction loss.
- ELBO is a trade-off and when the reconstruction loss is hard to learn (nearly no gradient from it), our model tends to learn another term.

# Challenges

Dataset is too intricate for VAE to learn

- Element-wise gaussian posterior is not appropriate
- Reconstruction loss is too hard to learn
- The limit of training method

Background

Challenges

Ideas

Experiments

# Ideas

| | |
|---|---|
| Element-wise gaussian posterior is not appropriate | Choose another posterior which is not element-wise |
| Reconstruction loss is too hard to learn | Relax the constraint of reconstruction loss |
| The limit of training method | Use adversarial training |

# Ideas

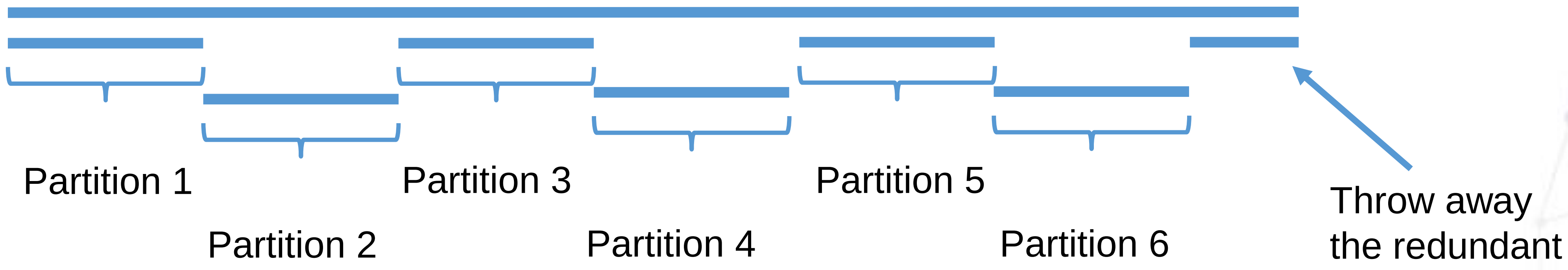
Choose another posterior which is not element-wise

- $p_\theta(\mathrm{x}|\mathrm{z}) = \frac{1}{Z(\lambda)} e^{-\lambda\|\mathrm{x}-G(z)\|}$
- $G(z)$ is the generative network and $\lambda$ is a learnable variable.
- $Z(\lambda)$ can be simply calculated when $\lambda$ is fixed.
- It is easy to check that it is not element-wise.

# Ideas

Relax the constraint of reconstruction loss

- Introduce a new notion: Partition
- Divide the whole KPI into several partitions, whose length are all L



