# Anomaly Detection in the Open World:
# Normality Shift Detection, Explanation, and Adaptation

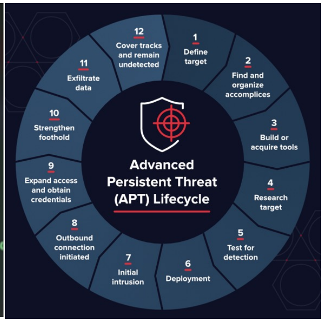
**Dongqi Han**, Zhiliang Wang, Wenqi Chen, Kai Wang, Rui Yu, Su Wang,

Han Zhang, Zhihua Wan, Minghui Jin, Jiahai Yang, Xingang Shi, and Xia Yin

# Anomaly Detection for Network Security

**Cyber crimes** are becoming more professional and coordinated

- Skilled cyber attackers can **bypass** approximately all the defense systems



# Anomaly Detection for Network Security

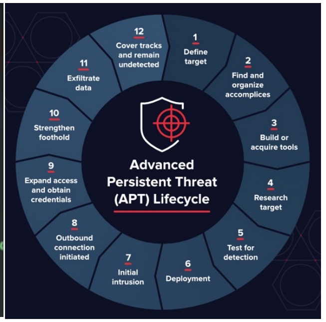
**Cyber crimes** are becoming more professional and coordinated

- Skilled cyber attackers can **bypass** approximately all the defense systems

**Anomaly Detection** has been widely used in diverse network security applications

- Learning **without knowledge of anomalies**
- Ability to detect **unforeseen threats**






# Anomaly Detection for Network Security

**Cyber crimes** are becoming more professional and coordinated

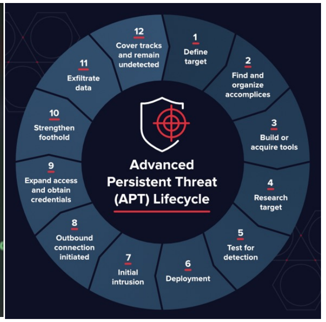
- Skilled cyber attackers can **bypass** approximately all the defense systems

**Anomaly Detection** has been widely used in diverse network security applications

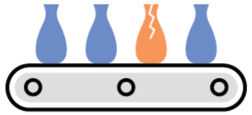
- Learning **without knowledge of anomalies**
- Ability to detect **unforeseen threats**

**Deep Learning** has shown a great potential to build network security applications

- Learn better **nonlinear and hierarchical** features
- Capture **complex** and **high-dimensional** structures



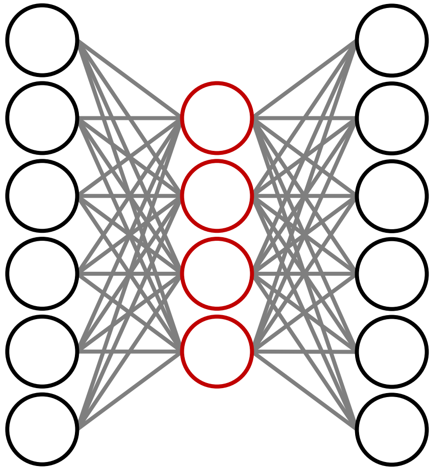


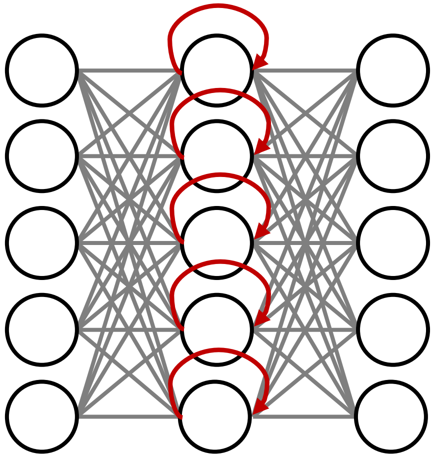
# Deep Learning based Anomaly Detection

***Zero-positive*** **Learning**

(trained with only normal data)

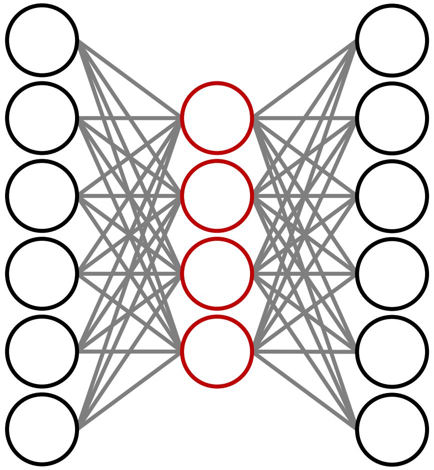


**Reconstruction-based**



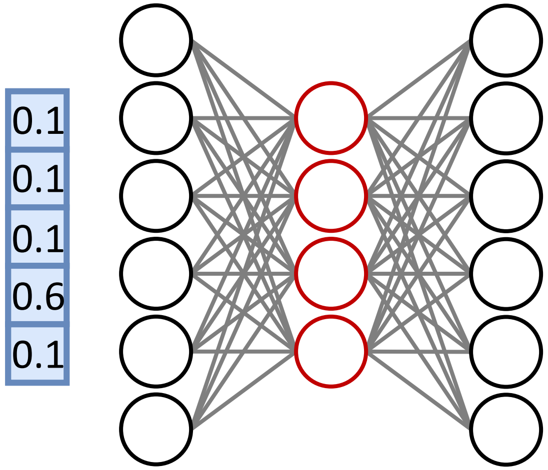
**Prediction-based**

# Deep Learning based Anomaly Detection



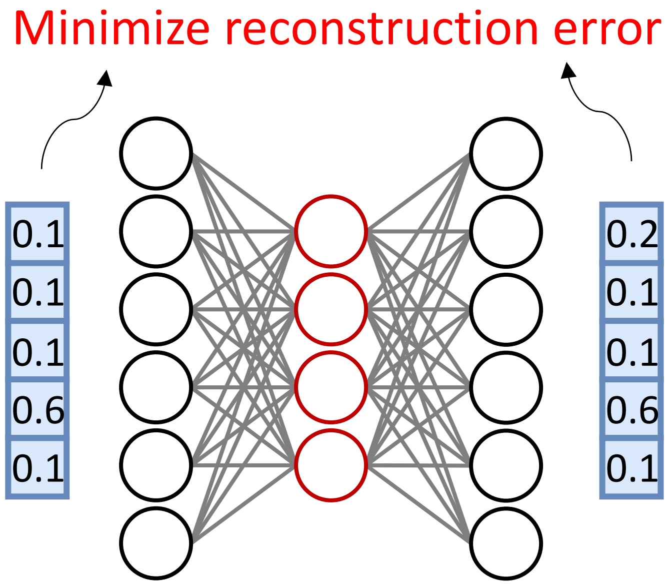
**Reconstruction-based**

# Deep Learning based Anomaly Detection



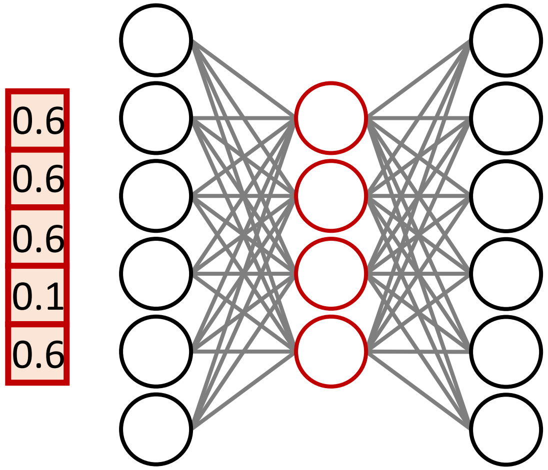

**Reconstruction-based Training**

# Deep Learning based Anomaly Detection



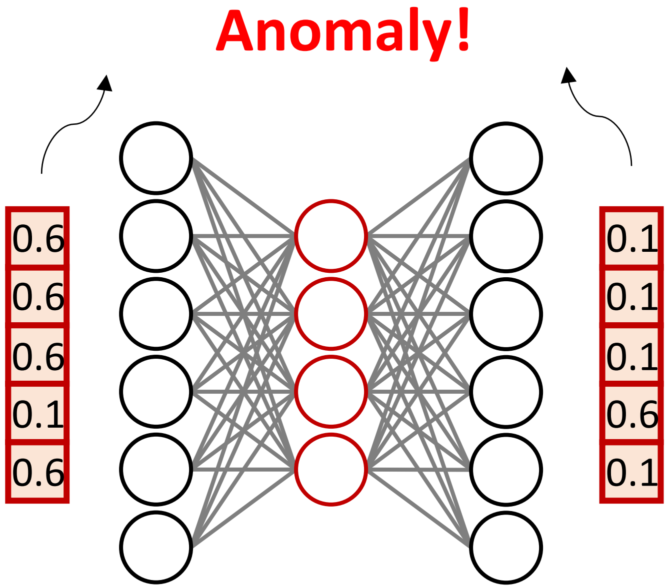

**Reconstruction-based Training**

# Deep Learning based Anomaly Detection




**Reconstruction-based Detection**

# Deep Learning based Anomaly Detection

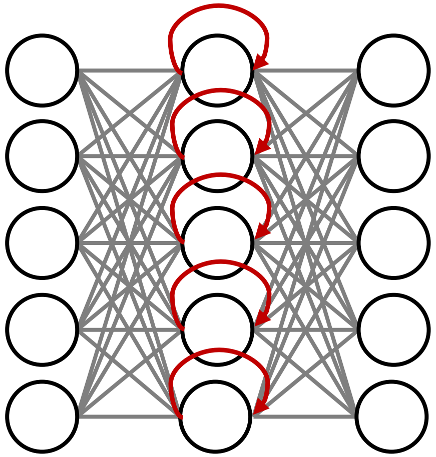



**Reconstruction-based Detection**

# Deep Learning based Anomaly Detection

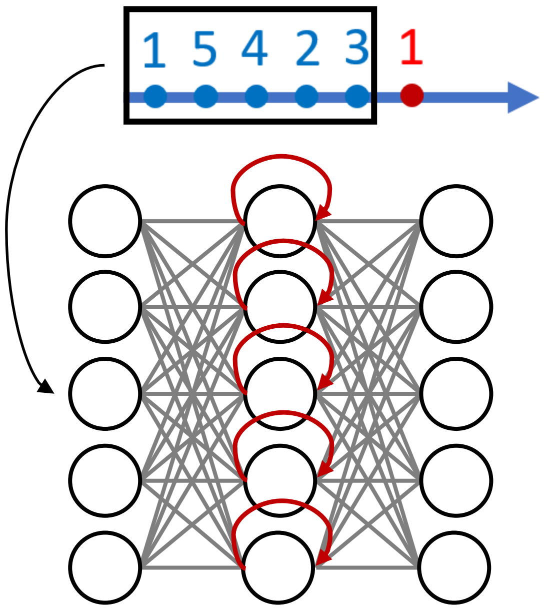


Reconstruction-based



Prediction-based

Reconstruction-based

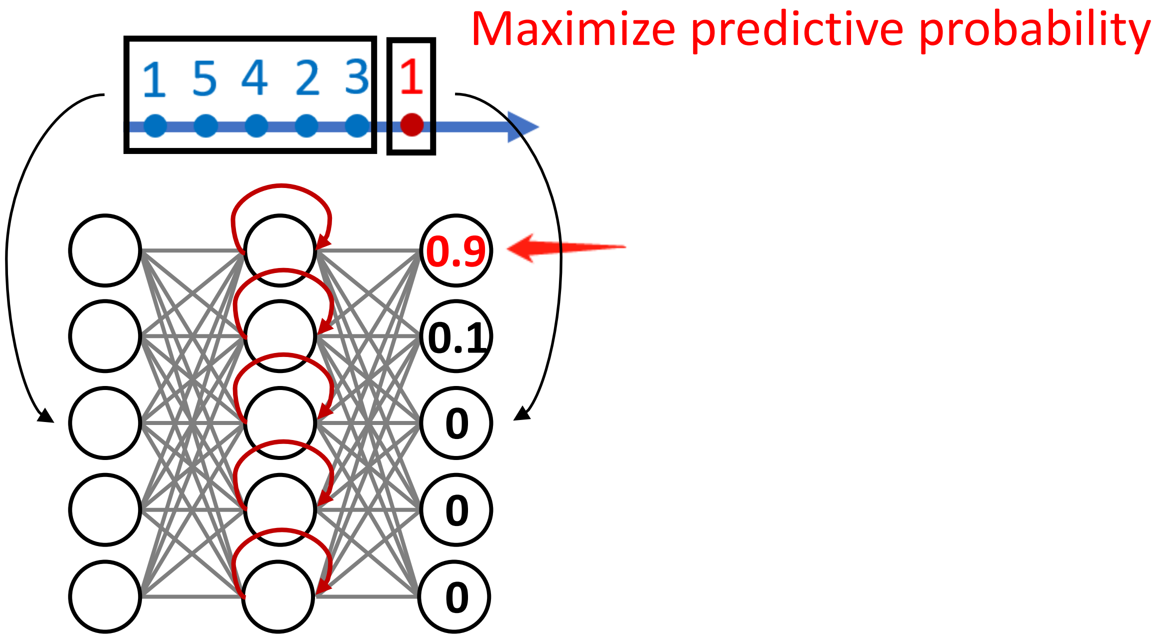



Prediction-based Training

# Deep Learning based Anomaly Detection



Reconstruction-based

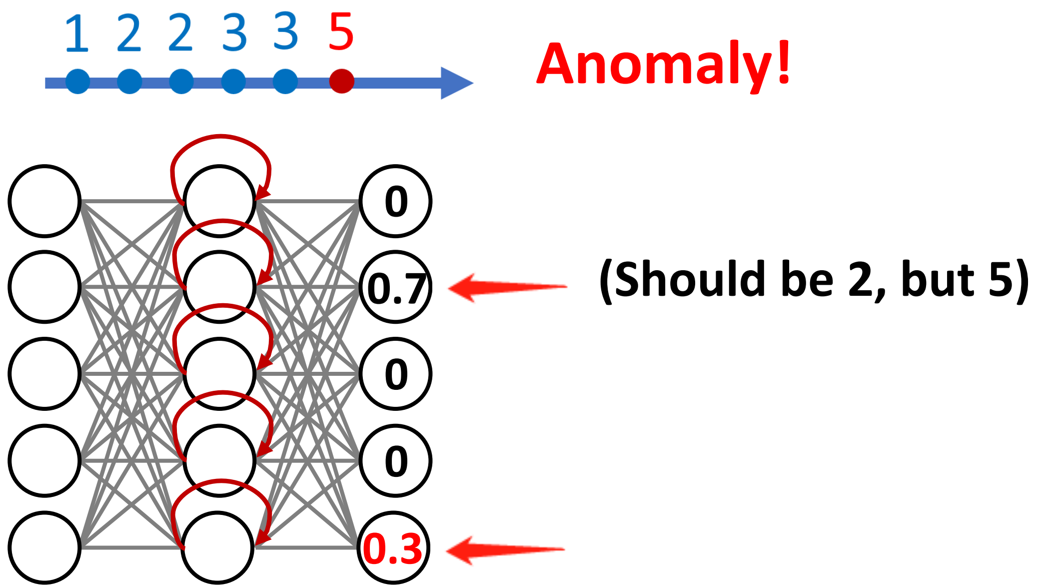



Prediction-based Training

# Deep Learning based Anomaly Detection



Reconstruction-based

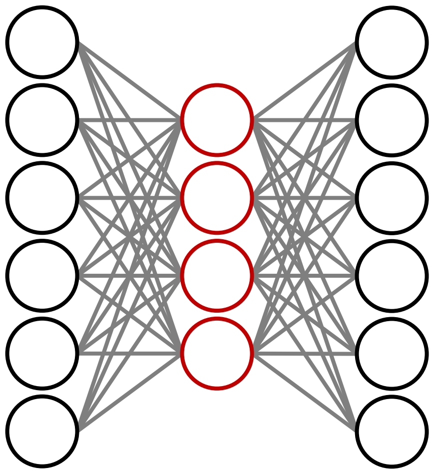



Prediction-based **Detection**

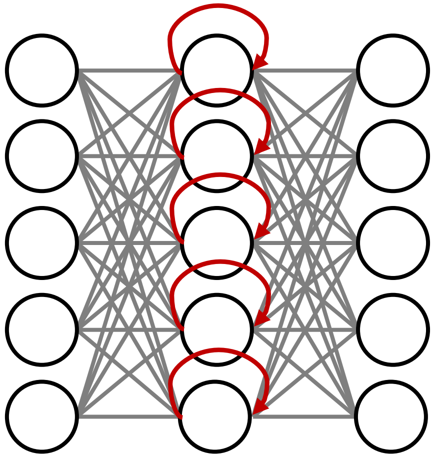
# Deep Learning based Anomaly Detection

***Zero-positive*** **Learning**

(trained with only normal data)



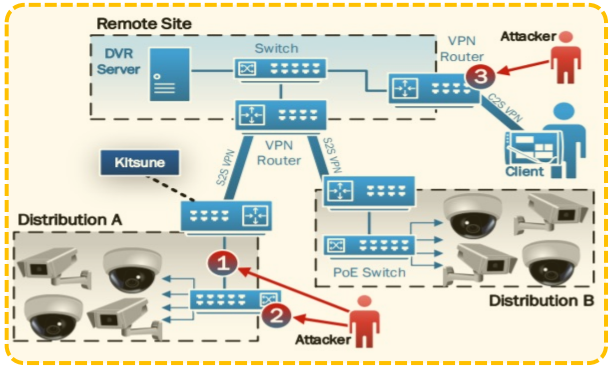
**Reconstruction-based**



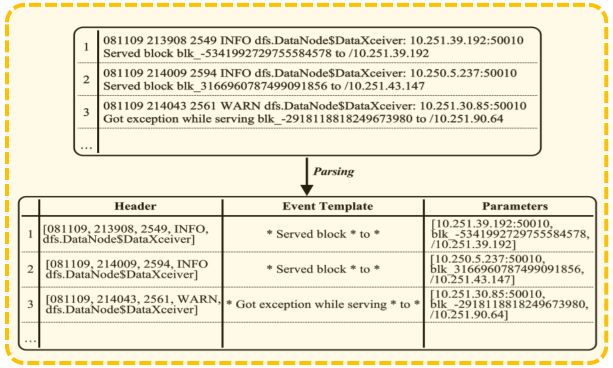
**Prediction-based**

# Anomaly Detection in Security Applications

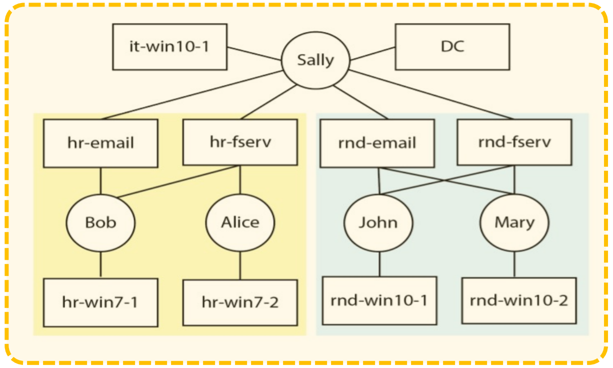
## Security Applications with Deep Learning based Anomaly Detection:



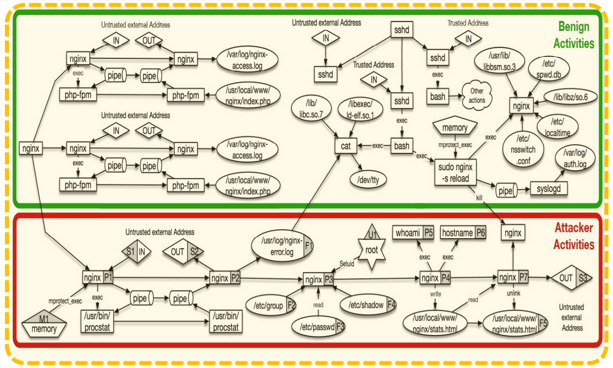

Network Intrusion Detection (*NDSS'18* , *CCS'23*)




Log Anomaly Detection (*CCS'17*, *CCS'19*)



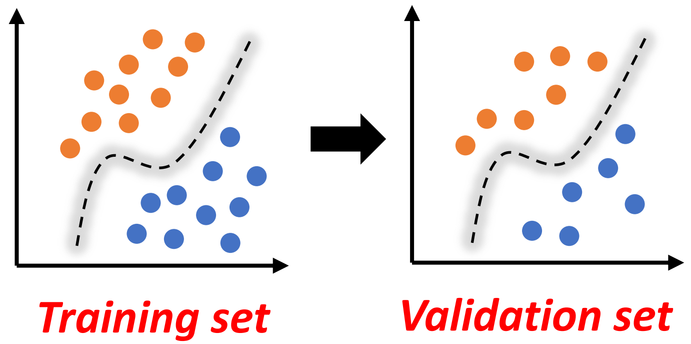

Lateral Movement Detection (*CCS'19*, *Security'23*)




Host-based Threat Detection (*NDSS'20*, *S&P'23*)

# Close World vs. Open World

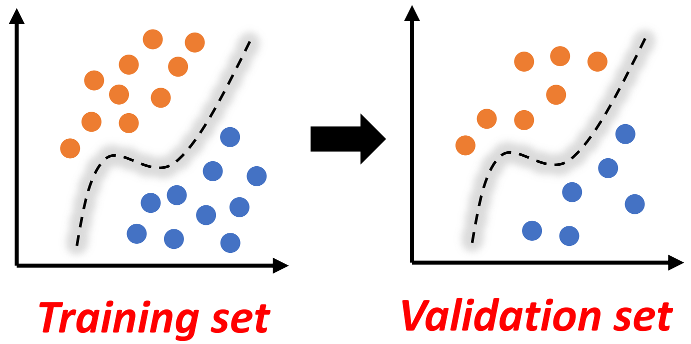
- The great success of machine/deep learning methods are based on the **Close-world** assumption— **testing data must be *similar* to the training data** (i.i.d. assumption)

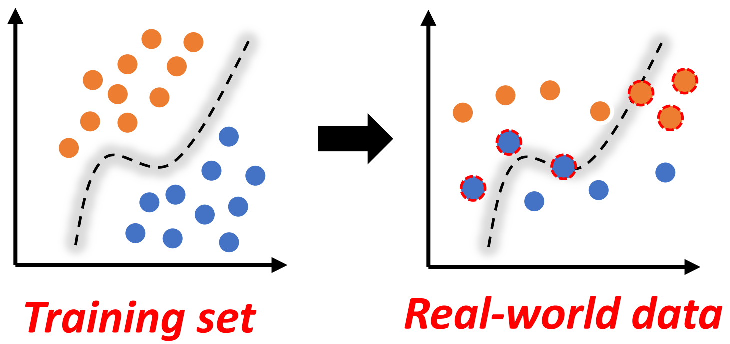


# Close World vs. Open World

- The great success of machine/deep learning methods are based on the **Close-world** assumption— **testing data must be *similar* to the training data** (i.i.d. assumption)

- However, in **Open-world** applications, the distribution of testing data can **change over time in unforeseen ways**
    - Concept Drift Problem
    - Example in security: the evolution of malware

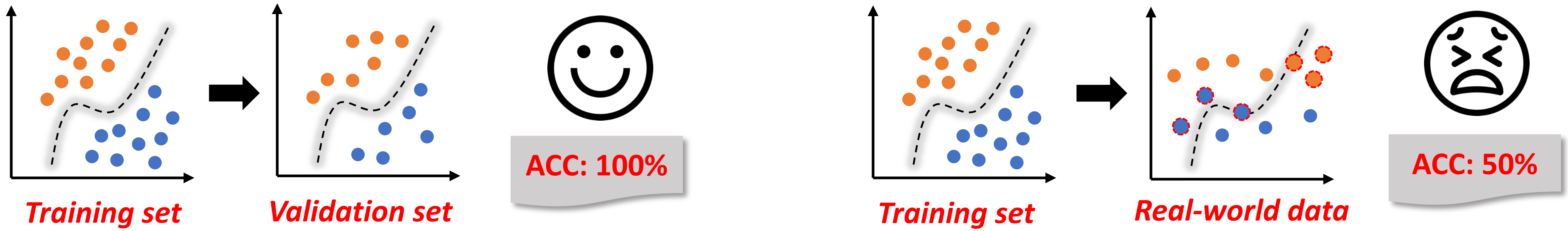


# Close World vs. Open World

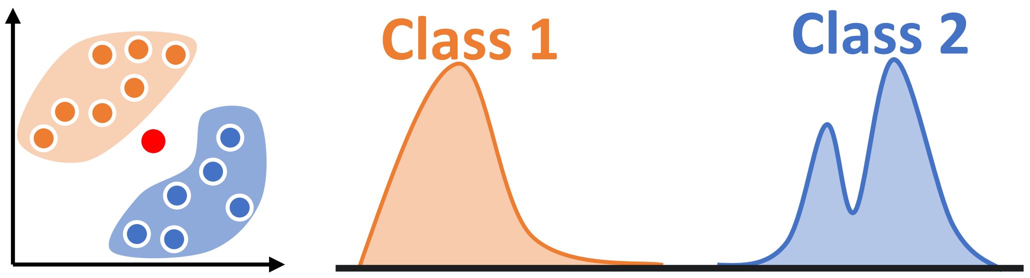
- The great success of machine/deep learning methods are based on the **Close-world** assumption— **testing data must be *similar* to the training data** (i.i.d. assumption)

- However, in **Open-world** applications, the distribution of testing data can **change over time in unforeseen ways**
  - Concept Drift Problem
  - Example in security: the evolution of malware
  - Model performance aging!



