No-Sweat Detective:
No Effort Anomaly Detection for Wi-Fi-Based Localization

Kota Tsubouchi† Kohei Yamamoto‡ Nobuhiko Nishio‡

† Yahoo Japan Research
‡ Ritsumeikan University

Outline

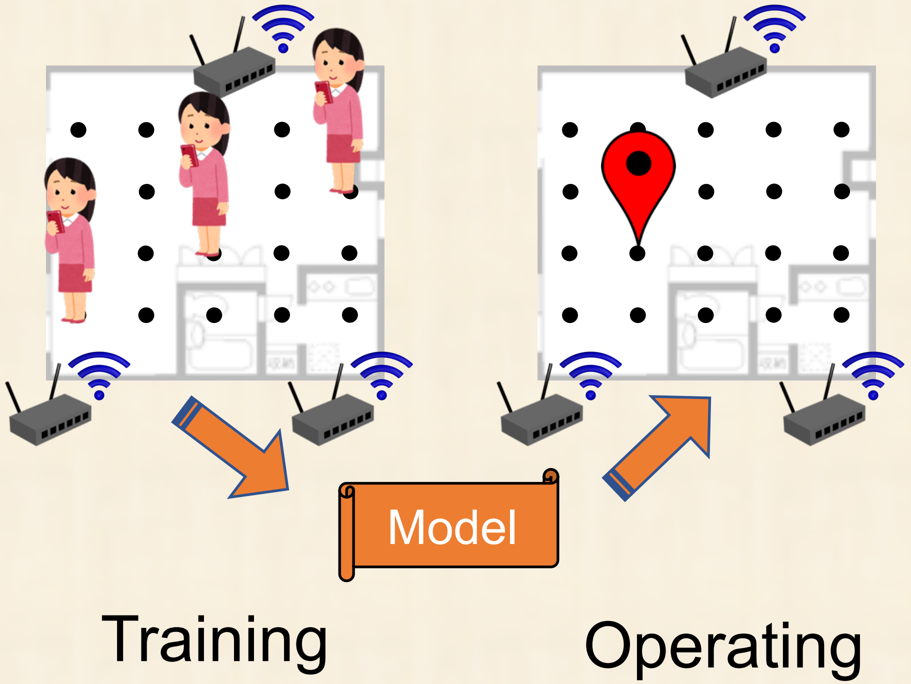
- Motivation
- State of the Art
- Methodology
- Evaluation
  - Laboratory environment
  - Wild environment
- Summary

Outline

- Motivation
- State of the Art
- Methodology
- Evaluation
  - Laboratory environment
  - Wild environment
- Summary

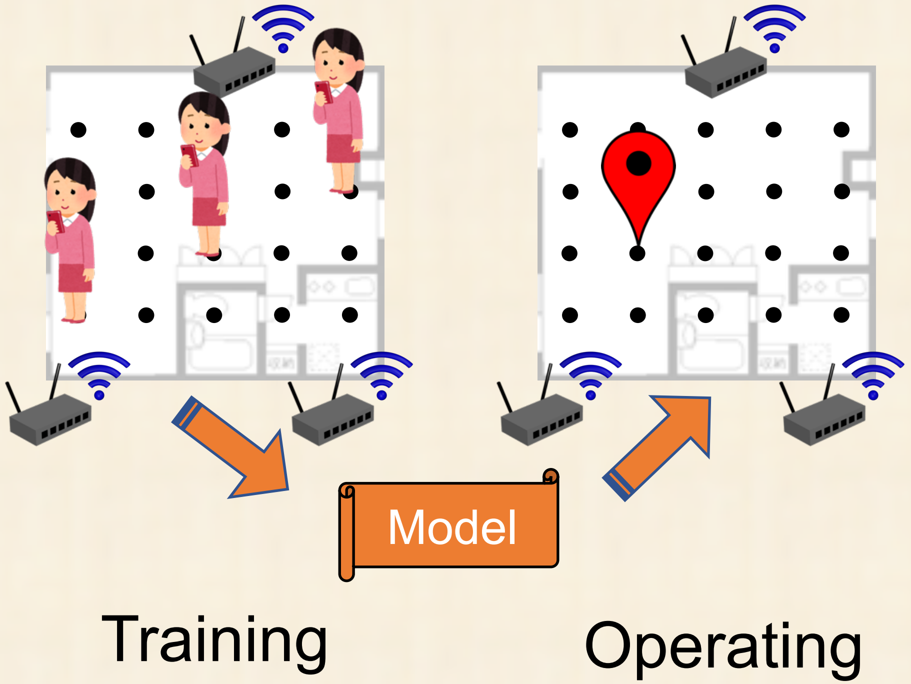
Motivation

- Wi-Fi fingerprinting is pervasive for indoor localization
- Wi-Fi localization consists of two phases:
  - Training phase
  - Operation phase



Motivation

- Wi-Fi fingerprinting is pervasive for indoor localization
- Wi-Fi localization consists of two phases:
  - Training phase
  - Operation phase



Environmental changes affect of the model (with aging)

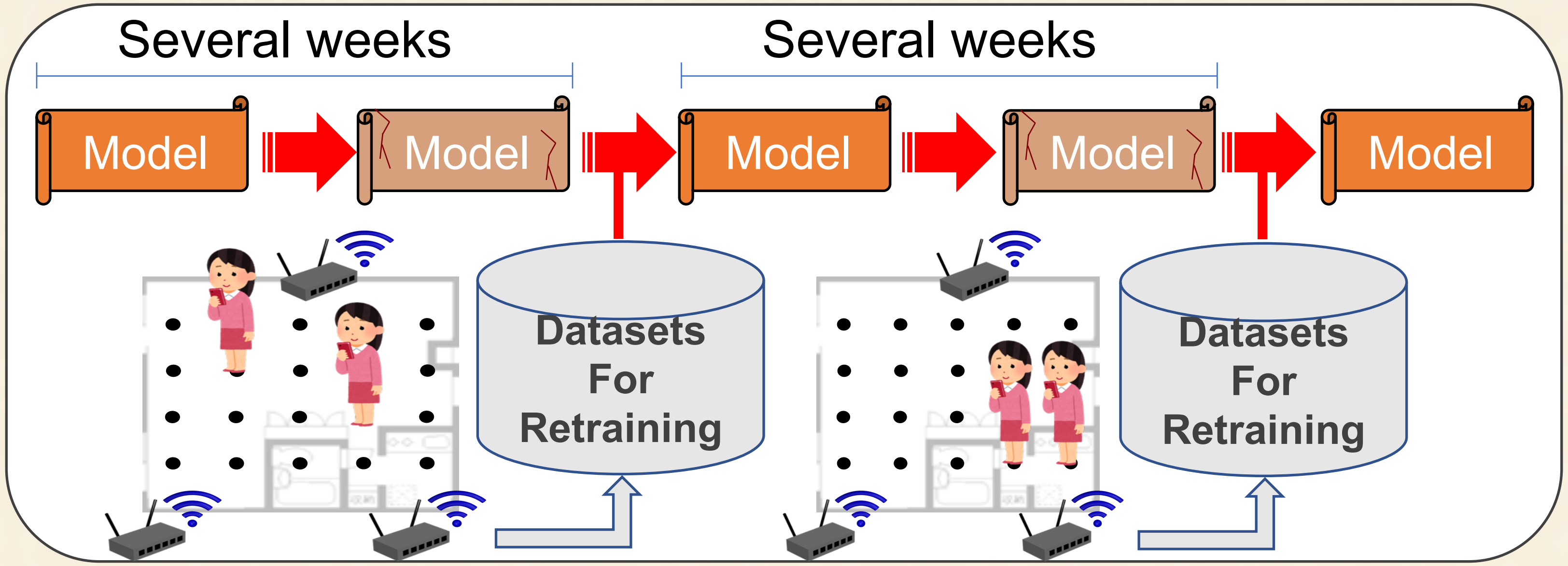
Need to remake the entire model

Outline

- Motivation
- State of the Art
- Methodology
- Evaluation
  - Laboratory environment
  - Wild environment
- Summary

State of the Art

- Transfer learning is employed to retrain the model periodically
  - Require small additional datasets but remaking of the entire model
- Reference points for additional datasets are randomly selected
  - Result in overfitting and instable accuracy



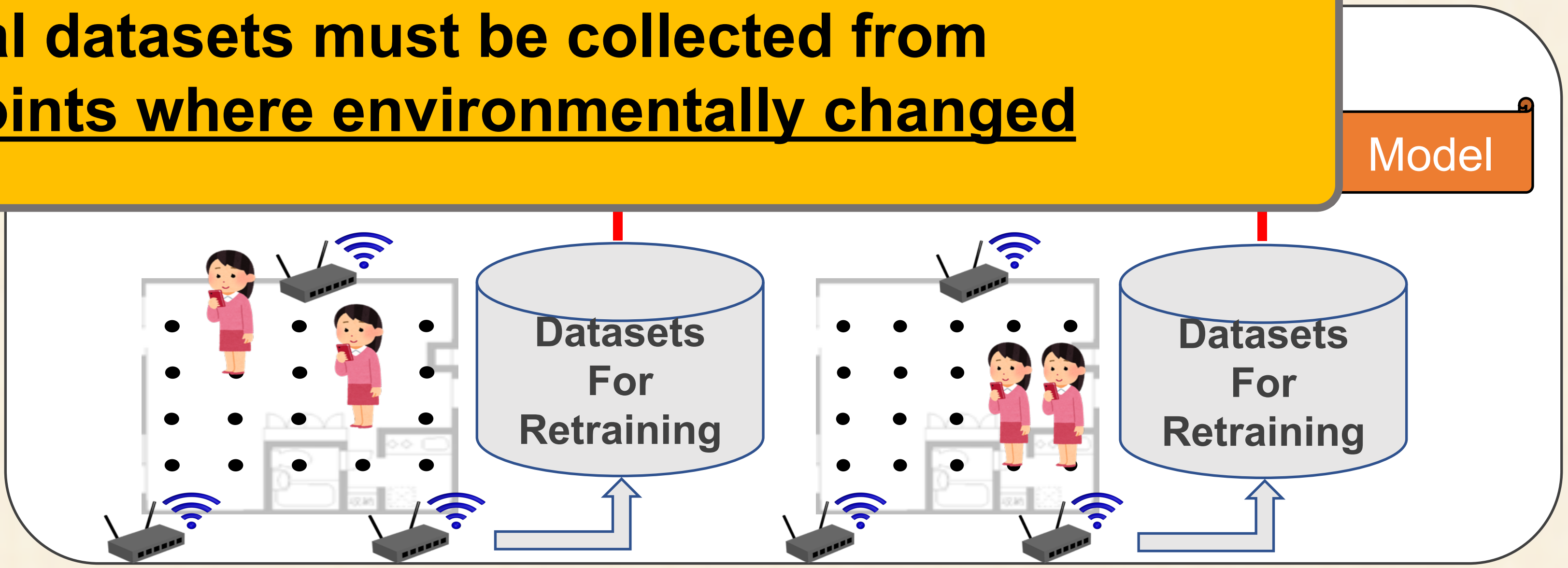
Retraining

[23]: Pengcheng Wu and Thomas G. Dietterich. “Improving SVM Accuracy by Training on Auxiliary Data Sources.”

State of the Art

- Transfer learning is employed to retrain the model periodically
  - Require small additional datasets but remaking of the entire model
- Reference points for additional datasets are randomly selected
  - Result in overfitting and instable accuracy

Additional datasets must be collected from
<u>reference points where environmentally changed</u>



Retraining

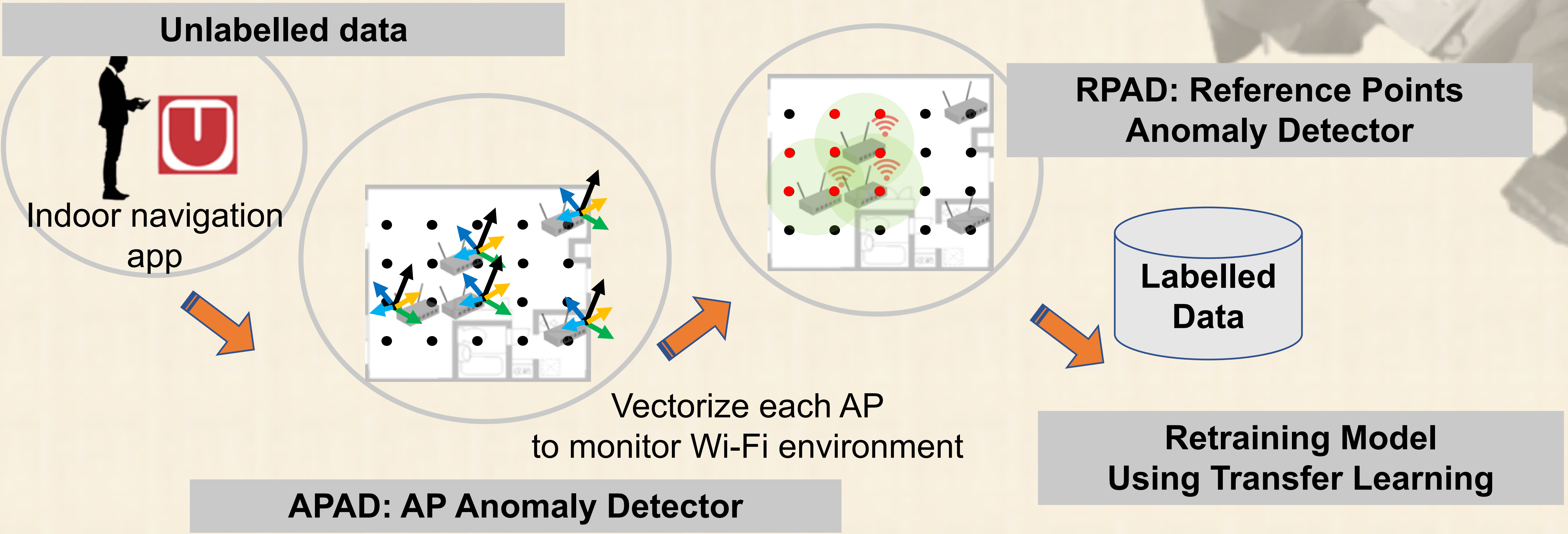
[23]: Pengcheng Wu and Thomas G. Dietterich. “Improving SVM Accuracy by Training on Auxiliary Data Sources.”

Outline

- Motivation
- State of the Art
- Methodology
- Evaluation
  - Laboratory environment
  - Wild environment
- Summary

No-Sweat Detective

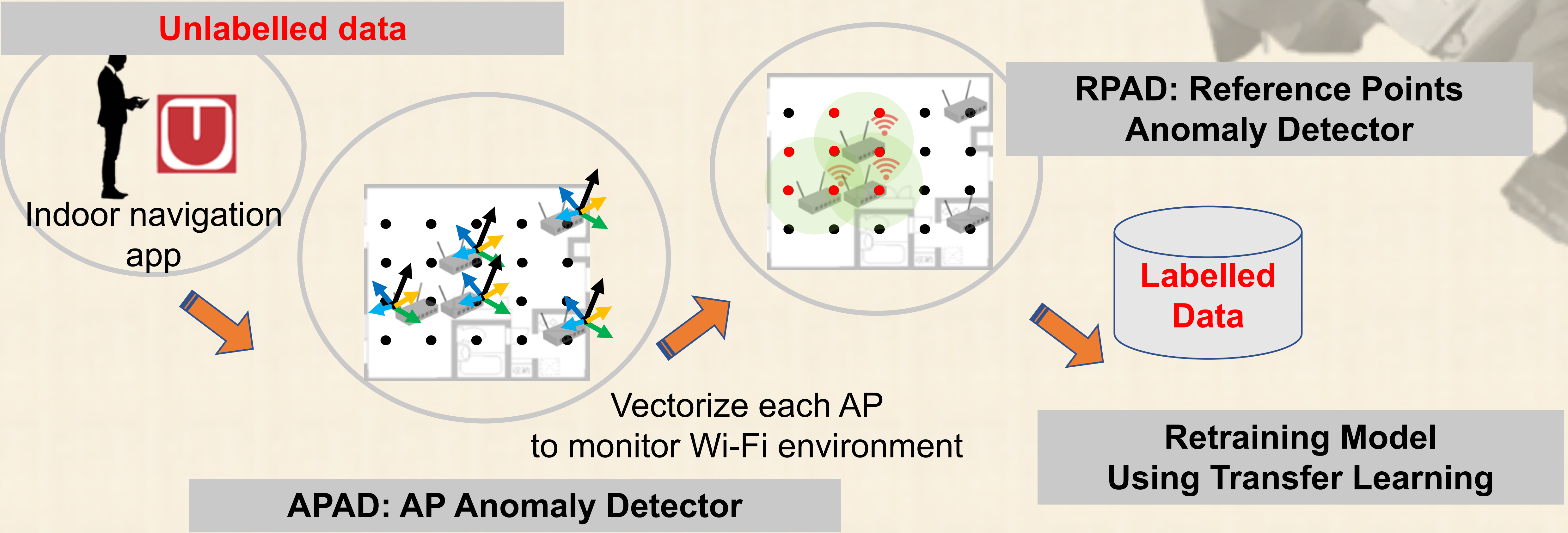
- Identify(detect) reference points where environmentally changed
  - Analyzing unlabelled data from off-the-shelf indoor navigation app



Umechika Navi: https://www.umechikanavi.jp/

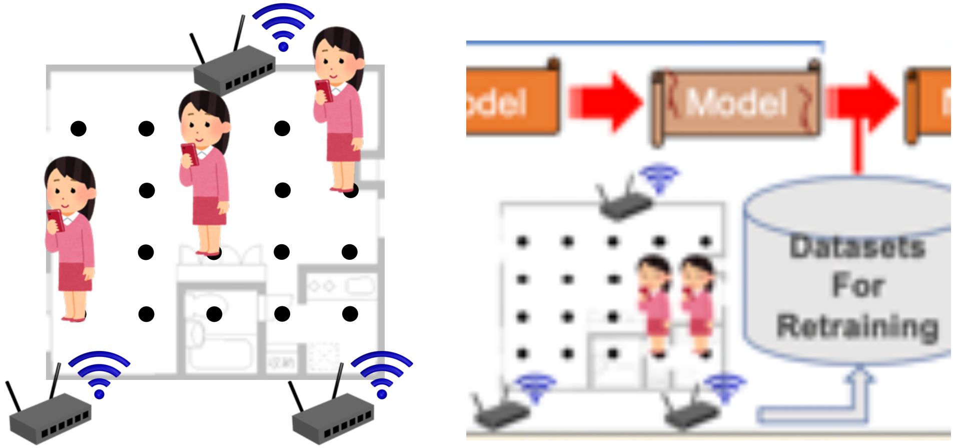
No-Sweat Detective

- Identify(detect) reference points where environmentally changed
  - Analyzing unlabelled data from off-the-shelf indoor navigation app



Umechika Navi: https://www.umechikanavi.jp/

Labelled Data



Training/Retraining of the model

<u>Location is labelled on fingerprints</u>

Unlabelled Data

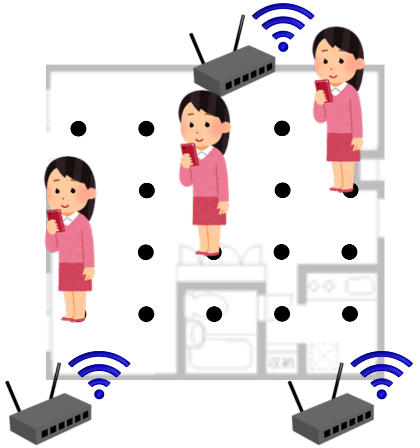
Indoor navigation/map app

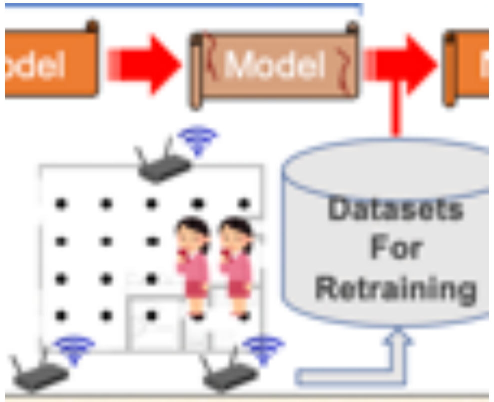
<u>Not labelled</u>

Can be obtained with no effort (sweat)

Umechika Navi: https://www.umechikanavi.jp/

Labelled Data





Training/Retraining of the model

<u>Location is labelled on fingerprints</u>

Specify reference points to obtain labeled data for retraining

←

Unlabelled Data

Indoor navigation/map app

<u>Not labelled</u>

Can be obtained with no effort (sweat)

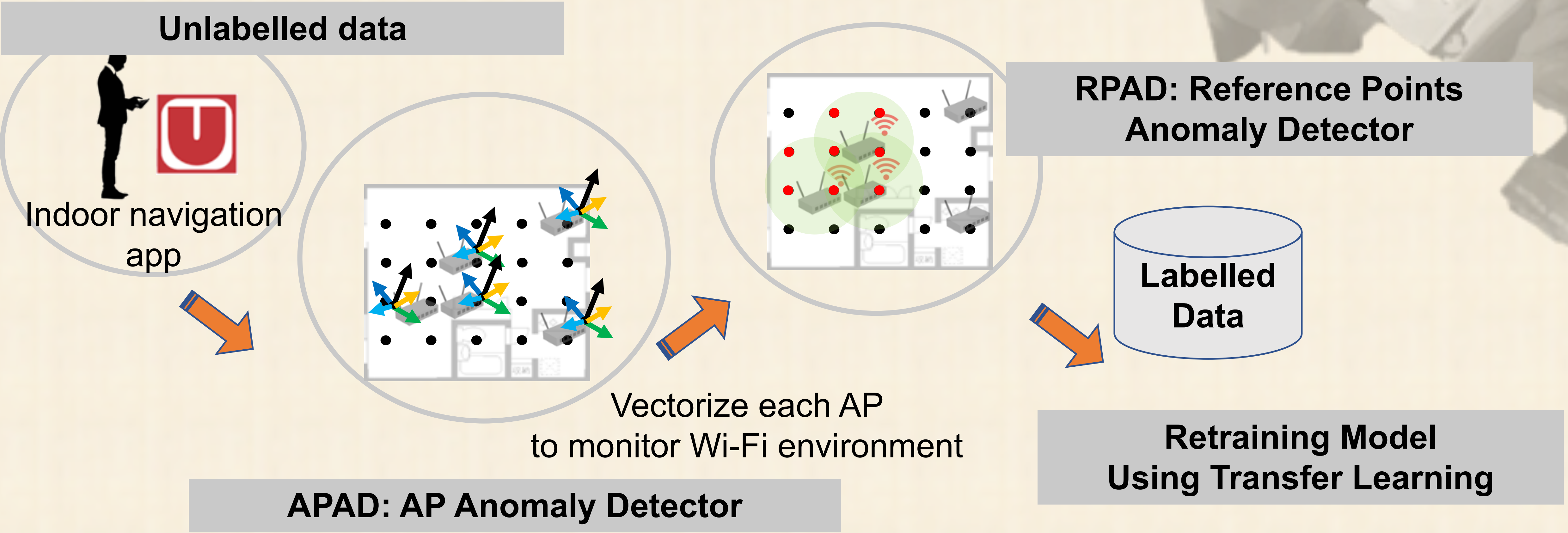
↓

Analysis

Umechika Navi: https://www.umechikanavi.jp/

No-Sweat Detective

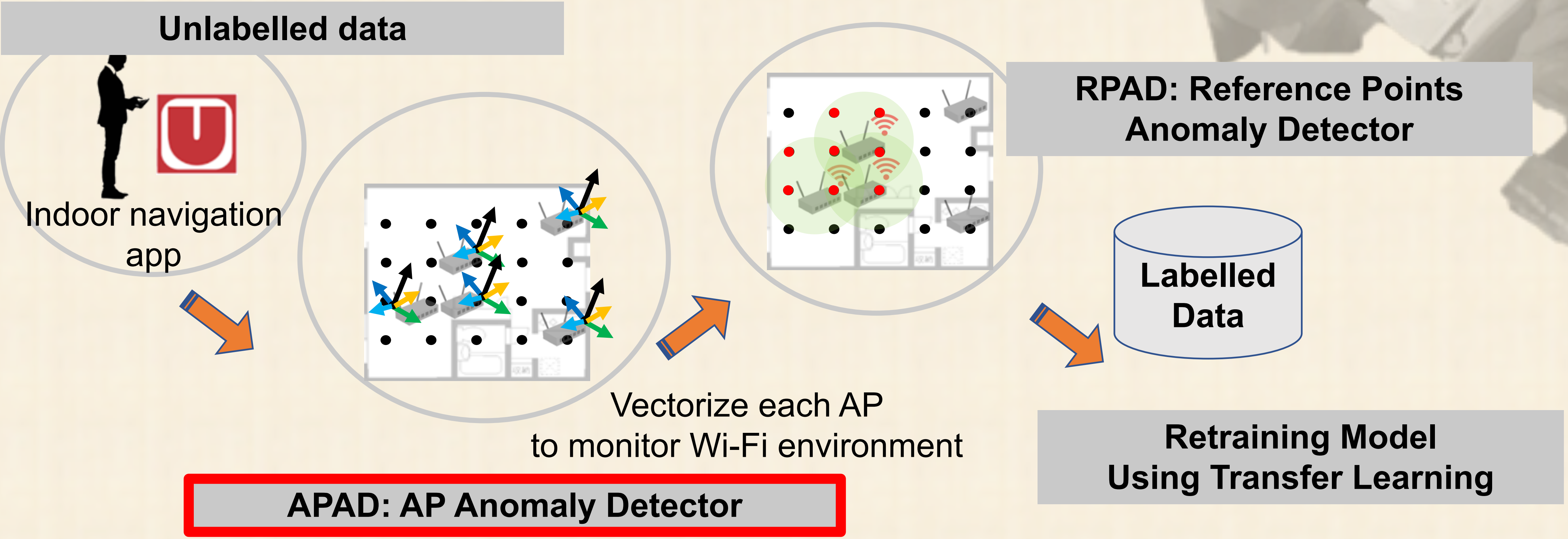
- Identify(detect) <u>reference points where environmentally changed</u>
  - Analyzing unlabelled data from off-the-shelf indoor navigation app



Umechika Navi: https://www.umechikanavi.jp/

No-Sweat Detective

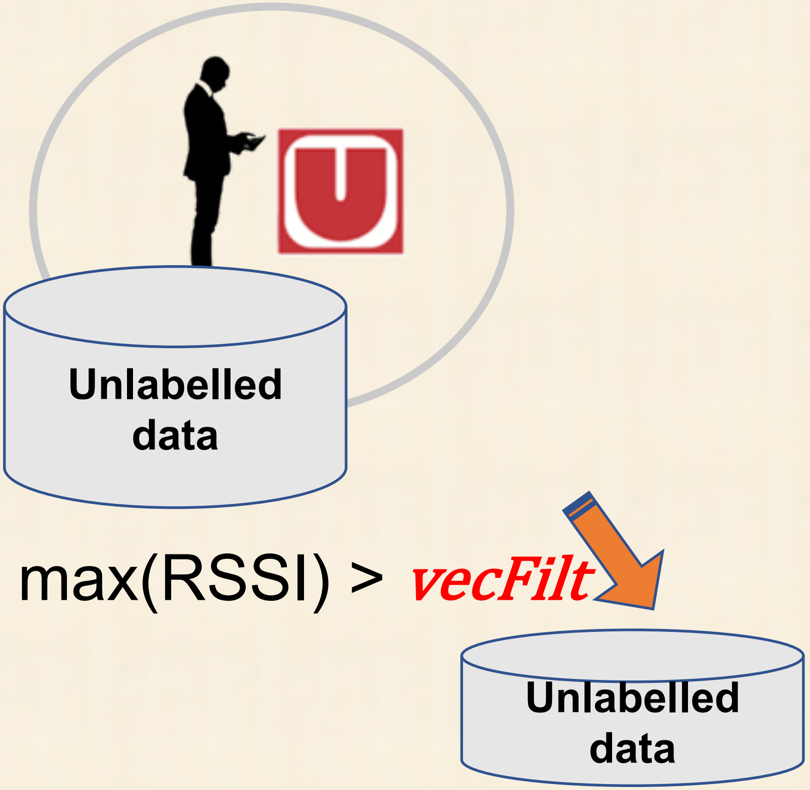
- Identify(detect) <u>reference points where environmentally changed</u>
  - Analyzing unlabelled data from off-the-shelf indoor navigation app