# Partition and Window

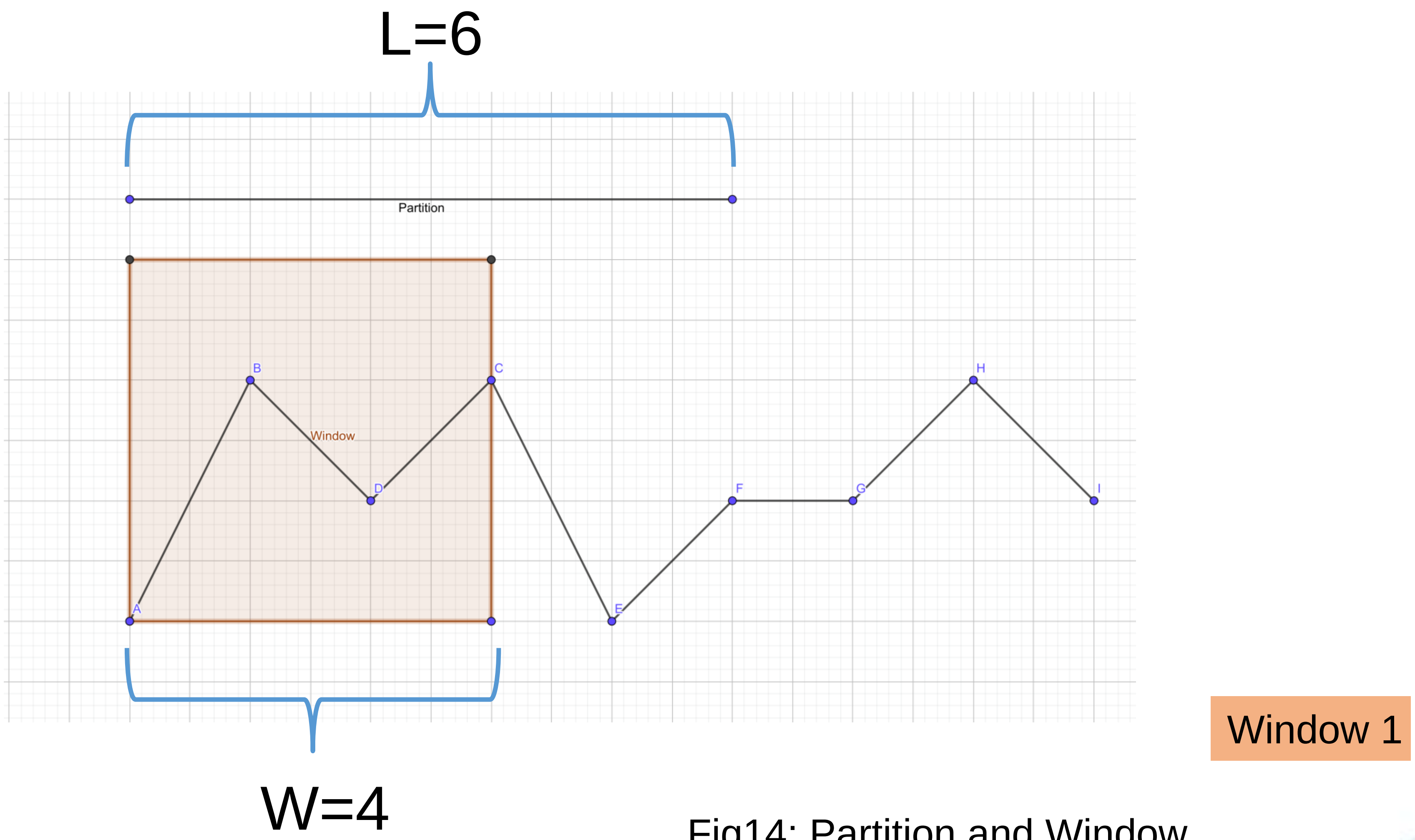
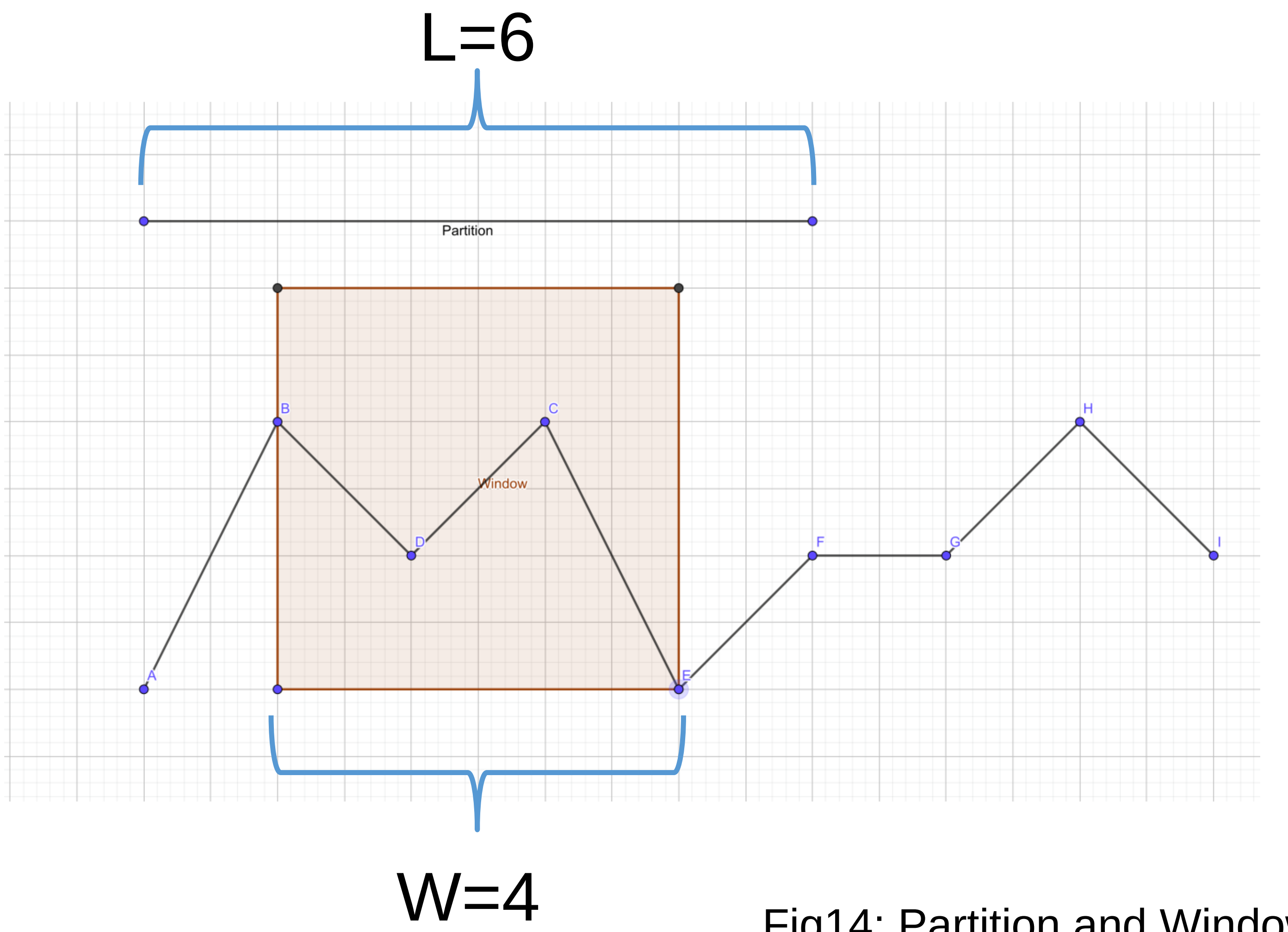