# Concept Drift vs. Normality Shift

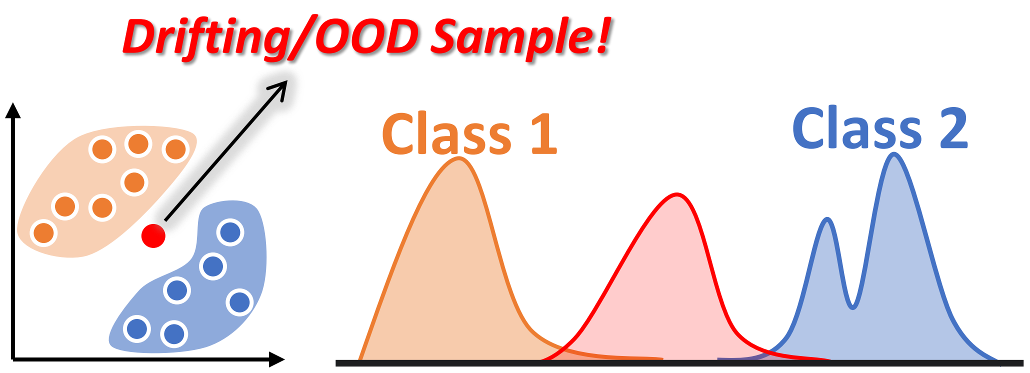
- **Concept drift** has been well-studied for supervised classification
  - **Security**: Transcend(*Usenix Sec'19*), CADE(*Usenix Sec'21*), Transcendent(*S&P'22*)




**Supervised Classification**

# Concept Drift vs. Normality Shift

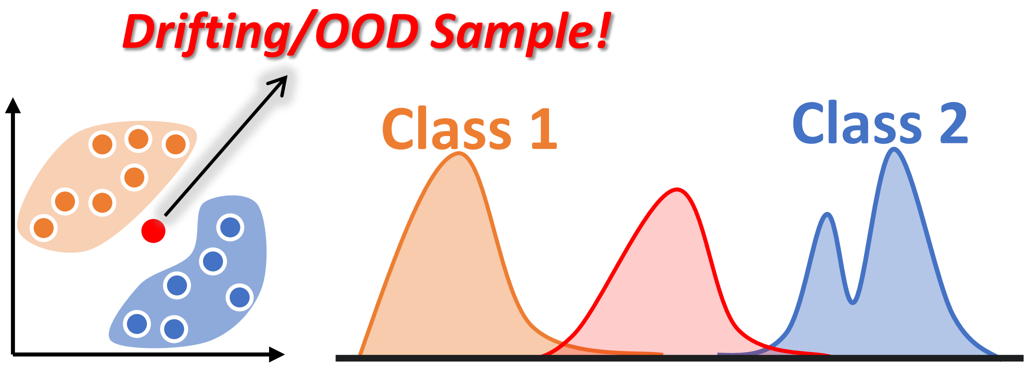
- **Concept drift** has been well-studied for supervised classification
  - **Security**: Transcend(*Usenix Sec'19*), CADE(*Usenix Sec'21*), Transcendent(*S&P'22*)
  - **Machine Learning**: Out-of-distribution (OOD) detection



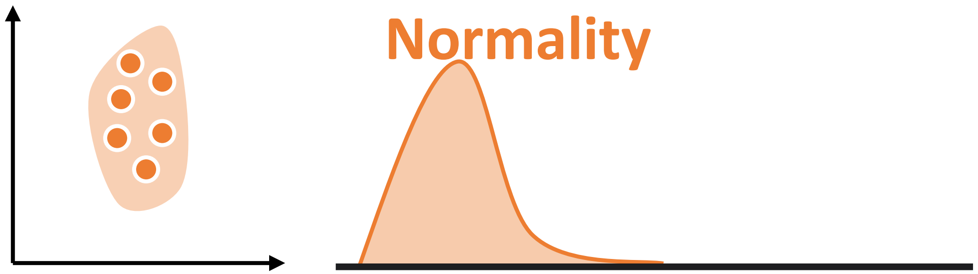

Supervised Classification

# Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - **Security**: Transcend(*Usenix Sec'19*), CADE(*Usenix Sec'21*), Transcendent(*S&P'22*)
  - **Machine Learning**: Out-of-distribution (OOD) detection
- **Anomaly detection** models are built upon purely normal data (normality)
  - Immune to the drift of malicious/abnormal behavior
  - More severe impact when the distribution of normality shifts
  - E.g., user behaviors and system themselves (patches, new devices)



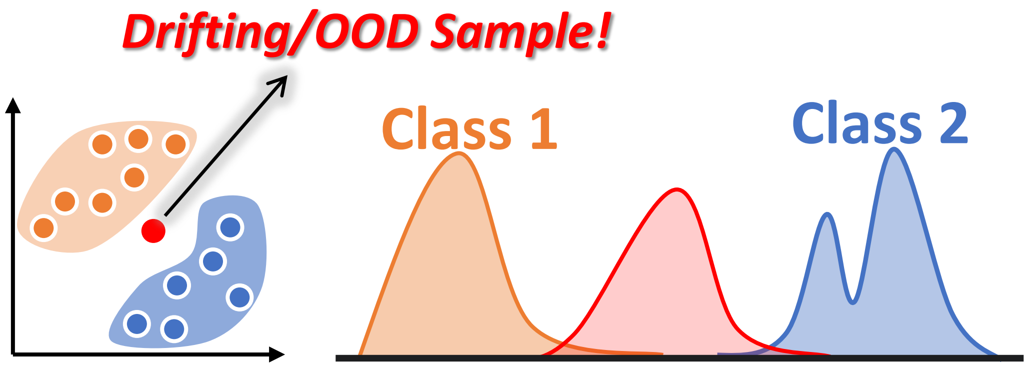

**Supervised Classification**



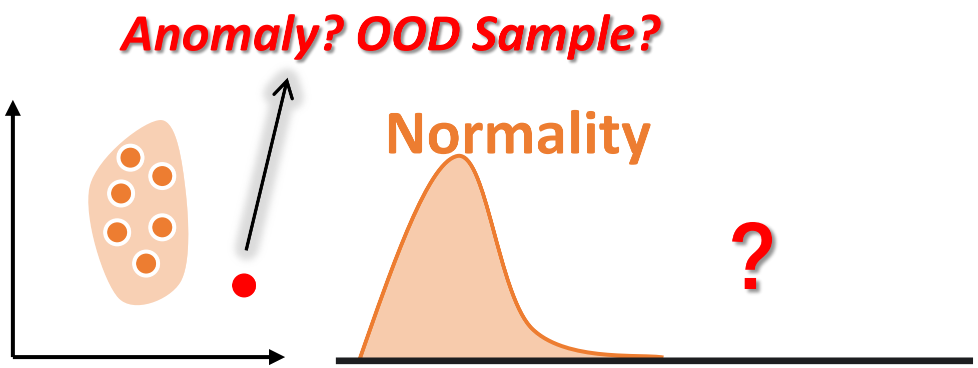

**Anomaly Detection**

# Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - **Security**: Transcend(*Usenix Sec'19*), CADE(*Usenix Sec'21*), Transcendent(*S&P'22*)
  - **Machine Learning**: Out-of-distribution (OOD) detection
- **Anomaly detection** models are built upon purely normal data (normality)
  - Immune to the drift of malicious/abnormal behavior
  - More severe impact when the distribution of normality shifts
  - E.g., user behaviors and system themselves (patches, new devices)




**Supervised Classification**



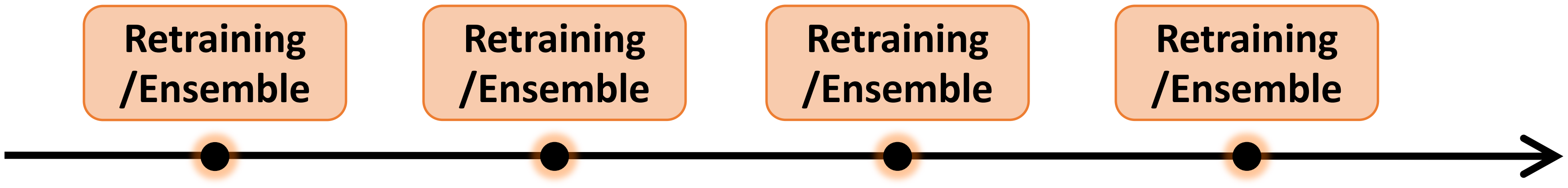

**Anomaly Detection**

# Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - **Security**: Transcend(*Usenix Sec'19*), CADE(*Usenix Sec'21*), Transcendent(*S&P'22*)
  - **Machine Learning**: Out-of-distribution (OOD) detection
- **Anomaly detection** models are built upon purely normal data (normality)
  - Immune to the drift of malicious/abnormal behavior
  - More severe impact when the distribution of normality shifts
  - E.g., user behaviors and system themselves (patches, new devices)

***Key Insight 1 – Without ground-truth label, a normality shift and real anomaly is not distinguishable for anomaly detection!***

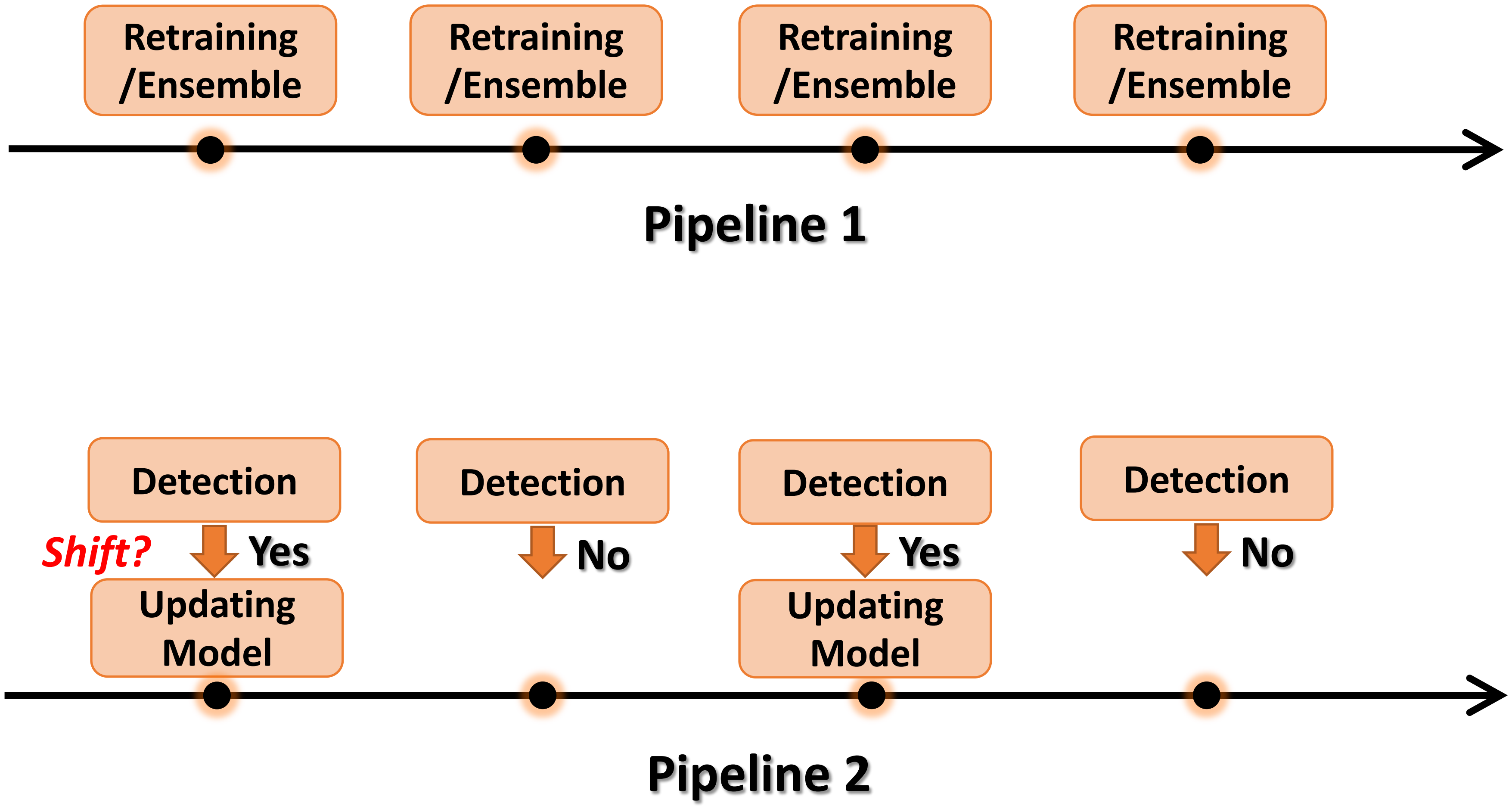
# Pipelines for Handling Shift




**Pipeline 1**

***Heavy Cost***
***Lack of Analysis***
***Delay update***

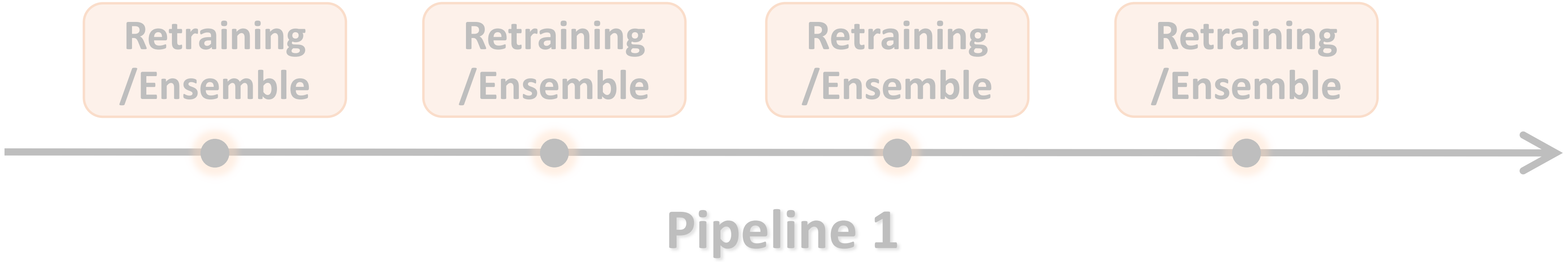
# Pipelines for Handling Shift




*Heavy Cost*
*Lack of Analysis*
*Delay update*

*Our Scope*

# Pipelines for Handling Shift




*Heavy Cost*
*Lack of Analysis*
*Delay update*

***Key Insight 2 – We need to decide <u>whether, when, and how shift</u> occurs before adapting models to the shift!***

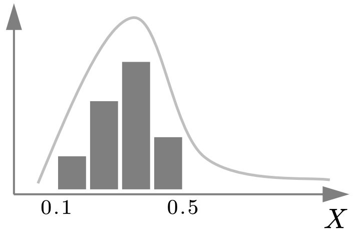
# Detecting Shift in Statistics

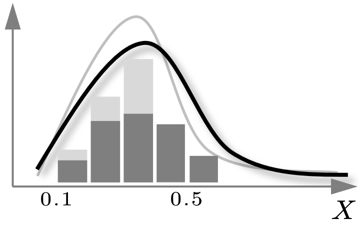
***Question: How to represent the distribution of normality?***