Umechika Navi: https://www.umechikanavi.jp/

APAD: AP anomaly detector

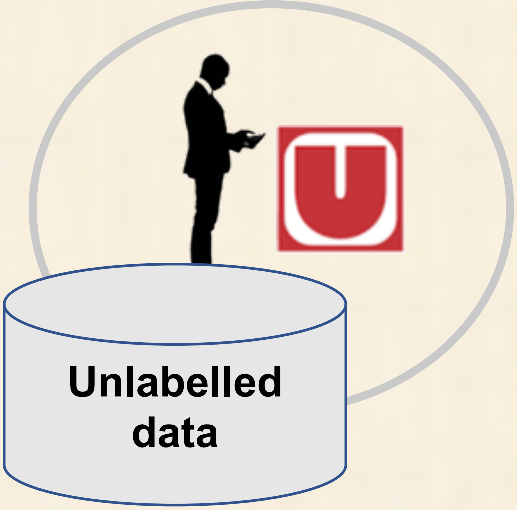
- Detect APs close to where Wi-Fi-environmentally changed
- Vectorize each AP to reflect each Wi-Fi environment

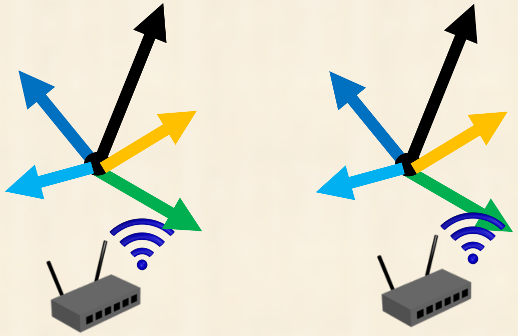


Filtered unlabelled data are observed close to a certain AP

APAD: AP anomaly detector

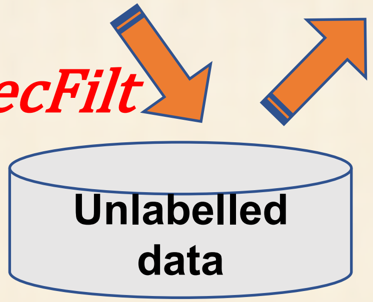
- Detect APs close to where Wi-Fi-environmentally changed
- Vectorize each AP to reflect each Wi-Fi environment





vecWidth to define size of vector

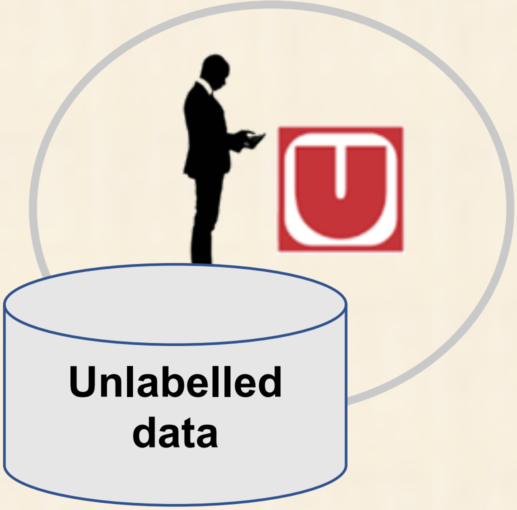
max(RSSI) > *vecFilt*

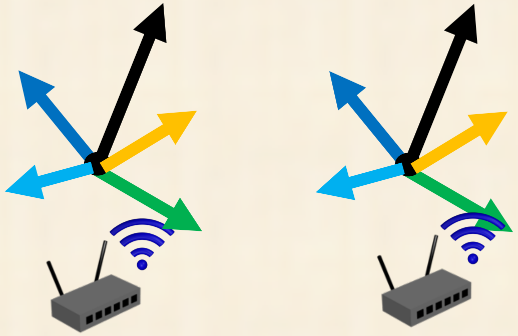


Filtered unlabelled data are observed close to a certain AP

APAD: AP anomaly detector

- Detect APs close to where Wi-Fi-environmentally changed
- Vectorize each AP to reflect each Wi-Fi environment



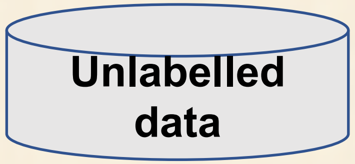


Calculate similarity over time for each AP vector

$$\cos(\vec{A},\vec{A}') = \frac{\vec{A}\cdot\vec{A}'}{|\vec{A}||\vec{A}'|} = \frac{\vec{A}}{|\vec{A}|}\cdot\frac{\vec{A}'}{|\vec{A}'|} = \frac{\sum_{i=1}^{|V|} A_i {A'}_i}{\sqrt{\sum_{i=1}^{|V|} A_i^2}\cdot\sqrt{\sum_{i=1}^{|V|} {A'}_i^2}} \quad \cdots (3)$$

vecWidth to define size of vector

max(RSSI) > *vecFilt*

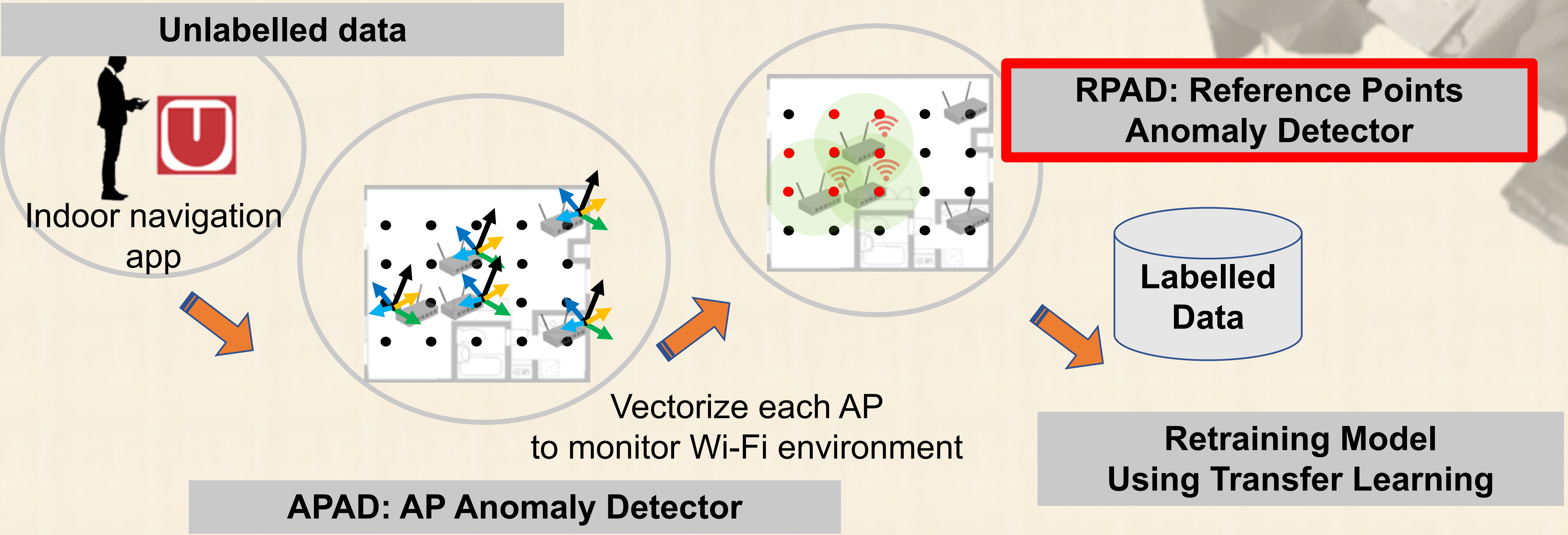


If similarity < *CosSim*:
detect the AP
as its surroundings changed

Filtered unlabelled data are observed close to a certain AP

No-Sweat Detective

- Identify(detect) reference points where environmentally changed
  - Analyzing unlabelled data from off-the-shelf indoor navigation app



Umechika Navi: https://www.umechikanavi.jp/

RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

Plural APs detected

RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

- Likely that small change affect the AP
- Nullify the AP + estimate its locus and collect labelled data

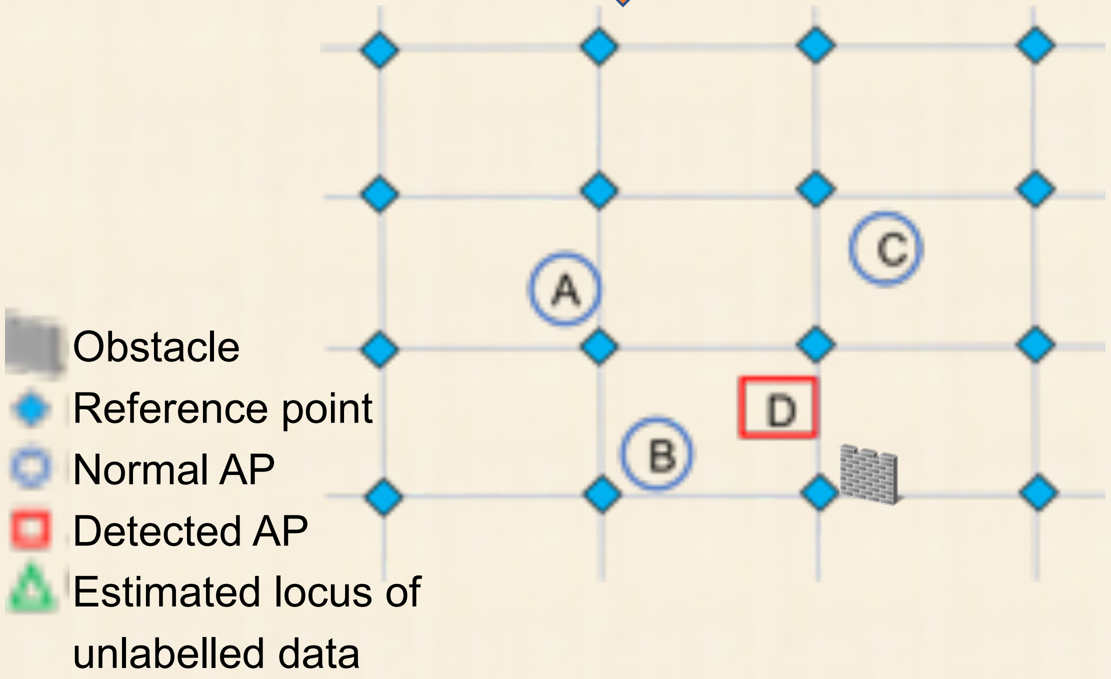
Plural APs detected

RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

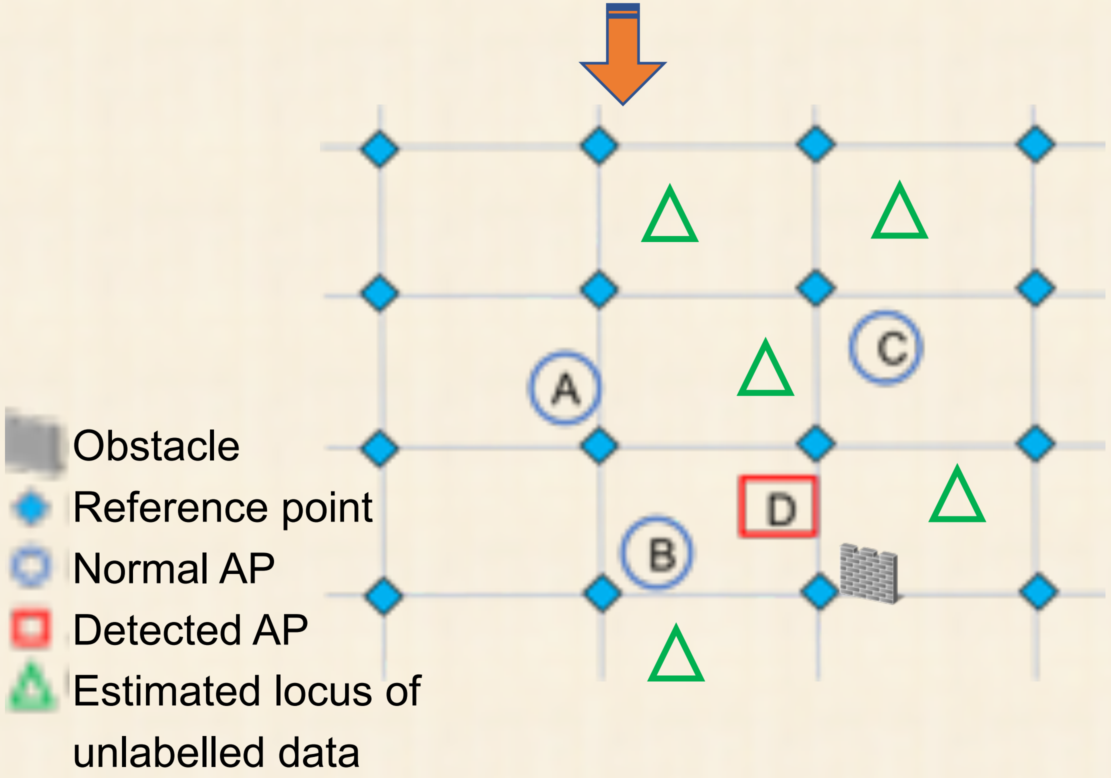
Plural APs detected



RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected



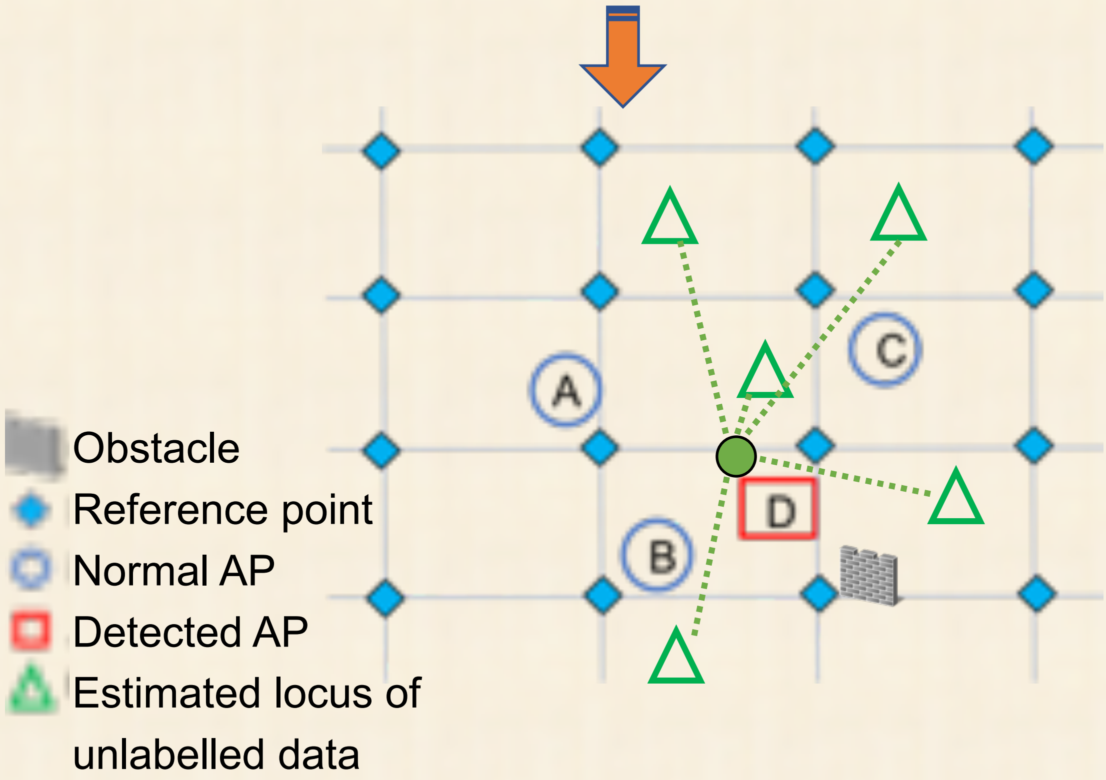
Plural APs detected

RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

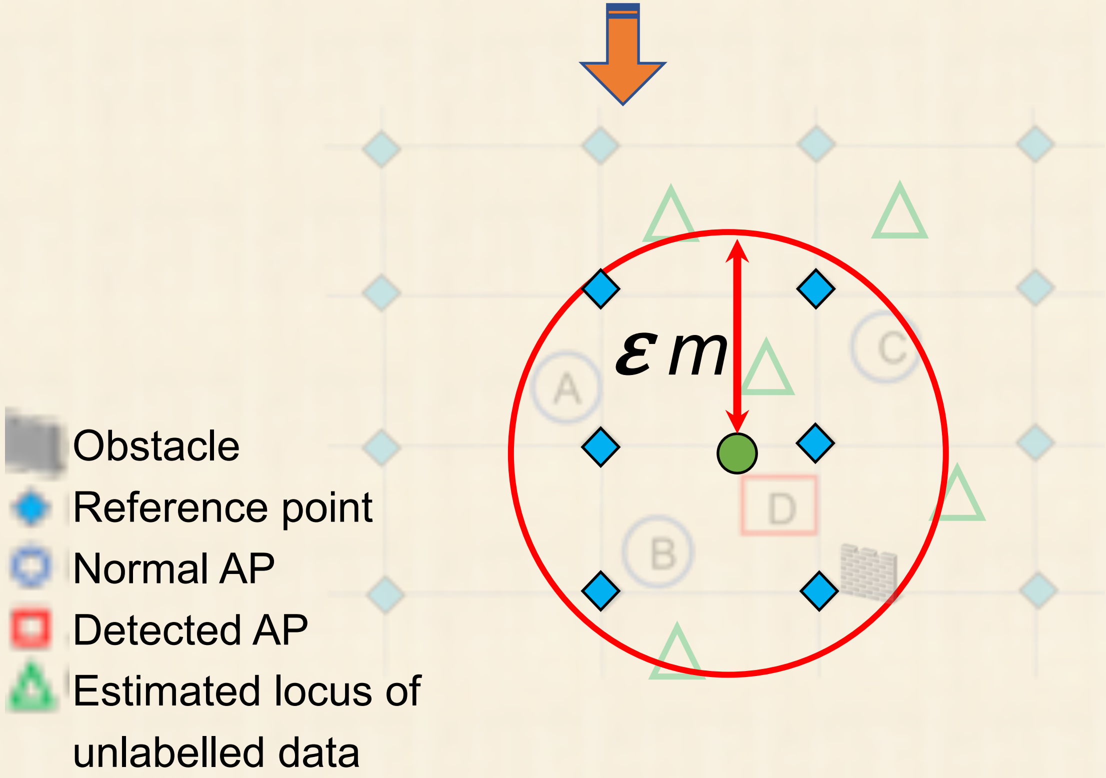
Plural APs detected



RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

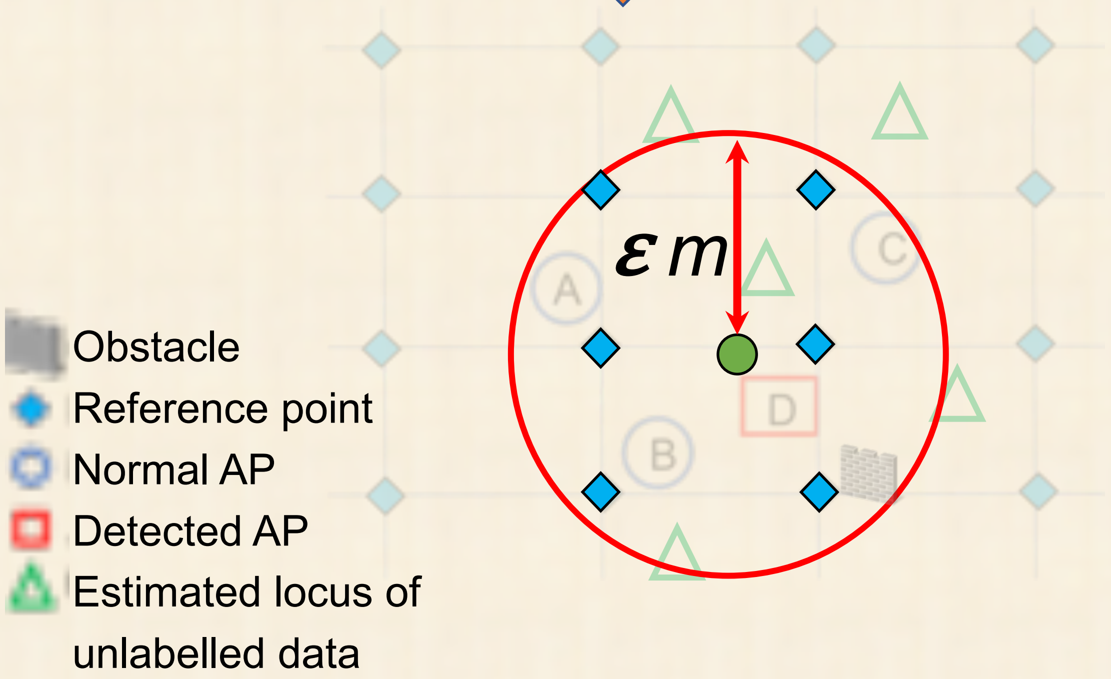


Plural APs detected

RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected



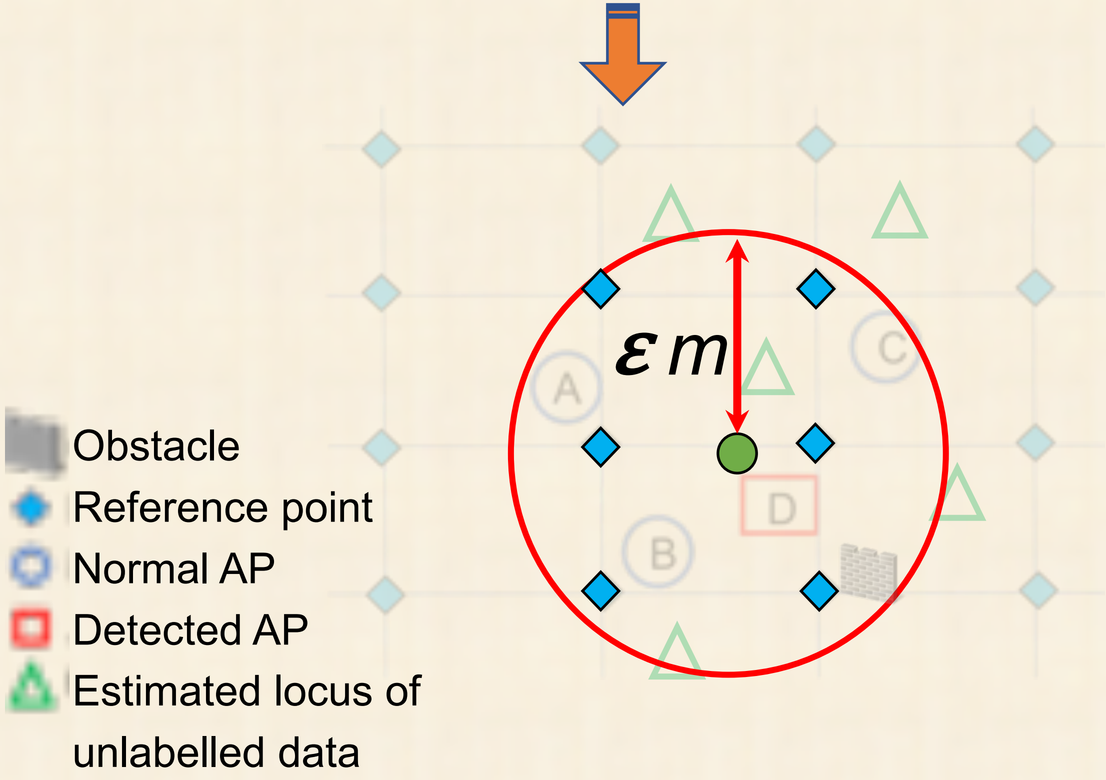
Plural APs detected

- Likely that huge change occurred
  - Spatial relationship of APs is clueless
- DBSCAN and collect labelled data

RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

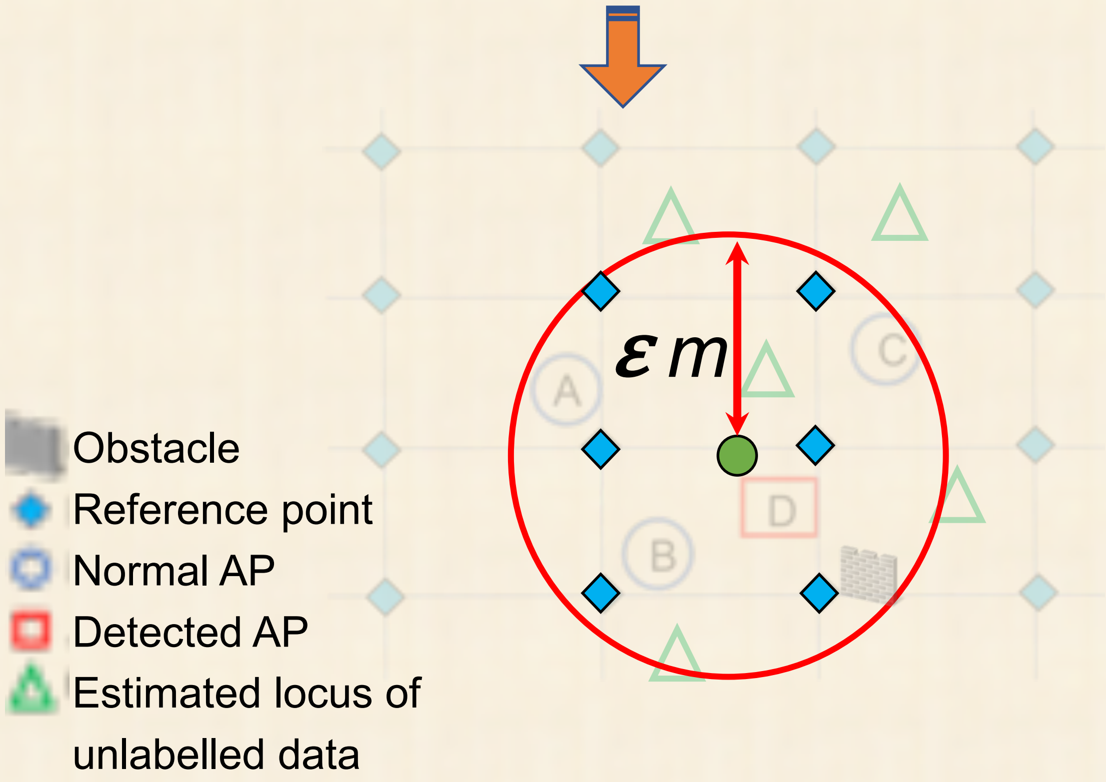


Plural APs detected

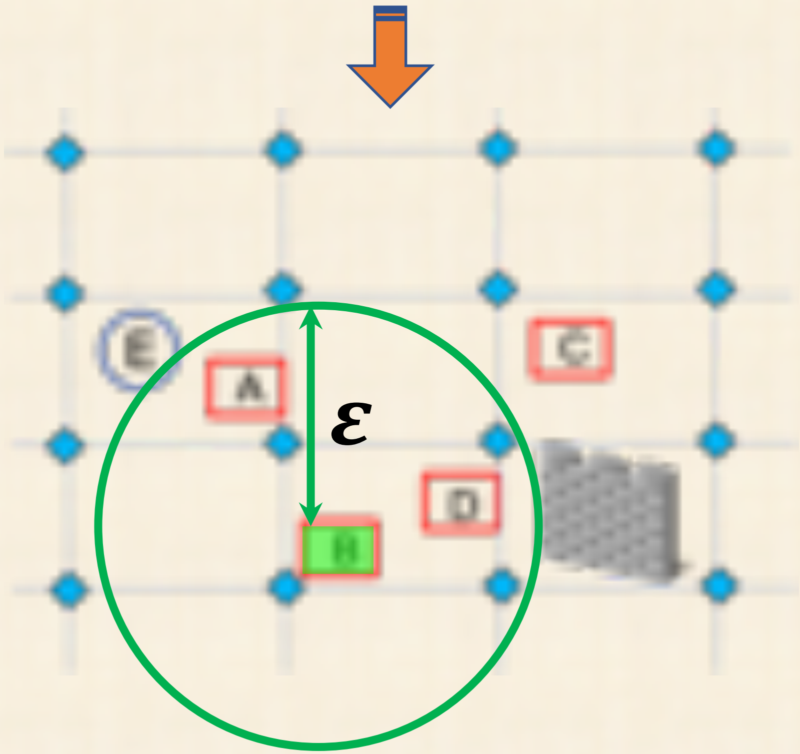
RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected



Plural APs detected

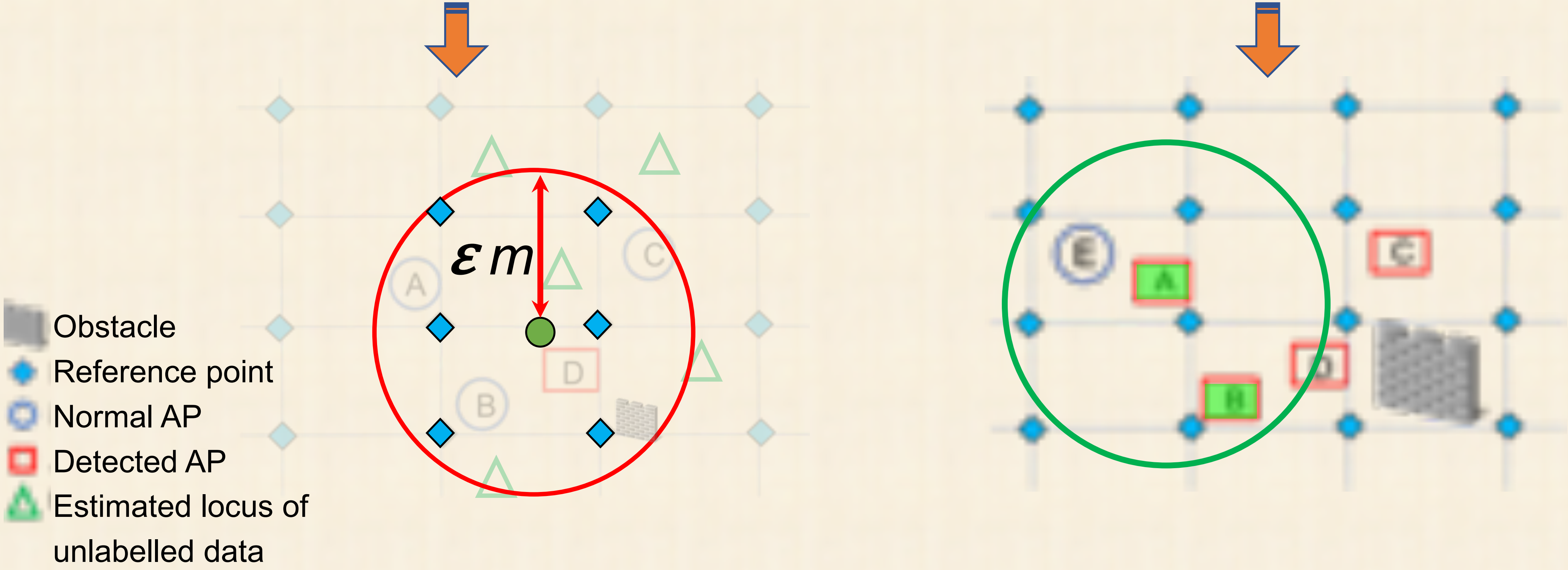


RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected

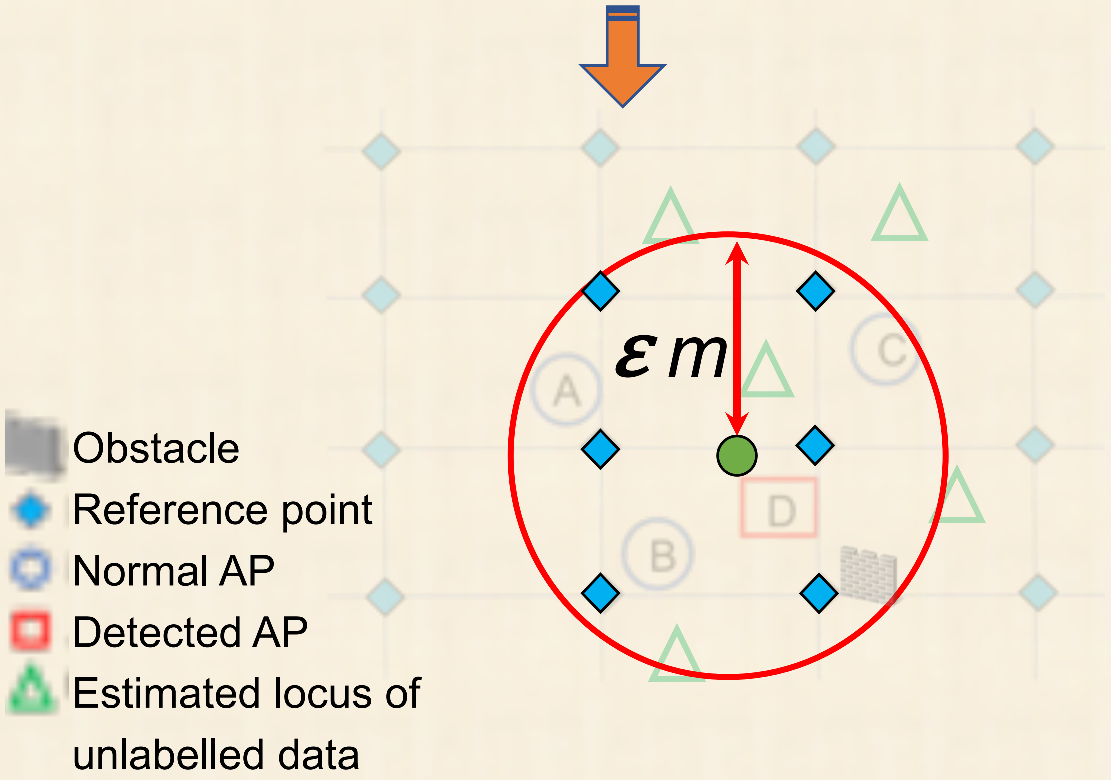
Plural APs detected



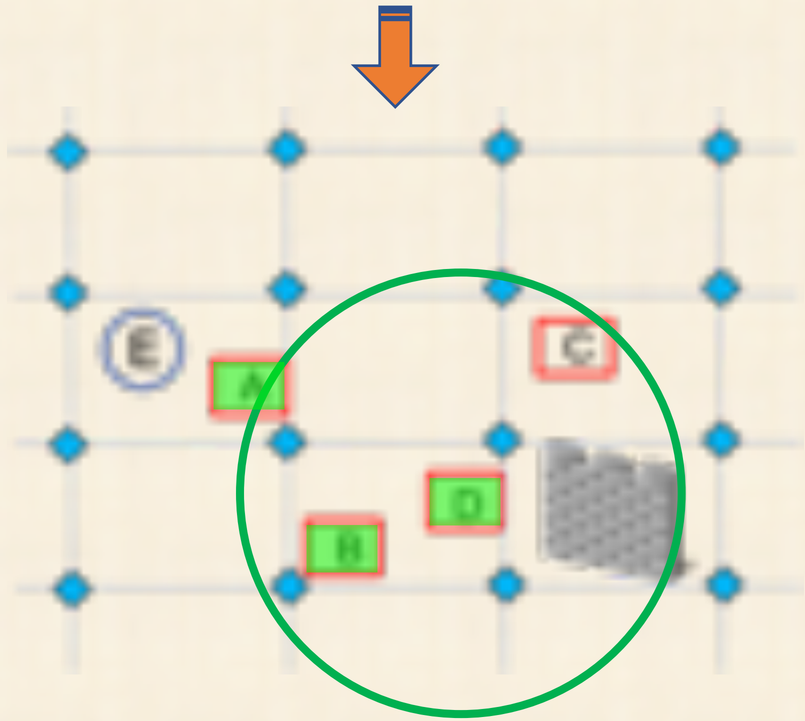
RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected



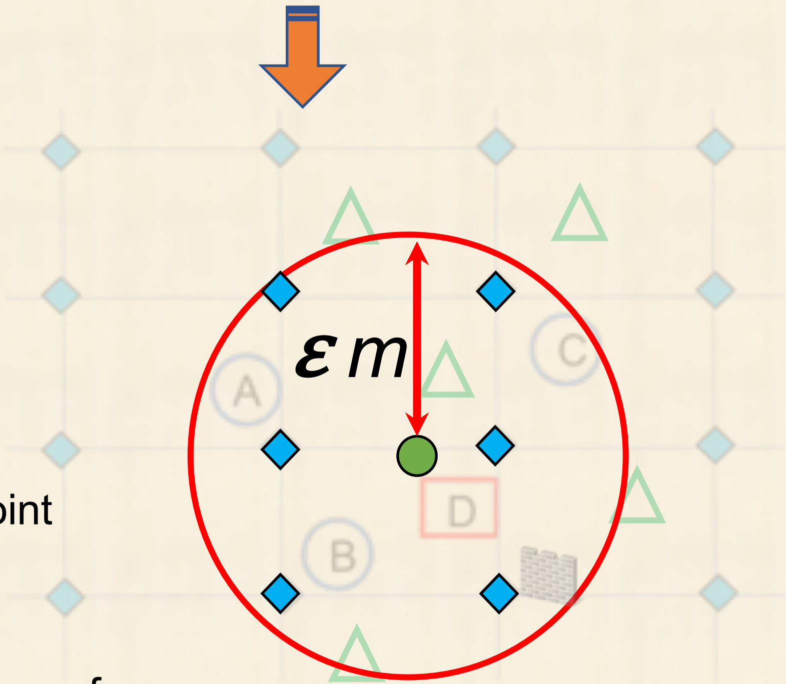
Plural APs detected



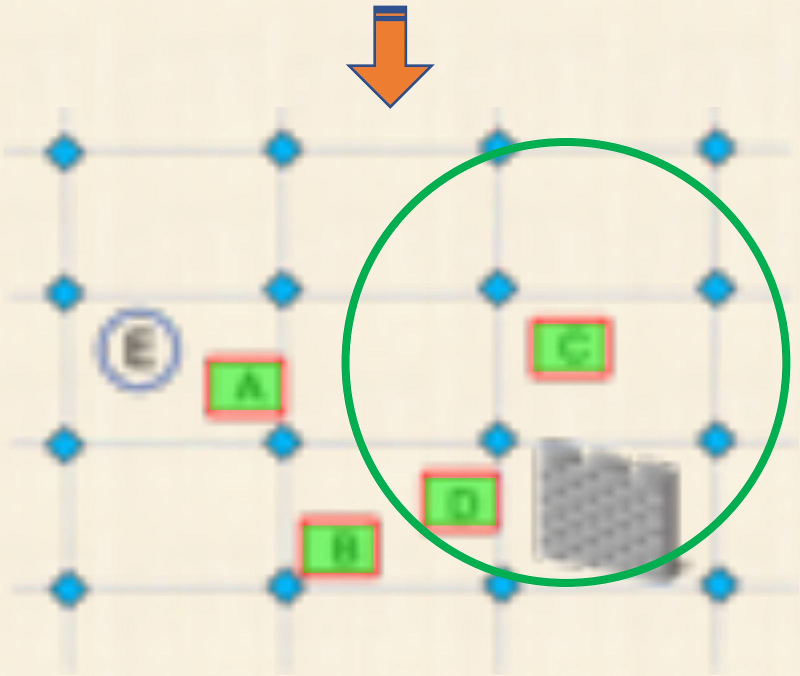
RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected



Plural APs detected

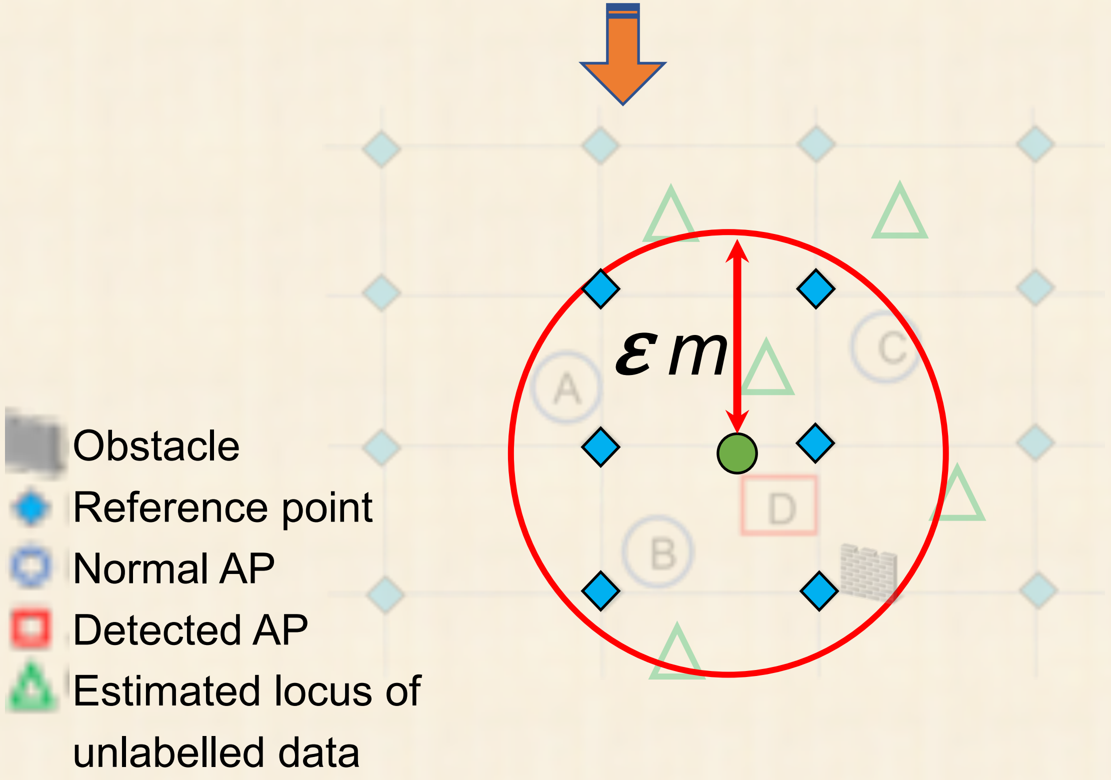


Obstacle
Reference point
Normal AP
Detected AP
Estimated locus of unlabelled data

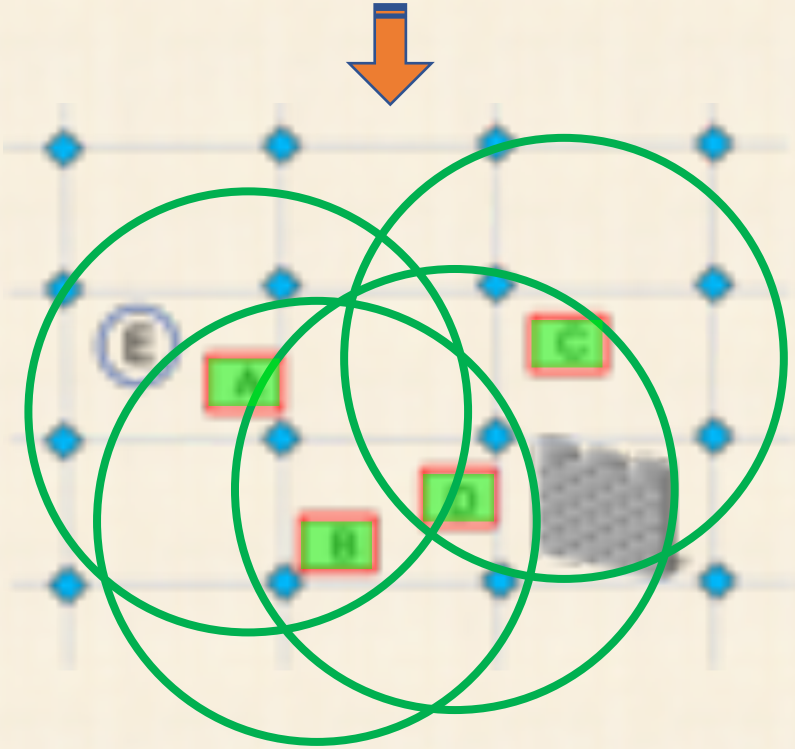
RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

Singular AP detected



Plural APs detected

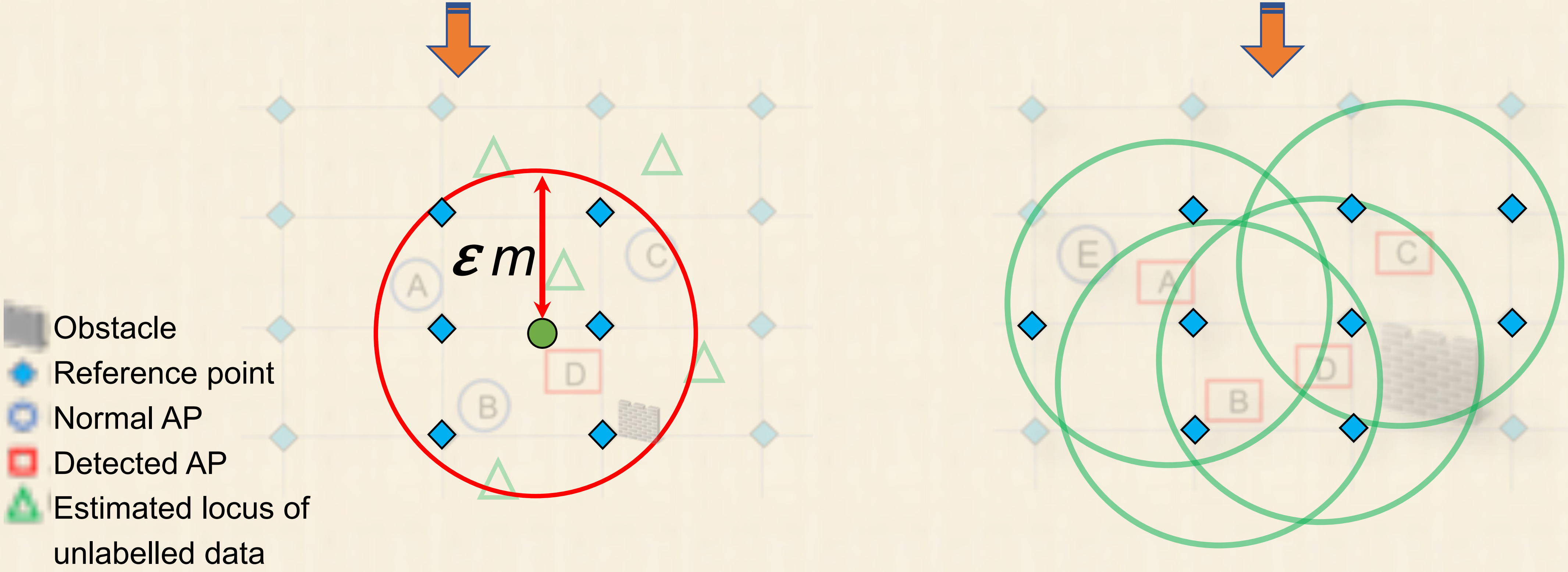


RPAD: Reference Points Anomaly Detector

- Identify reference points where Wi-Fi environmentally changed
- 2 cases to handle:

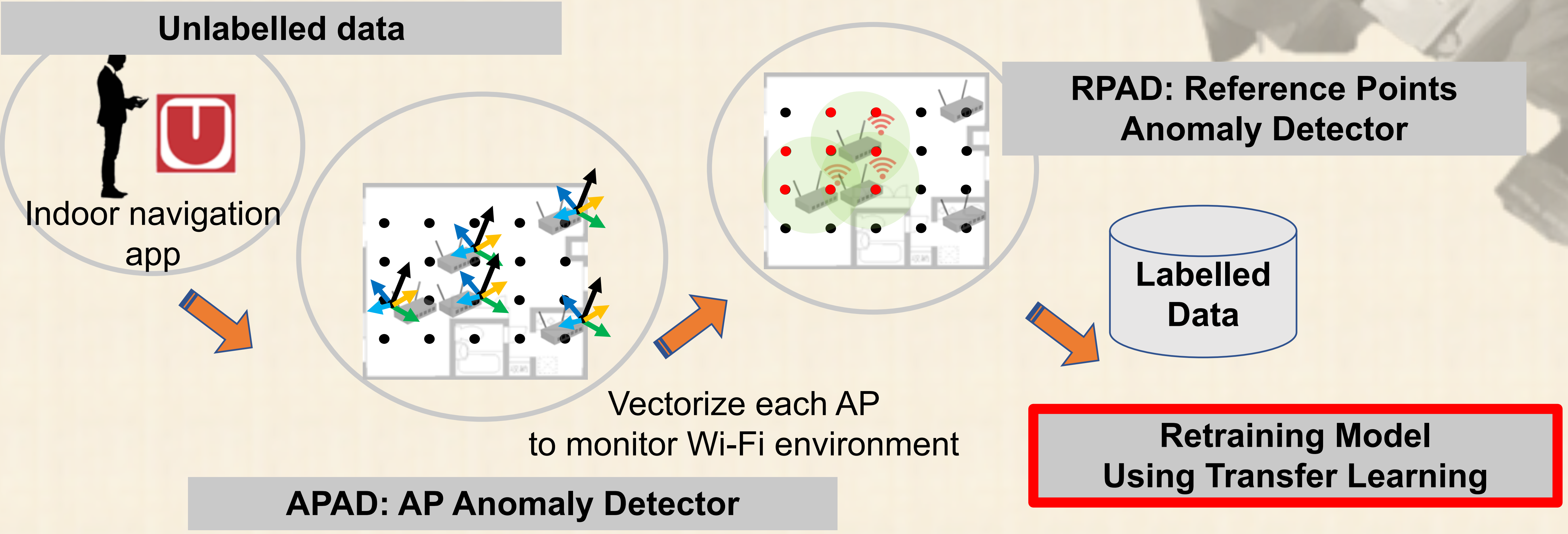
Singular AP detected

Plural APs detected



No-Sweat Detective

- Identify(detect) reference points where environmentally changed
  - Analyzing unlabelled data from off-the-shelf indoor navigation app



Umechika Navi: https://www.umechikanavi.jp/

Transfer Learning (TLM)

- Machine learning used for chronological datasets
- MixTrain method
  - Learn parameters θ from all the dataset
  - Update model by adding regularization term of L1 norm ($\sum_{i=1}^{|\theta|} |\theta i|$)
- Lasso method
  - Learn θ from variation of θ
  - Learn from L1 norm and regularization term $\left(\sum_{i=1}^{|\theta|} \left| \theta_i^{(k-1)} - \theta_i^{(k)} \right| \right)$
  - Regularization minimizing variation between θ at period $k-1$ and θ at period k

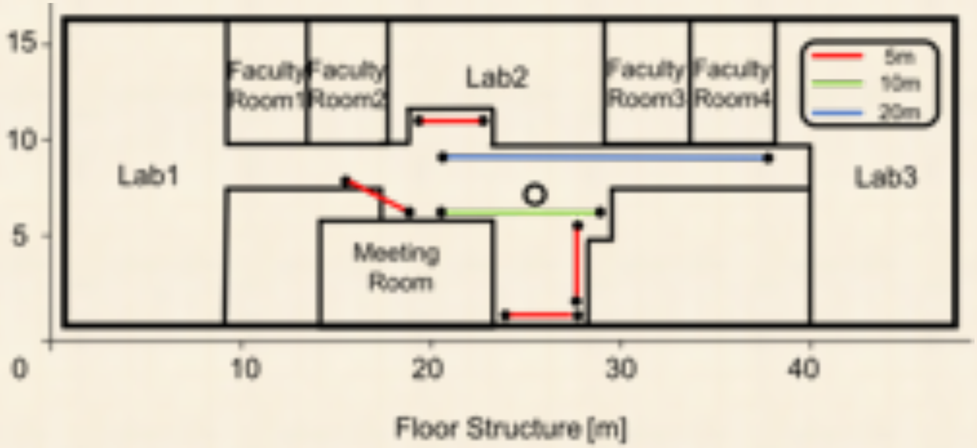
TLM can be replaceable

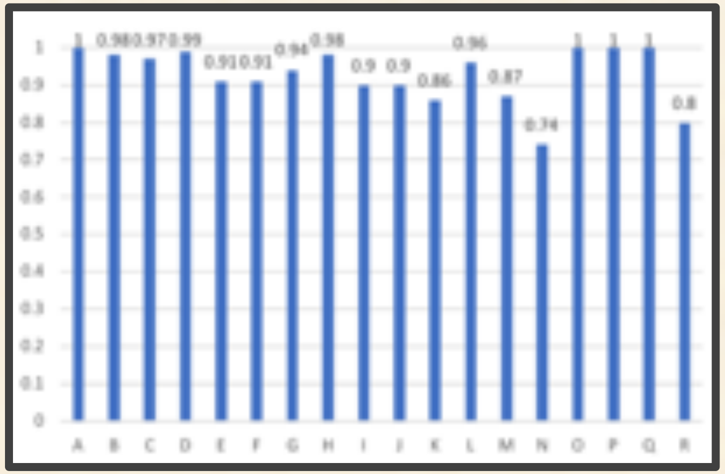
Outline

- Motivation
- State of the Art
- Methodology
- Evaluation
  - Laboratory environment
  - Wild environment
- Summary

Laboratory environment

- Simulate environmental changes by displacement of APs
  - Validate if anomalies are detected
- Settings
  - 5m*4, 10m*1, and 20m*1 displacements
  - 105 reference points with 1m interval
  - 2000 labelled data