Fig14: Partition and Window

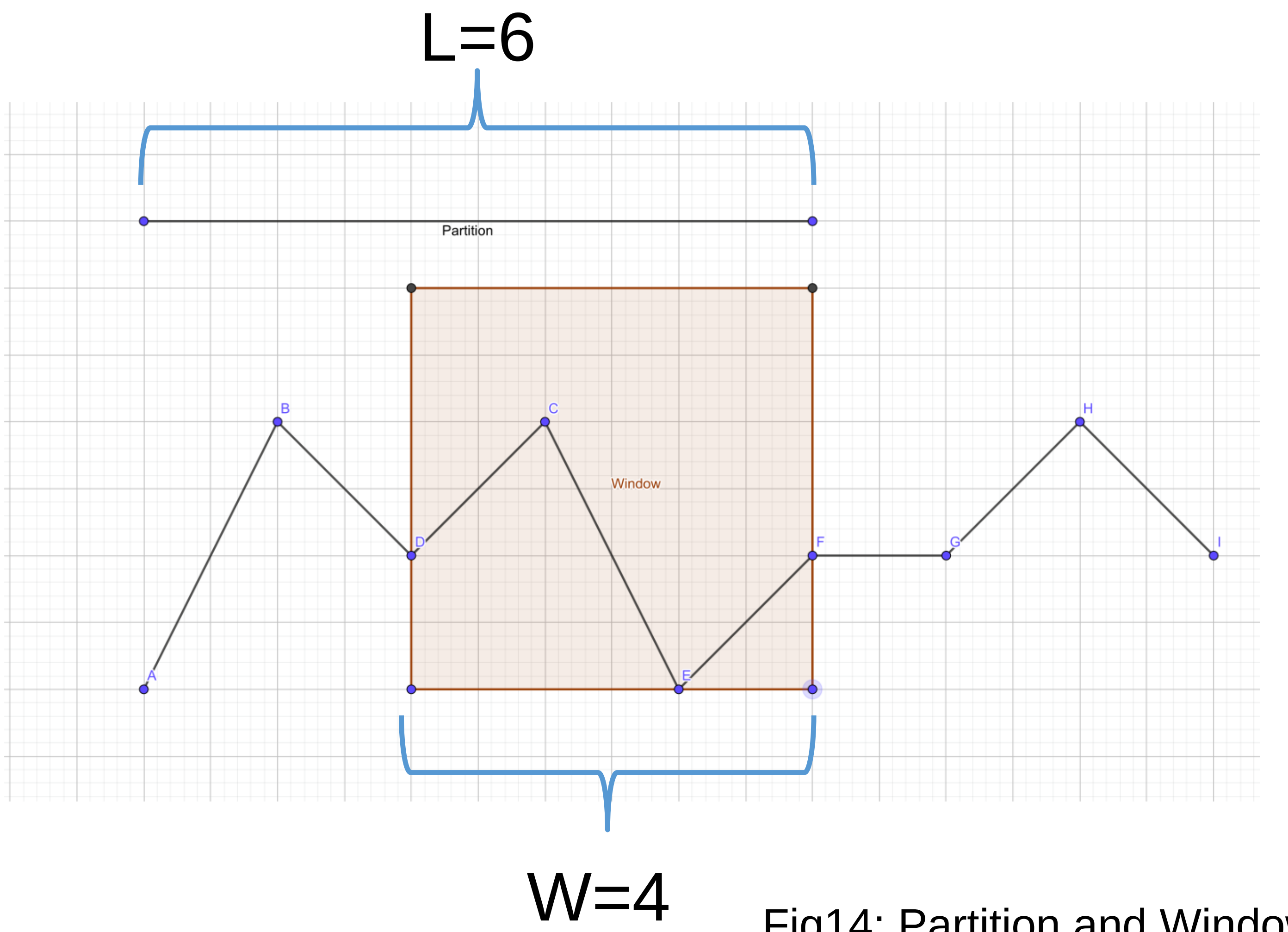
# Partition and Window




Window 2

Fig14: Partition and Window

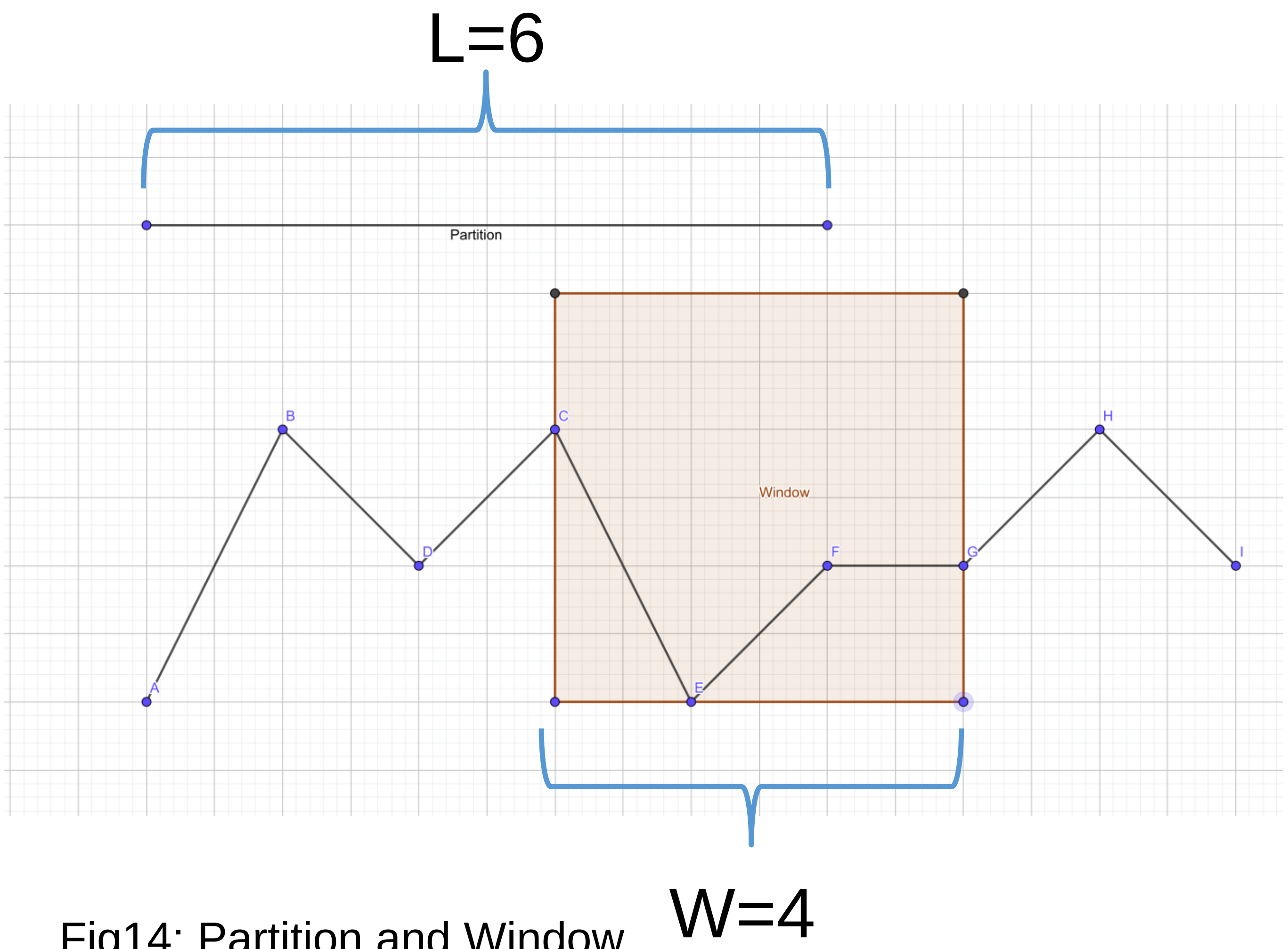
# Partition and Window




Window 3

Fig14: Partition and Window

# Partition and Window

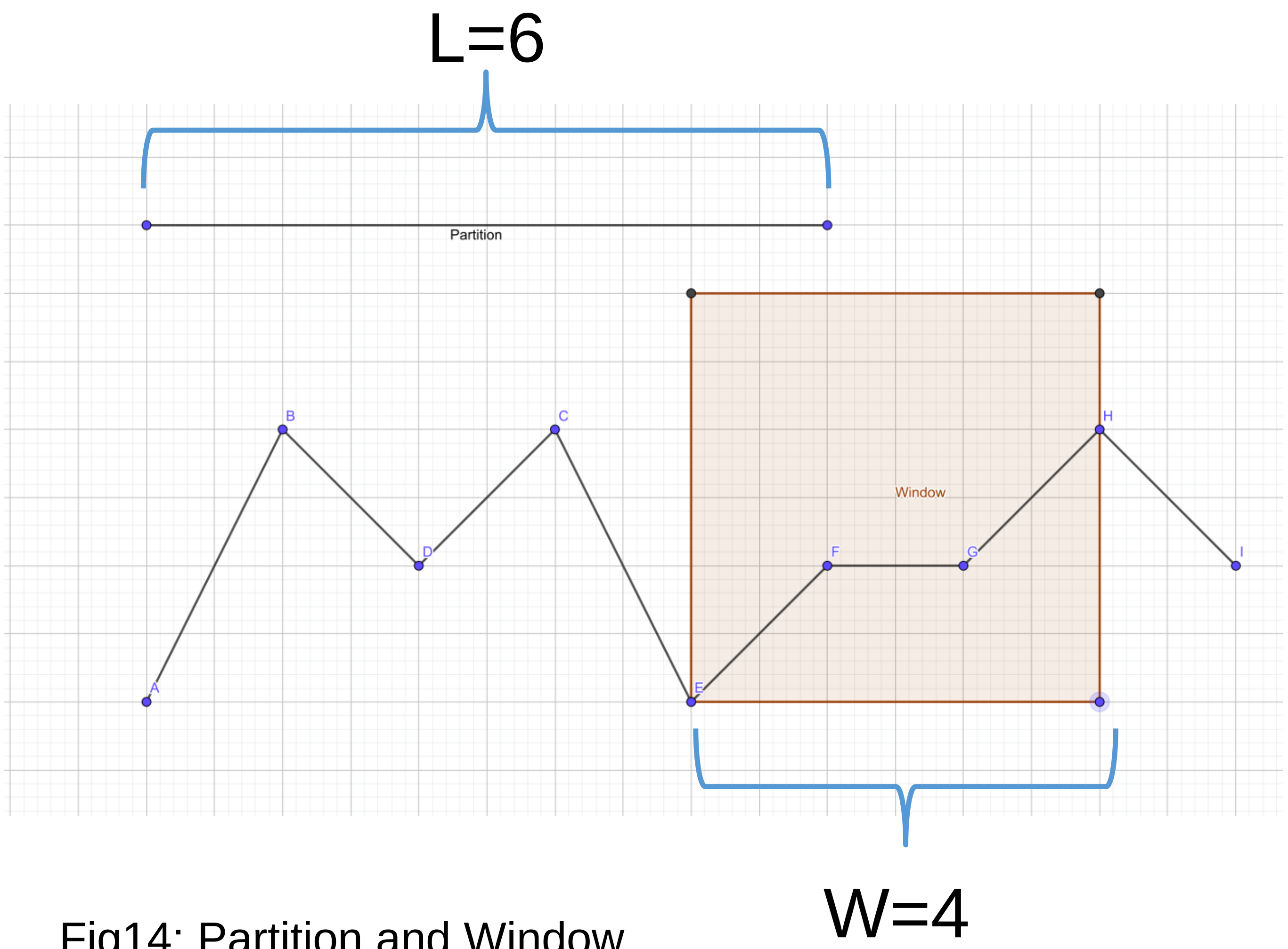



Window 4



Fig14: Partition and Window

# Partition and Window

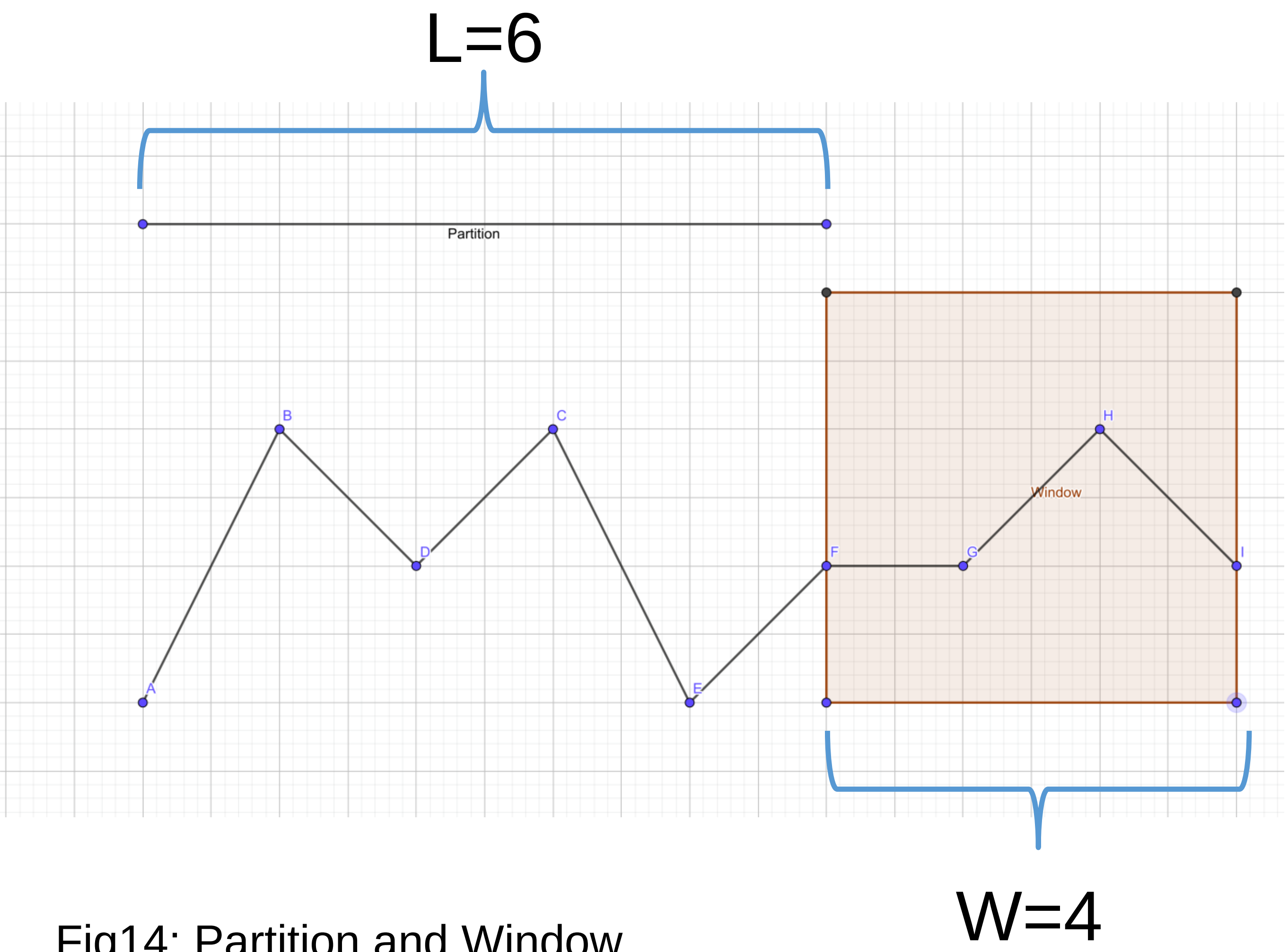



Window 5