**Original**

**Shifted**

**Distribution of feature-space data**



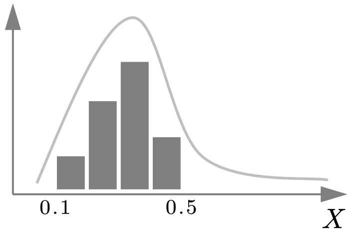

**1D**

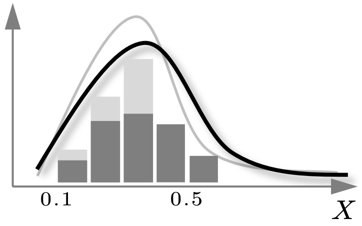
# Detecting Shift in Statistics

***Question: How to represent the distribution of normality?***

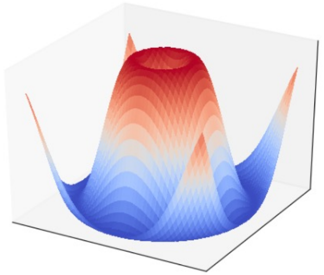
**Original** | **Shifted**

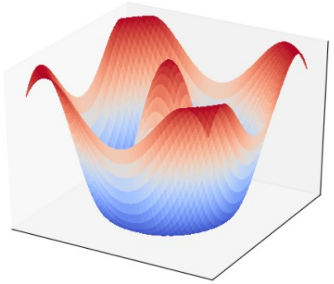
**Distribution of feature-space data**




**1D**





**2D**

**…**

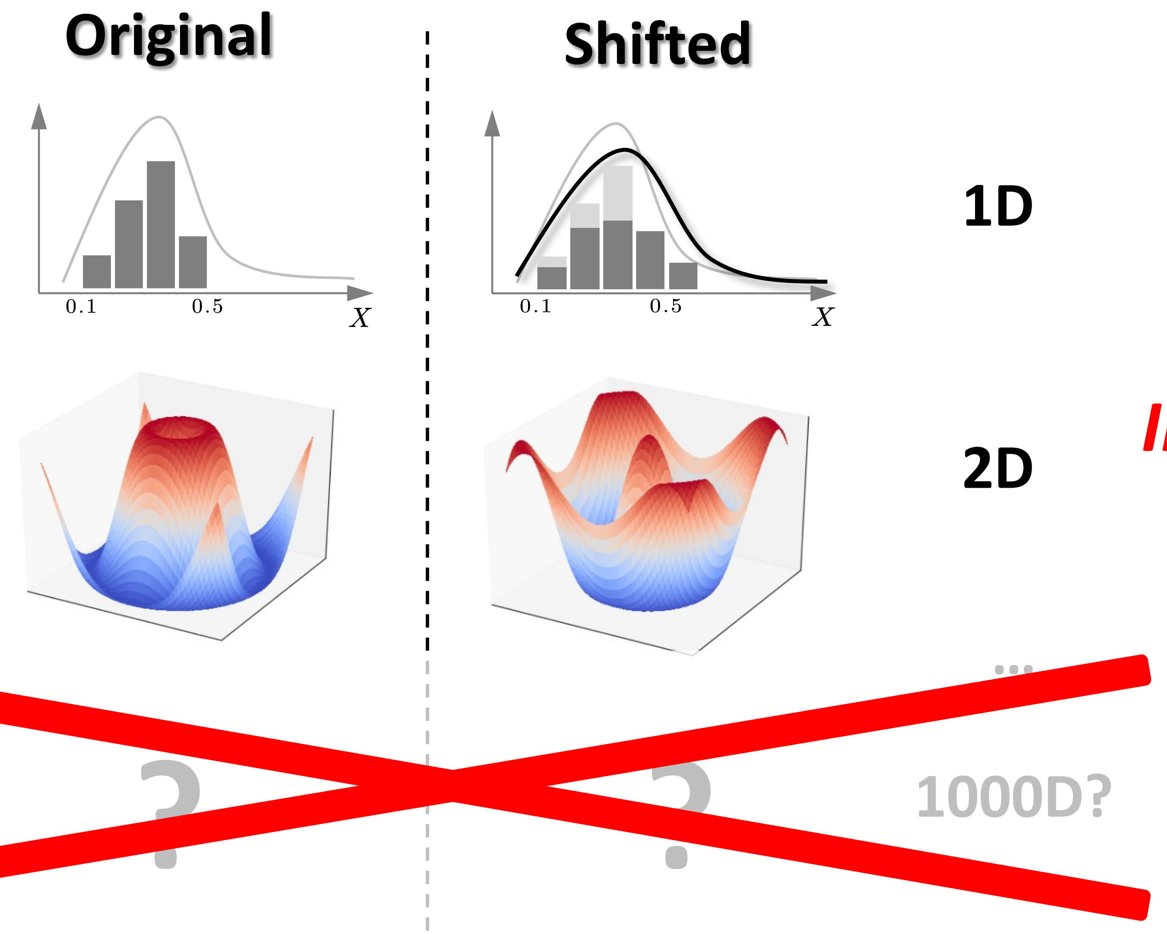
**?** | **?**

**1000D?**

# Detecting Shift in Statistics

***Question: How to represent the distribution of normality?***

**Distribution of feature-space data**



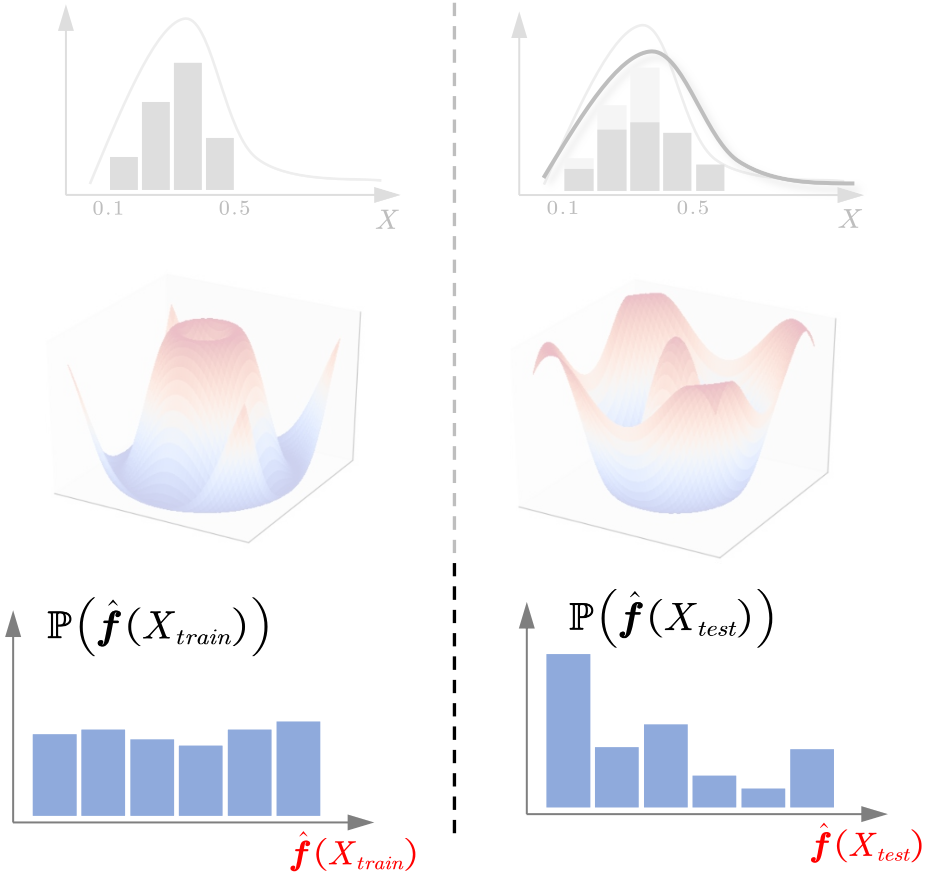

***Intractable for high-dimensional data!***

# Detecting Shift in Statistics

***Question: How to represent the distribution of normality?***

**Original** | **Shifted**

Distribution of feature-space data




1D

2D

*Intractable for high-dimensional data!*

**Distribution of model outputs**

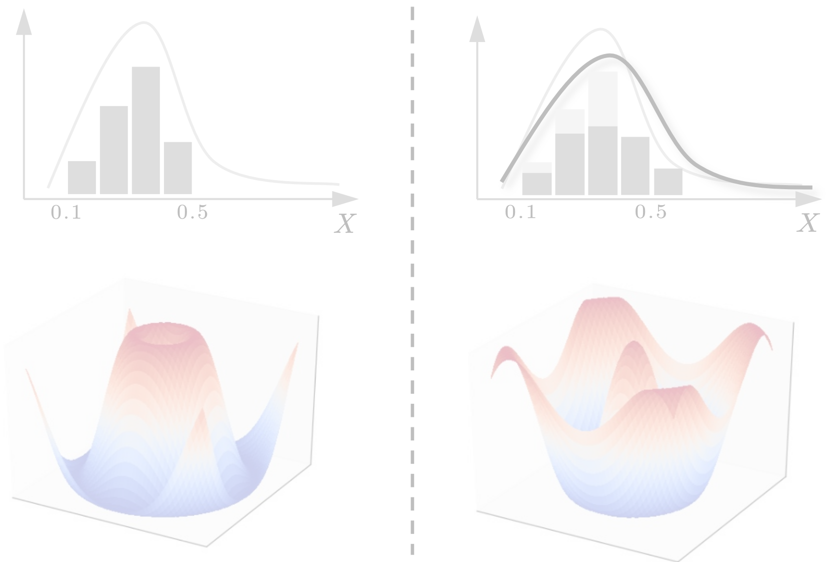
***Our Scope***

# Detecting Shift in Statistics

***Question: How to represent the distribution of normality?***

**Original** | **Shifted**

Distribution of feature-space data




1D

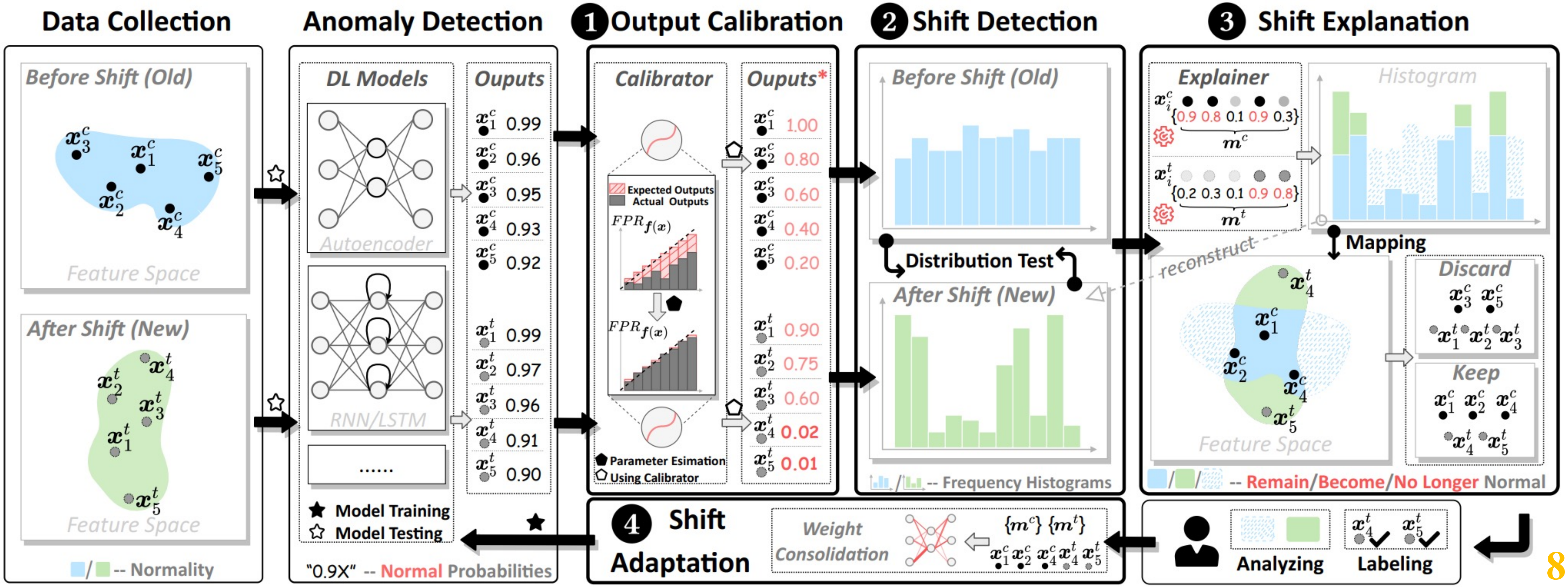
2D

*Intractable for high-dimensional data!*

***Key Insight 3 – Distribution of normality can be represented by the distribution of model outputs!***

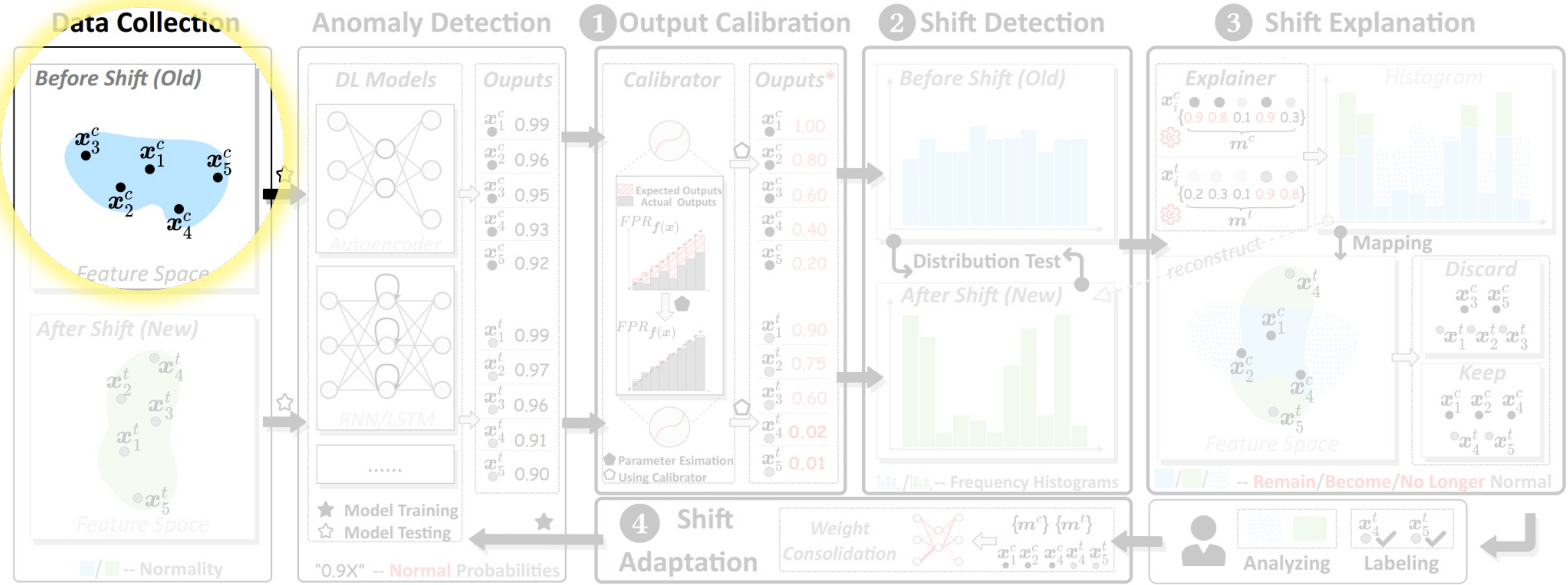
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



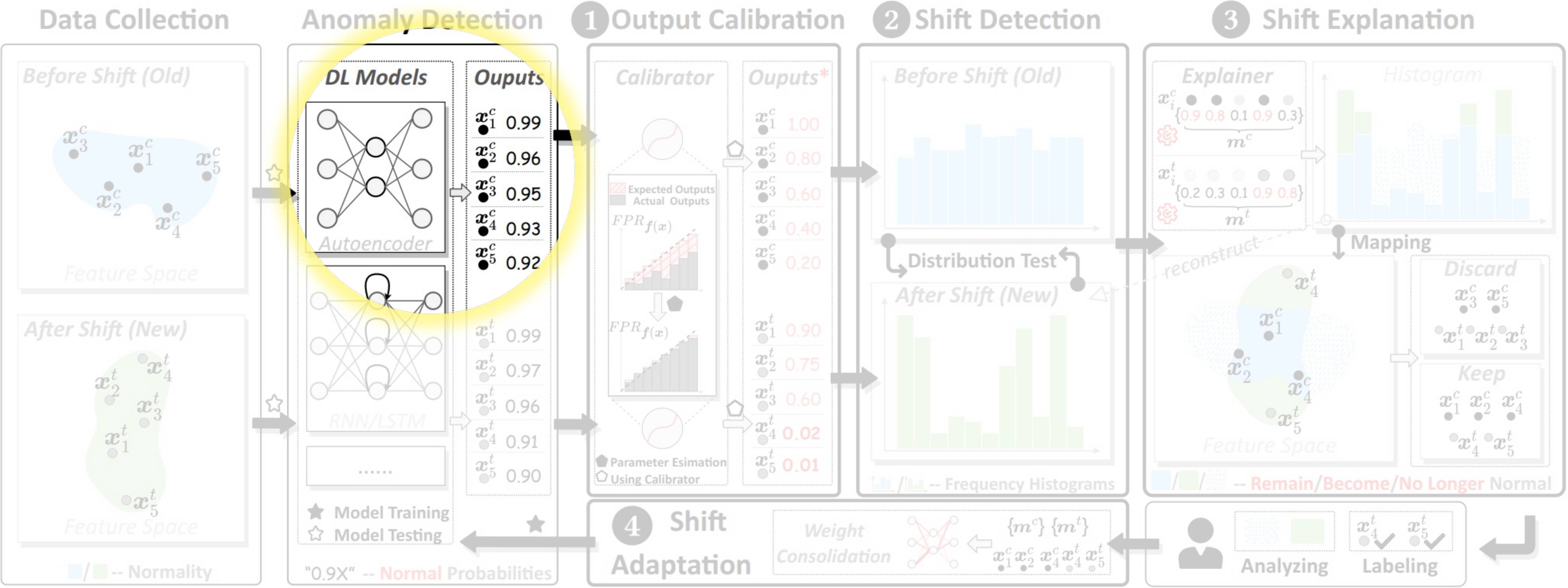
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



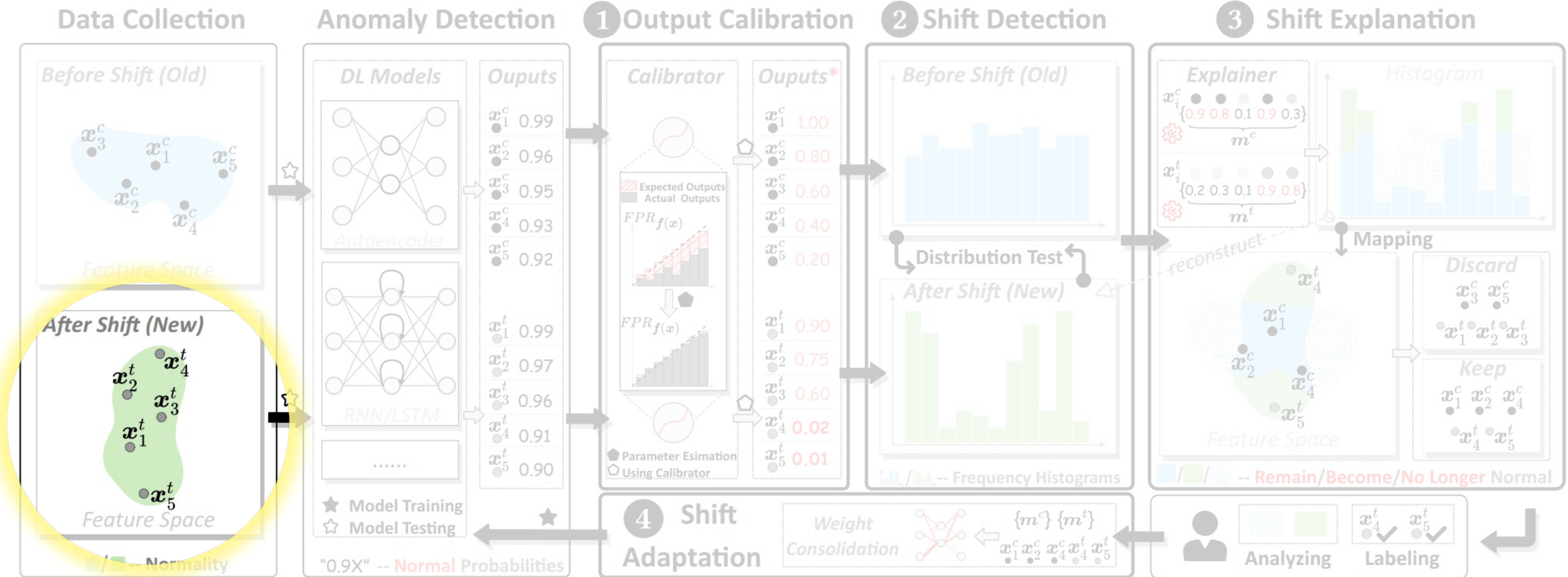
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and  Adapting to normality shift for DL-based anomaly detection.



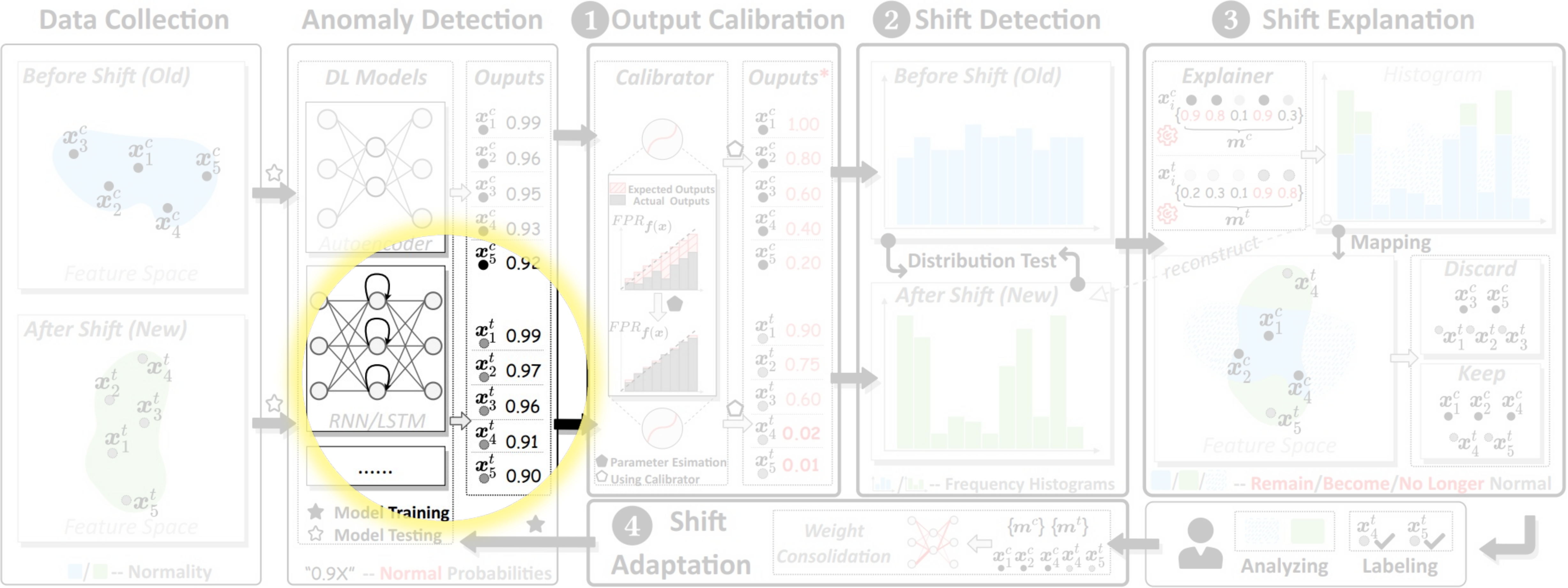
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



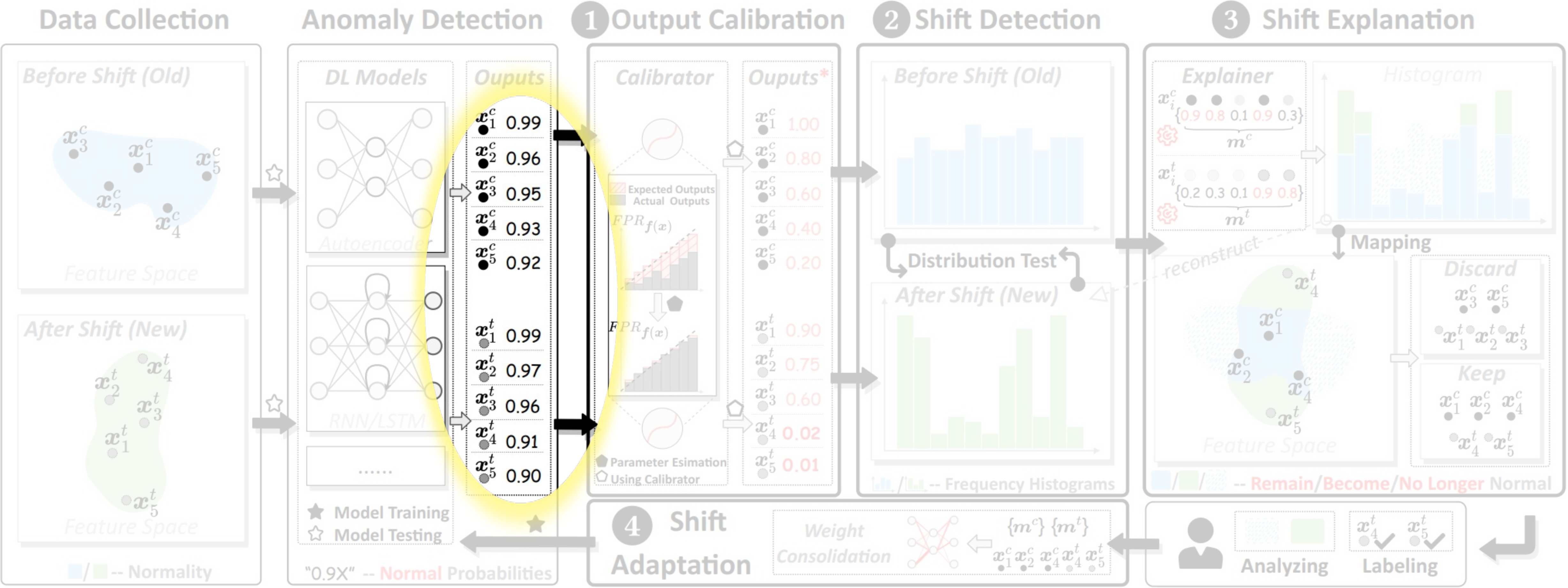
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



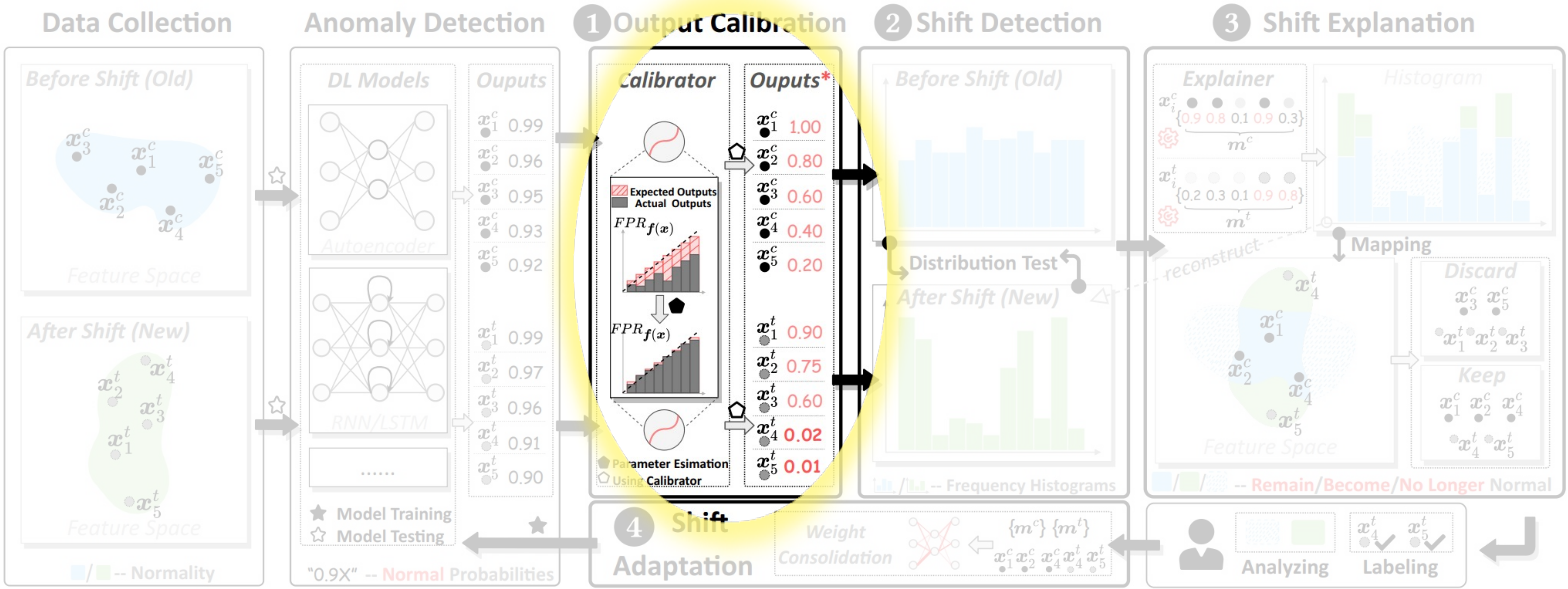
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and  Adapting to normality shift for DL-based anomaly detection.