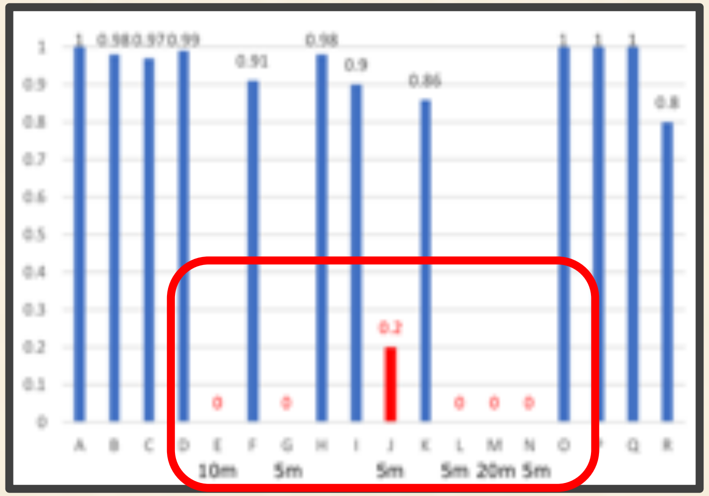




Un-simulated

Normal

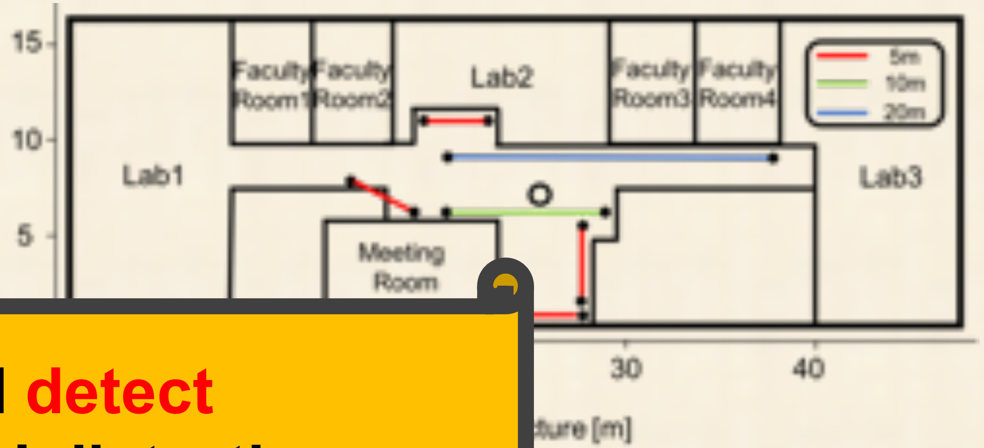
Anomalous



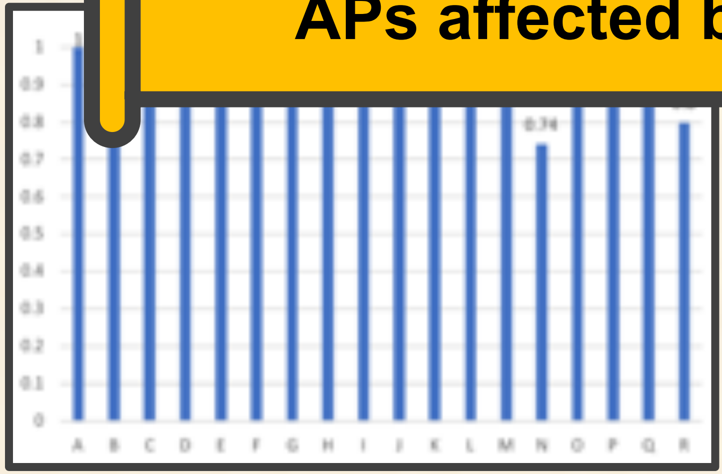
Simulated

Laboratory environment

- Simulate environmental changes by displacement of APs
  - Validate if anomalies are detected
- Settings
  - **5m***4, **10m***1, and **20m*1** displacements
  - 105 r
  - 2000



No-Sweat Detective could detect
APs affected by environmental distortion



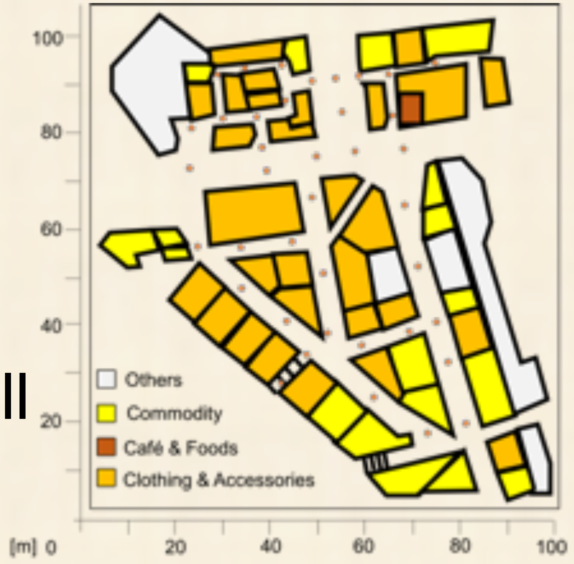
Un-simulated

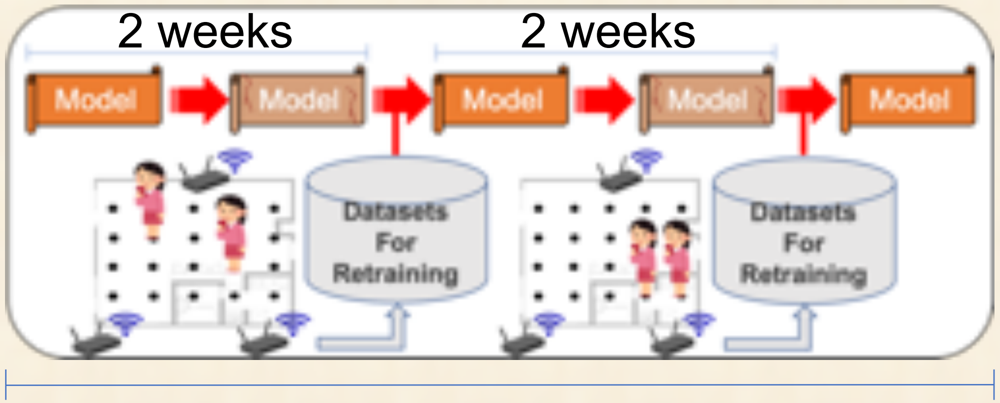
Anomalous

Simulated

Wild environment

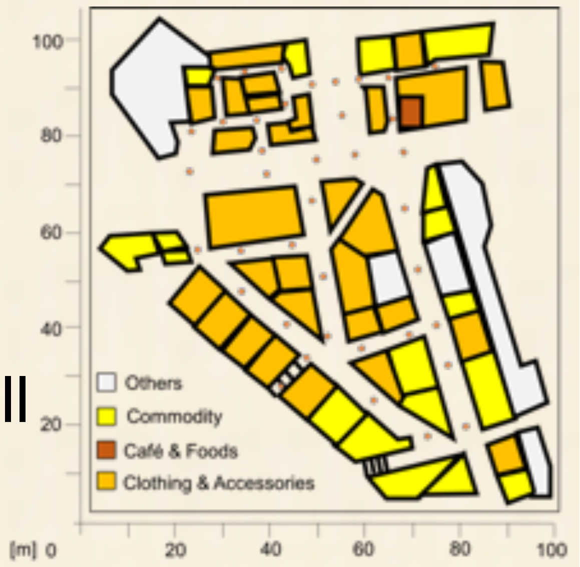
- Settings
  - Slide amount of reference points: 10, 20, 30, 60, 100% of all
  - 39 reference points with 8m interval
  - 764 unlabeled data, 2693 labelled data
    - Unlabelled data from indoor navigation app



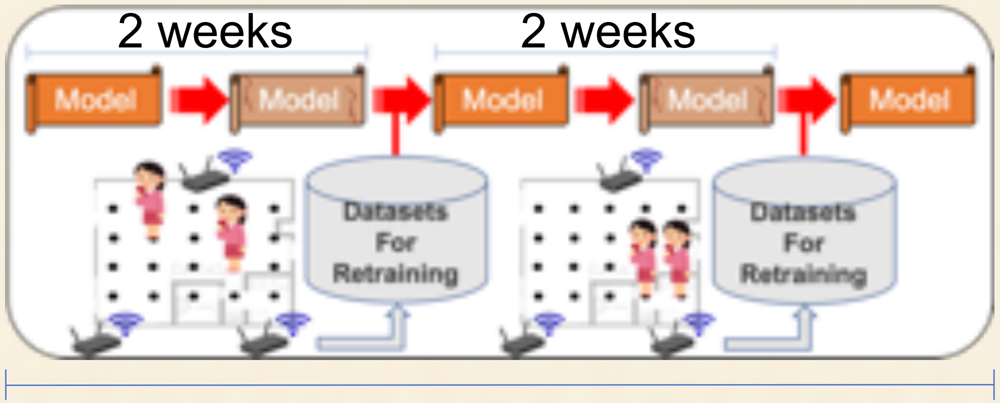


5 month-long observation

Wild environment



- Settings
  - Slide amount of reference points: 10, 20, 30, 60, 100% of all
  - 39 reference points with 8m interval
  - 764 unlabeled data, 2693 labelled data
    - Unlabelled data from indoor navigation app
- Evaluate the accuracy of the model at the end
  - Compare with Baseline and transfer-learning-but-random-sampling ways

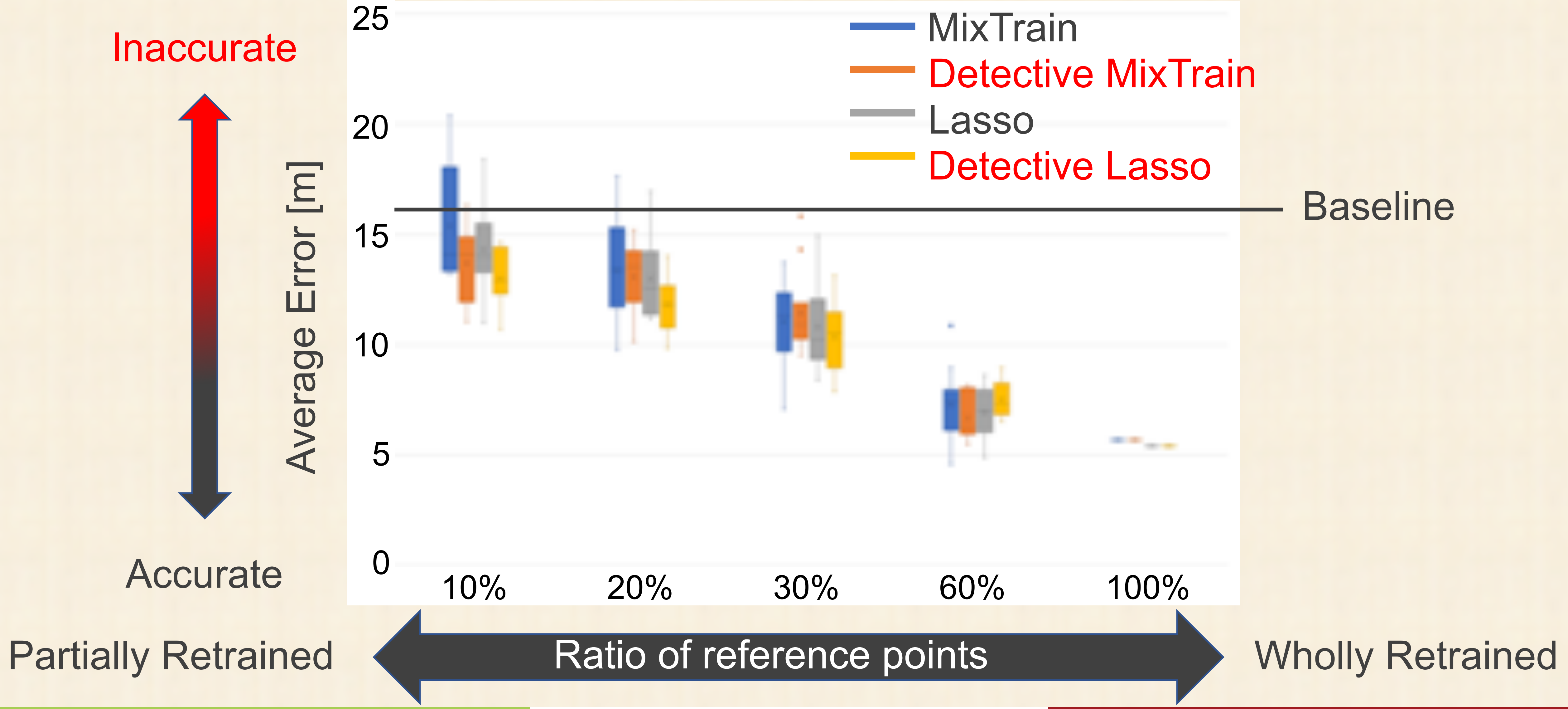


5 month-long observation

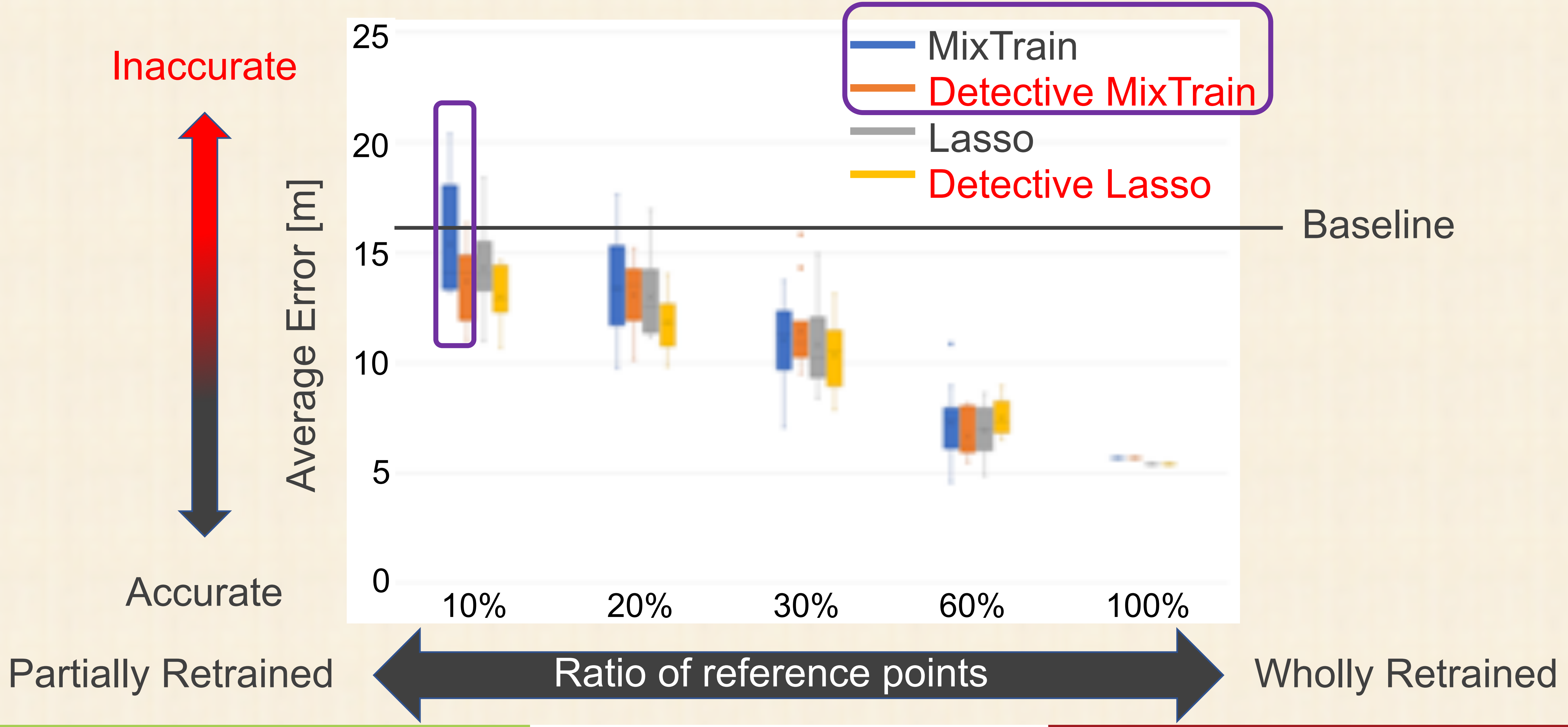
- Baseline: never retrained
- MixTrain: TLM
- Lasso: TLM

- **Detective MixTrain**: MixTrain + No-Sweat Detective
- **Detective Lasso**: Lasso + No-Sweat Detective

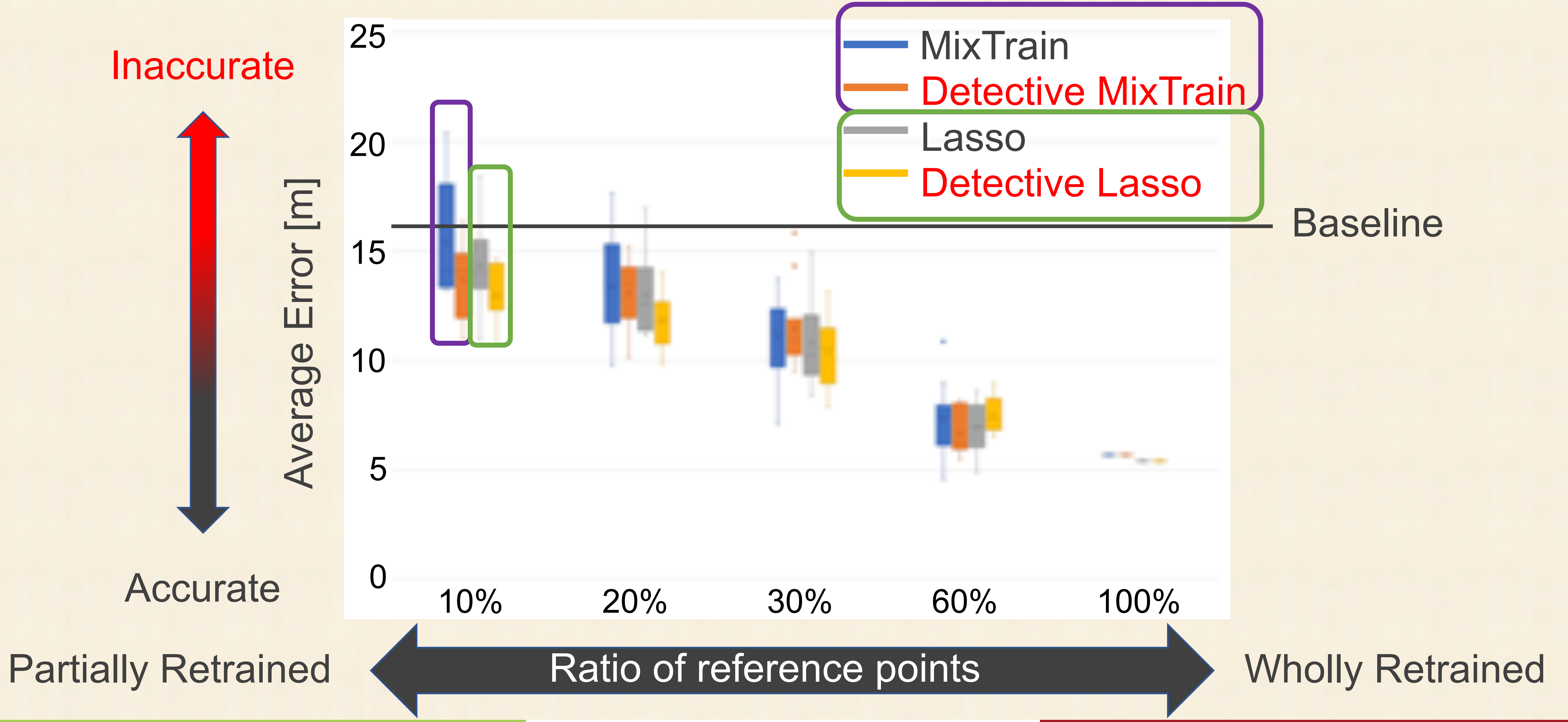
Results after 10 trials



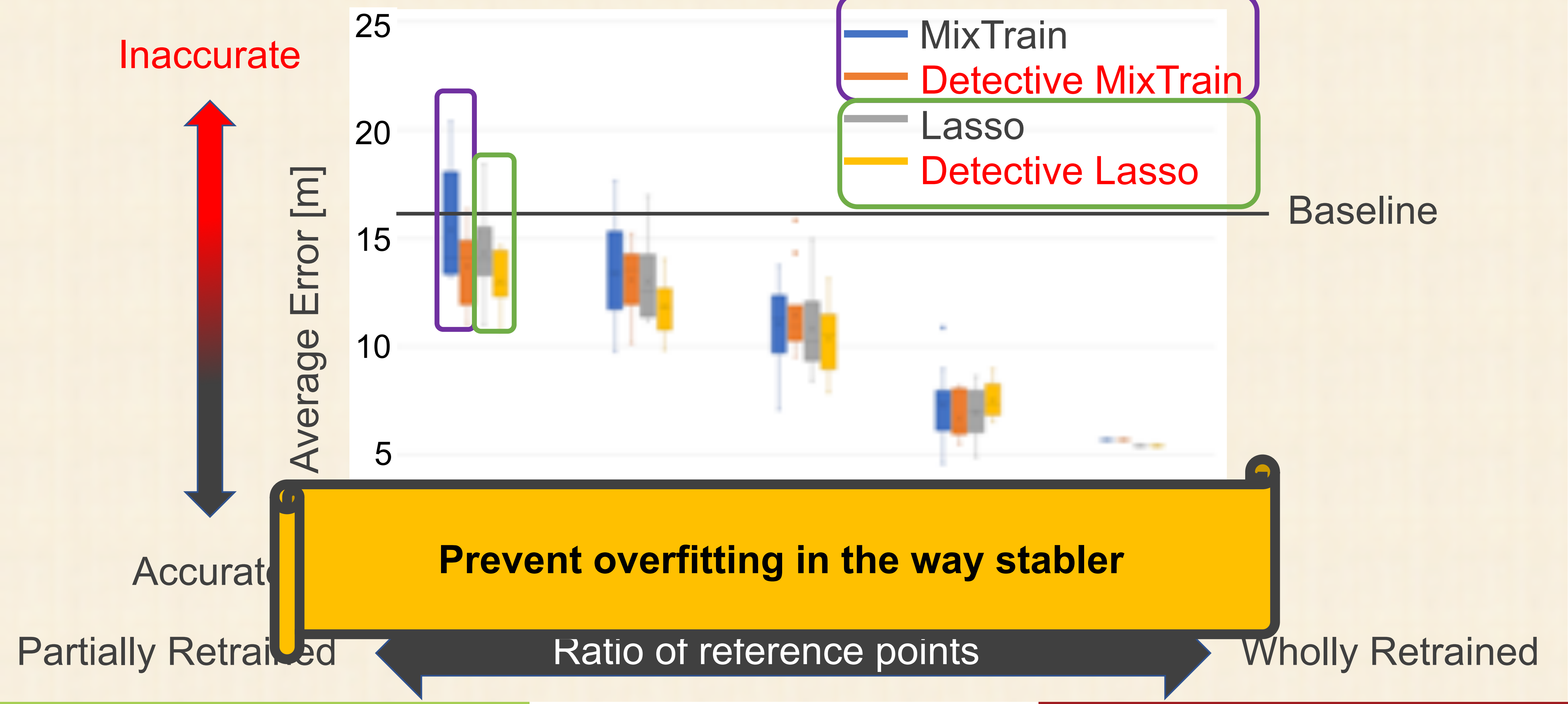
Results after 10 trials



Results after 10 trials



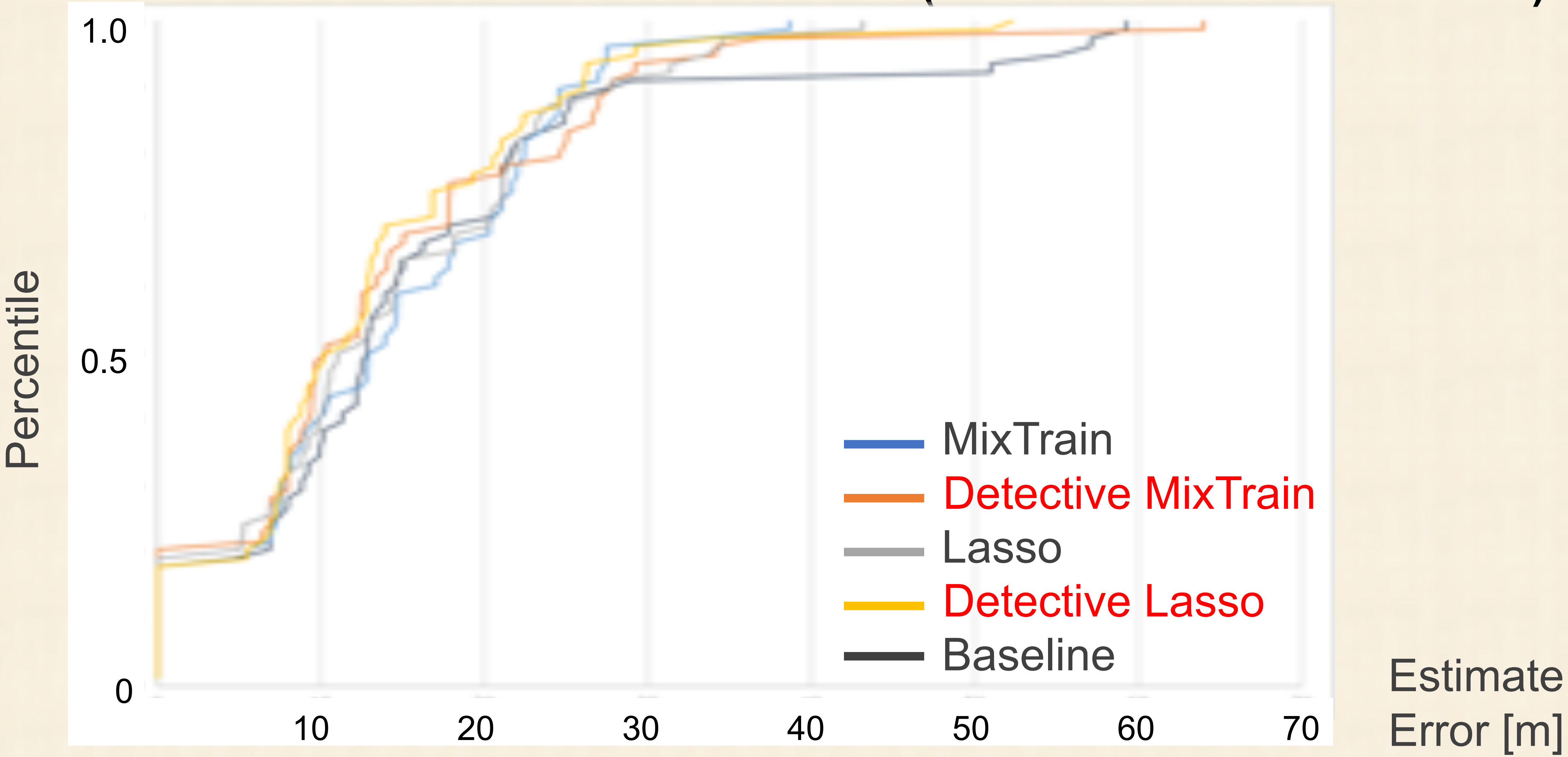
Results after 10 trials



Prevent overfitting in the way stabler

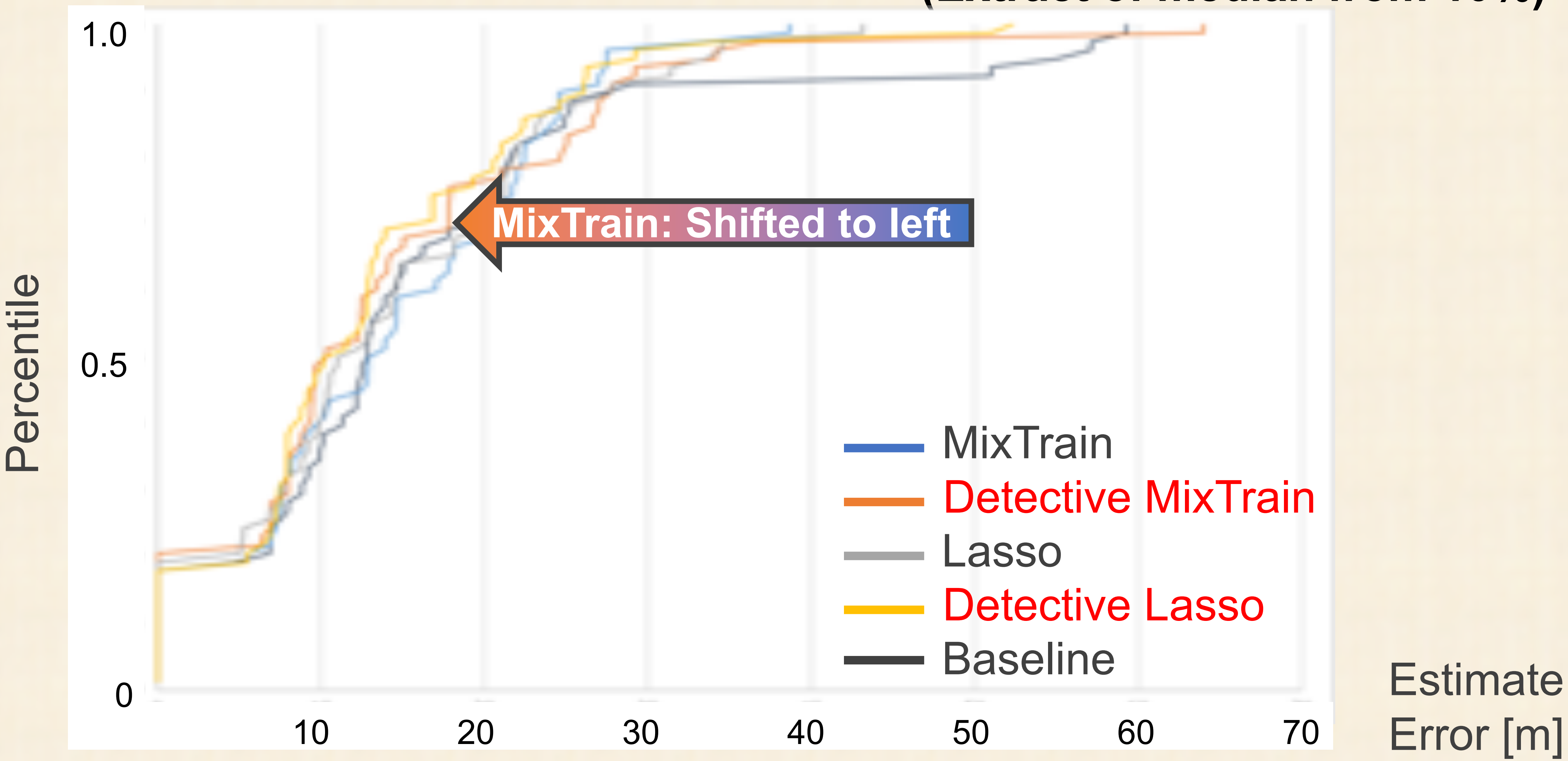
Cumulative Distribution of Estimate Error