Fig14: Partition and Window

# Partition and Window



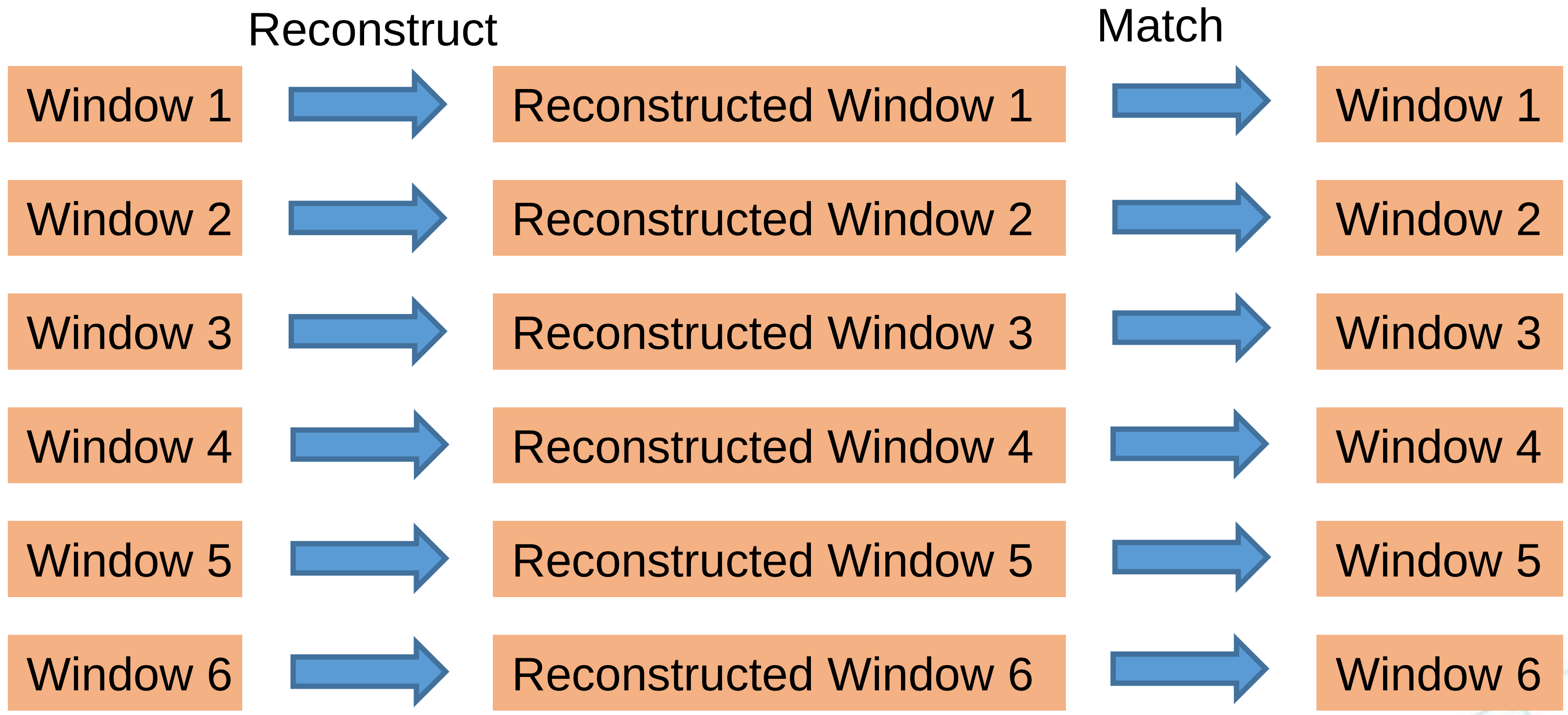

Window 6



Fig14: Partition and Window

# Match

- In a partition, we regard the reconstruction loss as distance between reconstructed window and original window.



# Match

- We relax the reconstruction loss by following way: we permit each reconstructed window to match one window in this partition and compute the sum of the distance between each pair.

| | Reconstruct | | Match | |
|---|---|---|---|---|
| Window 1 | → | Reconstructed Window 1 | → | Window 2 |
| Window 2 | → | Reconstructed Window 2 | → | Window 1 |
| Window 3 | → | Reconstructed Window 3 | → | Window 3 |
| Window 4 | → | Reconstructed Window 4 | → | Window 4 |
| Window 5 | → | Reconstructed Window 5 | → | Window 5 |
| Window 6 | → | Reconstructed Window 6 | → | Window 6 |