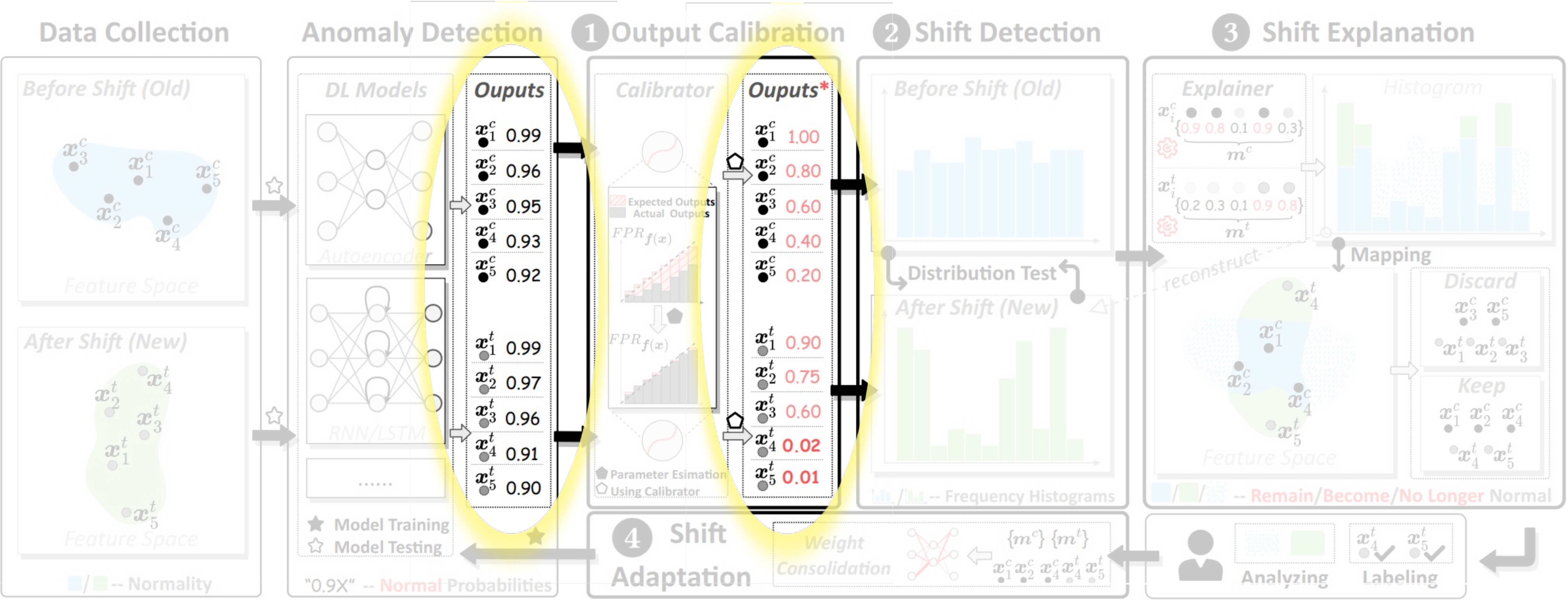
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



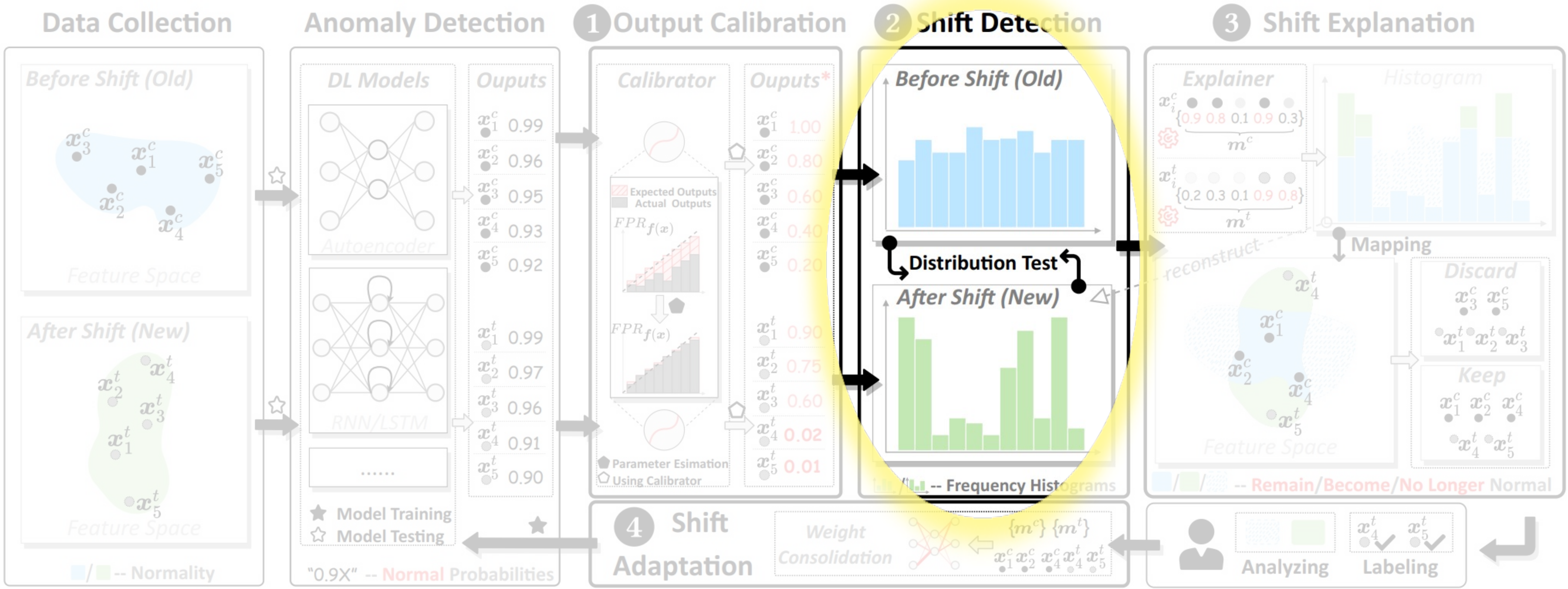
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and  Adapting to normality shift for DL-based anomaly detection.



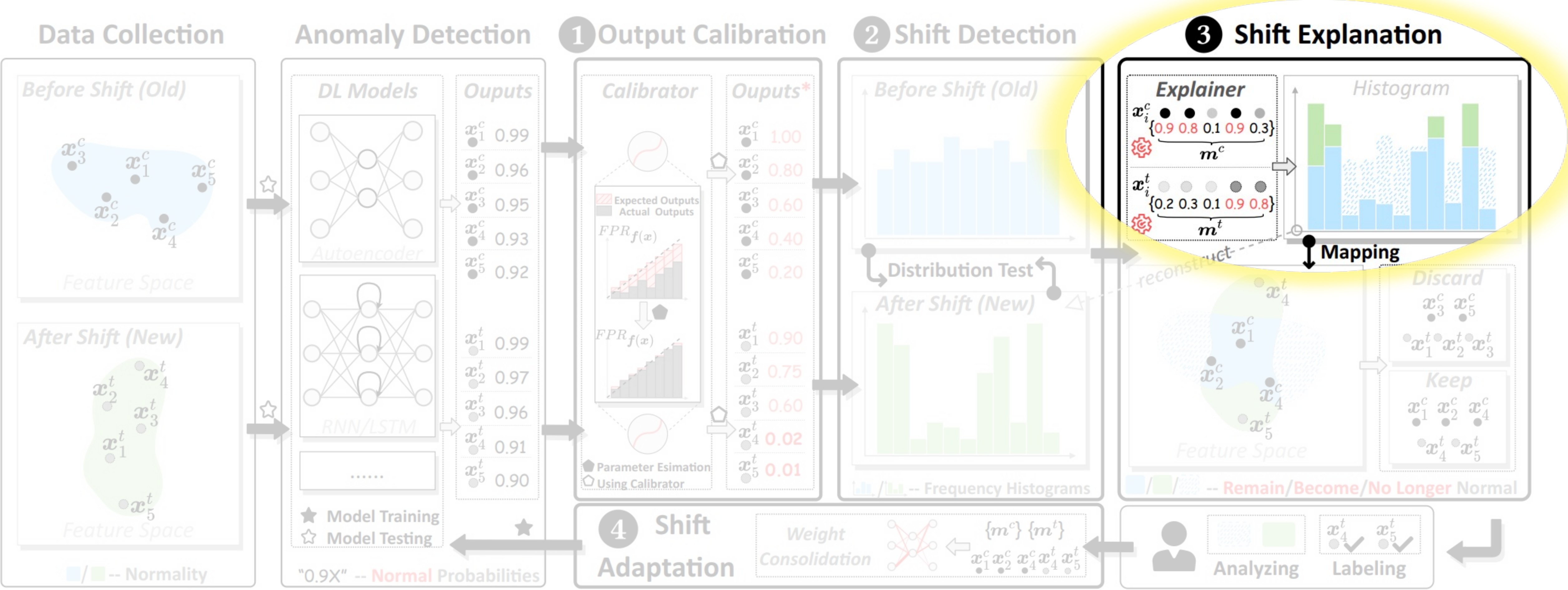
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



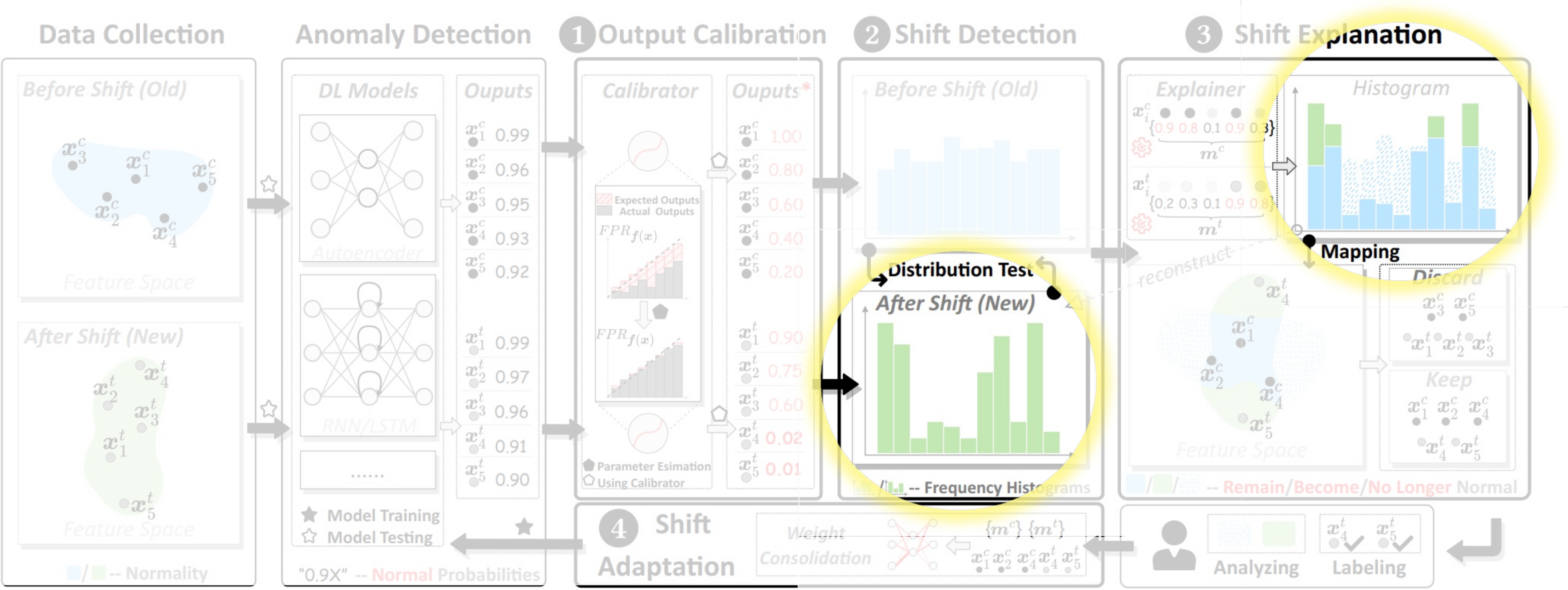
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and  Adapting to normality shift for DL-based anomaly detection.



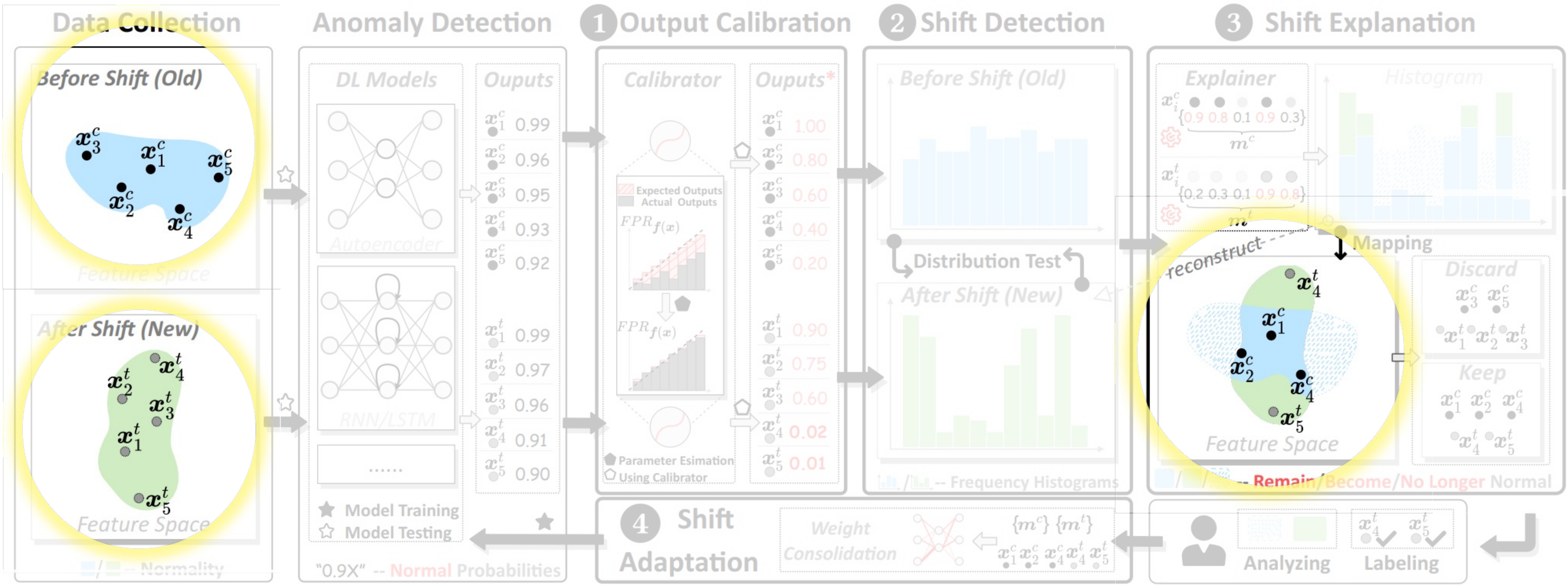
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



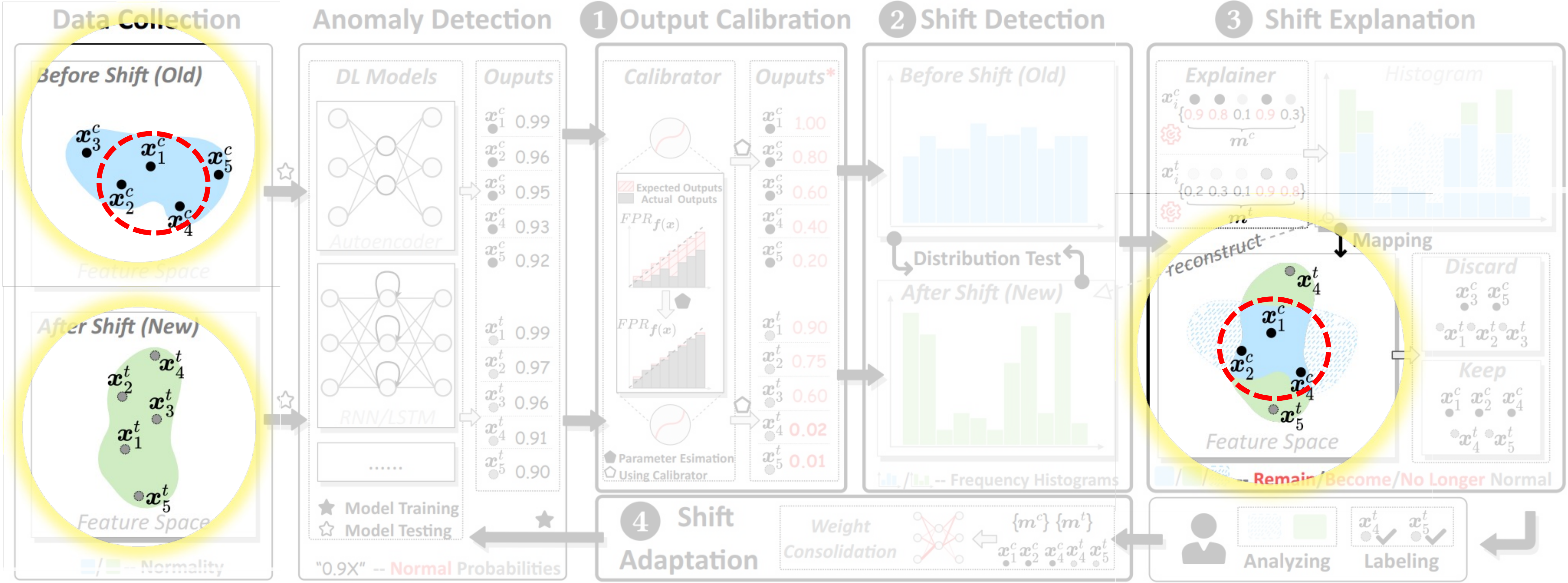
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



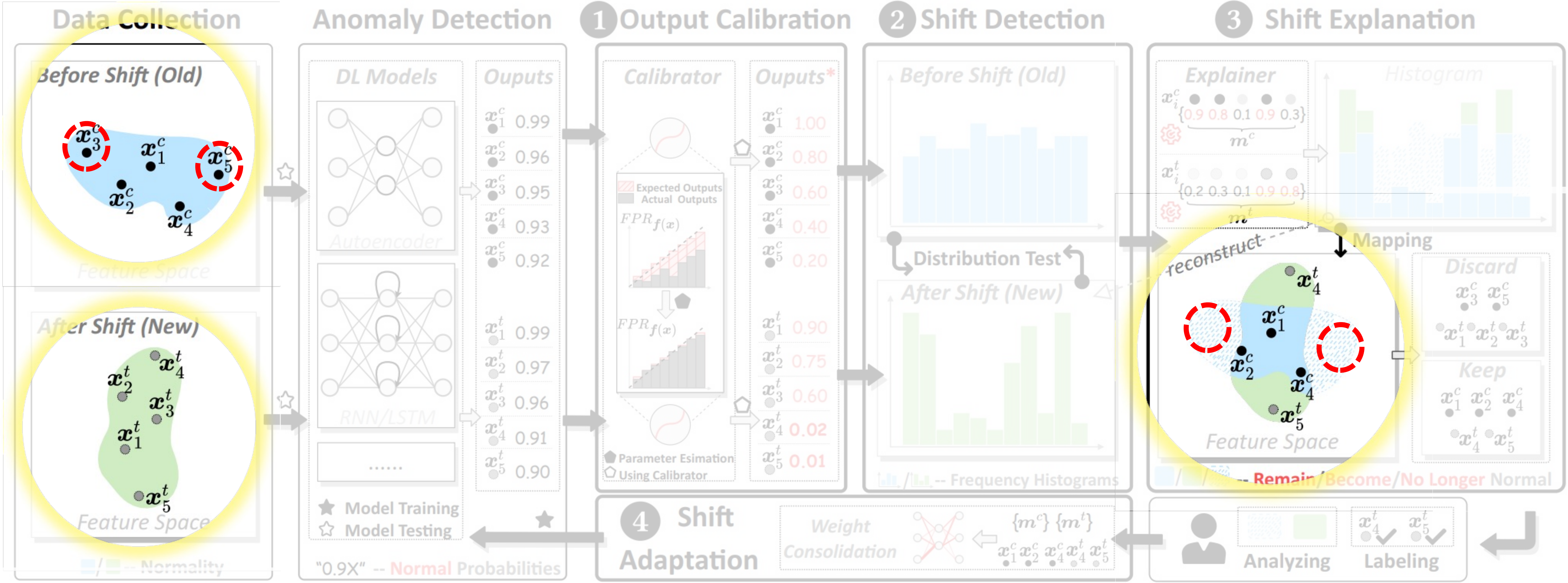
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



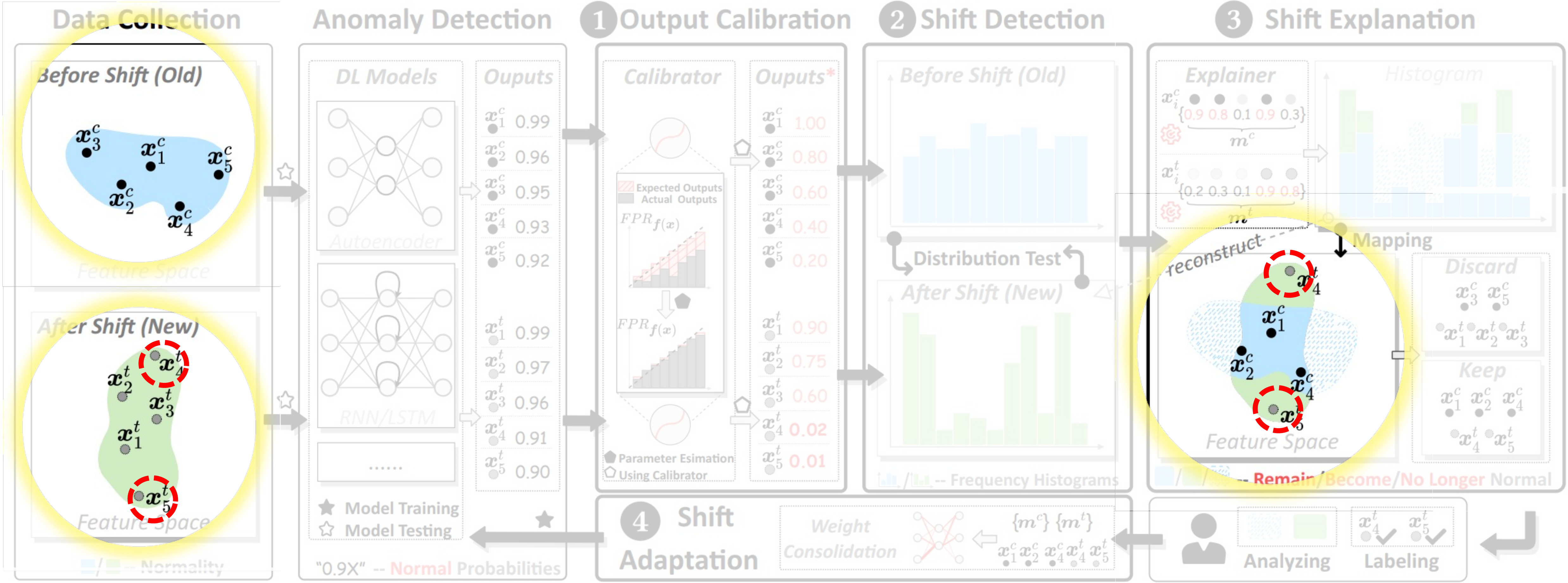
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



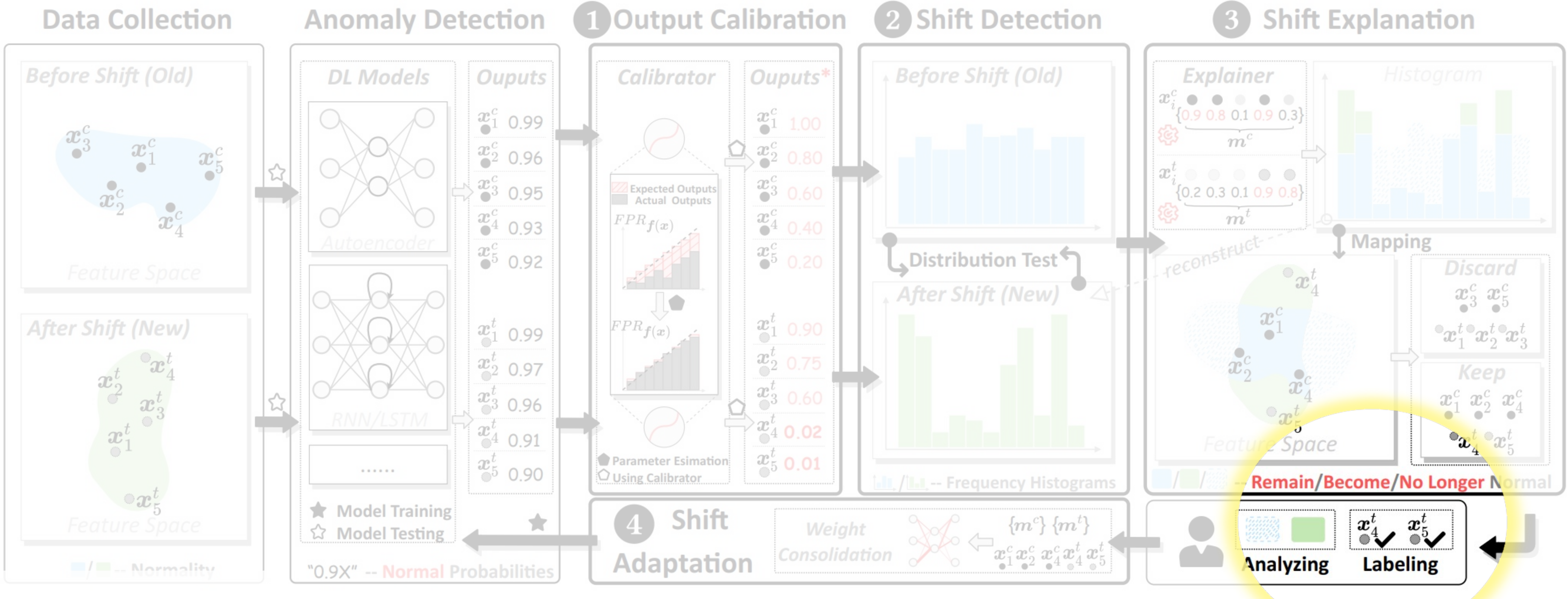
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



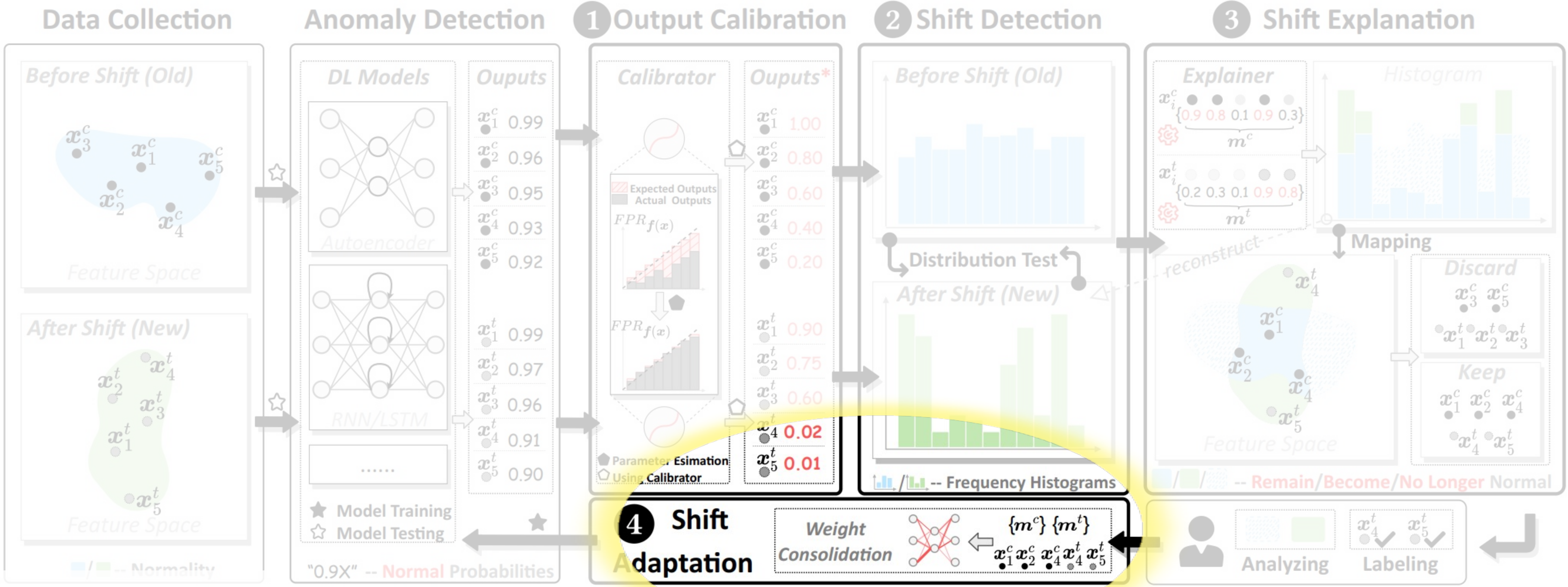
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and  Adapting to normality shift for DL-based anomaly detection.