(Extract of median from 10%)



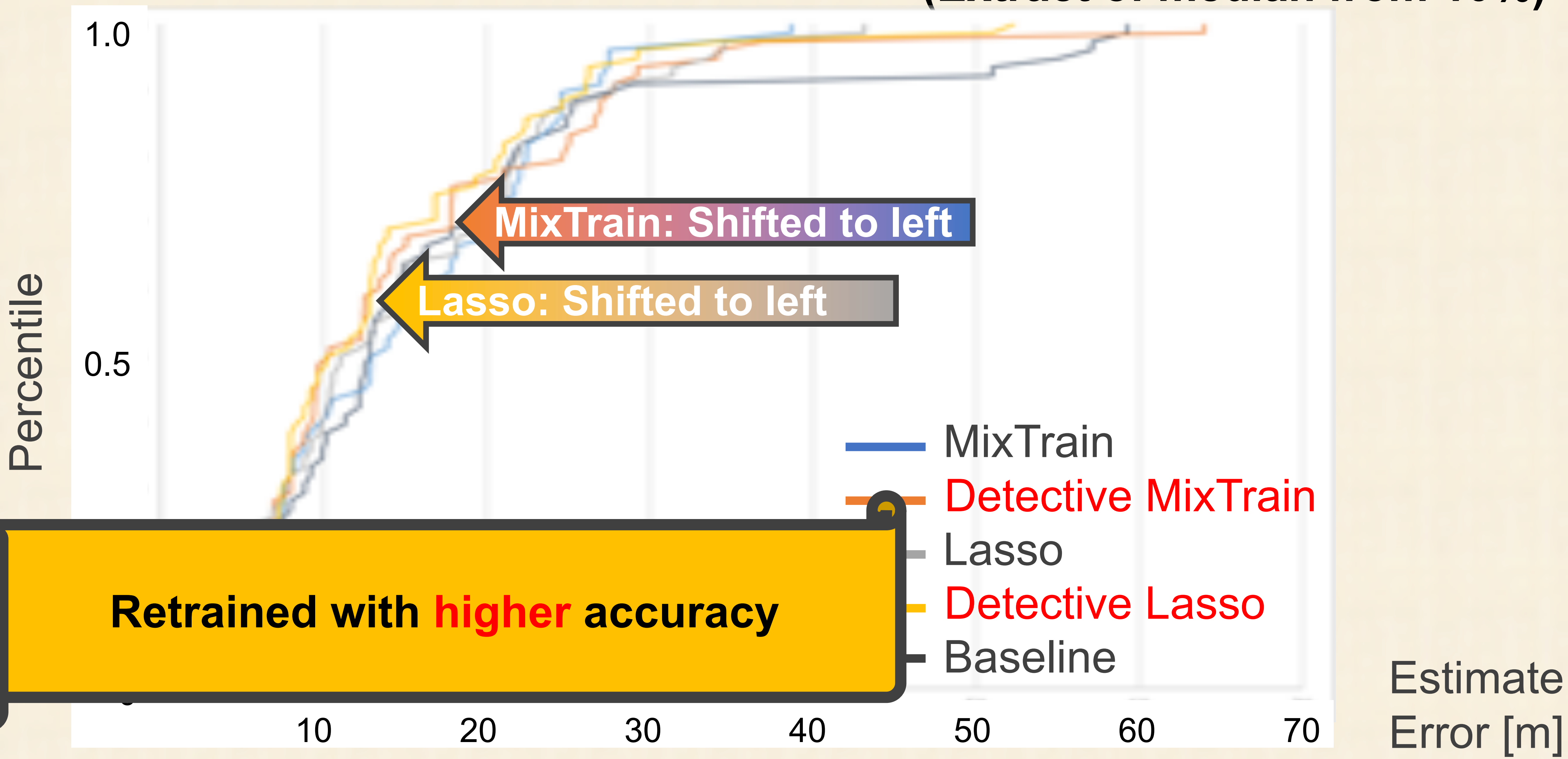
Cumulative Distribution of Estimate Error

(Extract of median from 10%)



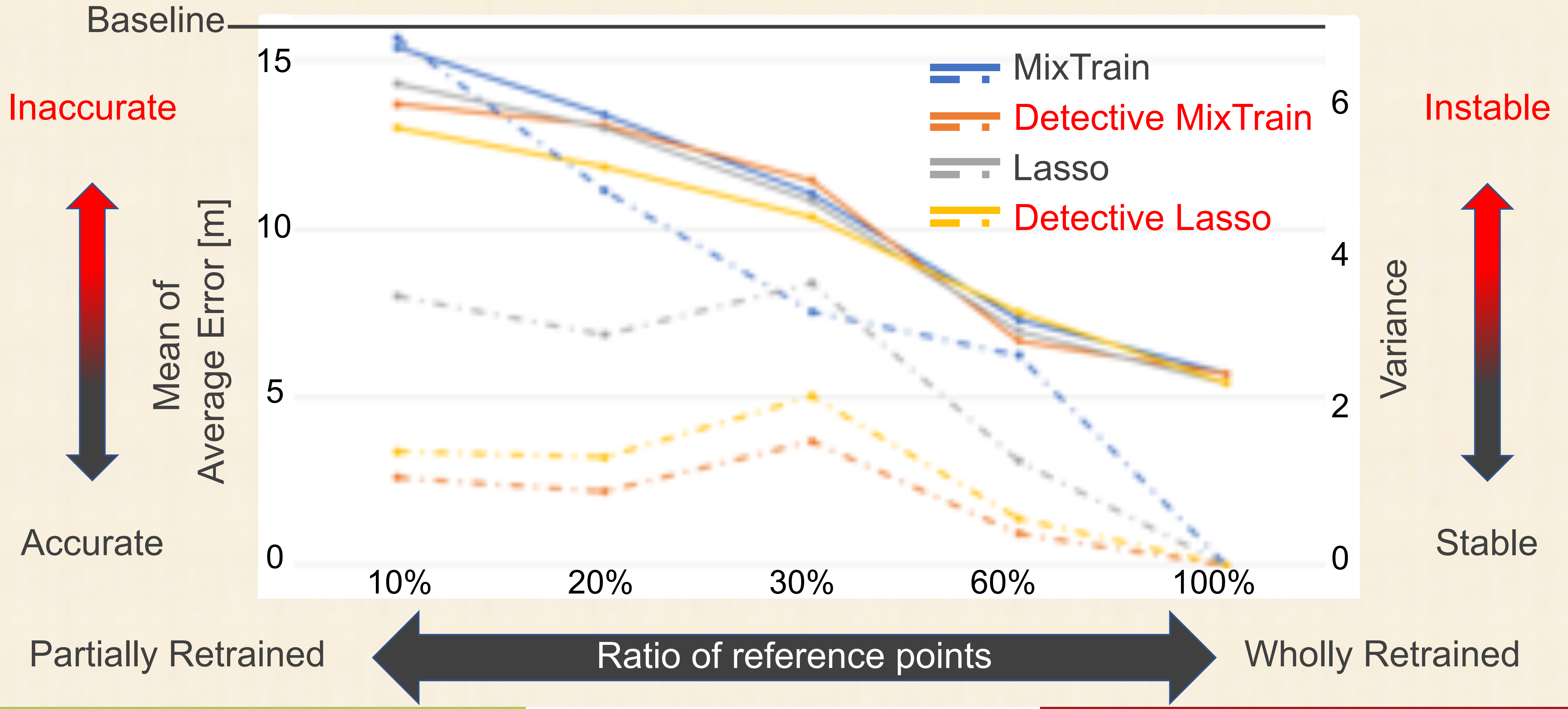
Cumulative Distribution of Estimate Error

(Extract of median from 10%)

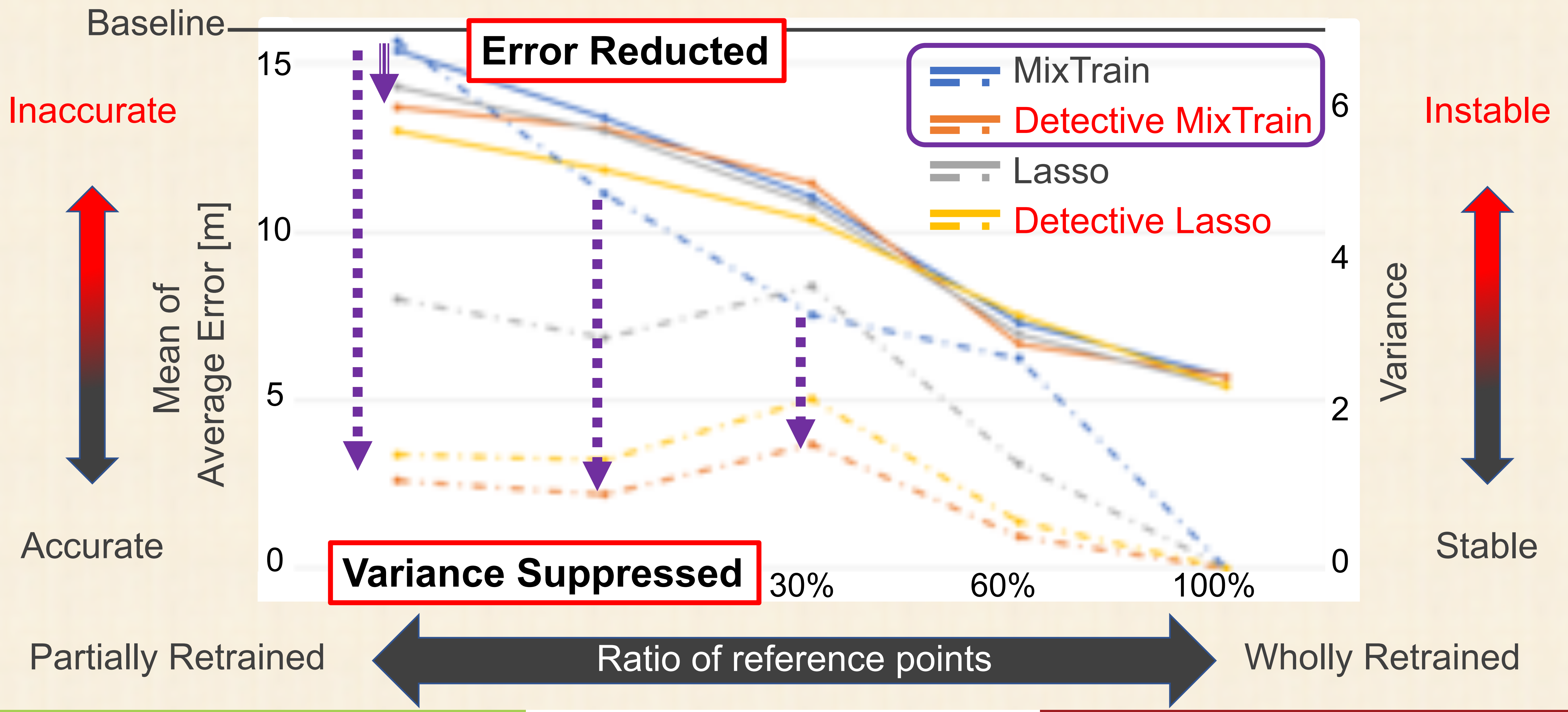


Retrained with higher accuracy

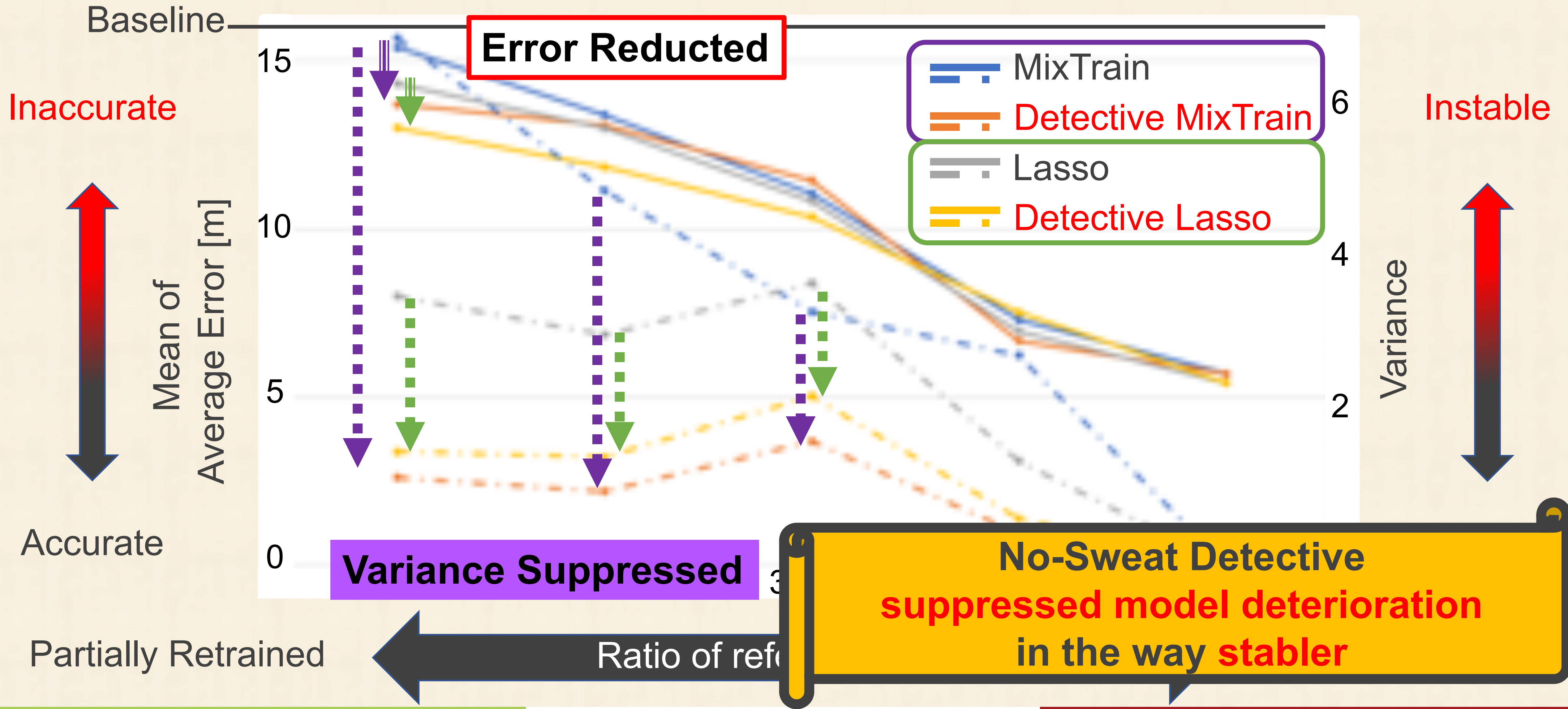
Means and Variance



Means and Variance



Means and Variance



No-Sweat Detective
suppressed model deterioration
in the way stabler

Outline

- Motivation
- State of the Art
- Methodology
- Evaluation
  - Laboratory environment
  - Wild environment
- Summary

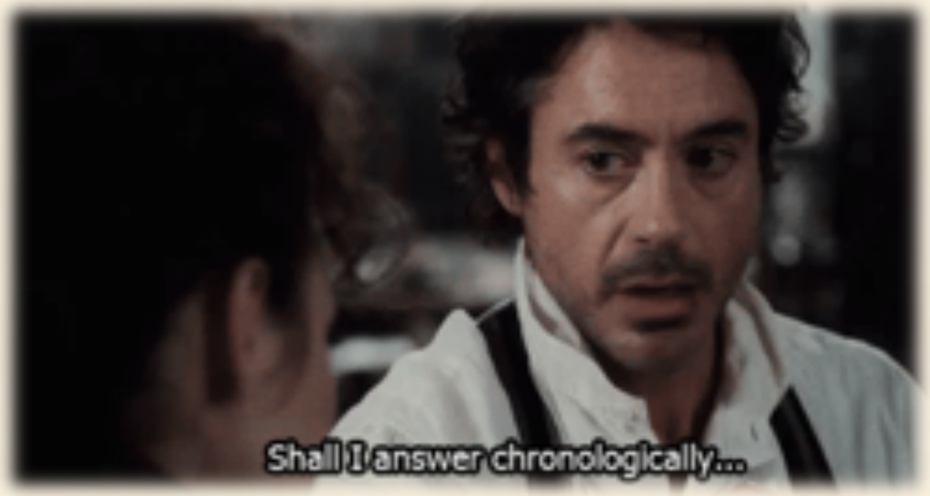
Summary

- Transfer learning is employed to retrain Wi-Fi localization model
- Additional Datasets are randomly selected for retraining
  - Result in overfitting and instable accuracy
- No-Sweat Detective
  - Identify reference points where environmentally changed
    - Analyzing unlabelled data obtained with no effort (sweat)
  - Prevent overfitting in the way stabler
- Outlook
  - Substitute unlabelled data as labelled data to retrain the model

Thank You!

Questions?

https://koheiyamamoto.net/



©Sherlock Holmes

Supplements

TABLE I

CONFUSION MATRIX OF DETECTED APS OUT OF DISPLACED APS IN SLIDING SCALE {VECFILT AND VECWIDTH}

VecFilt\VecWidth	-25	-30	-35	-40	-45	-50	-55	-60
-25	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
-30	-	4/4	4/4	4/4	3/4	2/4	1/4	0/4
-35	-	-	**6/6**	**6/6**	4/6	2/6	1/6	0/6
-40	-	-	-	**6/6**	4/6	2/6	1/6	1/6
-45	-	-	-	-	4/6	2/6	1/6	1/6
-50	-	-	-	-	-	2/6	1/6	1/6
-55	-	-	-	-	-	-	1/6	1/6
-60	-	-	-	-	-	-	-	1/6

Motivation

- Indoor position is now of considerable importance for IoT
  - To capture user contexts and her interests
- Fingerprinting localization based on Wi-Fi RSSI is major
  - Workflow consists of training and operational phases

I. Primary signal model created from primary datasets of labeled fingerprint
II. Estimate her location by comparing her current fingerprint with model

- **But** accuracy of model deteriorates with age
  - Caused by environmental changes
    - Movement of objects, construction of obstacles, automatic power adjustment, etc.

Requires **calibration** of model at certain periodic cycle

- Laborious effort and time-consuming overhead

State of the Art 1/2

- Reconstruction of model with less or no effort
  - TuRF, QRFC, and UnLoc can obtain labeled fingerprint installing sensors

- Detection of signal changes in wireless networks
  - Song et al. showed detecting node redeployment focusing neighborship
  - Ohara et al. showed detecting environmental changes observing CSI (Wi-Fi channel state information)
  - Meng et al. proposed probabilistic algorithm to detect distortion

State of the Art 1/2

- Reconstruction of model with less or no effort
  - TuRF, QRFC, and UnLoc can obtain labeled fingerprint installing sensors

Forcing user to put on sensors is far from practical use

- Detection of signal changes in wireless networks
  - Song et al. showed detecting node redeployment focusing neighborship

Forcing environment to have installment is not pragmatic

  - Ohara et al. showed detecting environmental changes observing CSI (Wi-Fi channel state information)

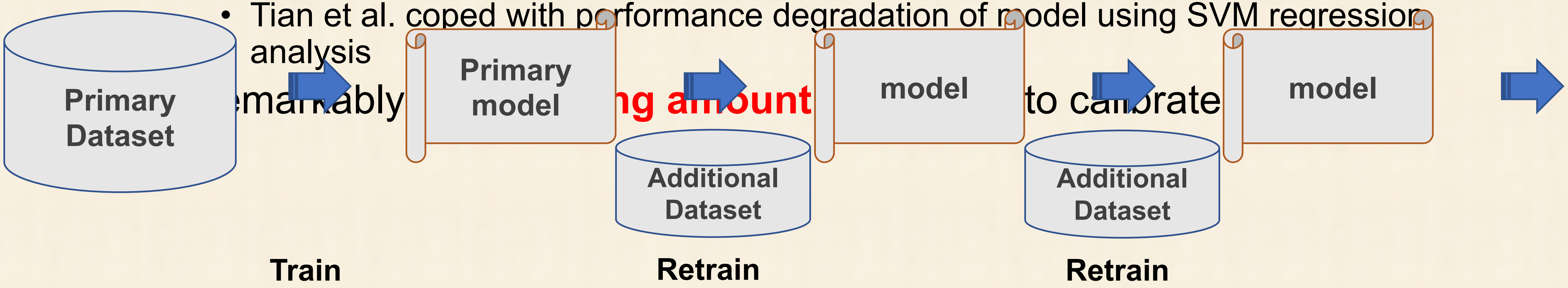
Requires pairs of connection and CSI is scarcely used

  - Meng et al. proposed probabilistic algorithm to detect distortion

Only proven in test-bed and does not target calibration

State of the Art

- Major current is to employ **transfer learning**[1)]
  - Reflects current Wi-Fi environment in model
  - Retrains model with adding a small amount of labeled fingerprint
    - Yang et al. proved higher accuracy with much less calibration effort
    - Yin et al. materialized regression and model-tree based algorithm
    - Tian et al. coped with performance degradation of model using SVM regression analysis
  - emarkably ... **ng amount** ... to calibrate



[1]: Pengcheng Wu and Thomas G. Dietterich. “Improving SVM Accuracy by Training on Auxiliary Data Sources.”

State of the Art

- Major current is to employ **transfer learning**[1)]
  - Reflects current Wi-Fi environment in model
  - Retrains model with adding a small amount of labeled fingerprint
    - Yang et al. proved higher accuracy with much less calibration effort
    - Yin et al. materialized regression and model-tree based algorithm
    - Tian et al. coped with performance degradation of model using SVM regression analysis

- Remarkably suppressing amount of dataset to calibrate model

- These methods pick up additional dataset (labeled fingerprint) randomly or comprehensively
  - Spilled into destabilization of accuracy recovery at every calibration
  - Recovery rate heavily relies on selection of labeled fingerprint

Prim
Data

Train Retrain Retrain

[1]: Pengcheng Wu and Thomas G. Dietterich. "Improving SVM Accuracy by Training on Auxiliary Data Sources."

State of the Art

- Major current is to employ **transfer learning**[1)]
  - Reflects current Wi-Fi environment in model
  - Retrains model with adding a small amount of labeled fingerprint
    - Yang et al. proved higher accuracy with much less calibration effort
    - Yin et al. materialized regression and model-tree based ... m
    - Tian et al. coped with performa... SVM regressi... analys...
  - Remarkab...

Prim...

Labeled fingerprint should be convergently collected from specific (anomalous) reference points

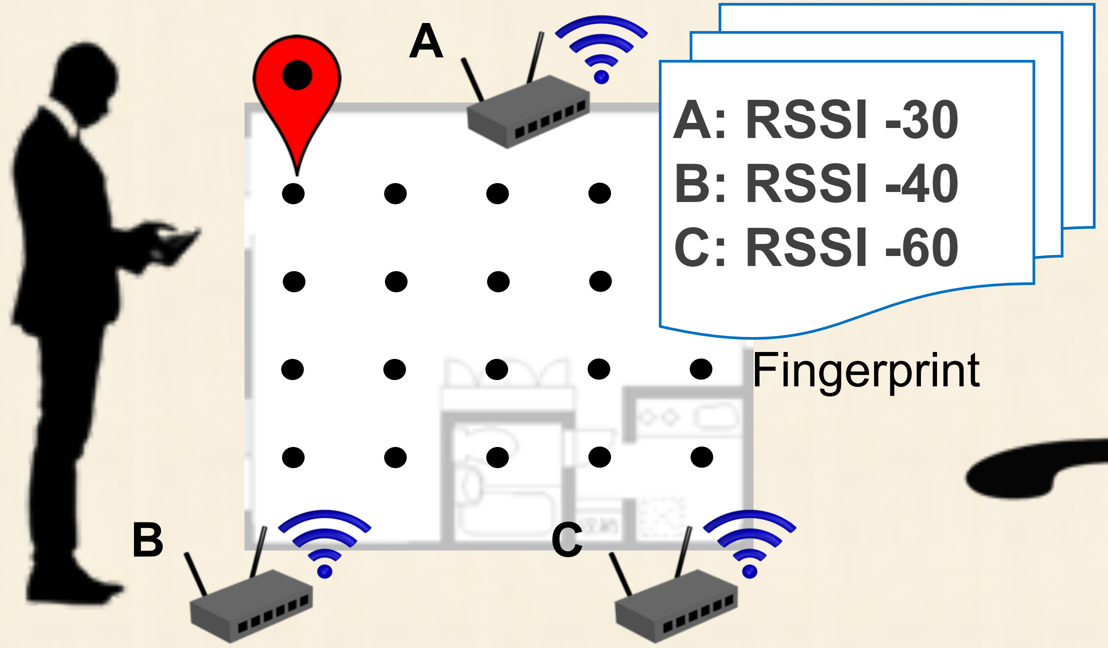
Tra... ...rain **Retrain**

[1]: Pe... Wu and Thomas G. Dietterich. "Improving SVM Accuracy by Training on Auxiliary Data Sources."