# Whole Process

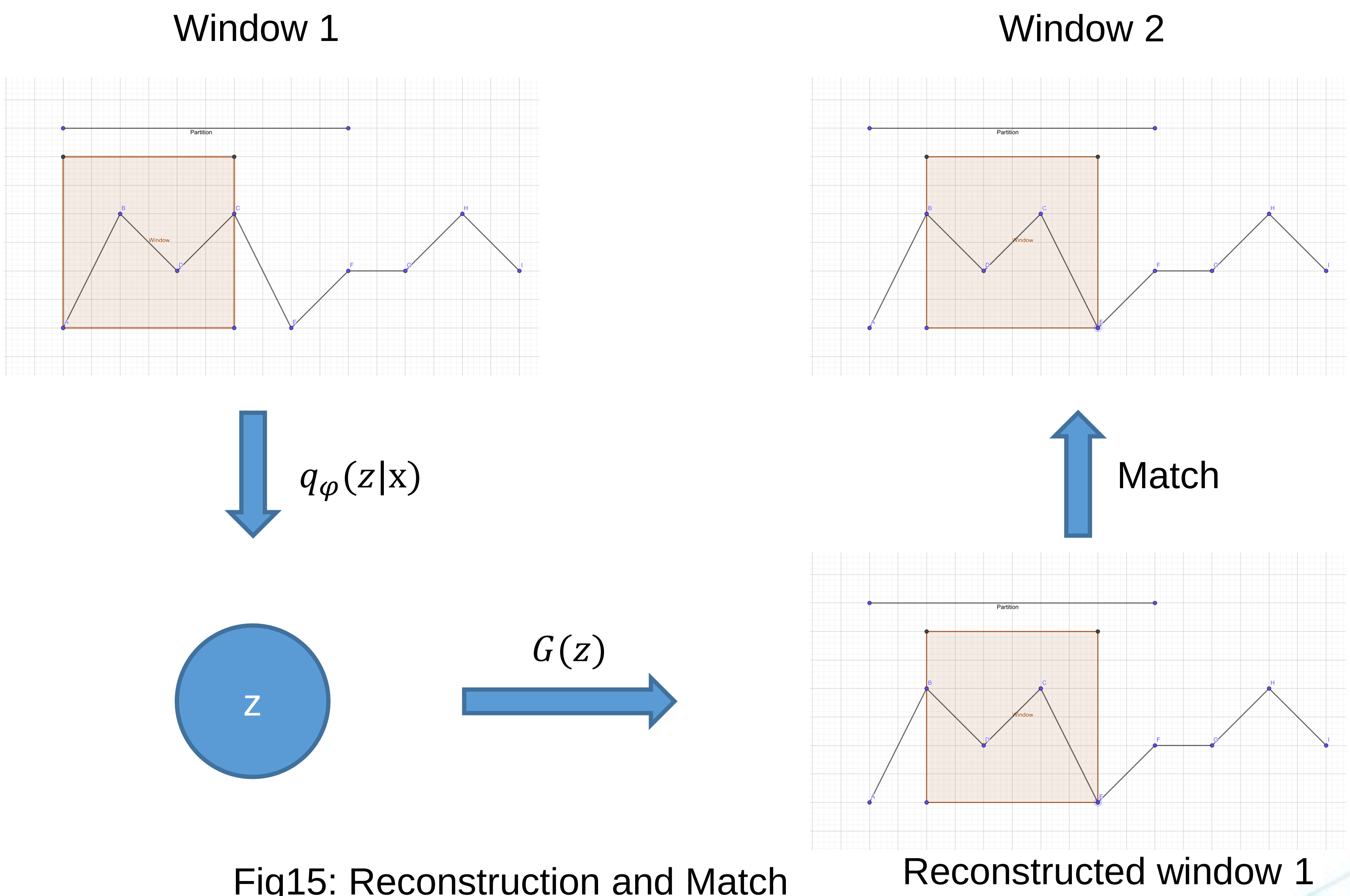



Fig15: Reconstruction and Match

# Relationship

- It is easy to see that match reconstruction loss is less than reconstruction loss (just trivial match).

- Reconstruction loss is the special case: L=1

- Understand it intuitively: L is our tolerance. We tolerate some errors of reconstruction.

# Ideas

Use adversarial training

- A **generative adversarial network** (**GAN**) is a class of machine learning systems. Two neural networks contest with each other in a zero-sum game framework.

- It works very well in image generation.

# Wasserstein distance

- **Wasserstein distance** used by WGAN[ICML2017].

$$W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)] = \inf_{\gamma\in\Gamma_w}\int_{\mathcal{X}\times\mathcal{X}}\|\mathbf{x}-\mathbf{y}\|\mathrm{d}\gamma(\mathbf{x},\mathbf{y})$$

$$= \sup_{Lip(f)\leq 1}\left\{\int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x}|w)\mathrm{d}\mathbf{x} - \int_{\mathcal{X}} f(\mathbf{y})p_G(\mathbf{y}|w)\mathrm{d}\mathbf{y}\right\}$$

- $P(\mathrm{x}|\omega)$ is the distribution of windows in Partition $\omega$
- $P_G(\mathrm{y}|\omega)$ is the distribution of reconstructed windows in Partition $\omega$
- $\gamma$ represents the matches

# Training

$$W^1[P(\mathbf{x}|w)\|P_G(\mathbf{y}|w)] = \inf_{\gamma\in\Gamma_w}\int_{\mathcal{X}\times\mathcal{X}}\|\mathbf{x}-\mathbf{y}\|\mathrm{d}\gamma(\mathbf{x},\mathbf{y})$$

$$= \sup_{Lip(f)\leq 1}\left\{\int_{\mathcal{X}} f(\mathbf{x})p(\mathbf{x}|w)\mathrm{d}\mathbf{x} - \int_{\mathcal{X}} f(\mathbf{y})p_G(\mathbf{y}|w)\mathrm{d}\mathbf{y}\right\}$$

- We train another network $D(\mathrm{x})$ to find the optimal $f$ above, with a penalty on the gradient norm for random samples (WGAN-GP[NIPS2017]).
- Decrease the size of each partition during training.
- We complete an adversarial training algorithm of VAE.