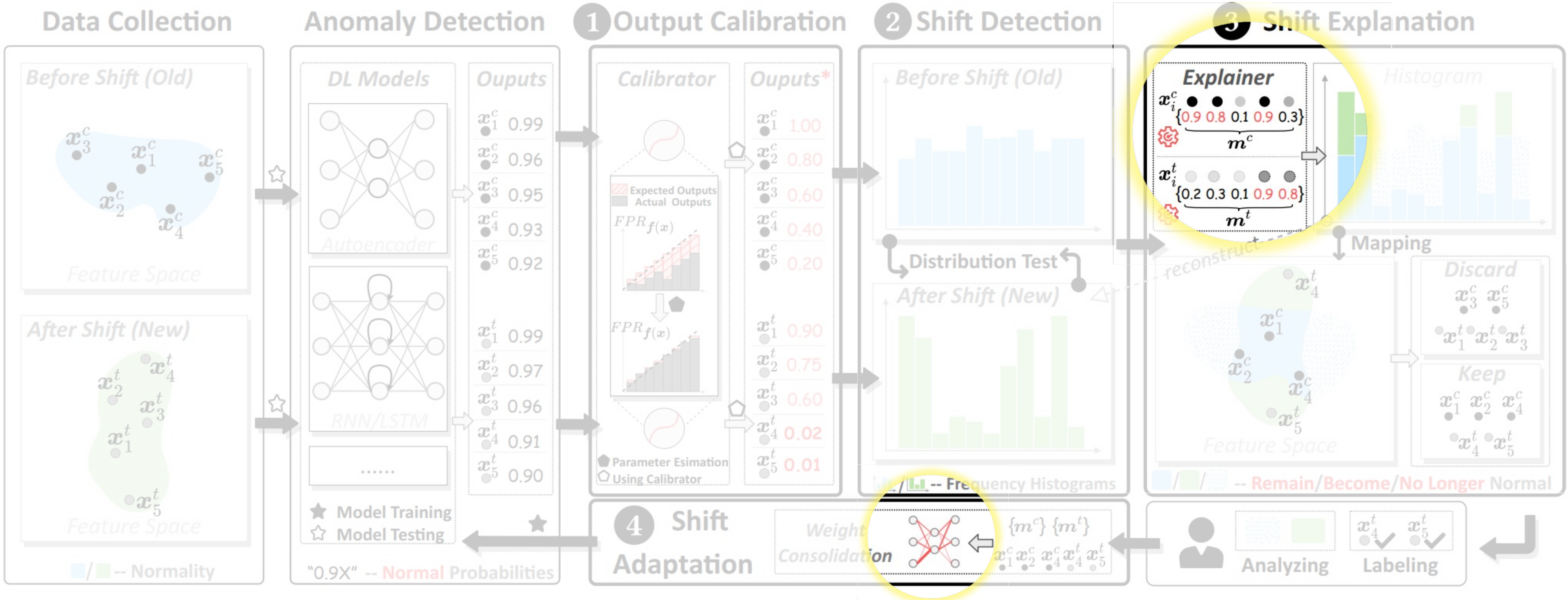
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



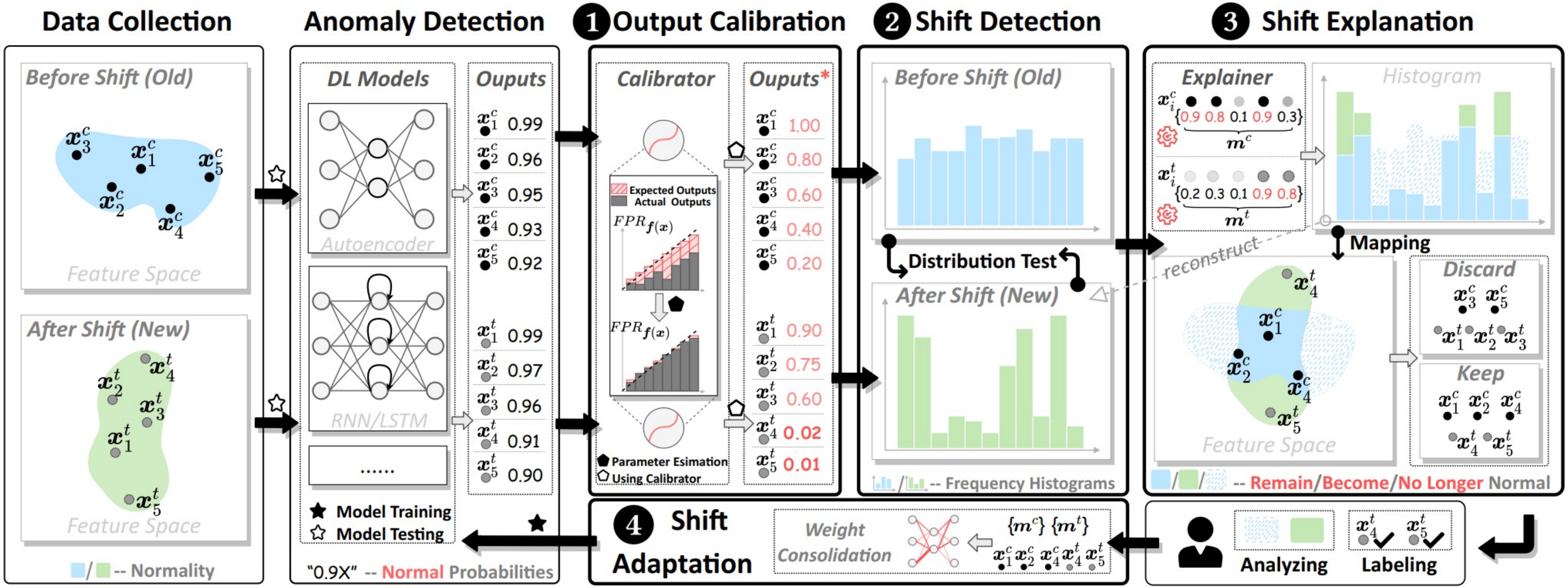
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



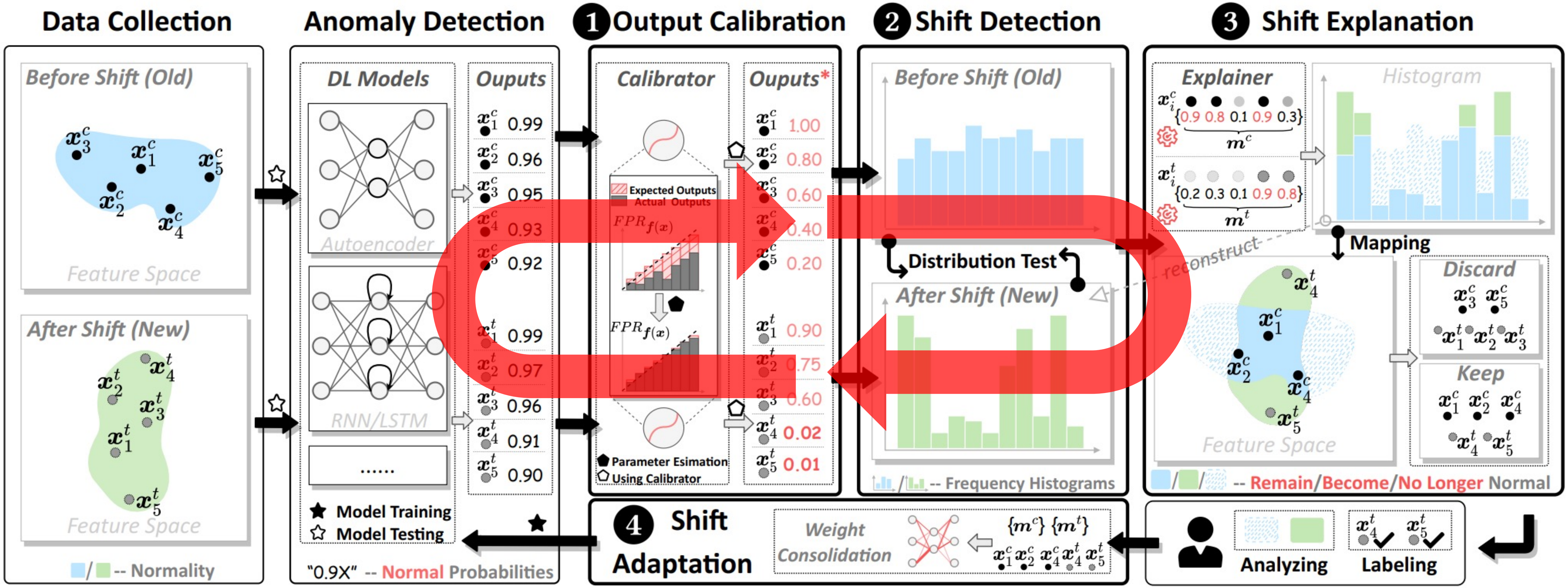
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



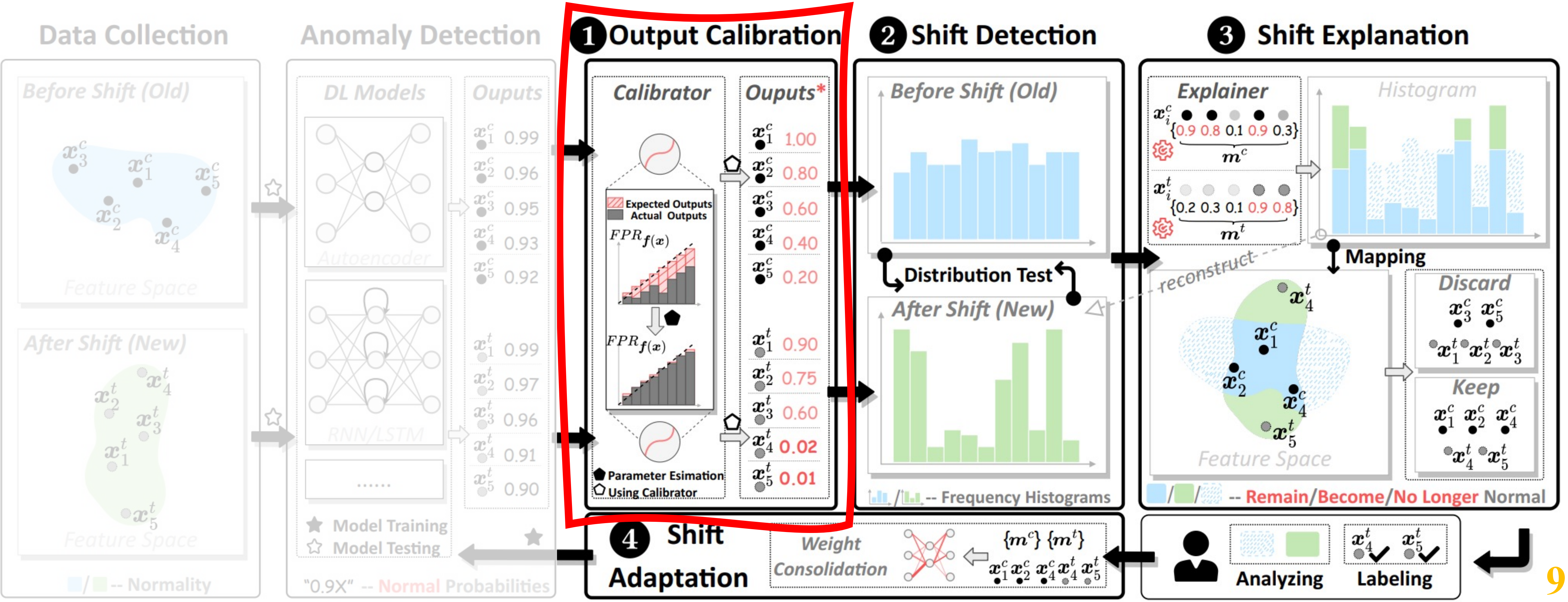
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and  Adapting to normality shift for DL-based anomaly detection.



# Step 1 — Output Calibration

**Model Calibration** for Classification

- Transform classifier scores into class membership probabilities
- E.g., given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.

# Step 1 — Output Calibration

**Model Calibration** for Classification

- Transform classifier scores into class membership probabilities
- E.g., given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.

Calibration for **Anomaly Detection**

- Expected Meaning: the **percentile** of model outputs (also FPR if threshold is itself)
- E.g., Original: [0.7, 0.8, 0.9, 1.0], Calibrated: [0.25, 0.5, 0.75. 1.0]

# Step 1 — Output Calibration

**Model Calibration** for Classification

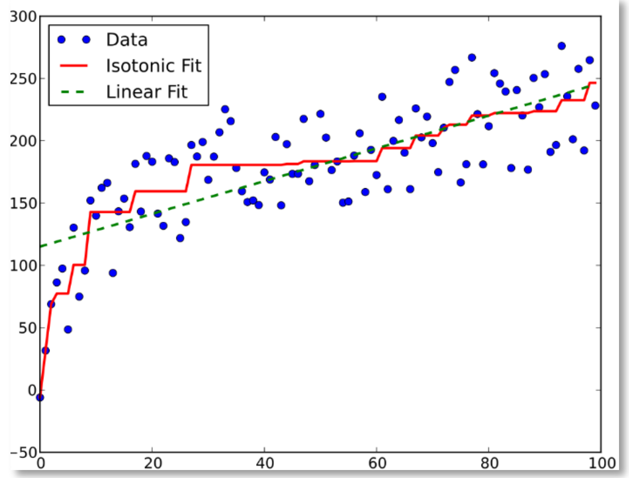
- Transform classifier scores into class membership probabilities
- E.g., given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.

Calibration for **Anomaly Detection**

- Expected Meaning: the **percentile** of model outputs (also FPR if threshold is itself)
- E.g., Original: [0.7, 0.8, 0.9, 1.0], Calibrated: [0.25, 0.5, 0.75. 1.0]

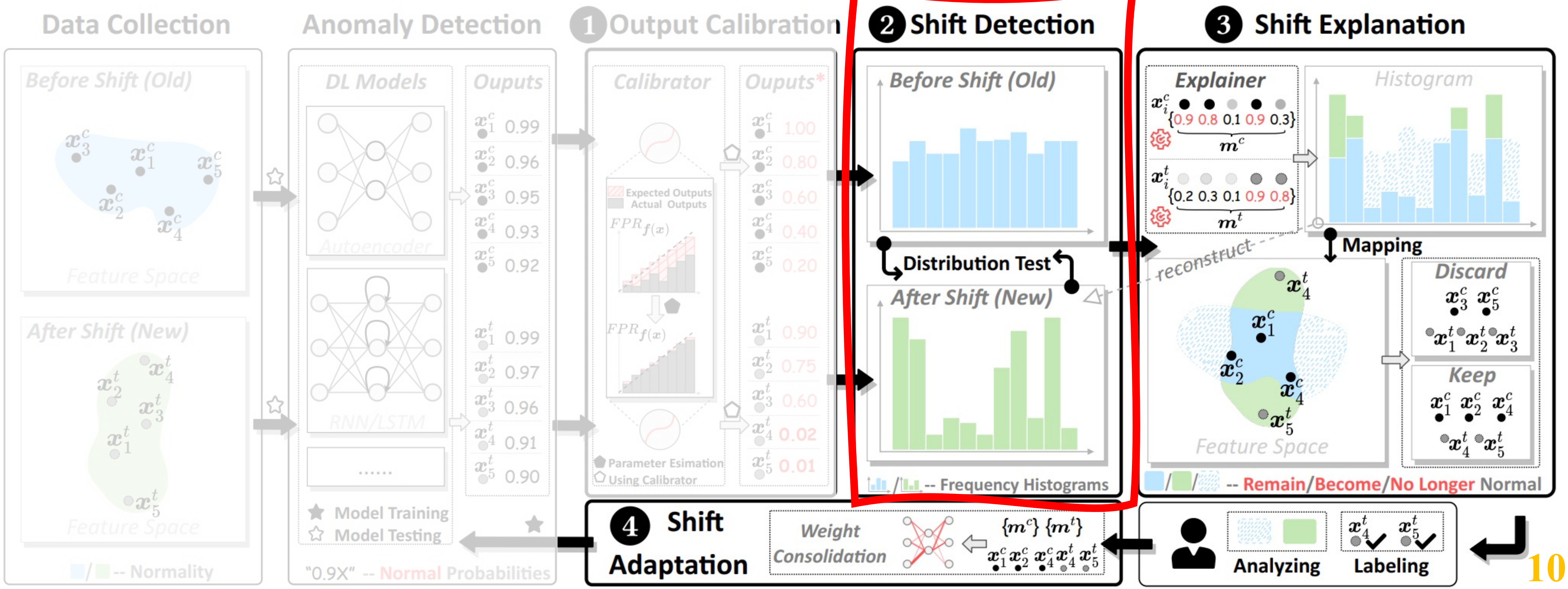
Calibration Function – **Isotonic Regression**

- **Probabilistic** legality: Convert Anomaly Score into [0,1]
- **Monotonicity**: Without affecting detection performance
- **Non-linear**: Linear transformation of distribution is meaningless



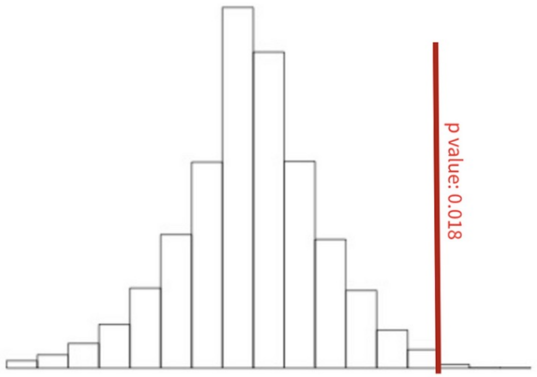
# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



# Step 2 — Shift Detection

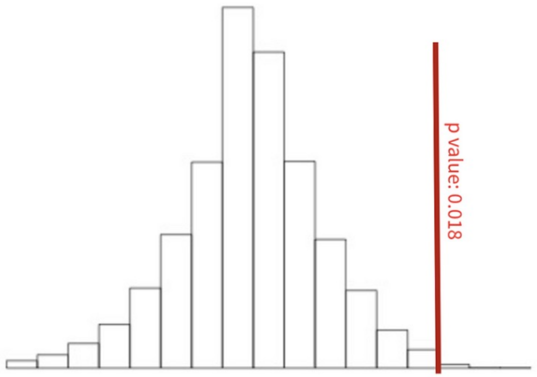
- **Hypothesis Test**
  - **H0**: Two data follow the same distribution (No drift happen)
  - **H1**: Two data do not follow same distribution (drift happens)




Ref:https://towardsdatascience.com/how-to-use-permutation-tests-bacc79f45749

# Step 2 — Shift Detection

- **Hypothesis Test**
  - **H0**: Two data follow the same distribution (No drift happen)
  - **H1**: Two data do not follow same distribution (drift happens)
- **Permutation Test**
  - **Pros:** Distribution-free, support any test statistic, and suitable for small set
  - Test Statistic: ***KL divergence*** of original and shifted distribution
  - P-value: $\frac{1+\Sigma_i^N[(KL(P||Q))<\Delta]}{N+1} < \delta$



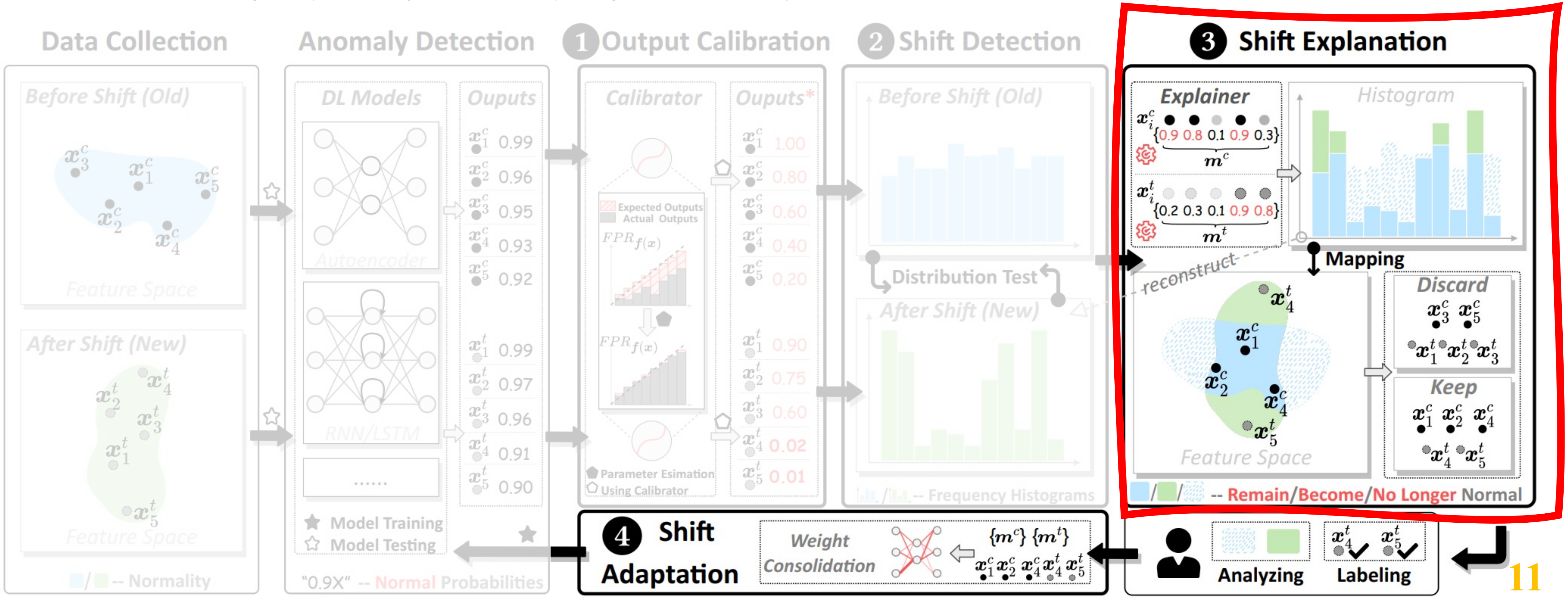

Ref:https://towardsdatascience.com/how-to-use-permutation-tests-bacc79f45749

**Algorithm 1:** Procedure for shift detection

**Input:** $\boldsymbol{x}^c \in \mathcal{X}_N^c$, $\boldsymbol{x}^t \in \mathcal{X}^t$; $K$; permutation number $N_p$
**Output:** P-value $p$ indicating the probability of non-shift
$\triangledown$ *getting original discrete distributions (histograms)*
$P_{org} \leftarrow \mathbb{H}_K(\mathcal{C}(\boldsymbol{f}(\boldsymbol{x}^c)))$; $Q_{org} \leftarrow \mathbb{H}_K(\mathcal{C}(\boldsymbol{f}(\boldsymbol{x}^t)))$;
$s_{org} \leftarrow D_{KL}(P_{org}||Q_{org})$ ; $\triangleright$ *original test statistics*
$\{\boldsymbol{P}'_i, Q'_i\}_{i=1}^{N_p} \leftarrow$ Permutating/Resampling and recomputing two histograms ($\mathbb{H}_K$) from $\{\mathcal{C}(\boldsymbol{f}(\boldsymbol{x}^c))\} \cup \{\mathcal{C}(\boldsymbol{f}(\boldsymbol{x}^t))\}$;
$p \leftarrow \frac{1+\sum_{i=1}^{N_p} \mathbb{1}[s_{org} \le D_{KL}(P'_i||Q'_i)]}{N_p+1}$ ; $\triangleright$ *p-value of test*
**return** $p$ $\triangleright$ *confidence of non-shift*

# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



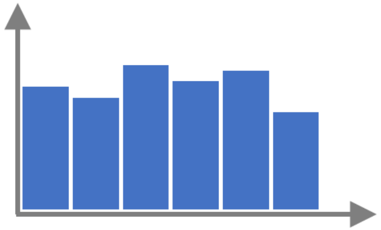
# Step 3 — Shift Explanation

$$\min_{\boldsymbol{m}_{c\oplus t}=\boldsymbol{m}_c\oplus\boldsymbol{m}_t} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)],\mathbb{D}[\boldsymbol{p}_t]\}$$

$$+\lambda_1||\boldsymbol{m}_{c\oplus t}|| - \lambda_2 \mathop{\mathbb{E}}_{m\in\boldsymbol{m}_{c\oplus t}}[m\log m+(1-m)\log(1-m)]$$

($\odot$:hadamard product, $\oplus$:vector concatenation)

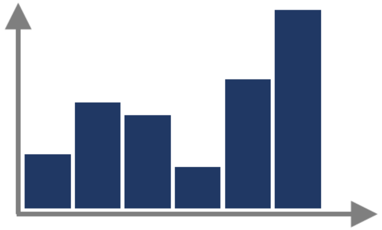
$\mathbb{D}[\boldsymbol{p}_c]$



$\boldsymbol{p}_c$

***Old Distribution***

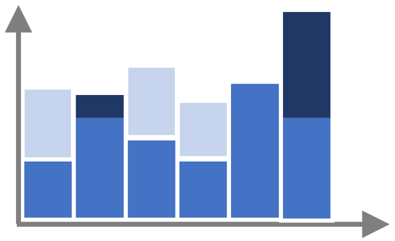
$\mathbb{D}[\boldsymbol{p}_t]$



$\boldsymbol{p}_t$

***New Distribution***

$\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)]$



$((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)$

***Mixed Distribution***

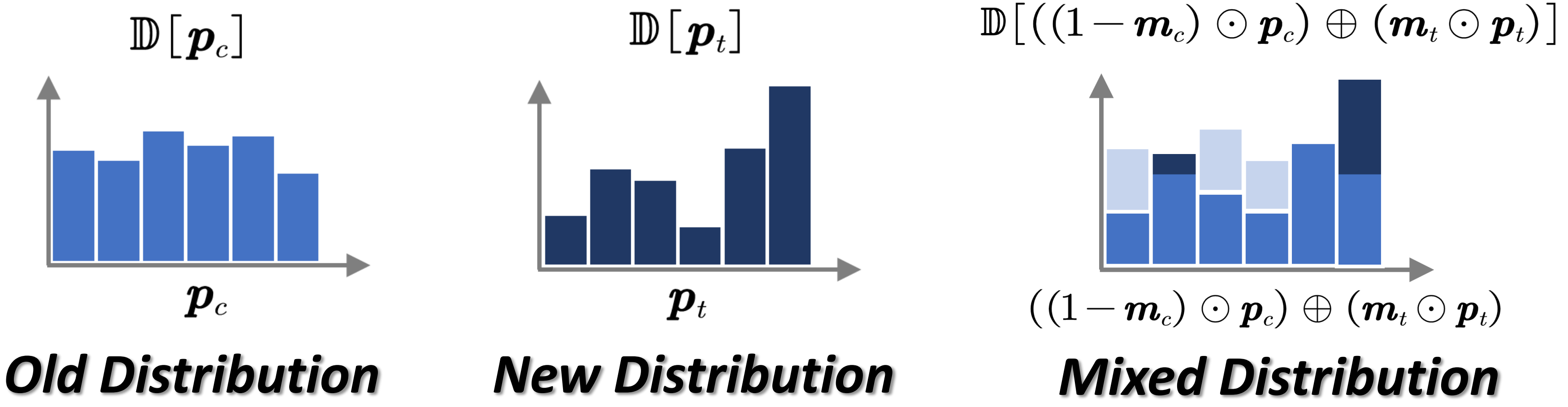
# Step 3 — Shift Explanation

Accuracy Loss

$$\min_{\boldsymbol{m}_{c\oplus t}=\boldsymbol{m}_c\oplus\boldsymbol{m}_t} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)],\mathbb{D}[\boldsymbol{p}_t]\}$$
$$+\lambda_1||\boldsymbol{m}_{c\oplus t}|| - \lambda_2 \mathop{\mathbb{E}}_{m\in\boldsymbol{m}_{c\oplus t}}[m\log m+(1-m)\log(1-m)]$$

*Mixed samples should accurately reconstruct the new distribution*

($\odot$:hadamard product, $\oplus$:vector concatenation)



# Step 3 — Shift Explanation

*Mixed samples should accurately reconstruct the new distribution*

**Accuracy Loss**

$$\min_{\boldsymbol{m}_{c\oplus t}=\boldsymbol{m}_c\oplus\boldsymbol{m}_t} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)],\mathbb{D}[\boldsymbol{p}_t]\}$$

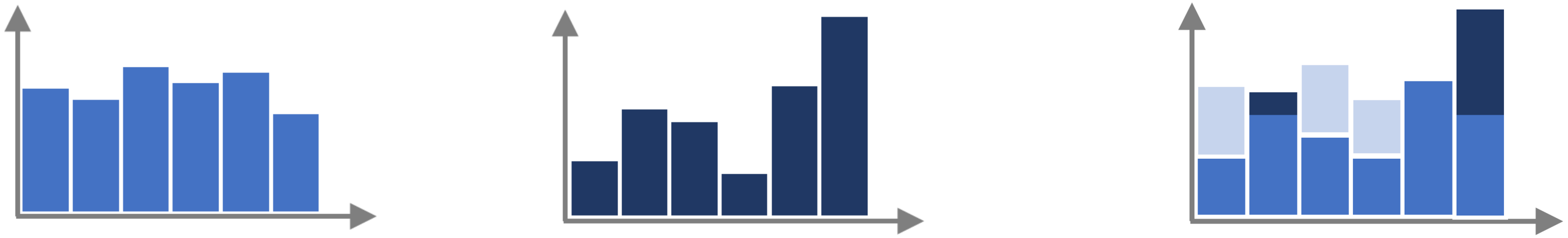
$$+\lambda_1||\boldsymbol{m}_{c\oplus t}|| - \lambda_2 \mathop{\mathbb{E}}_{m\in\boldsymbol{m}_{c\oplus t}}[m\log m+(1-m)\log(1-m)]$$

**Overhead Loss**

*Choose as few samples from the new distribution as possible*

($\odot$:hadamard product, $\oplus$:vector concatenation)

$\mathbb{D}[\boldsymbol{p}_c]$ $\mathbb{D}[\boldsymbol{p}_t]$ $\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)]$



$\boldsymbol{p}_c$ $\boldsymbol{p}_t$ $((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)$

**Old Distribution** **New Distribution** **Mixed Distribution**

# Step 3 — Shift Explanation

Accuracy Loss

$$\min_{\boldsymbol{m}_{c\oplus t}=\boldsymbol{m}_c\oplus\boldsymbol{m}_t} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)],\mathbb{D}[\boldsymbol{p}_t]\}$$
$$+\lambda_1||\boldsymbol{m}_{c\oplus t}|| - \lambda_2 \mathop{\mathbb{E}}_{m\in\boldsymbol{m}_{c\oplus t}}[m\log m + (1-m)\log(1-m)]$$

Overhead Loss

Determinism Loss

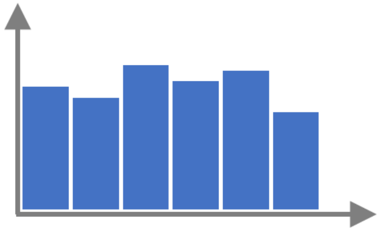
($\odot$:hadamard product, $\oplus$:vector concatenation)

*Mixed samples should accurately reconstruct the new distribution*

*Choose as few samples from the new distribution as possible*

*Expect $m_c$ or $m_t$ to be deterministic (either close to 0 or close to 1)*

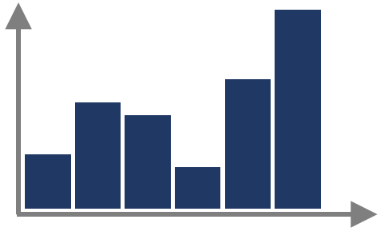
$\mathbb{D}[\boldsymbol{p}_c]$



$\boldsymbol{p}_c$

***Old Distribution***

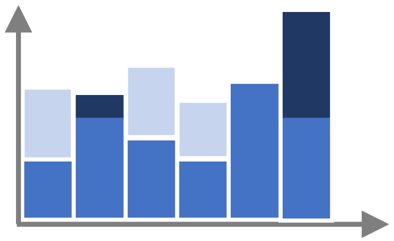
$\mathbb{D}[\boldsymbol{p}_t]$



$\boldsymbol{p}_t$

***New Distribution***

$\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)]$



$((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)$

***Mixed Distribution***

# Step 3 — Shift Explanation

Accuracy Loss

$$\min_{\boldsymbol{m}_{c\oplus t}=\boldsymbol{m}_c\oplus\boldsymbol{m}_t} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)],\mathbb{D}[\boldsymbol{p}_t]\}$$

$$+\lambda_1||\boldsymbol{m}_{c\oplus t}|| - \lambda_2 \mathop{\mathbb{E}}_{m\in\boldsymbol{m}_{c\oplus t}}[m\log m + (1-m)\log(1-m)]$$

Overhead Loss

Determinism Loss

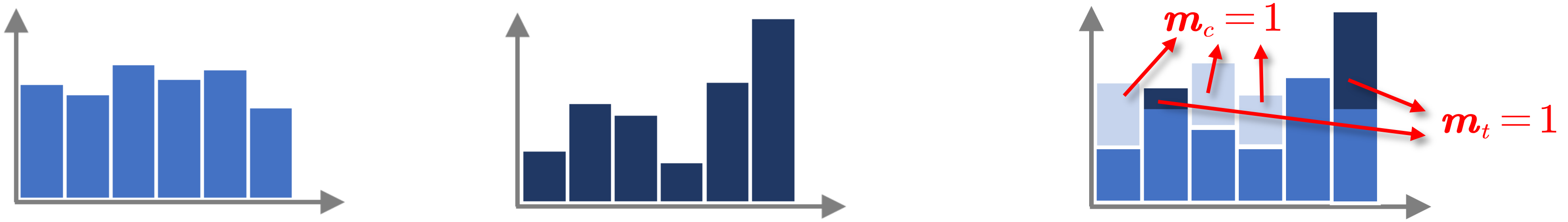
($\odot$:hadamard product, $\oplus$:vector concatenation)

*Mixed samples should accurately reconstruct the new distribution*

*Choose as few samples from the new distribution as possible*

*Expect $m_c$ or $m_t$ to be deterministic (either close to 0 or close to 1)*

$\mathbb{D}[\boldsymbol{p}_c]$ $\mathbb{D}[\boldsymbol{p}_t]$ $\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)]$

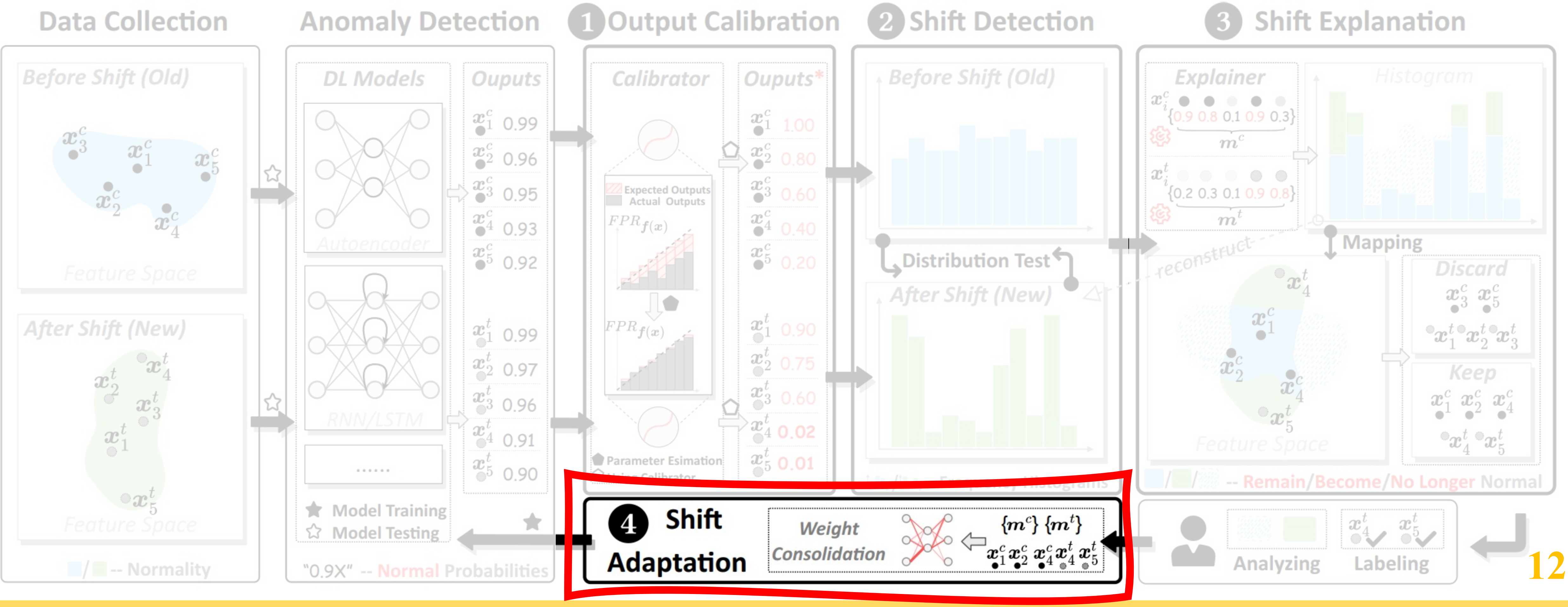



$\boldsymbol{p}_c$ $\boldsymbol{p}_t$ $((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)$

**Old Distribution** **New Distribution** **Mixed Distribution**

# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.



# Step 4 — Shift Adaptation

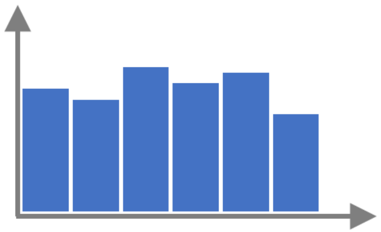
$$\min_{\theta^*} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c(\theta^*))\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t(\theta^*))],\mathbb{D}[\boldsymbol{p}_c(\theta)]\}$$

$$+\lambda\sum_{i,j}\Omega_{ij}(\theta_{ij}-\theta^*_{ij})^2$$

$$\text{where}\quad \Omega_{ij}=\sum_{P(\boldsymbol{x})\sim\boldsymbol{p}_c}\left\|\frac{\partial[\ell_2^2(F(\boldsymbol{x};\theta))]}{\partial\theta_{ij}}\right\|\cdot\boldsymbol{m}_c(\boldsymbol{x})$$

($\odot$:hadamard product, $\oplus$:vector concatenation)

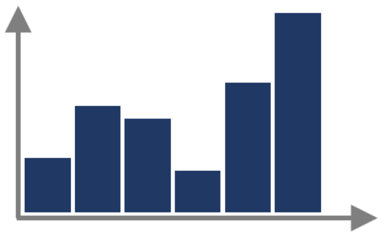
$\mathbb{D}[\boldsymbol{p}_c]$



$\boldsymbol{p}_c$

***Old Distribution***

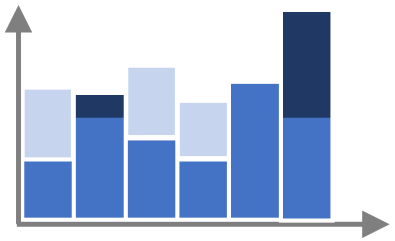
$\mathbb{D}[\boldsymbol{p}_t]$



$\boldsymbol{p}_t$

***New Distribution***

$\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)]$



$((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)$

***Mixed Distribution***

# Step 4 — Shift Adaptation

$$\min_{\theta^*} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot \boldsymbol{p}_c(\theta^*))\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t(\theta^*))],\mathbb{D}[\boldsymbol{p}_c(\theta)]\}$$

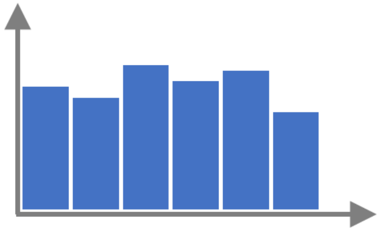
**Distributional Shift Adaptation**

$$+\lambda\sum_{i,j}\Omega_{ij}(\theta_{ij}-\theta^*_{ij})^2$$

$$\text{where} \quad \Omega_{ij}=\sum_{P(\boldsymbol{x})\sim \boldsymbol{p}_c}\left\|\frac{\partial[\ell_2^2(F(\boldsymbol{x};\theta))]}{\partial\theta_{ij}}\right\|\cdot \boldsymbol{m}_c(\boldsymbol{x})$$

($\odot$:hadamard product, $\oplus$:vector concatenation)

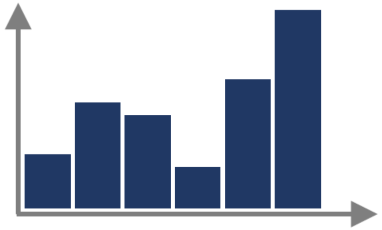
$\mathbb{D}[\boldsymbol{p}_c]$



$\boldsymbol{p}_c$

***Old Distribution***

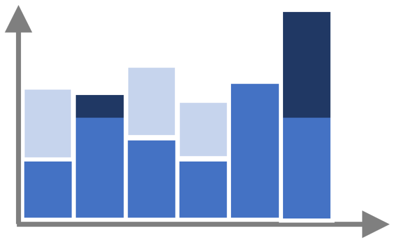
$\mathbb{D}[\boldsymbol{p}_t]$



$\boldsymbol{p}_t$

***New Distribution***

$\mathbb{D}[((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)]$



$((1-\boldsymbol{m}_c)\odot\boldsymbol{p}_c)\oplus(\boldsymbol{m}_t\odot\boldsymbol{p}_t)$

***Mixed Distribution***

# Step 4 — Shift Adaptation

$$\min_{\theta^*} \mathcal{L}\{\mathbb{D}[((1-\boldsymbol{m}_c)\odot \boldsymbol{p}_c(\theta^*)) \oplus (\boldsymbol{m}_t \odot \boldsymbol{p}_t(\theta^*))], \mathbb{D}[\boldsymbol{p}_c(\theta)]\}$$

**Distributional Shift Adaptation**

$$+\lambda \sum_{i,j} \Omega_{ij}(\theta_{ij}-\theta_{ij}^*)^2$$

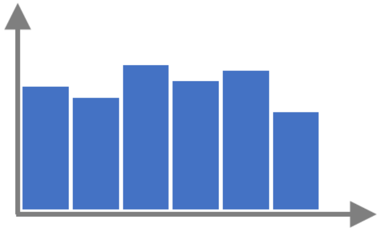
**Elastic Weight Consolidation**

$$\text{where} \quad \Omega_{ij} = \sum_{P(\boldsymbol{x})\sim \boldsymbol{p}_c} \left|\left|\frac{\partial[\ell_2^2(F(\boldsymbol{x};\theta))]}{\partial \theta_{ij}}\right|\right| \cdot \boldsymbol{m}_c(\boldsymbol{x})$$

***Evaluate the importance of model parameters***

($\odot$:hadamard product, $\oplus$:vector concatenation)

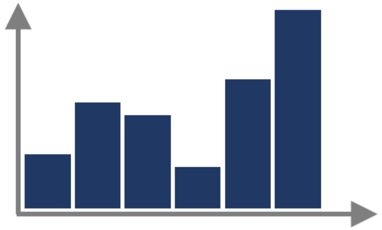
$\mathbb{D}[\boldsymbol{p}_c]$



$\boldsymbol{p}_c$

***Old Distribution***

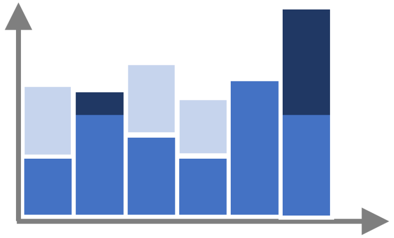
$\mathbb{D}[\boldsymbol{p}_t]$



$\boldsymbol{p}_t$

***New Distribution***

$\mathbb{D}[((1-\boldsymbol{m}_c)\odot \boldsymbol{p}_c) \oplus (\boldsymbol{m}_t \odot \boldsymbol{p}_t)]$

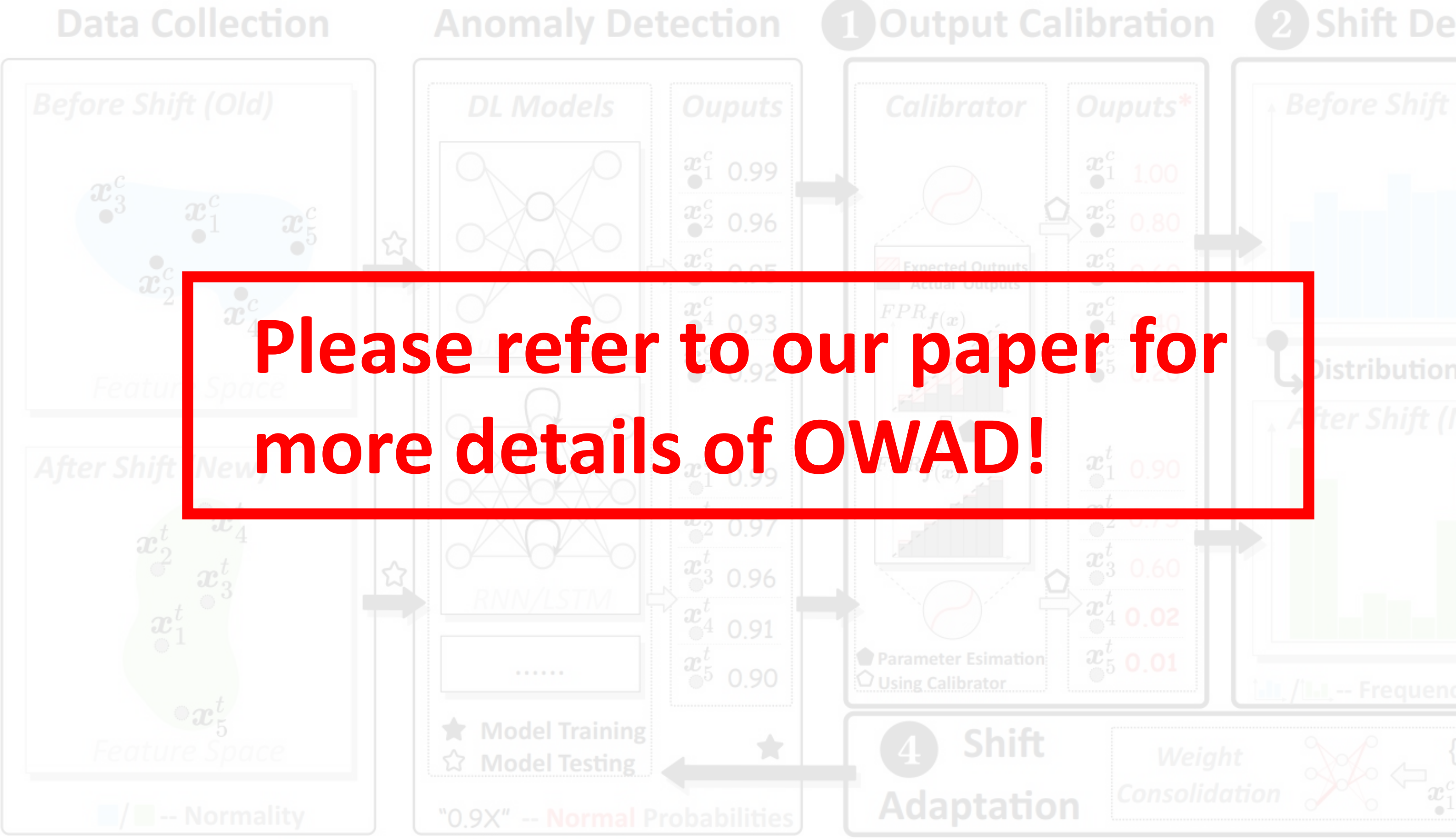


$((1-\boldsymbol{m}_c)\odot \boldsymbol{p}_c) \oplus (\boldsymbol{m}_t \odot \boldsymbol{p}_t)$

***Mixed Distribution***

# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.