Two Types of Fingerprint

Labeled

- Used for training primary model
- Small amount used for retraining model
- Manual collection by administrator



Unlabeled

- Collected from user of location services
- Automatically but not labeled

Plenty

Two Types of Fingerprint

Identify reference points where environmentally changed

Unlabeled

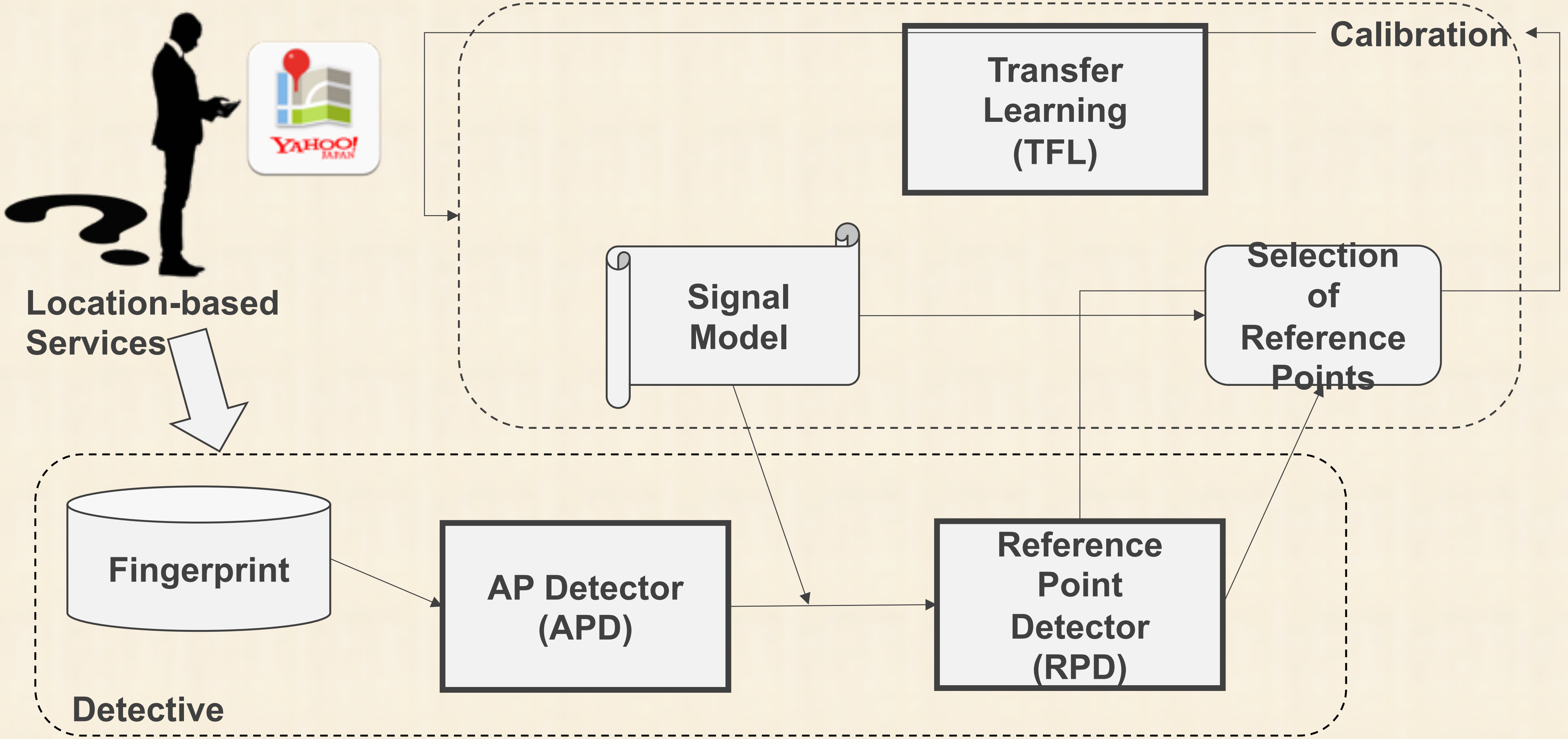
- Collected from user of location services
- Automatically but not labeled

Plenty

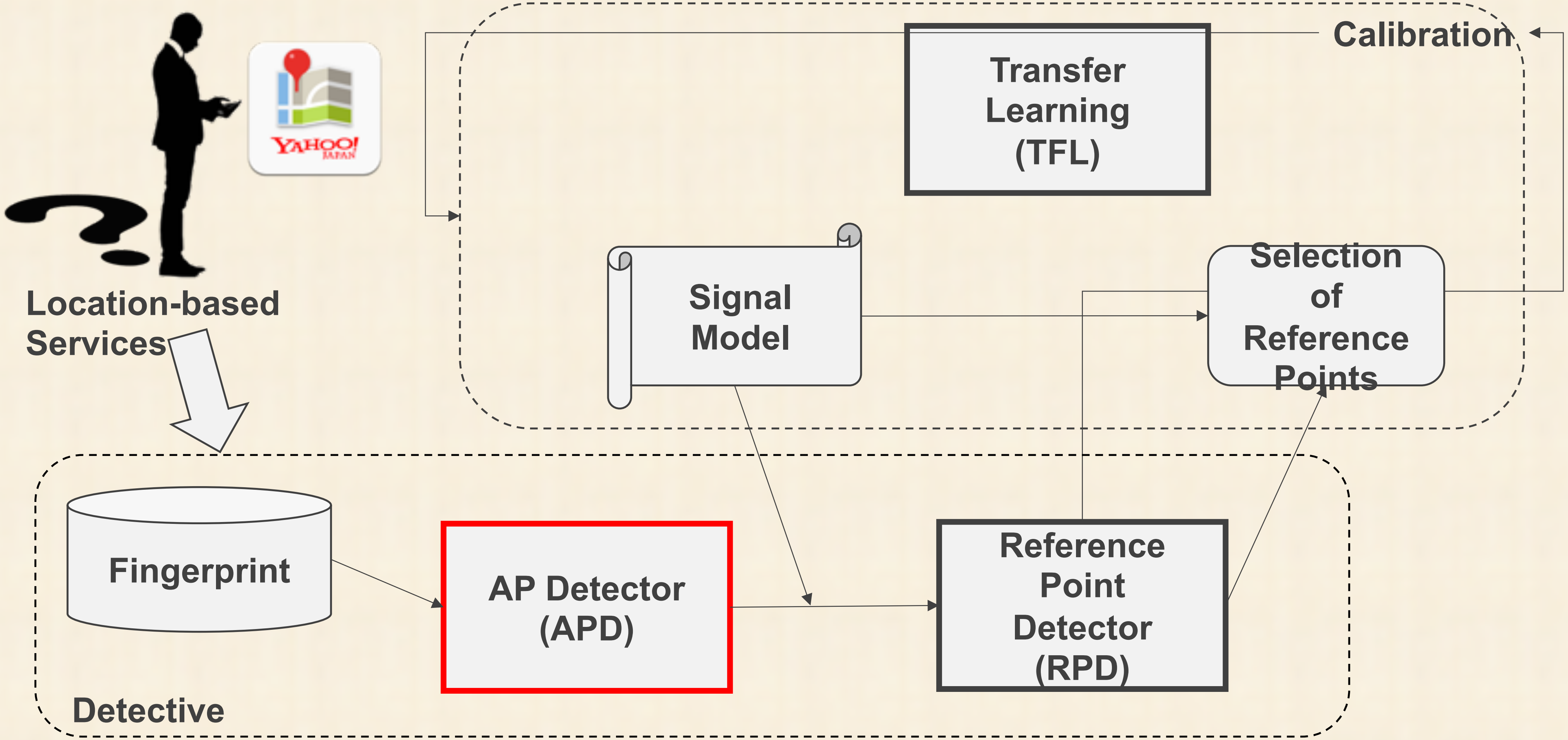
Methods

- Identify reference points where environmentally changed as anomaly
  - Build a system named **No-sweat Detective**
  - Perform with **no effort** using unlabeled fingerprint
  - Work strongly even in a **real world** with **long**-period observation
  - Achieve **higher accuracy recovery** in comparison with existing transfer learning methods using the **same amount** of labeled fingerprint

No-sweat Detective



No-sweat Detective



AP Detector (APD)

- Detects anomalous AP
  - Utilizes co-occurrence and analyzes relative position with other AP
- Workflow

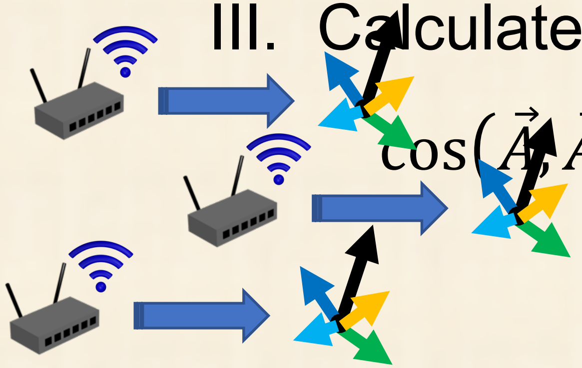
I. Picks up unlabeled fingerprint having maximum RSSI over ***vecFilt***

$(R1_{r1}, R2_{r2}, R3_{r3}, \cdots, Rx_{rx})$ {***vecFilt*** < max(r1,rx)} [R: AP, r: RSSI] ⋯ (1)

II. Vectorizes unlabeled fingerprint in sparse space with ***vecWidth***

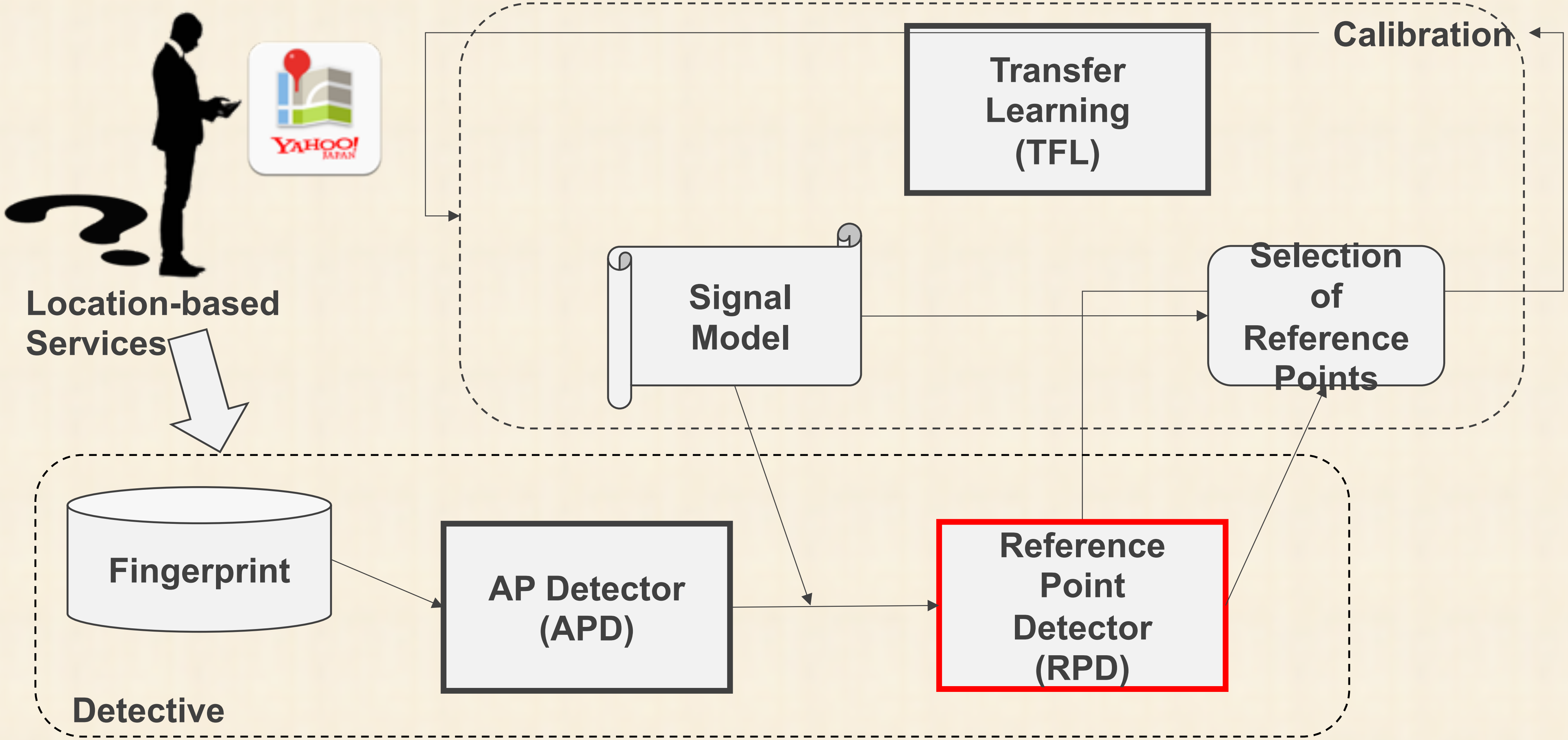
$\vec{A} = (R1_{r1}, R2_{r2}, R3_{r3}, \cdots, Rx_{rx})$ {***vecFilt*** < max(r1,rx), **vecWidth** < rx} ⋯ (2)

III. Calculates similarity of vector model



$$\cos(\vec{A}, \vec{A}') = \frac{\vec{A} \cdot \vec{A}'}{|\vec{A}||\vec{A}'|} = \frac{\vec{A}}{|\vec{A}|} \cdot \frac{\overrightarrow{A'}}{|\vec{A}'|} = \frac{\sum_{i=1}^{|V|} A_i A'_i}{\sqrt{\sum_{i=1}^{|V|} A_i^2} \cdot \sqrt{\sum_{i=1}^{|V|} A'^2_i}} \quad \cdots (3)$$

No-sweat Detective



Reference Point Detector (RPD)

- Detects anomalous Reference Point
  - Environmentally changed around detected Wi-Fi source
- Workflow

I. Singularly detected:

   i. Roughly estimates location where each unlabeled fingerprint were observed by ignoring distorted radio wave
   ii. Gets weighted average center
   iii. Recollect labeled fingerprint at reference points where within radius $\boldsymbol{\varepsilon}$ m from center

II. Plurally detected:

   - Employ DBSCAN to Wi-Fi source location retrieved from past model with radius $\boldsymbol{\varepsilon}$ m
     - Regarding collective changes hardly caused by displacement of Wi-Fi source

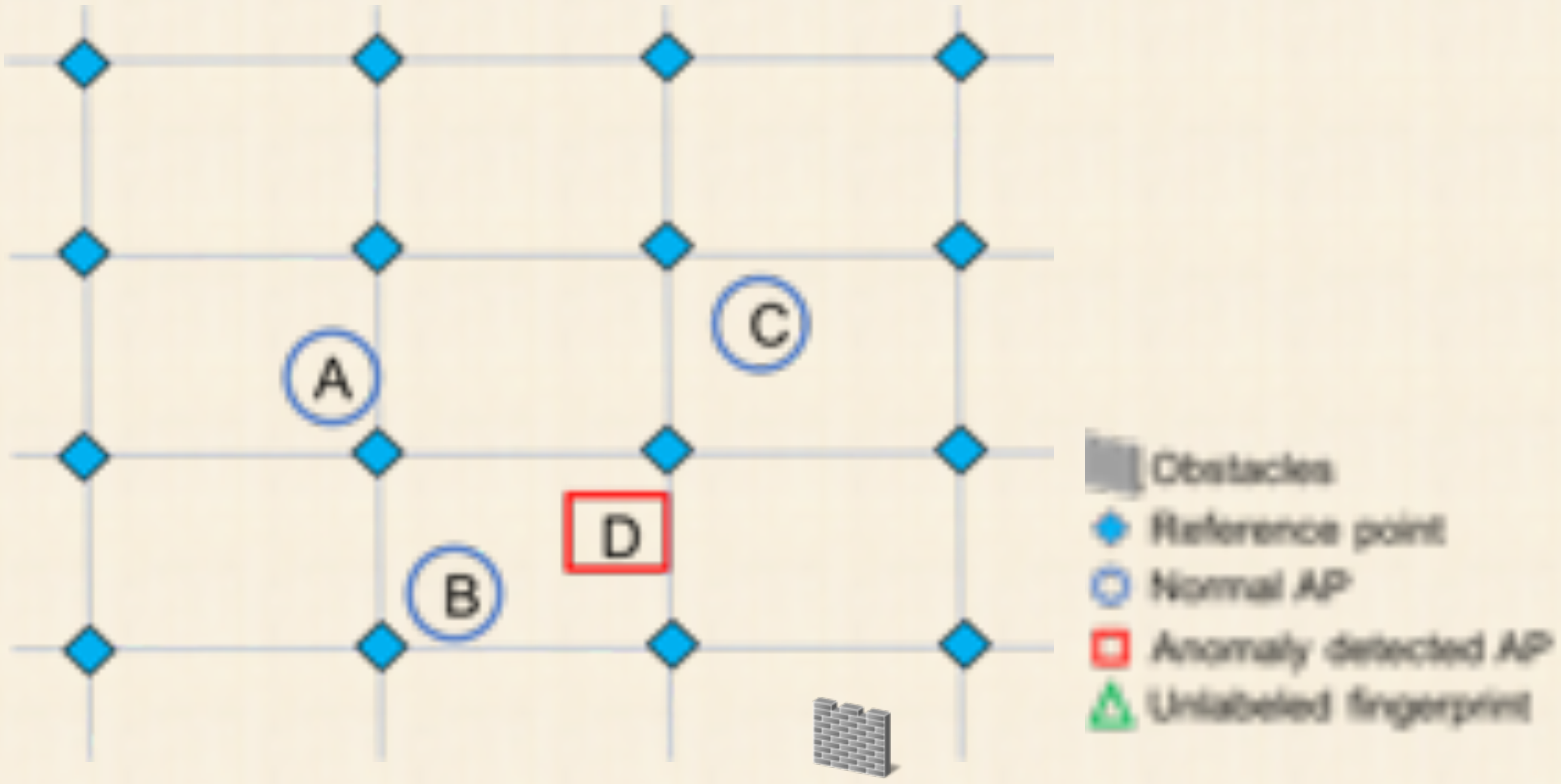
Reference Point Detector (RPD)

- Detects anomalous Reference Point
  - Environmentally changed around detected Wi-Fi source
- Workflow

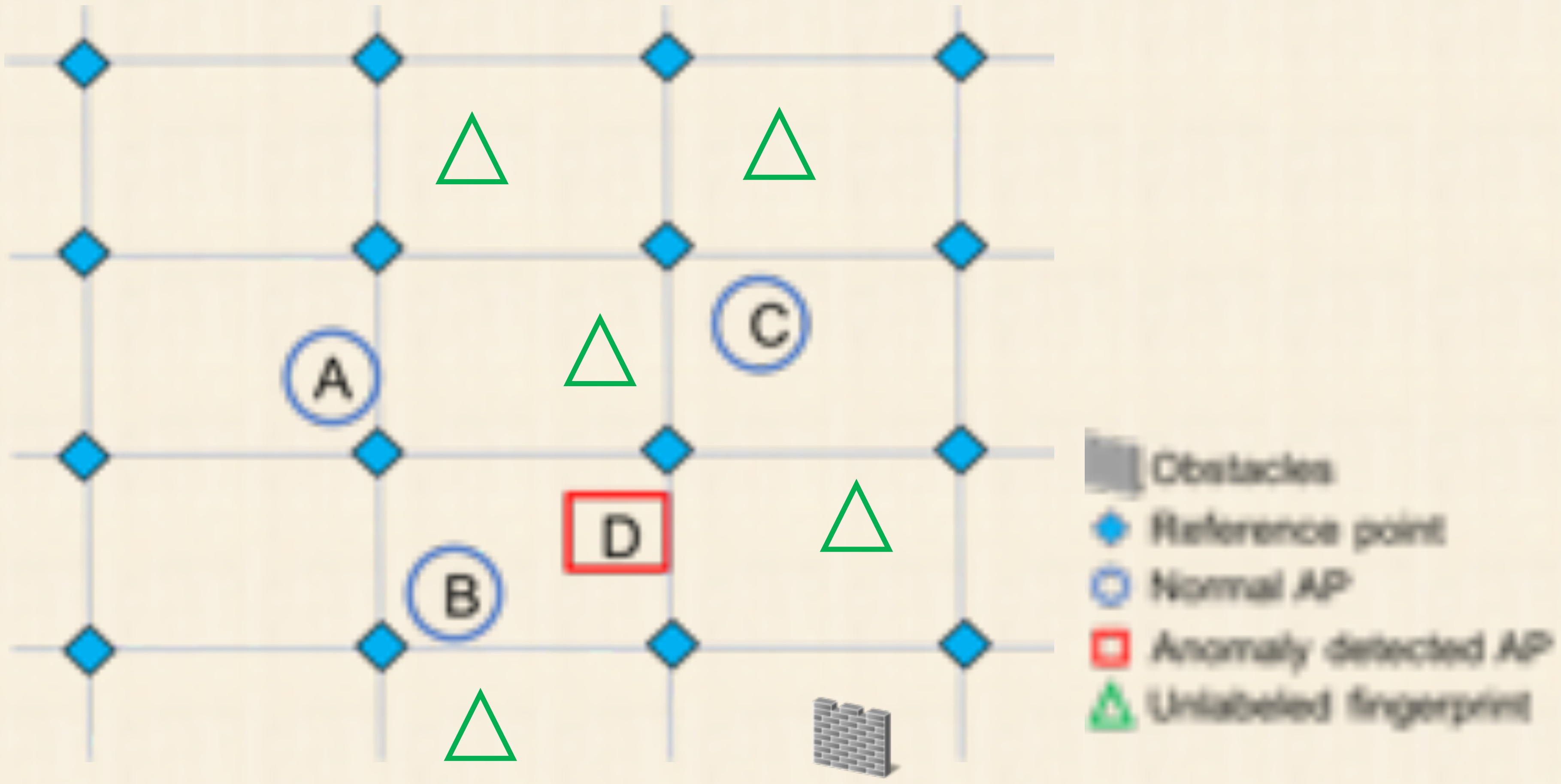
I. Singularly detected:
   i. Roughly estimates location where each unlabeled fingerprint were observed by ignoring distorted radio wave
   ii. Gets weighted average center
   iii. Recollect labeled fingerprint at reference points where within radius ε m from center

II. Plurally detected:
   - Employ DBSCAN to Wi-Fi source location retrieved from past model with radius ε m
     - Regarding collective changes hardly caused by displacement of Wi-Fi source