# Review

Use adversarial training

Which

Choose another posterior which is not diagonal

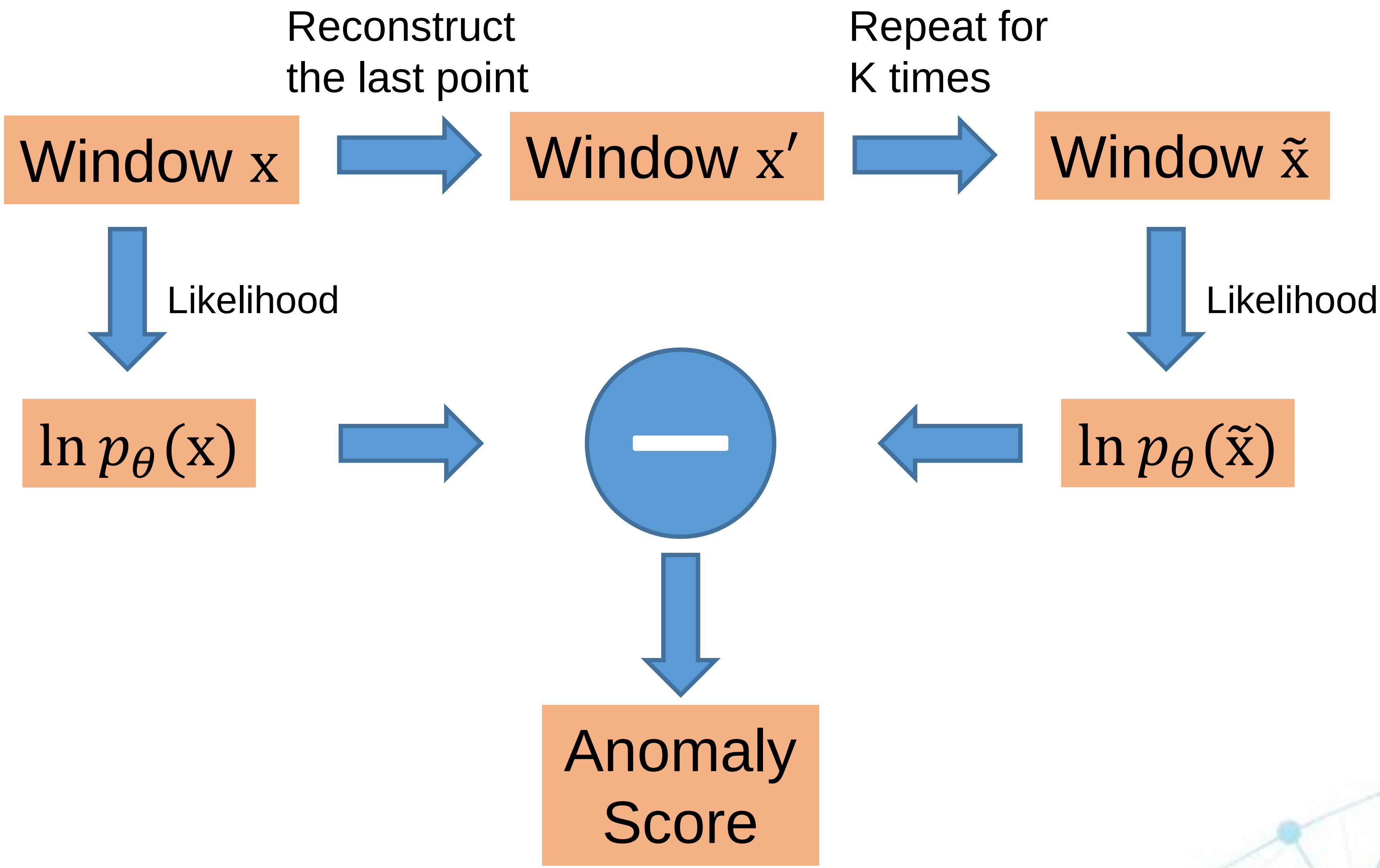
1-Wasserstein Distance

degenerate

Exponential posterior distribution

Robust metric

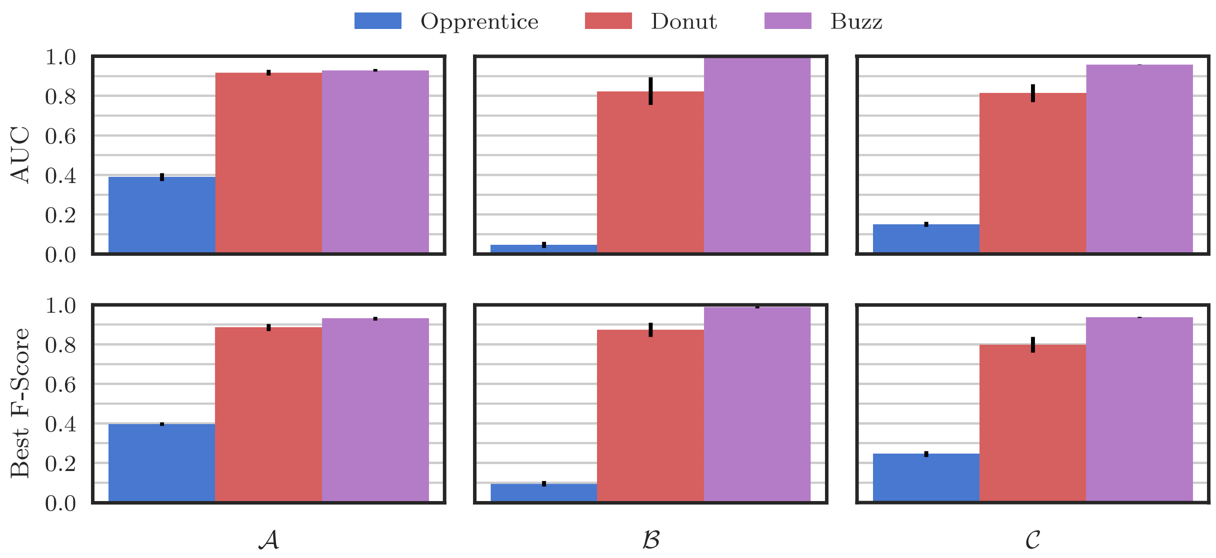
# Detection



Reconstruct
the last point
Repeat for
K times
Window $\mathrm{x}$
Window $\mathrm{x}'$
Window $\tilde{\mathrm{x}}$
Likelihood
Likelihood
$\ln p_\theta(\mathrm{x})$
$\ln p_\theta(\tilde{\mathrm{x}})$
Anomaly
Score