**Please refer to our paper for more details of OWAD!**

Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation

Dongqi Han*†, Zhiliang Wang *†‡, Wenqi Chen*†, Kai Wang*†, Rui Yu§, Su Wang†¶, Han Zhang*†‡, Zhihua Wang‖, Minghui Jin‖, Jiahai Yang*†‡, Xingang Shi*† and Xia Yin†¶
*Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing, China
†Zhongguancun Laboratory, Beijing, China
‡Quan Cheng Laboratory, Jinan, Shandong, China
§Tsinghua Shenzhen International Graduate School, Tsinghua University, Beijing, China
¶Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing, China
‖State Grid Shanghai Municipal Electric Power Company, Shanghai, China
{handq19, chenwq19, k-wang20, yur20, wangsu17}@mails.tsinghua.edu.cn, {wzl, yang, shixg}@cernet.edu.cn

***Abstract*—Concept drift is one of the most frustrating challenges for learning-based security applications built on the close-world assumption of identical distribution between training and deployment. Anomaly detection, one of the most important tasks in security domains, is instead immune to the drift of *abnormal* behavior due to the training without any abnormal data (known as *zero-positive*), which however comes at the cost of more severe impacts when *normality shifts*. However, existing studies mainly focus on concept drift of abnormal behaviour and/or supervised learning, leaving the normality shift for zero-positive anomaly detection largely unexplored.**

**In this work, we are the first to explore the normality shift for deep learning-based anomaly detection in security applications, and propose OWAD, a general framework to *detect*, *explain*, and *adapt to* normality shift in practice. In particular, OWAD outperforms prior work by detecting shift in an *unsupervised* fashion, reducing the overhead of manual labeling, and providing better adaptation performance through distribution-level tackling. We demonstrate the effectiveness of OWAD through several realistic experiments on three security-related anomaly detection applications with long-term practical data. Results show that OWAD can provide better adaptation performance of normality shift with less labeling overhead. We provide case studies to analyze the normality shift and provide operational recommendations for security applications. We also conduct an initial real-world deployment on a SCADA security system.**

I. INTRODUCTION

Anomaly detection is one of the most important tasks in security domains [13], trained with normal data and detecting anomalies that deviates from the distribution of *normality*. Recently, the adoption of Deep Learning (DL) enables anomaly detection to extract more complex features from massive data [12], [76], as well as detect unforeseen threats such as zero-day attacks through learning with only normal data, known as *zero-positive learning* [20]. Heretofore, researchers have applied DL-based anomaly detection for various security applications,

Network and Distributed System Security (NDSS) Symposium 2023
27 February - 3 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.24830
www.ndss-symposium.org

TABLE I: Comparison of representative related works.

| Features | CADE[81] | TRANSCENDENT[37], [5] | UNLEARN[20] | OWAD |
|---|---|---|---|---|
| Support Timeseries | ○ | ● | ● | ● |
| Unsupervised* | ○ | ◐ | ● | ● |
| Label-efficient† | ◐ | ◐ | ○ | ● |
| Distribution-level‡ | ○ | ◐ | ○ | ● |

(● = true, ◐ = partially true, ○ = false);
* TRANSCENDENT *can be used but requires non-trivial adjustments for unsupervised (zero-positive) cases;* † *Measured in* §V-B;
‡ TRANSCENDENT *statistically considers distributional information, but tackles drift in a sample-level fashion (i.e., rejection).*

such as detecting network intrusions [54], [72], finding threats from system logs [21], [53], tracing advanced persistent threats (APT) [9], [77], which all achieved satisfactory performance.

Unfortunately, the superior performance of learning-based applications is built on the *close-world* assumption of independent and identically distributed (i.i.d.) between training and test samples [68]. Such assumption often does not hold in *open-world* settings due to the divergence of incoming test distribution from the original one, known as *concept drift*. In security domains, concept drift is pervasive as the malicious patterns are switched suddenly and dramatically over time in the hostile environment [4].

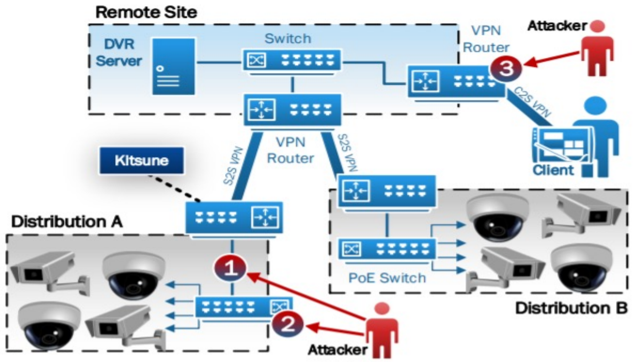
In this context, *anomaly detection* is instead immune to the drift of *malicious/abnormal* behavior due to zero-positive learning, which however comes at the price of more severe impact when the distribution of normality shifts. In real-world deployment, the way users interact with systems under monitoring can differ and evolve over time, so do the systems themselves. For example, the involvement of new patches, devices, and protocols all have the potential to shift the normal pattern. Such *normality shift*[1], if not detected and adapted, will induce a large number of false positives (FP) and false negatives (FN), suggested by anecdotal evidence in practice.

In recent years, several studies have been proposed to tackle concept drift for learning-based security applications in the community, which can be divided into two approaches: The first is to periodically retrain the models in the dynamic environment [14], [15], [38], [61], [36], [30], [57], regard-

[1]Normality shift intuitively refers to the change of distribution of normal data (detailed definition is in §II-C). In this paper, we interchangeably use terms "drift" and "shift". We tend to use "normality shift" as a whole term.

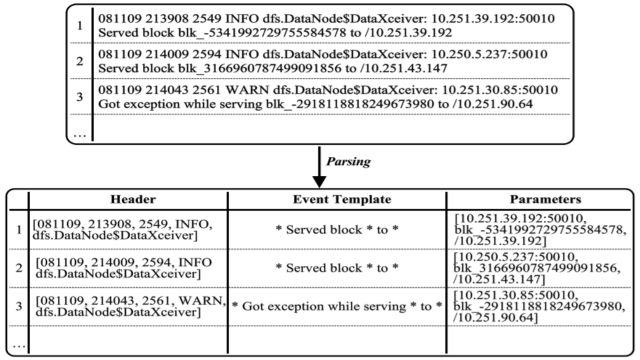
# Evaluation

## Network Intrusion



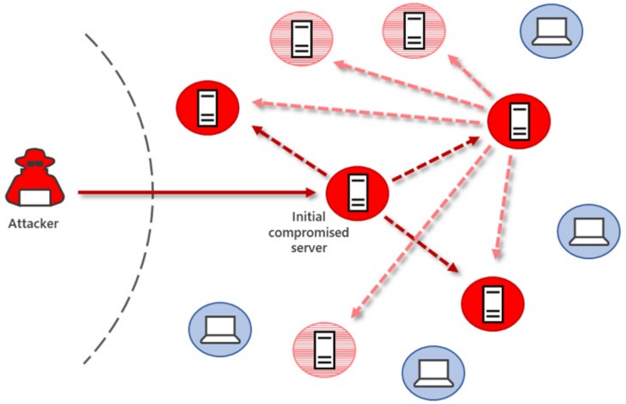

- Kitsune [NDSS’18]

## Log Anomaly




- DeepLog [CCS’17]

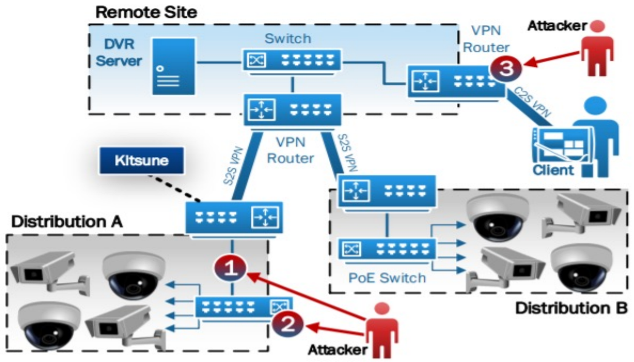
## Lateral Movement




- GL-GV [RAID’20]

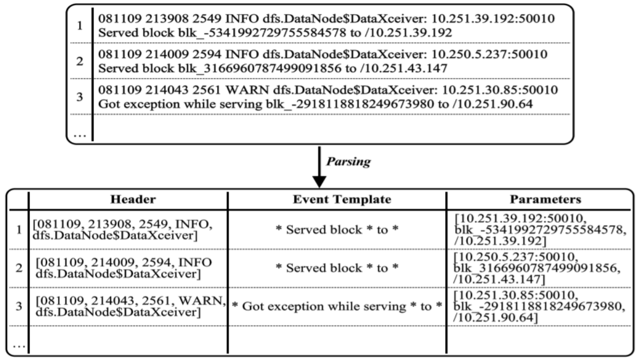
# Evaluation

### Network Intrusion



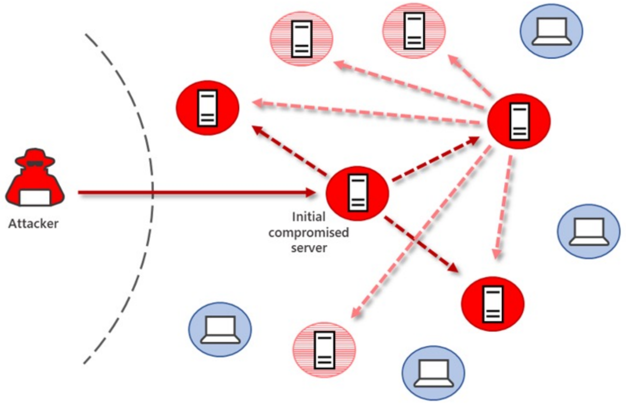

- Kitsune [NDSS’18]
- Anoshift Benchmark [NIPS’22]
- honey pot and campus network traffic

### Log Anomaly




- DeepLog [CCS’17]
- BGL Dataset [DSN’07]
- BlueGene/L supercomputer group Logs

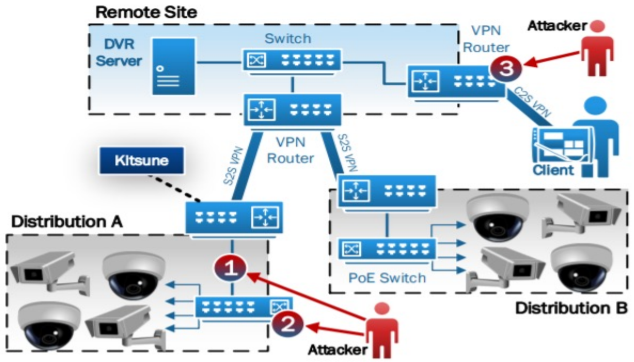
### Lateral Movement




- GL-GV [RAID’20]
- LANL-CMSCSE Dataset
- login events from corporate internal network

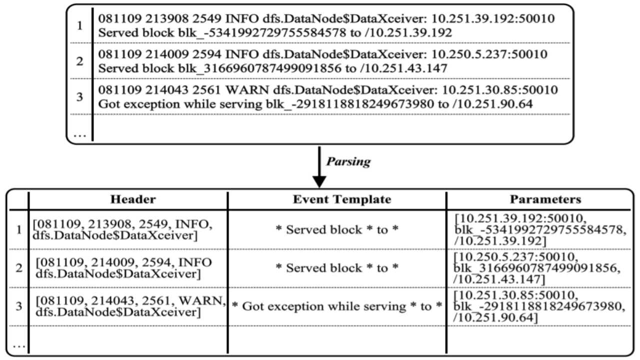
# Evaluation

### Network Intrusion



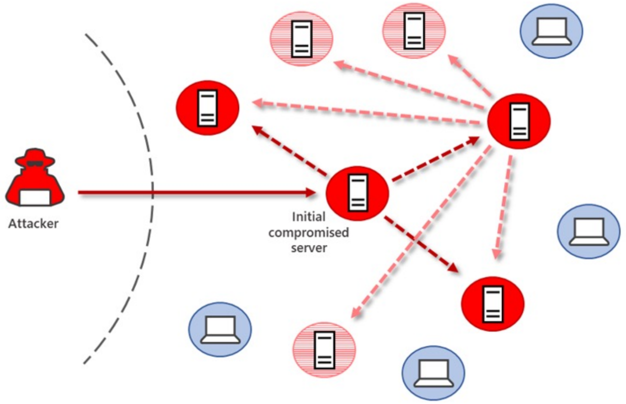

- Kitsune [NDSS’18]
- Anoshift Benchmark [NIPS’22]
- honey pot and campus network traffic
- 10 years
- detect once a year

### Log Anomaly



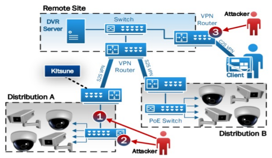

- DeepLog [CCS’17]
- BGL Dataset [DSN’07]
- BlueGene/L supercomputer group Logs
- 7 months
- detect once a month

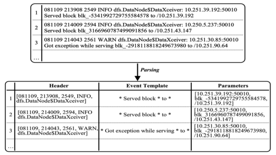
### Lateral Movement

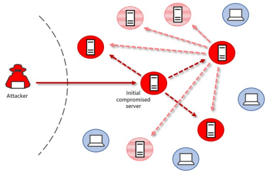



- GL-GV [RAID’20]
- LANL-CMSCSE Dataset
- login events from corporate internal network
- 58 days
- detect once a week

# Normality Shift in Security Applications



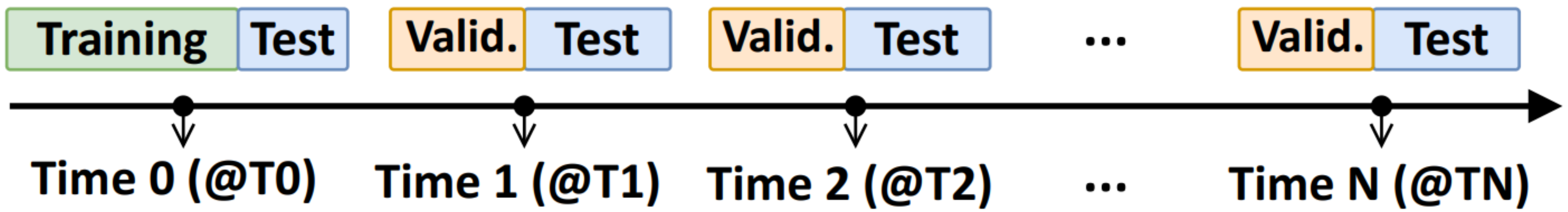




# Normality Shift in Security Applications


**Normality shift in security domain is quite common and different for each applications (case-by-case)**

# End-to-end Performance Evaluation

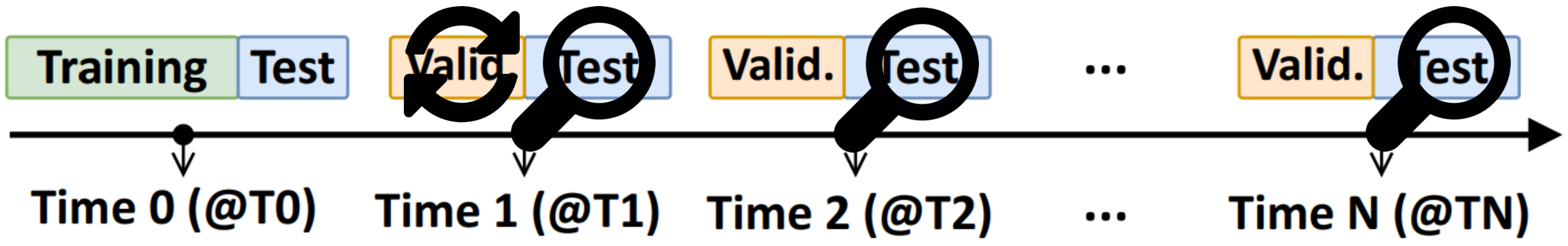
- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, …, N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, …, N



# End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, …, N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, …, N

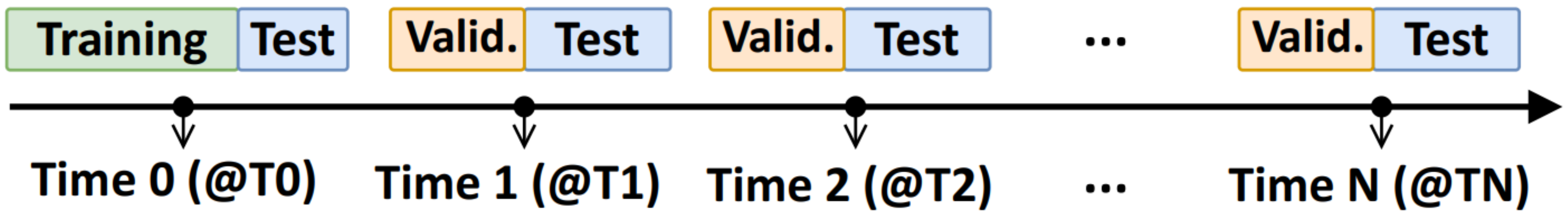
- **Experimental setup**
  - **Single Adaptation**: Update model at Time 1, Test mode at Time 2, 3, …



# End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, …, N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, …, N

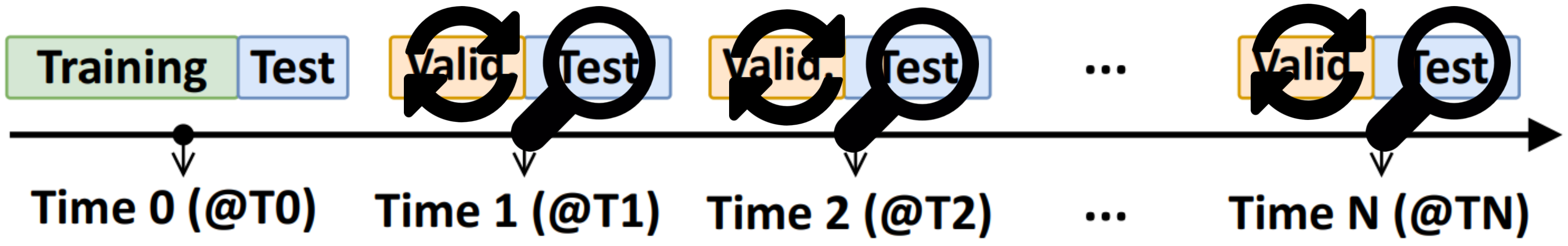
- **Experimental setup**
  - **Single Adaptation**: Update model at Time 1, Test mode at Time 2, 3, …



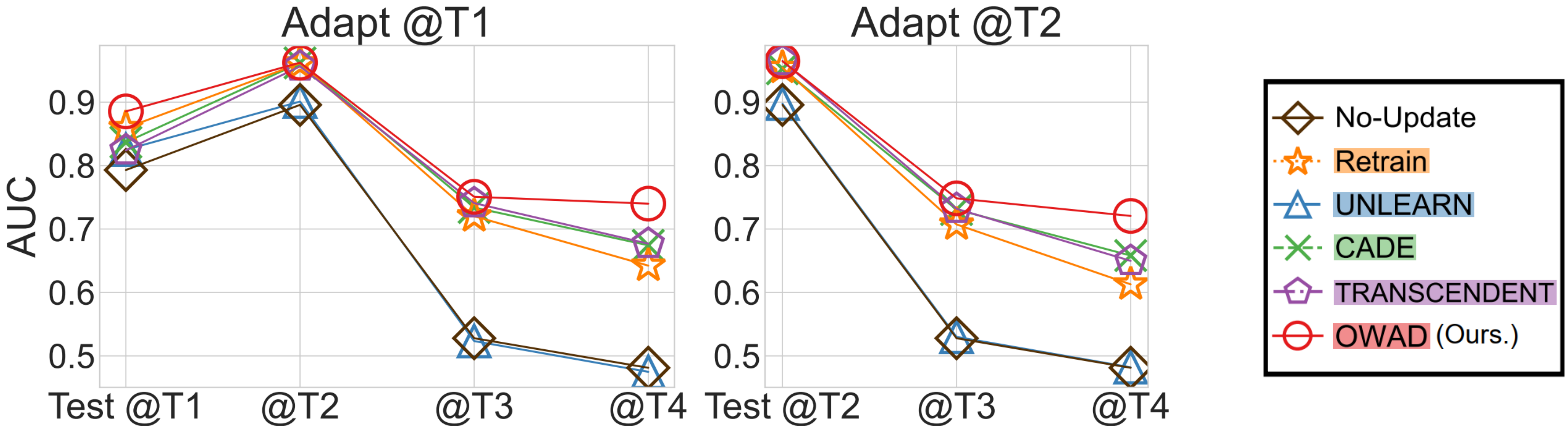
# End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, …, N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, …, N

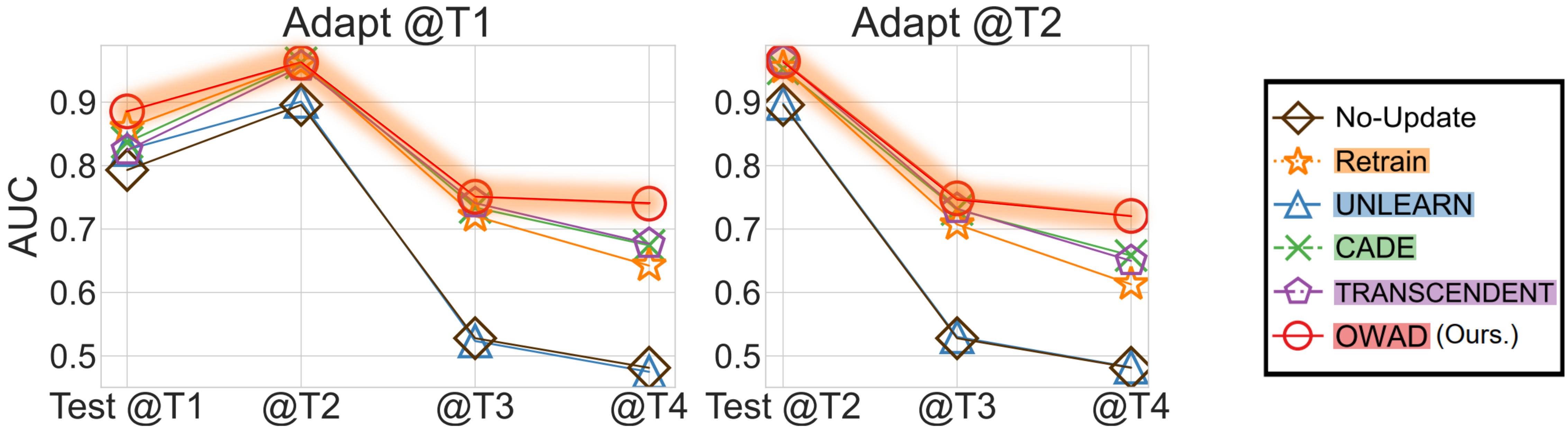
- **Experimental setup**
  - **Single Adaptation**: Update model at Time 1, Test mode at Time 2, 3, …
  - **Multiple Adaptations**: Update model at Time 1, 2, 3, …, Test mode at the same time