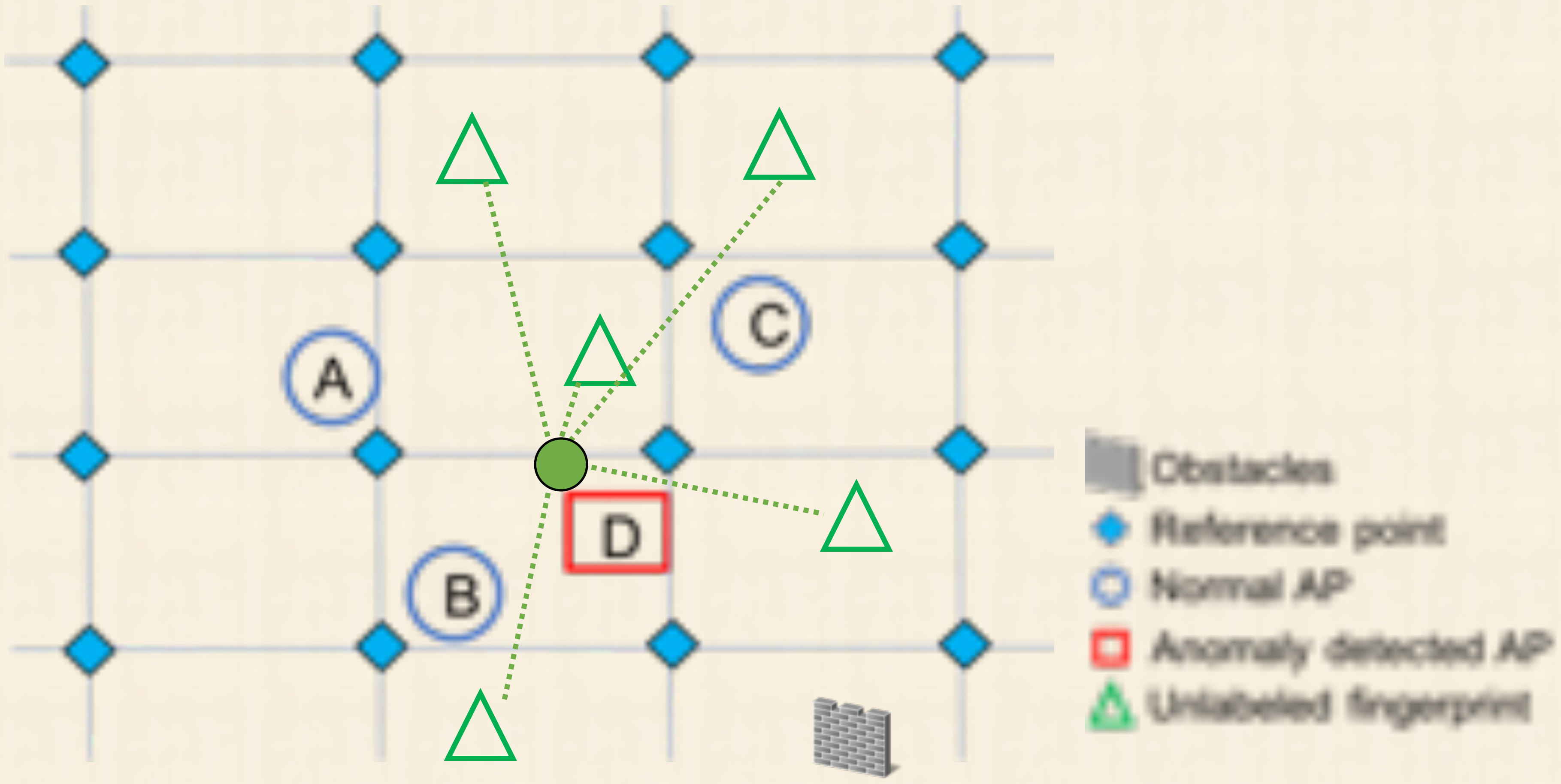
When Singularly Detected



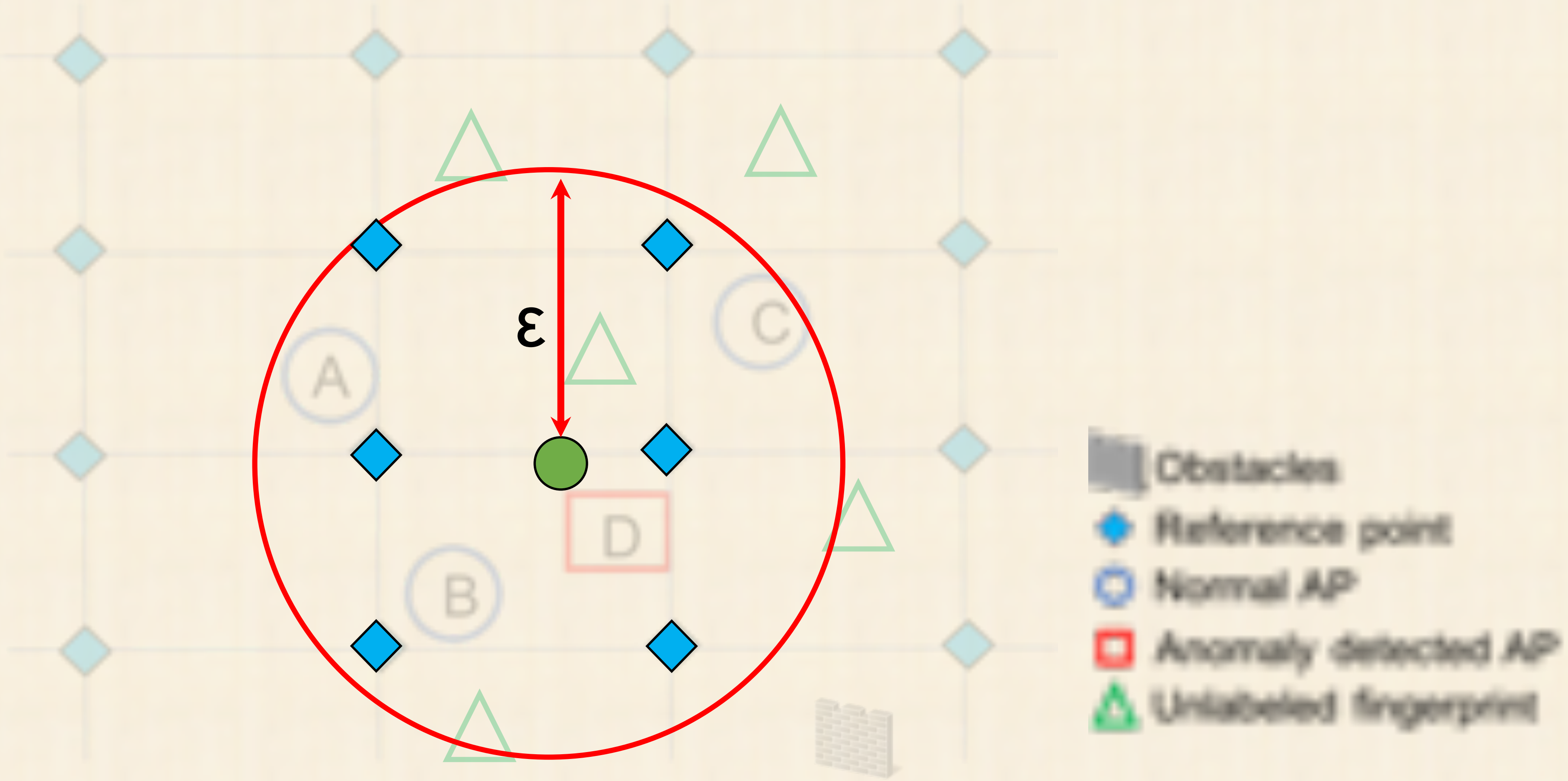
When Singularly Detected



When Singularly Detected



When Singularly Detected



When Singularly Detected

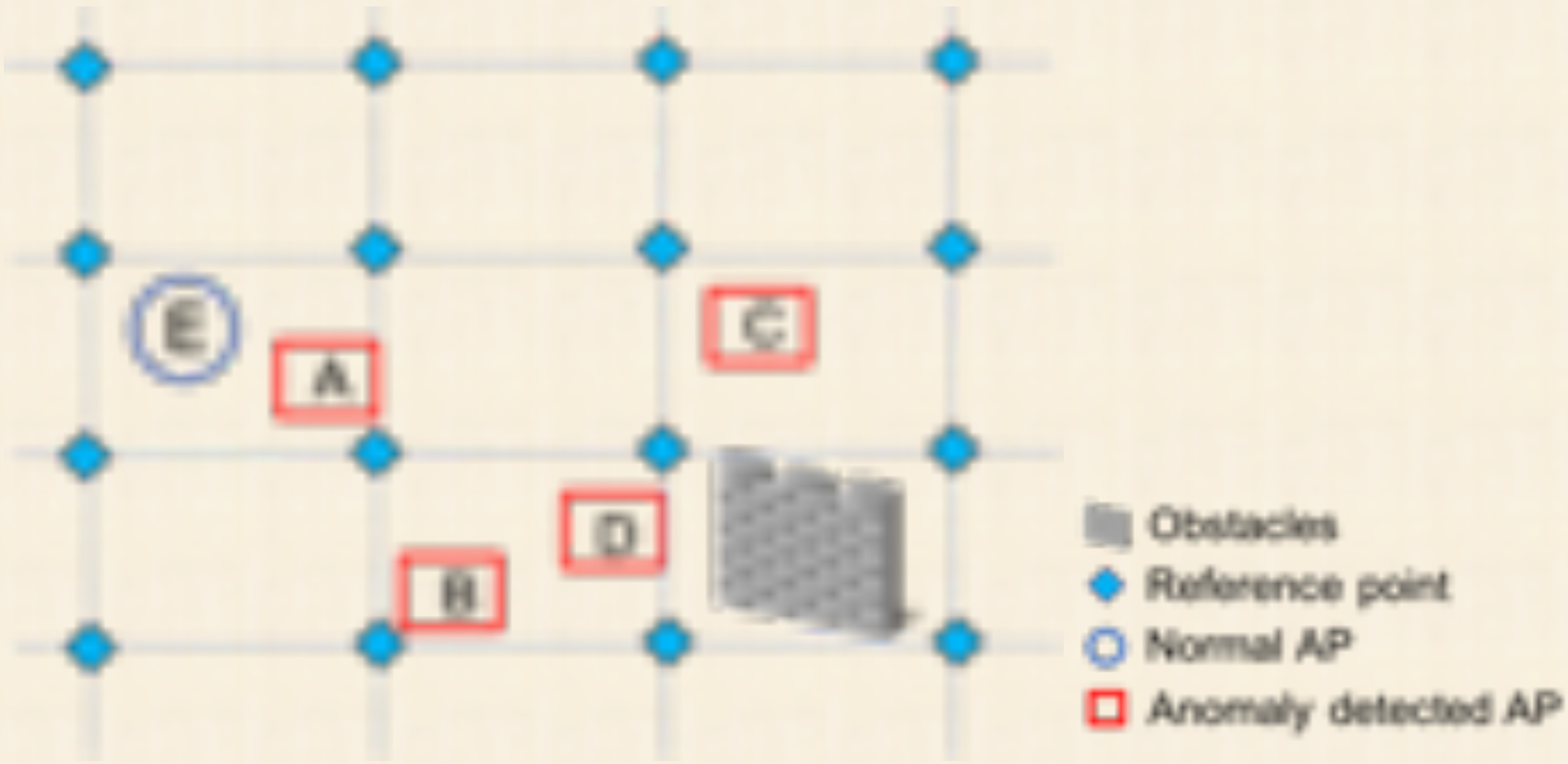
Reference Point Detector (RPD)

- Detects anomalous Reference Point
  - Environmentally changed around detected Wi-Fi source
- Workflow

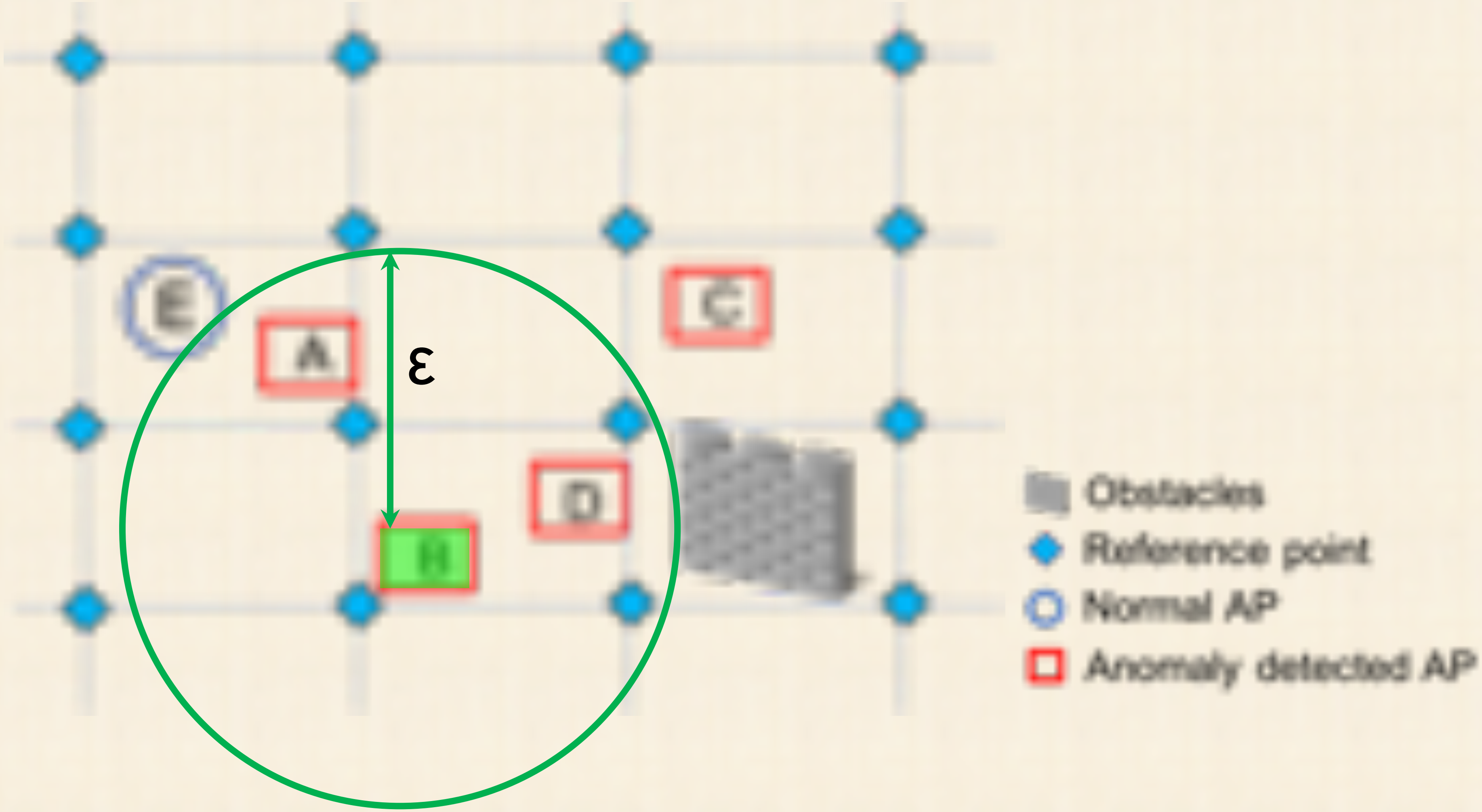
I. Singularly detected:
   i. Roughly estimates location where each unlabeled fingerprint were observed by ignoring distorted radio wave
   ii. Gets weighted average center
   iii. Recollect labeled fingerprint at reference points where within radius ε m from center

II. Plurally detected:
   - Employ DBSCAN to Wi-Fi source location retrieved from past model with radius ε m
     - Regarding collective changes hardly caused by displacement of Wi-Fi source