- Background
- Challenges
- Ideas
- Experiments

# Experiments



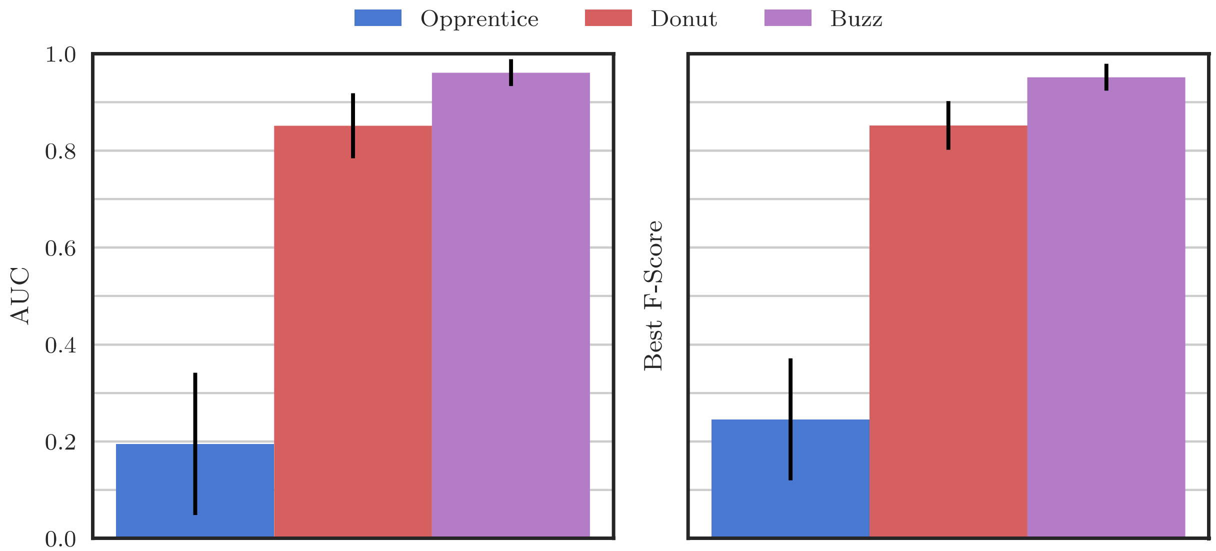




Fig16: Performance

# Experiments

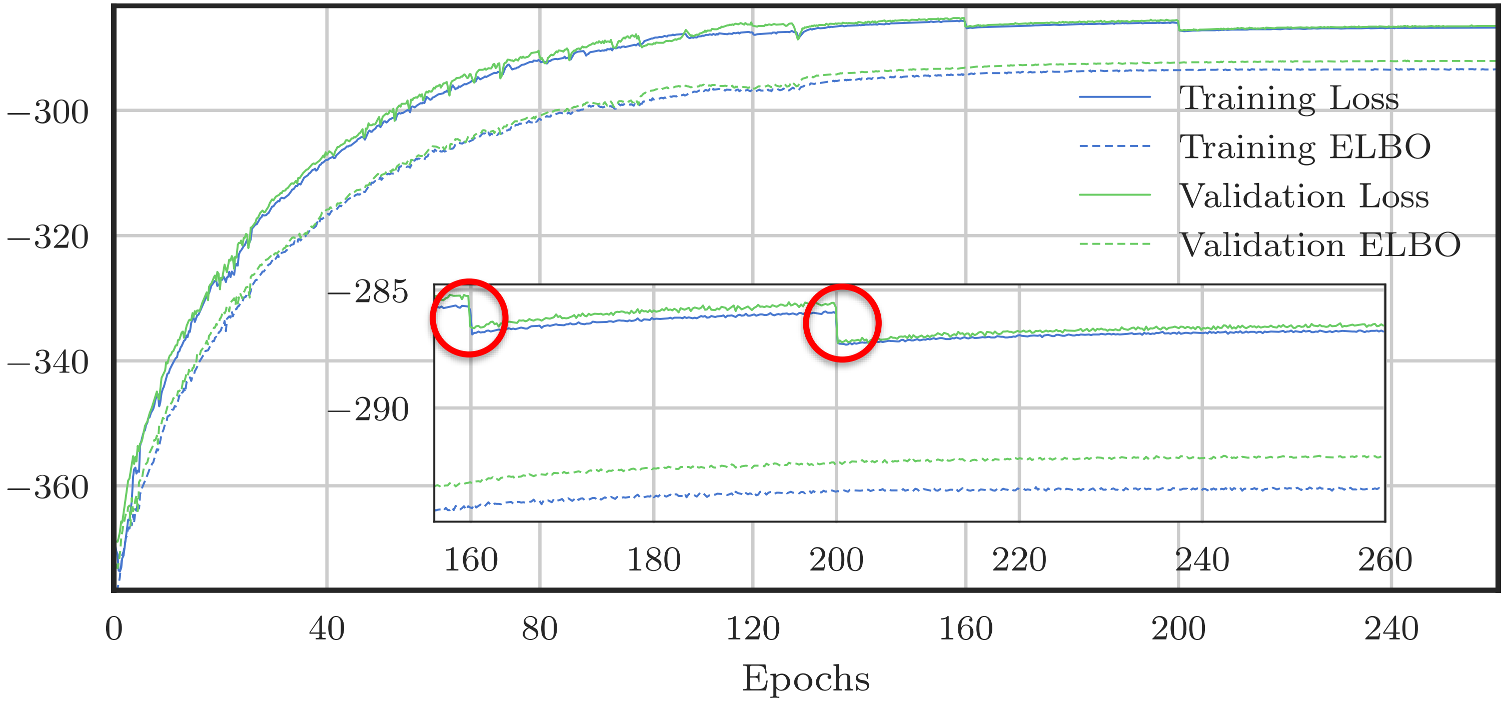



Fig17: Loss and ELBO

The fact that ELBO increases during the training, indicates that our model maximizes the ELBO indeed.

# Conclusion

- The first unsupervised anomaly detection algorithm via deep generative model on intricate KPIs

- The first adversarial training method for VAE, based on partitions analysis

- Our deduction build the bridge between VAE and Wasserstein Distance

# Thank you

# Q&A