# Performance of Single Adaptation

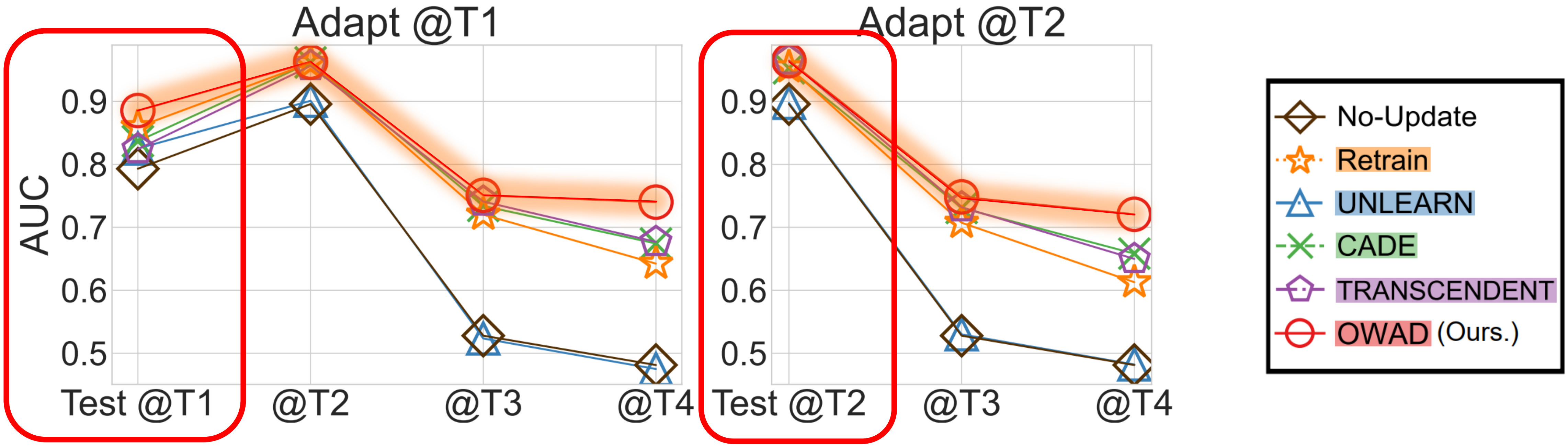


# Performance of Single Adaptation



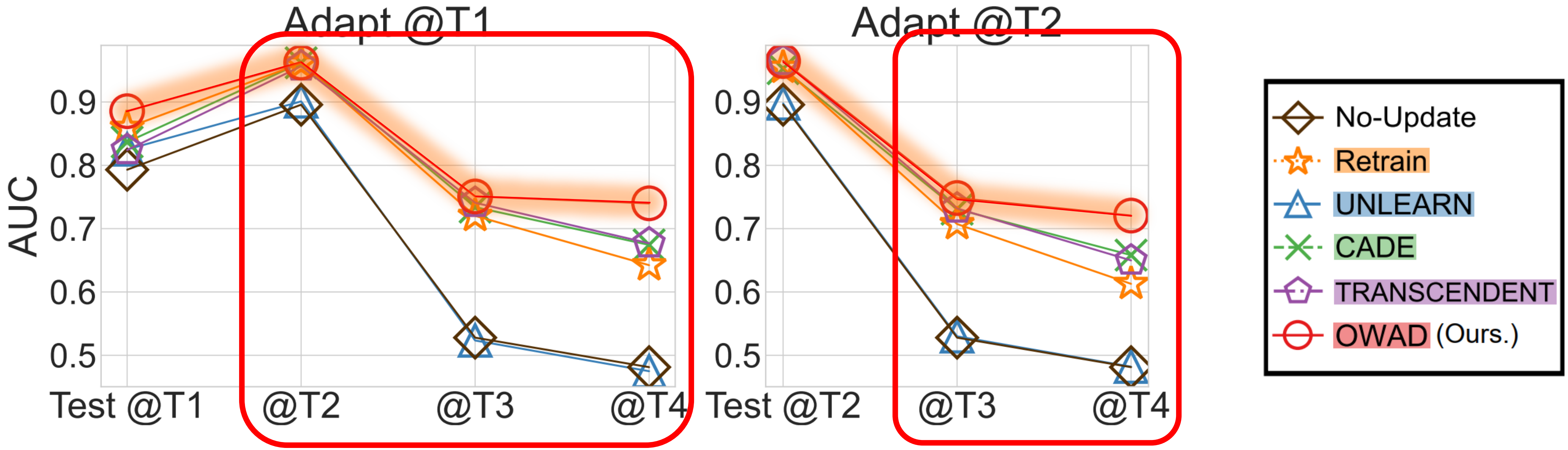

**OWAD outperforms other approaches at the adaptation time, and can also mitigate the performance degradation in subsequent time**

# Performance of Single Adaptation



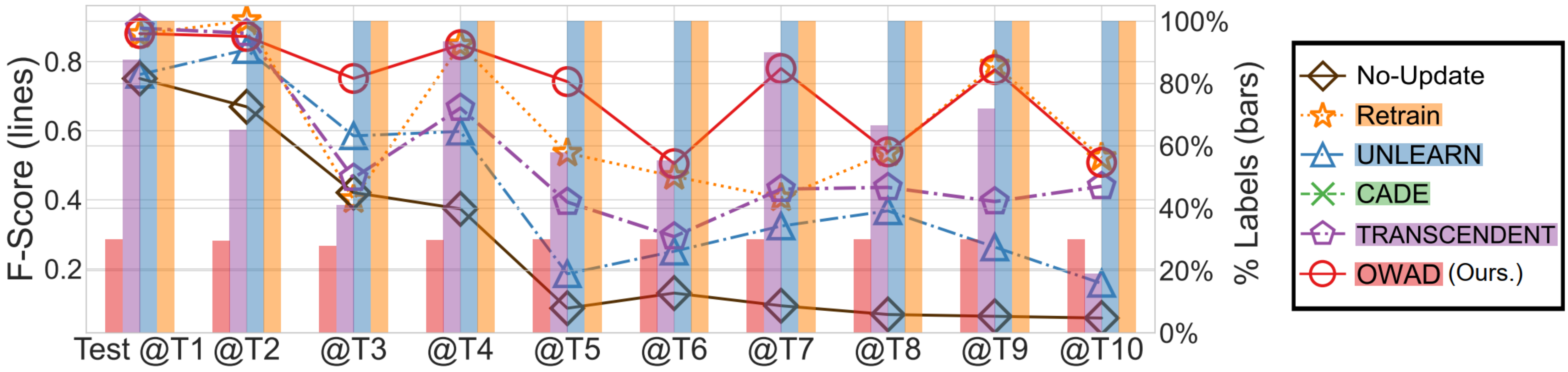

**OWAD outperforms other approaches *at the adaptation time*, and can also mitigate the performance degradation in subsequent time**

# Performance of Single Adaptation

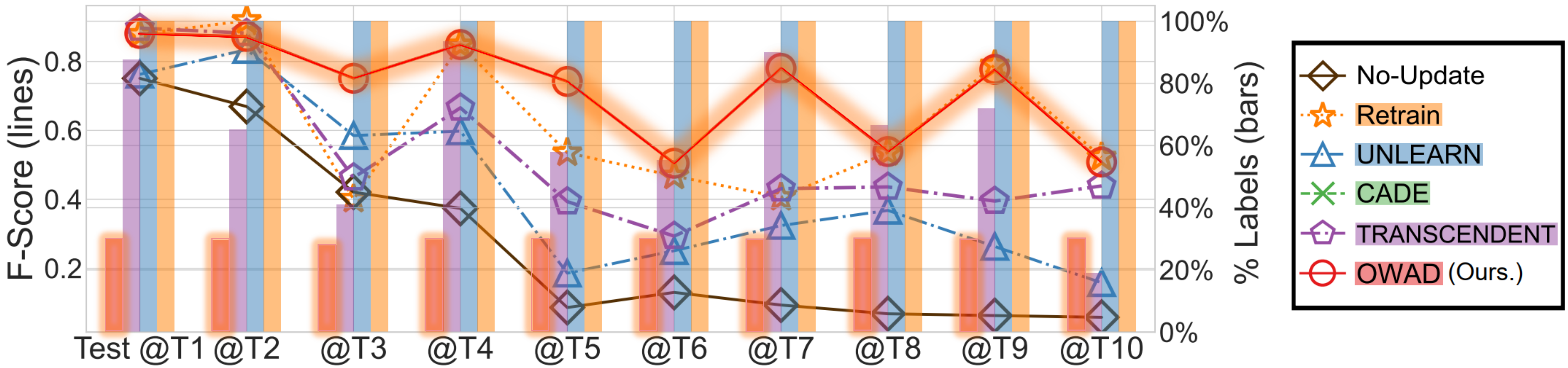



**OWAD outperforms other approaches at the adaptation time, and can also mitigate the performance degradation *in subsequent time***

# Performance of Multiple Adaptations



# Performance of Multiple Adaptations




**OWAD can achieve better results with significantly less required labels**

# Performance of FP/FNs

| Methods | # FPs (Lower is Better) | | | # FNs (Lower is Better) | | |
|---|---|---|---|---|---|---|
| | @T1 | @T2 | @T3 | @T1 | @T2 | @T3 |
| No-Update | 2404 | 903 | 6585 | 135 | 34 | 39 |
| Retrain | 2238 | 933 | 6213 | 233 | 32 | 28 |
| UNLEARN | 3350 | 1293 | 7369 | **105** | **27** | **26** |
| TRANS. | 1508 | 849 | 3237 | 552 | 197 | 106 |
| **OWAD** | **1491** | **701** | **2519** | 120 | 34 | 35 |

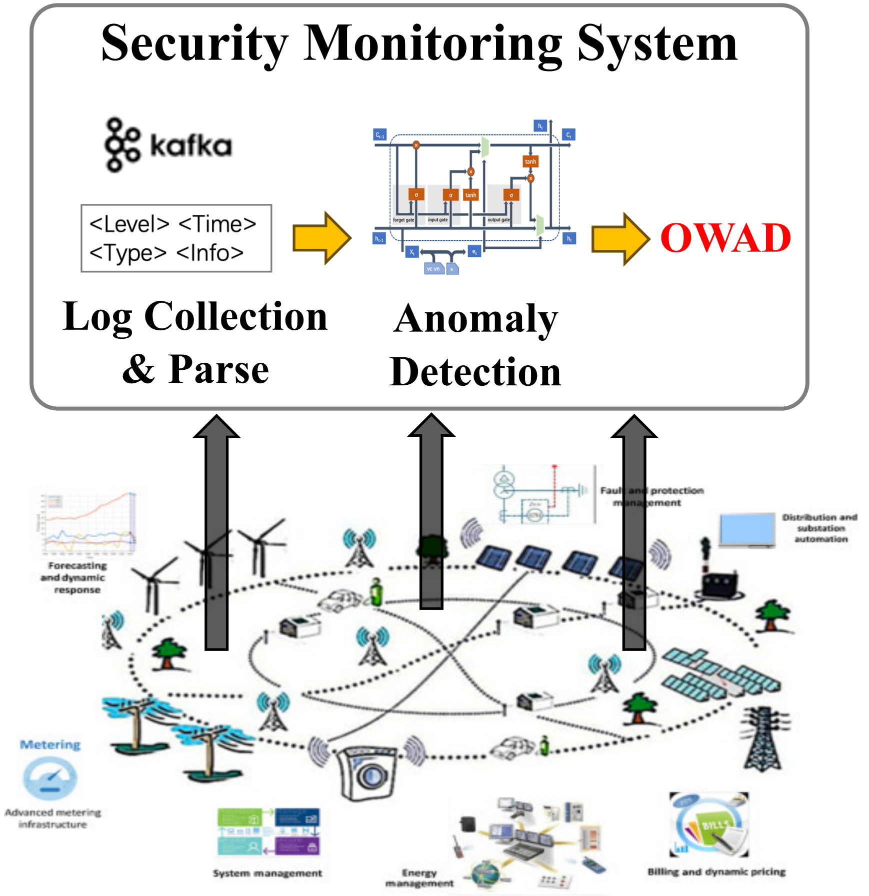
# Performance of FP/FNs

| Methods | # FPs (Lower is Better) | | | # FNs (Lower is Better) | | |
|---|---|---|---|---|---|---|
| | @T1 | @T2 | @T3 | @T1 | @T2 | @T3 |
| No-Update | 2404 | 903 | 6585 | 135 | 34 | 39 |
| Retrain | 2238↓ | 933↑ | 6213↓ | 233↑ | 32↓ | 28↓ |
| UNLEARN | 3350↑ | 1293↑ | 7369↑ | **105**↓ | **27**↓ | **26**↓ |
| TRANS. | 1508↓ | 849↓ | 3237↓ | 552↑ | 197↑ | 106↑ |
| **OWAD** | **1491**↓ | **701**↓ | **2519**↓ | 120↓ | 34 – | 35↓ |

**OWAD is the only approach that can reduce both FPs and FNs**

# Real-world Deployment

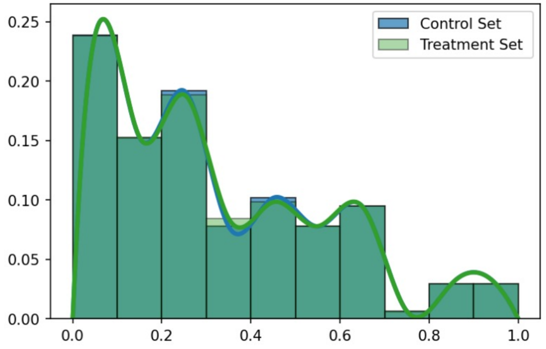
- **Background**
  - SCADA in State Grid Shanghai Electric Power Company
  - Security Monitoring System (device logs and events)
  - LSTM-based Log Anomaly Detection
  - Performance degradation in long-term deployment
  - **Data**: >10M logs from 20 devices in 5 months (2022)



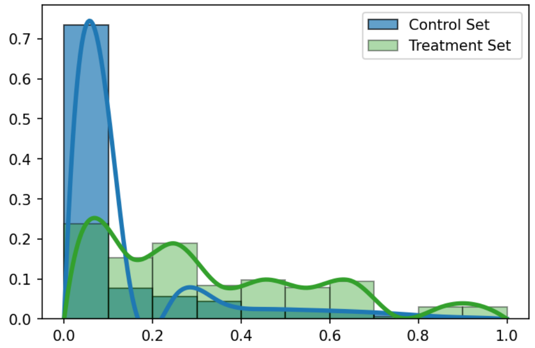

Ref:https://www.sciencedirect.com/science/article/abs/
780128053430000188

# Real-world Deployment

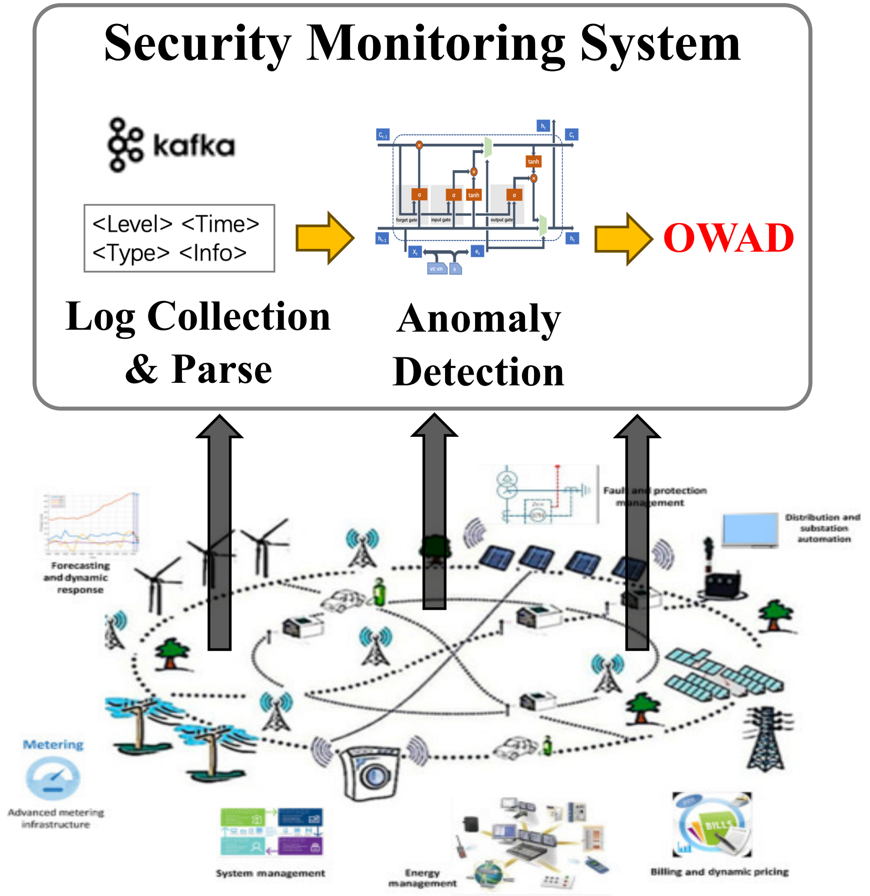
- **Background**
  - SCADA in State Grid Shanghai Electric Power Company
  - Security Monitoring System (device logs and events)
  - LSTM-based Log Anomaly Detection
  - Performance degradation in long-term deployment
  - **Data**: >10M logs from 20 devices in 5 months (2022)
- **OWAD Shift Detection**




**01/28/2022 vs 02/20/2022**
**(3 weeks)**



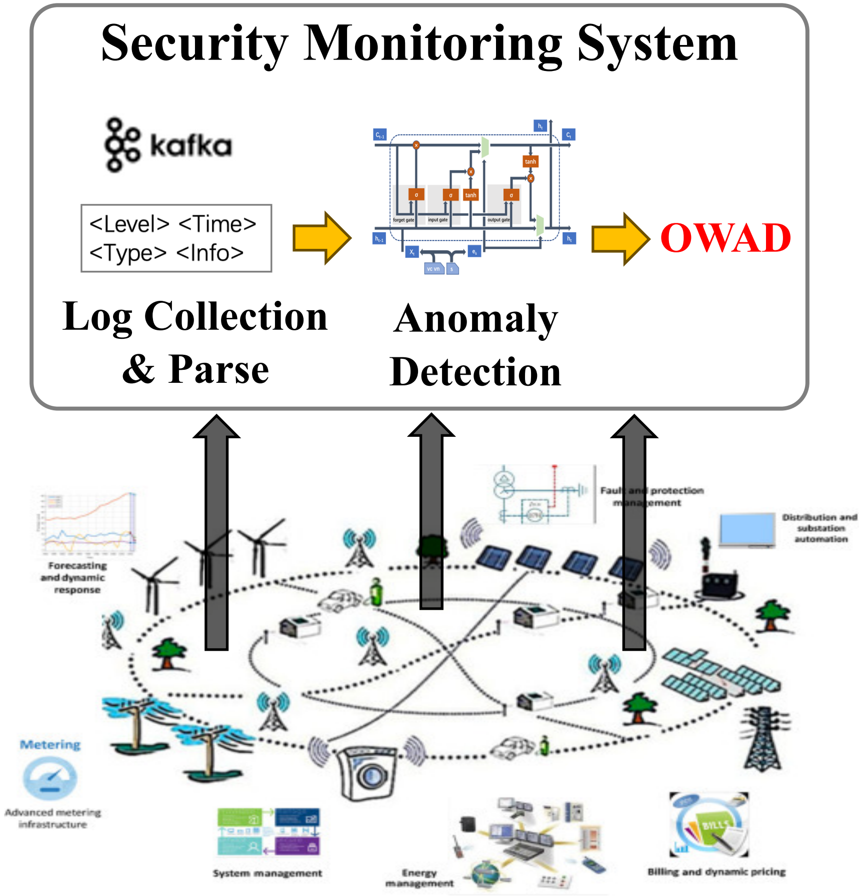

**10/19/2021 vs 02/20/2022**
**(18 weeks)**




Ref:https://www.sciencedirect.com/science/article/abs/pii/B9780128053430000188

# Real-world Deployment

- **OWAD Shift Explanation**
  - Identify 2 key logs inducing the normality shift
  - 1) network volume increases for specific devices
    - > SVR 4 4 eth3 0 0 0 eth2 1 29098502414 30822806215 eth0 1 752064 2107538
  - 2) new service continuously launches
    - > SVR 4 13 tcp 0.0.0.0 36387 0.0.0.0 0 LISTEN 1129 rpc.statd
  - Find the key reason of shift:
    - FTP service error due to system update & restart (Jan. 2022)

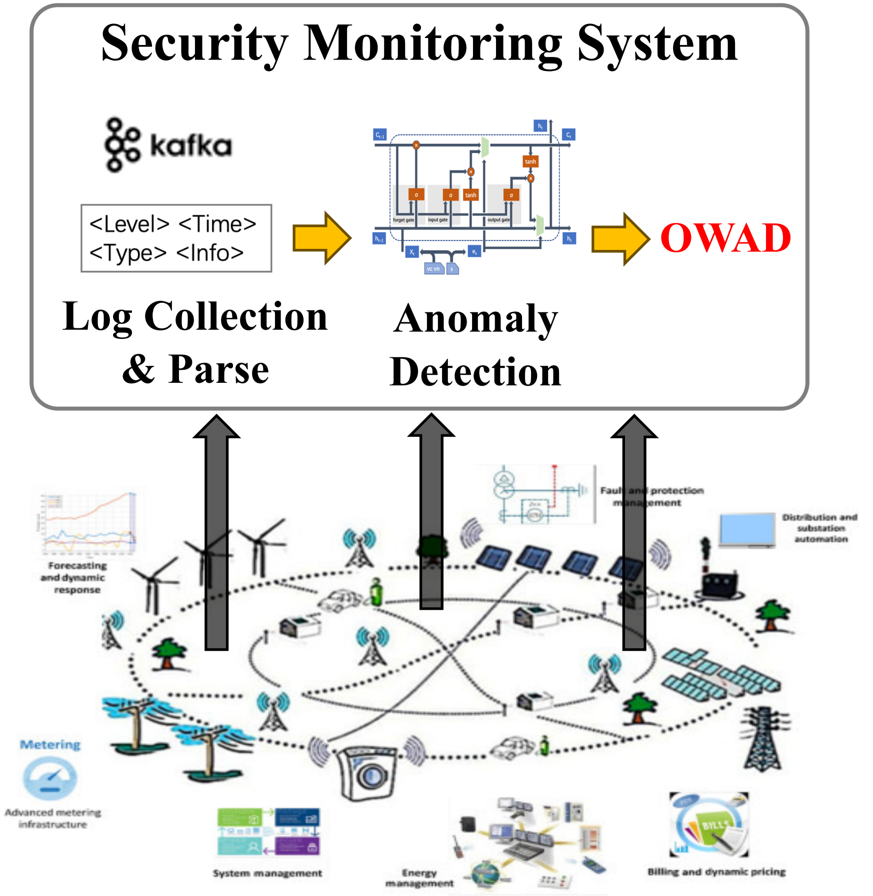



Ref:https://www.sciencedirect.com/science/article/abs/pii/B9780128053430000188

# Real-world Deployment

- **OWAD Shift Explanation**
  - Identify 2 key logs inducing the normality shift
  - 1) network volume increases for specific devices
    - > SVR 4 4 eth3 0 0 0 eth2 1 29098502414 30822806215 eth0 1 752064 2107538
  - 2) new service continuously launches
    - > SVR 4 13 tcp 0.0.0.0 36387 0.0.0.0 0 LISTEN 1129 rpc.statd
  - Find the key reason of shift:
    - FTP service error due to system update & restart (Jan. 2022)
- **OWAD Shift Adaptation**
  - Reduce >90% False Positives

| | Week 1 | Week 9 (@T1) | | Week 18 (@T2) | | Test @T2 (Adapt@T1) |
|---|---|---|---|---|---|---|
| | #FP | #FP | P-value | #FP | P-value | #FP |
| Device A | 14 | 25 | 0.999 | 79 | 0.257 | Unshift |
| Device B | 45 | 1,027 | 0.000 | 1,678 | 0.000 | 154 |
| Device C | 68 | 3,071 | 0.000 | 3,103 | 0.000 | 98 |




Ref:https://www.sciencedirect.com/science/article/abs/pii/B9780128053430000188

# Takeaways

- Normality shift is quite common and complicated in network security domains
- After calibration, model outputs can effectively to represent the normality distribution
- Labeling is inevitable for handling normality shift.
  Nevertheless, OWAD can achieve better performance with lower labels
- OWAD is shown to be able to reduce both False Positives and False Negatives

https://github.com/dongtsi/OWAD

Anomaly Detection in the Open World:
Normality Shift Detection, Explanation, and Adaptation

# Thank you! Questions?

**Presenter: Dongqi Han**

https://github.com/dongtsi

handq19@mails.tsinghua.edu.cn

www.dongqihan.top

# OWAD Design

- We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.