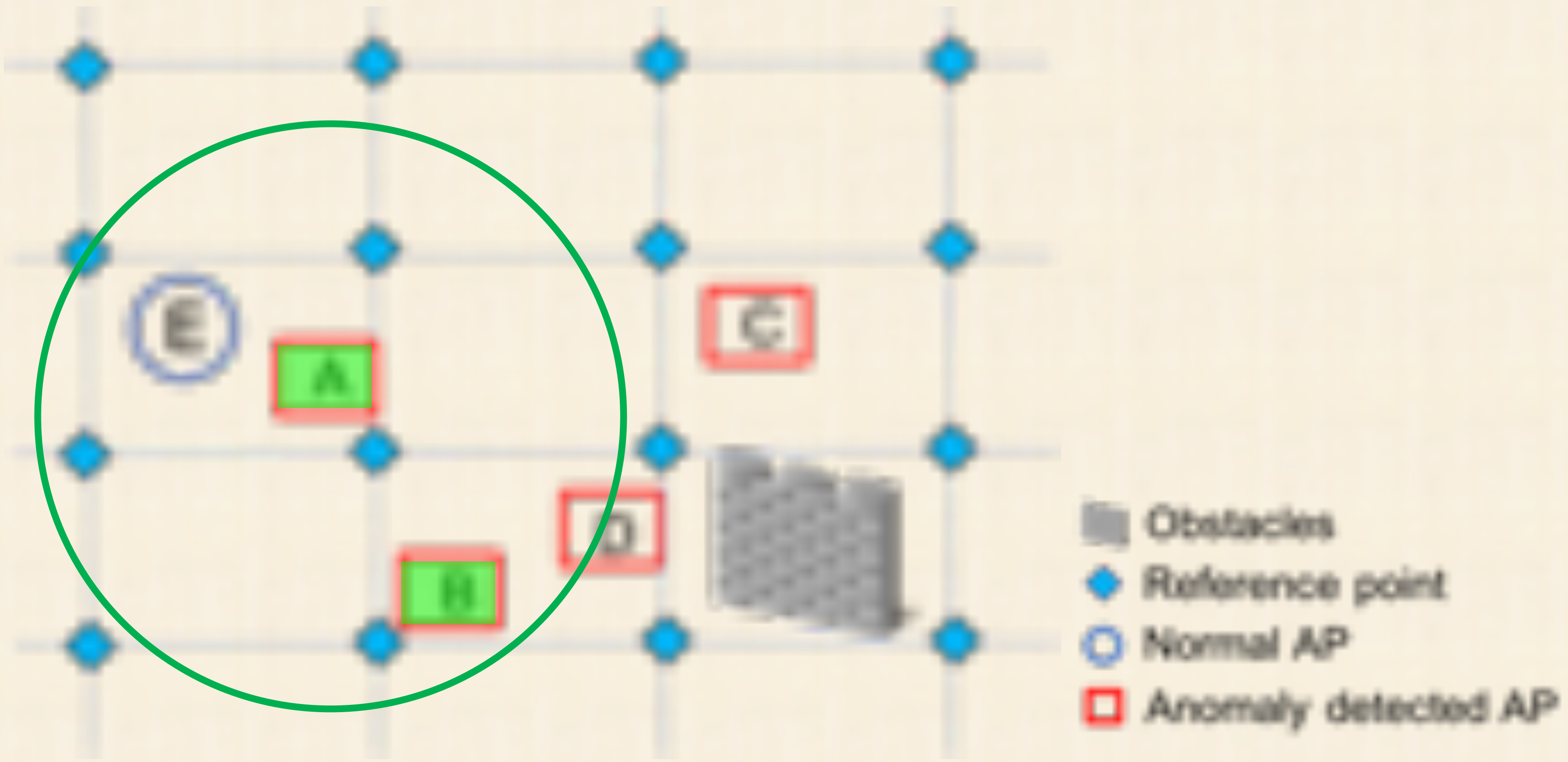
When Plurally Detected



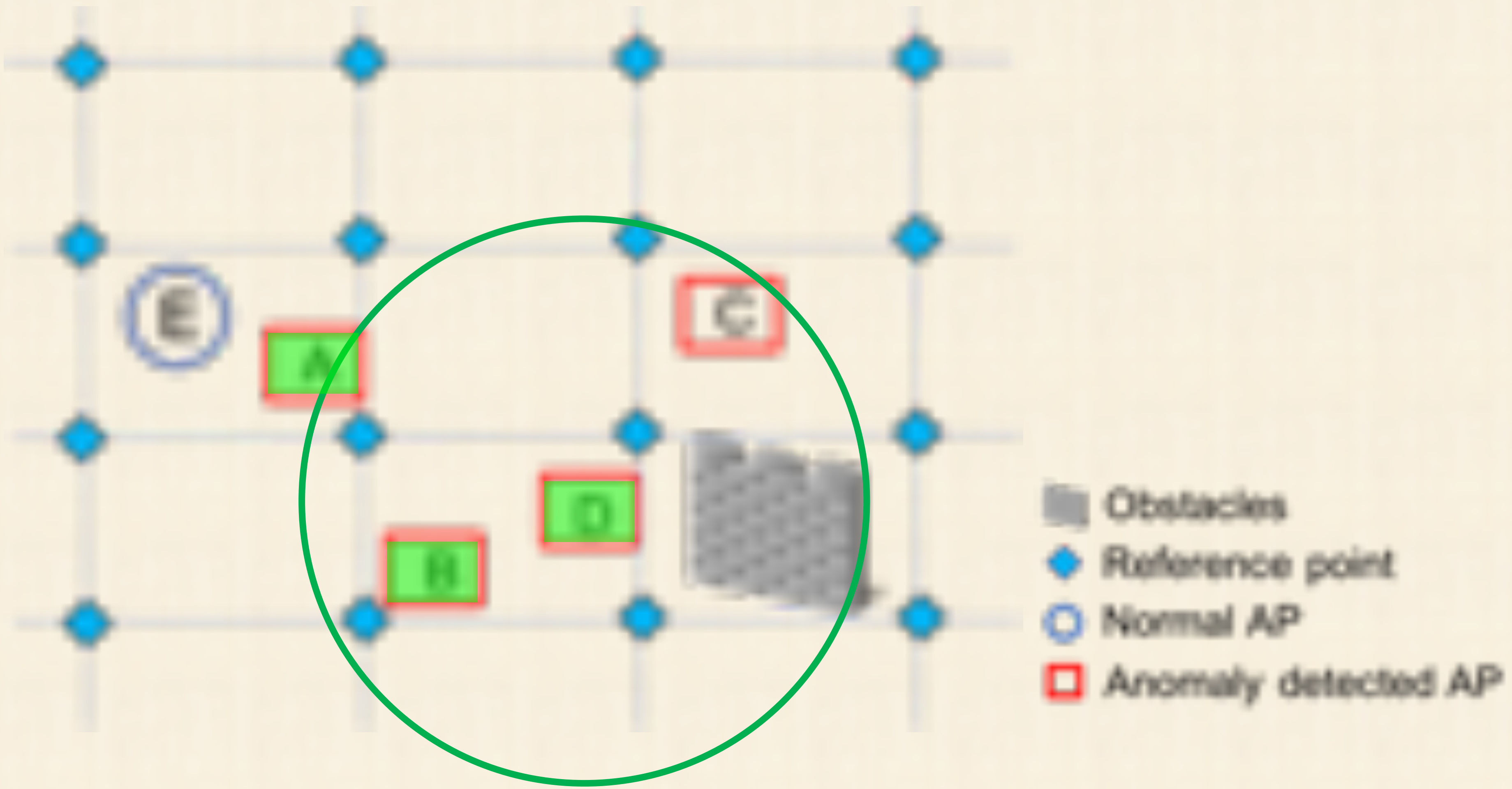
When Plurally Detected



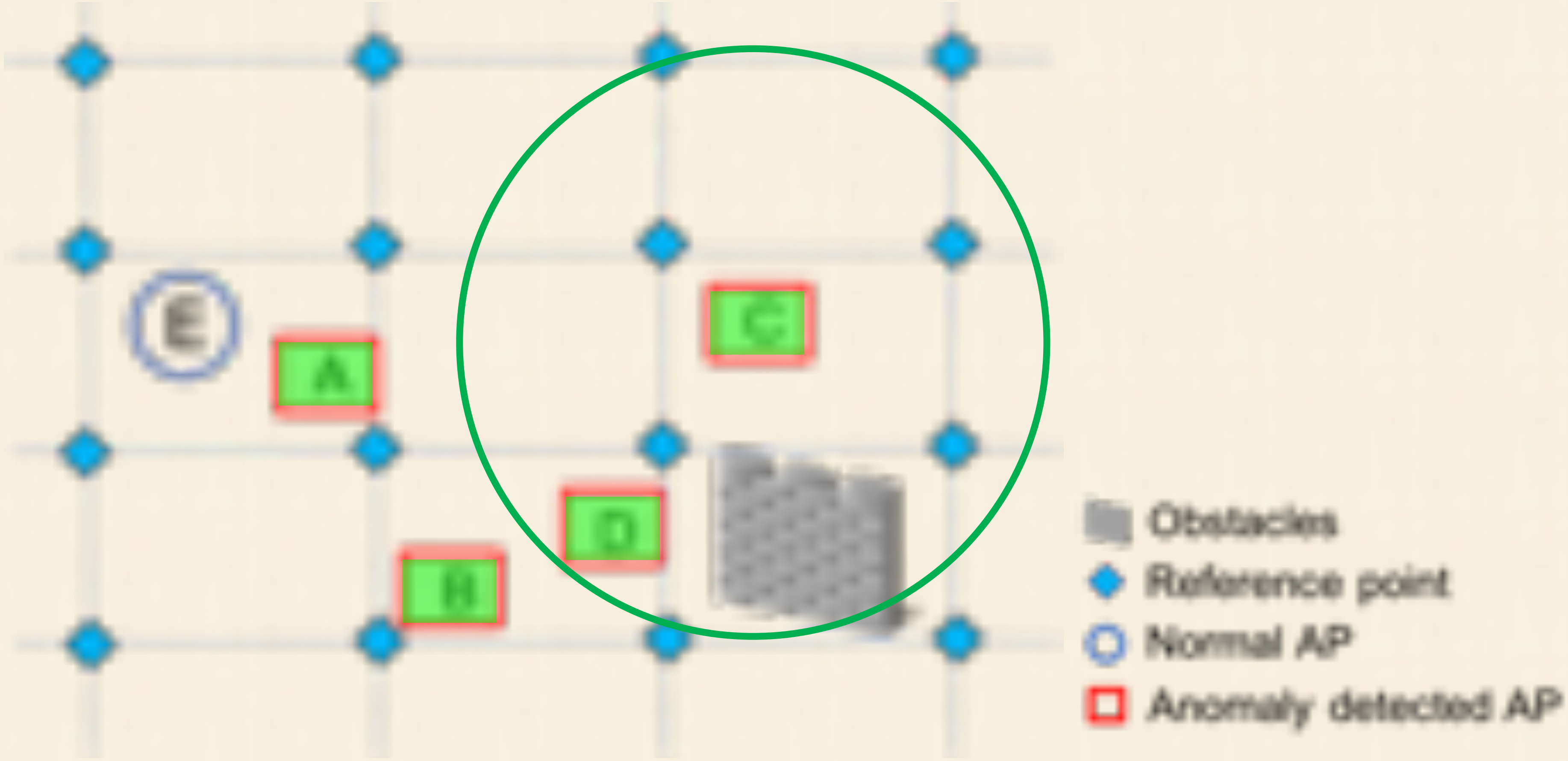
When Plurally Detected



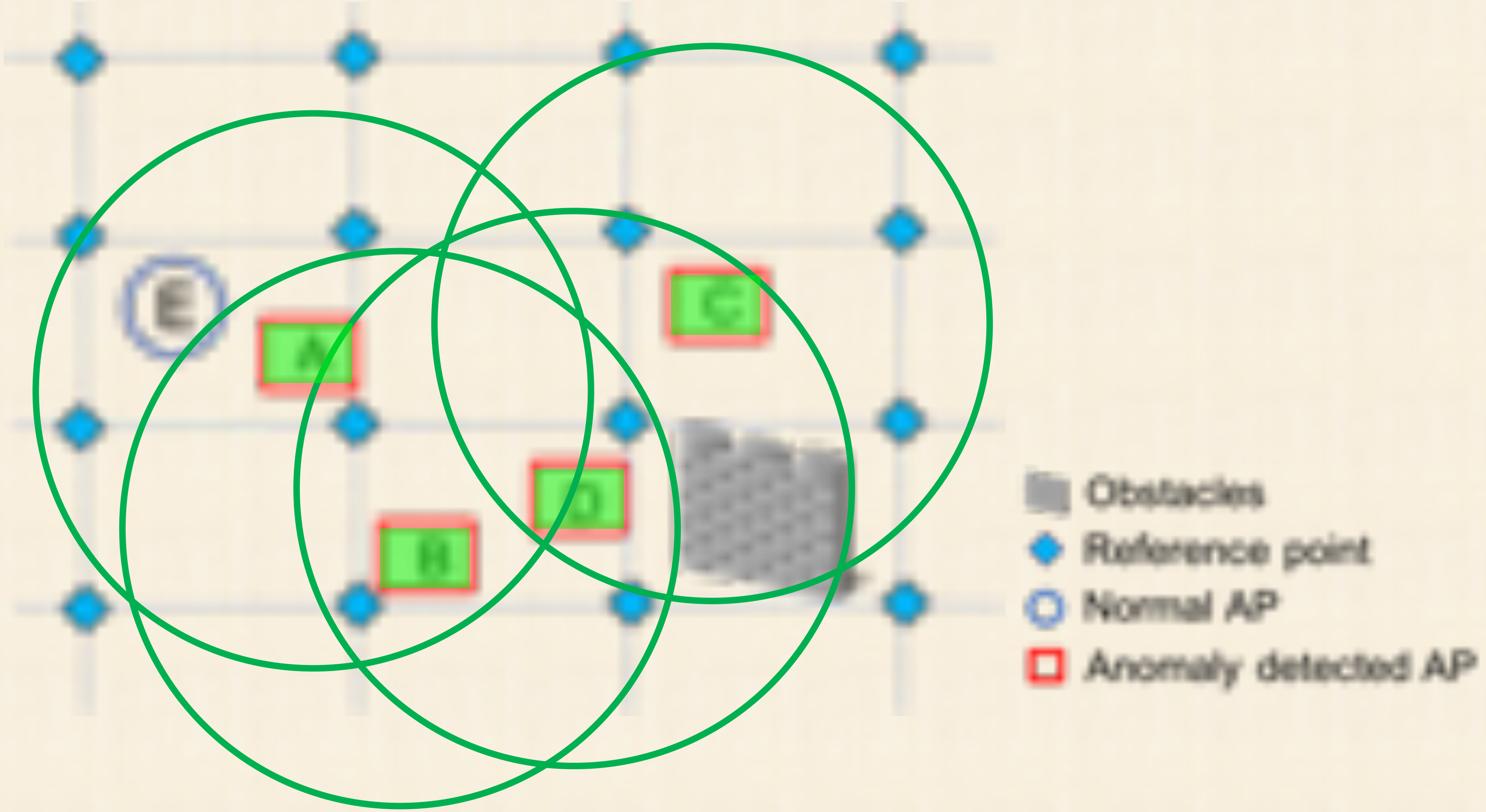
When Plurally Detected



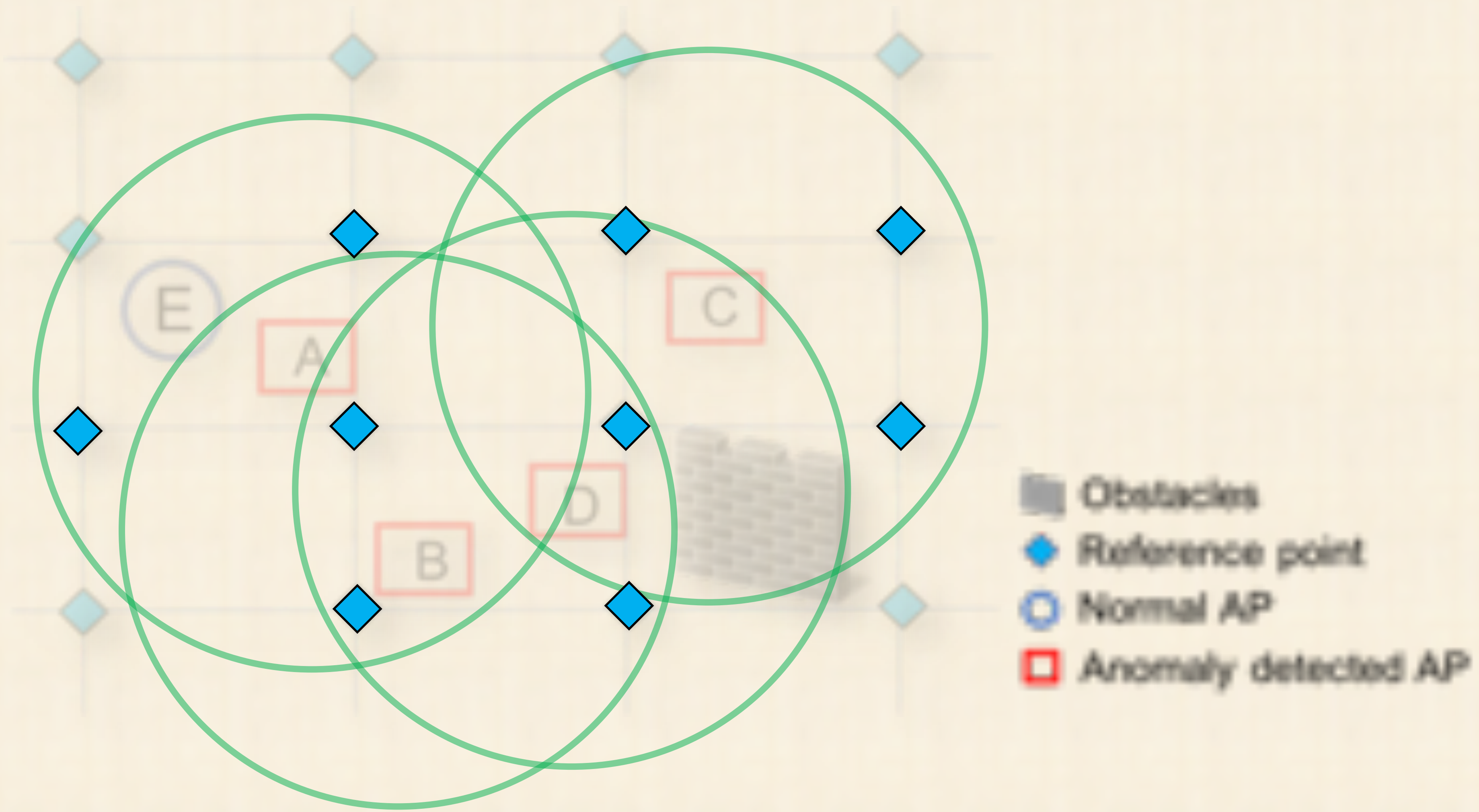
When Plurally Detected



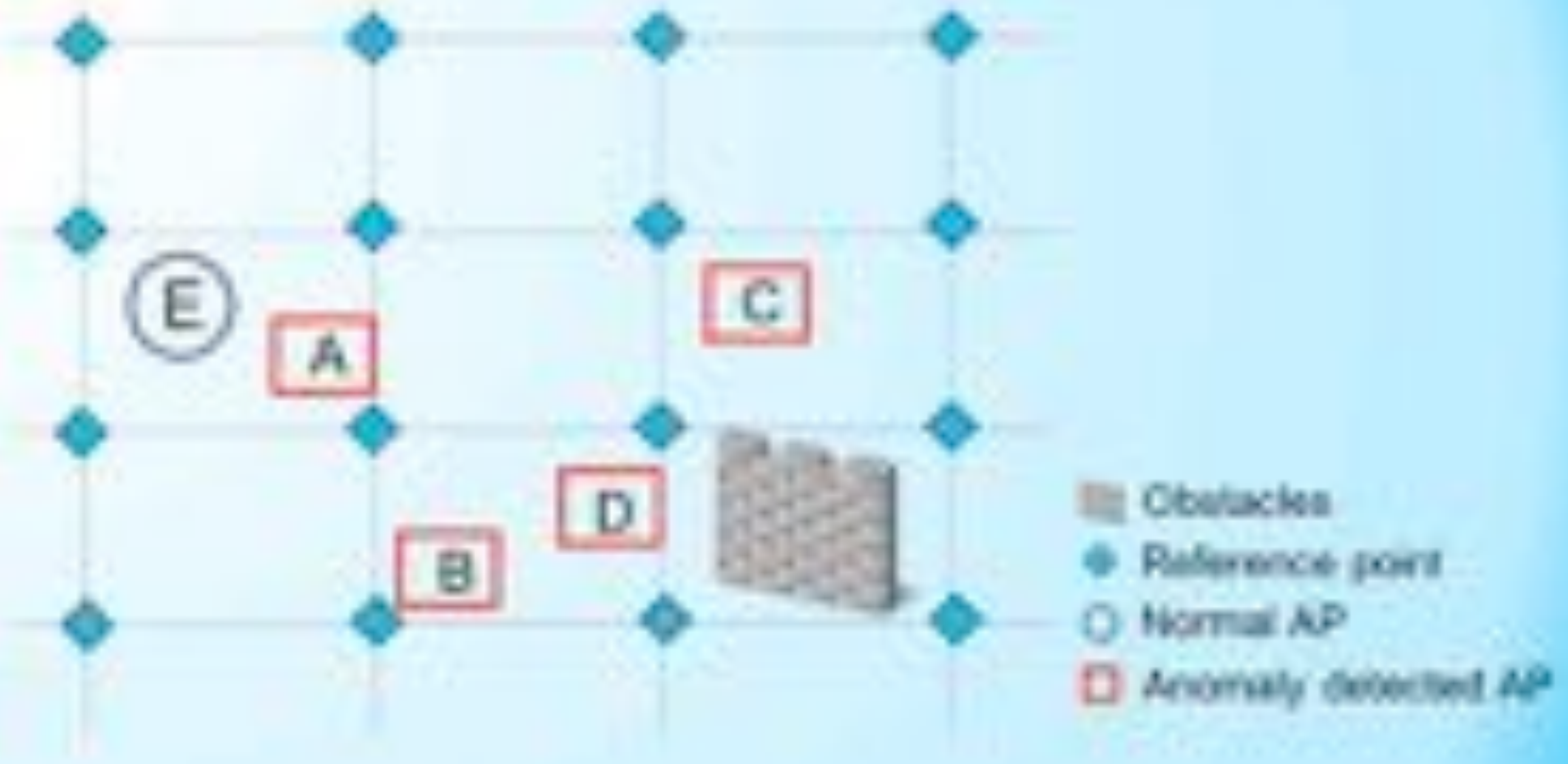
When Plurally Detected



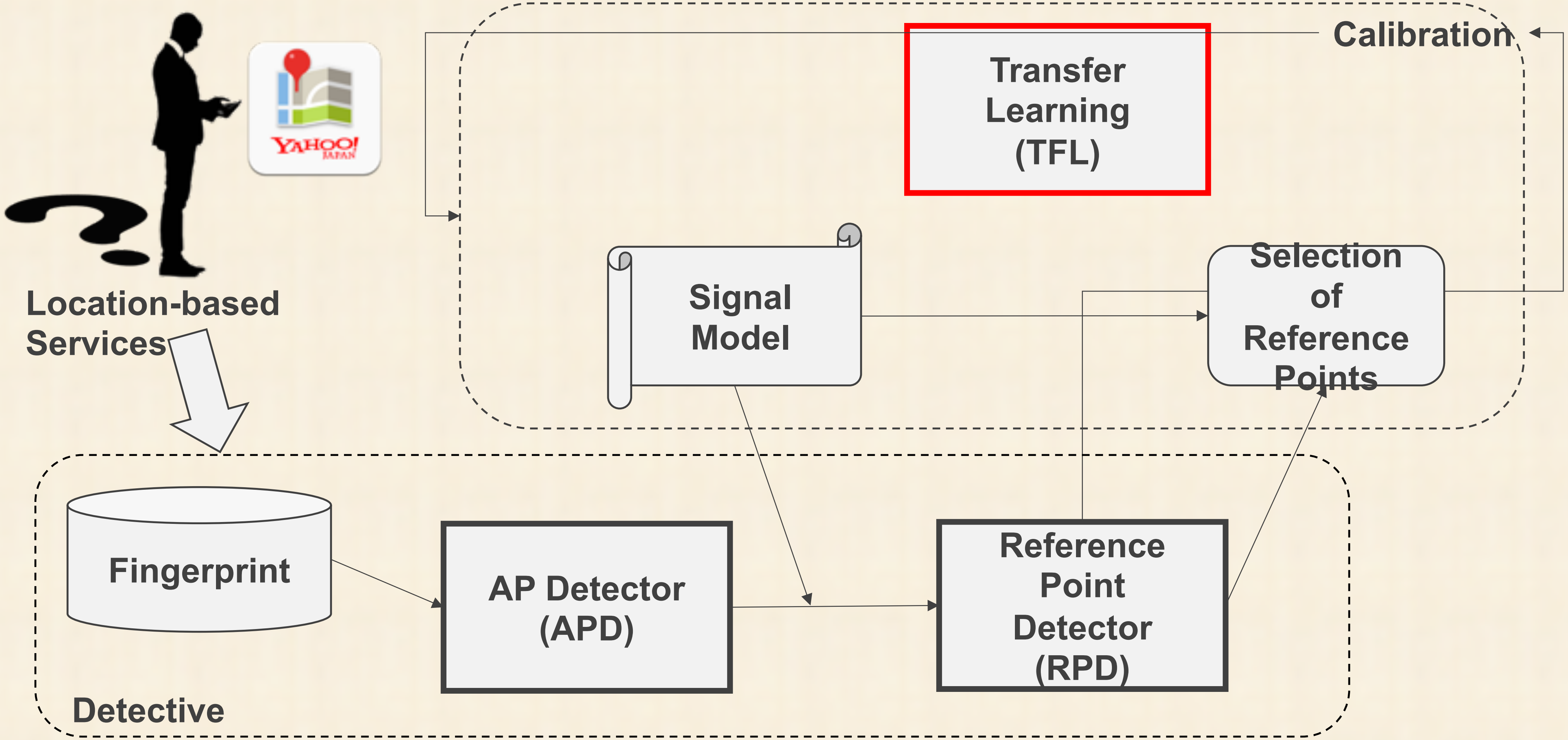
When Plurally Detected



When Plurally Detected



No-sweat Detective



Transfer Learning (TFL)

- TFL can be any colors and replaceable
  - Testify No-sweat Detective's performance by being applied to MixTrain and Lasso transfer methods[1)]
- MixTrain method
  - Closer to basis of transfer learning
  - Learns parameters $\boldsymbol{\theta}$ itself utilizing all the dataset from primary to additional dataset at each calibration
  - Updates model adding regularization term of L1 norm ($\sum_{i=1}^{|\theta|} |\theta i|$)
  - Simply for keeping weights given to features from being hyper-activated by usual L1 norm
- Lasso method
  - Learns parameters from variation of parameters
  - Does not store anything other than previous parameters and additional dataset
  - Learns from L1 norm and regularization term $\left(\sum_{i=1}^{|\theta|} \left| \theta_i^{(k-1)} - \theta_i^{(k)} \right| \right)$
  - Regularization minimizing variation between $\boldsymbol{\theta}$ at period $\boldsymbol{k-1}$ and $\boldsymbol{\theta}$ at period $\boldsymbol{k}$

[1]: Pengcheng Wu and Thomas G. Dietterich. "Improving SVM Accuracy by Training on Auxiliary Data Sources."

Evaluation

- Tested in two different situations
  - Laboratory dataset
    - Testify anomaly detection modules
    - Simulate environmental changes
  - Underground dataset
    - Testify performance in real world scenario
    - With five-month-long labeled fingerprint and unlabeled fingerprint from Umechika-navi[1)]

[1]: http://www.umechikanavi.jp/

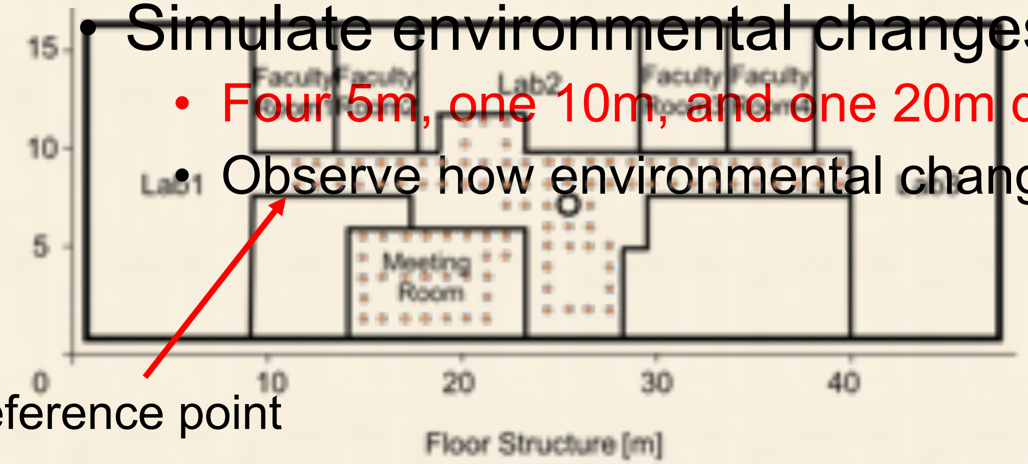
Evaluation

- Tested in two different situations
  - Laboratory dataset
    - Testify anomaly detection modules
    - Simulate environmental changes
  - Underground dataset
    - Testify performance in real world scenario
    - With five-month-long labeled fingerprint and unlabeled fingerprint from Umechika-navi[1)]

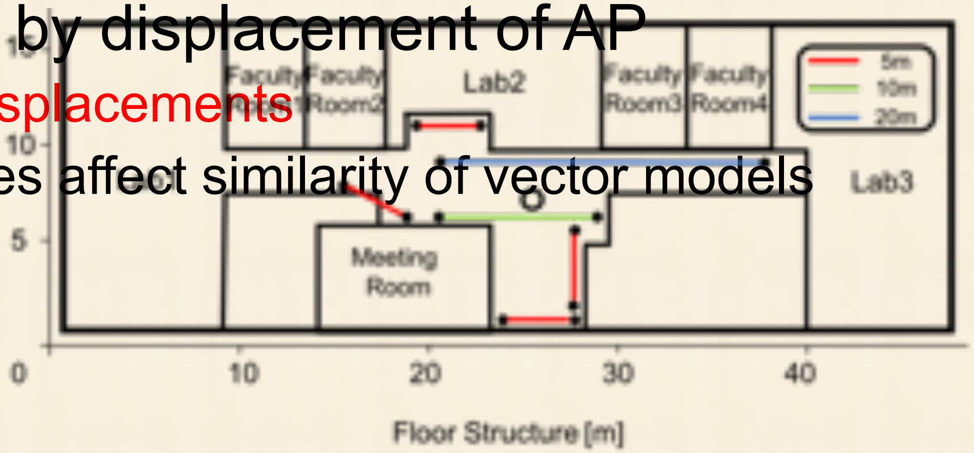
[1]: http://www.umechikanavi.jp/

Laboratory Dataset

- Details:
  - Set 105 reference points within $1m^2$ of floor covering $348m^2$ ($12m \times 29m$)
    - Meeting room and hallway used as test-bed
  - 2,100 labeled fingerprint in total
    - One scan per point and repeatedly ten times a day with Nexus5 over two days
  - Simulate environmental changes by displacement of AP
    - Four 5m, one 10m, and one 20m displacements
    - Observe how environmental changes affect similarity of vector models

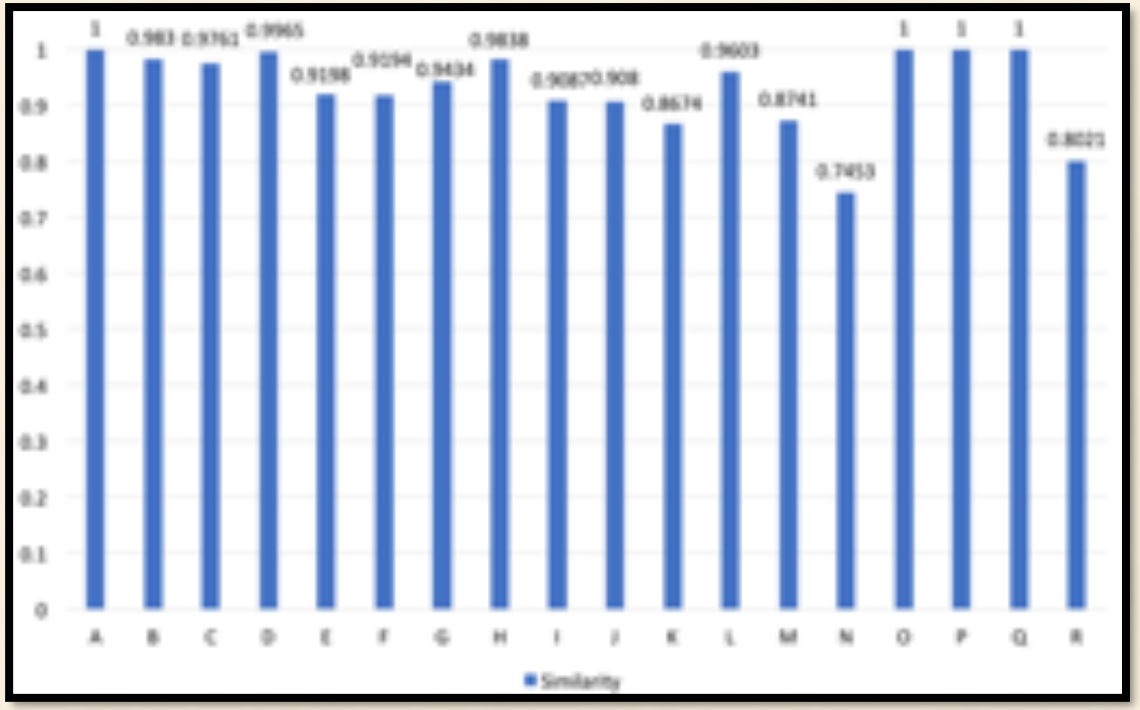


Reference point

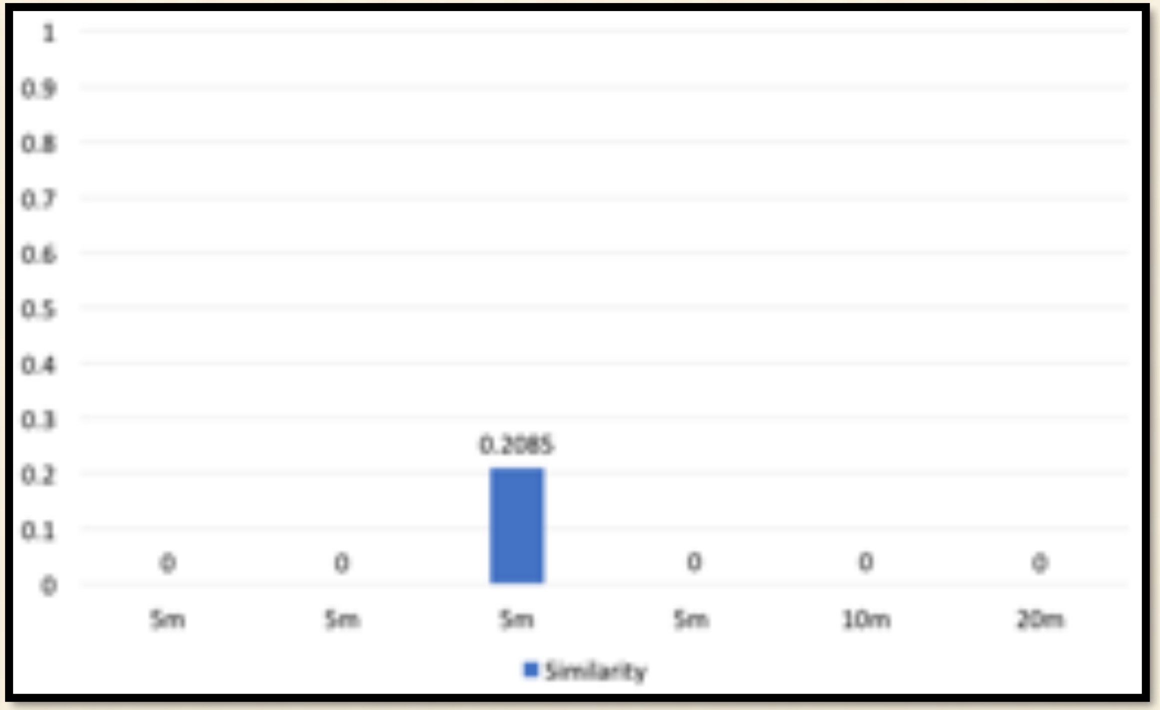


Results – Laboratory Dataset

- 18 vector models were created in both days



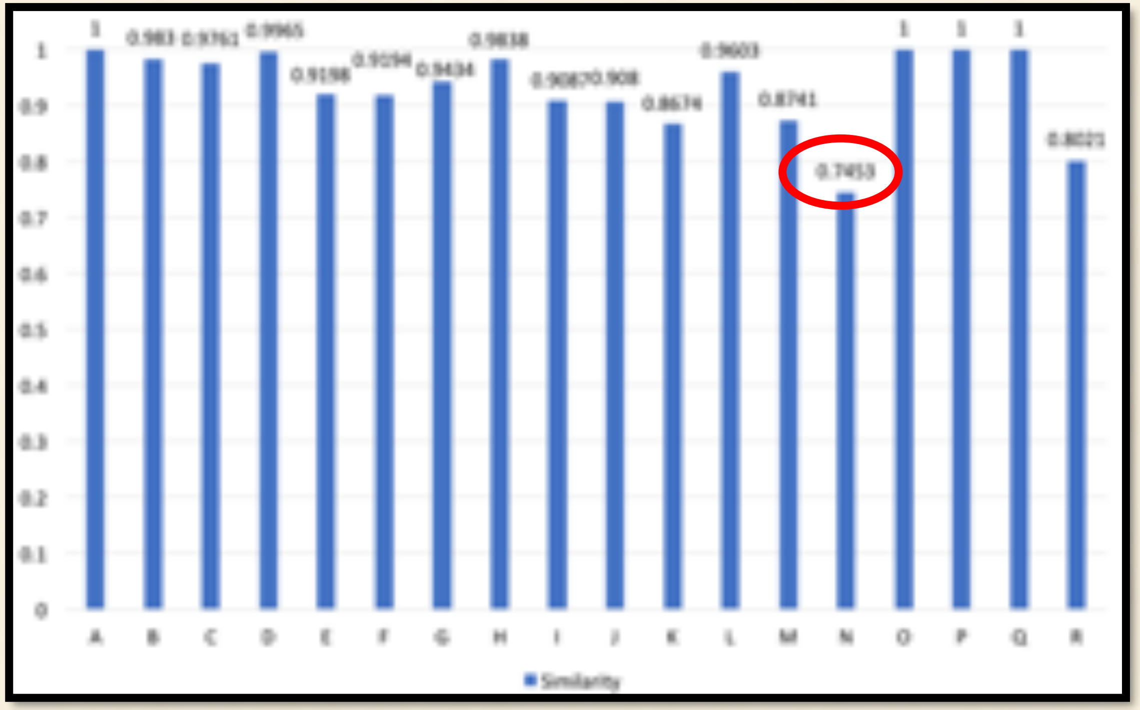
Non-distorted 18 AP's similarity



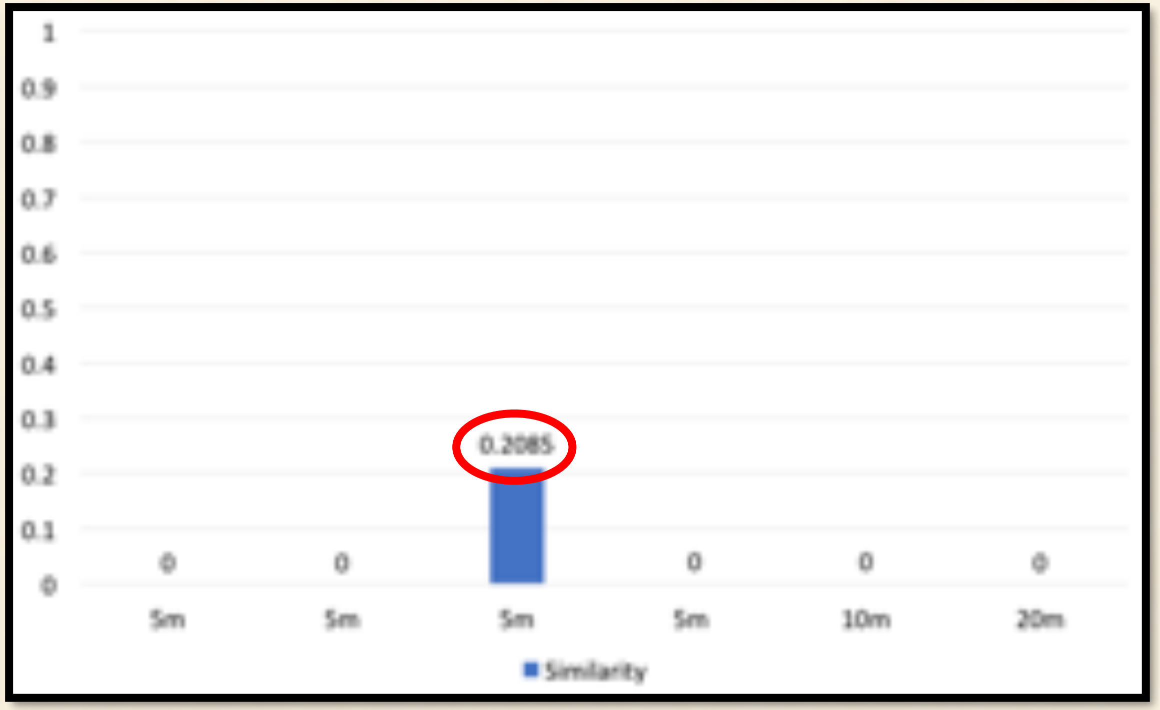
Distorted 5 AP's similarity

Results – Laboratory Dataset

- 18 vector models were created in both days



Non-distorted 18 AP's similarity



Distorted 5 AP's similarity

Validated significant difference in similarity

No-sweat Detective could detect environmental distortion

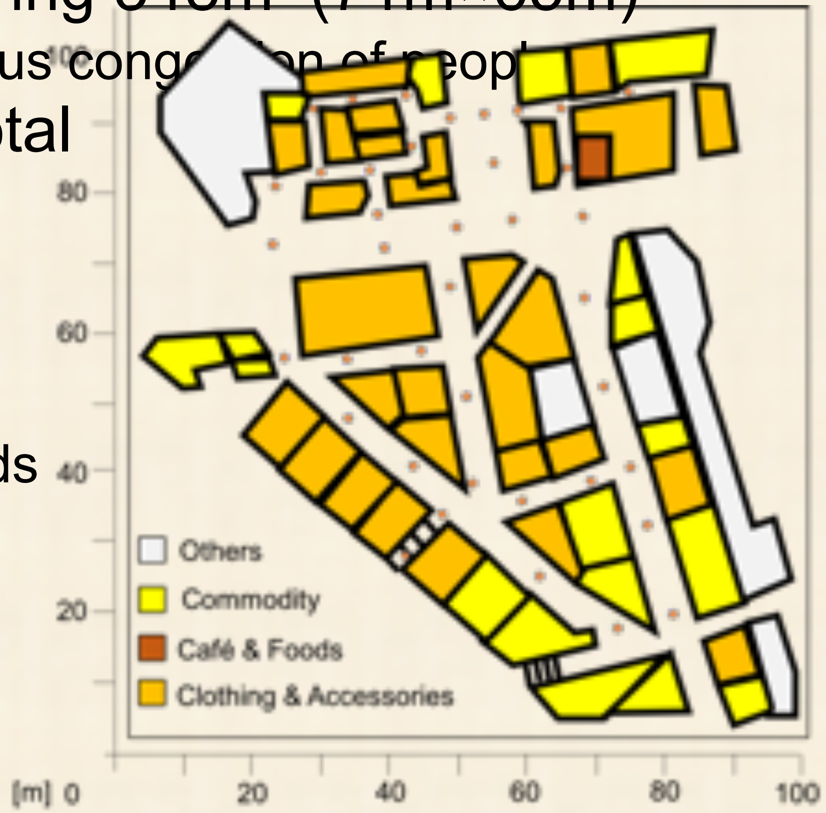
Evaluation

- Tested in two different situations
  - Laboratory dataset
    - Testify anomaly detection modules
    - Simulate environmental changes
  - Underground dataset
    - Testify performance in real world scenario
    - With five-month-long labeled fingerprint and unlabeled fingerprint from Umechika-navi[1)]

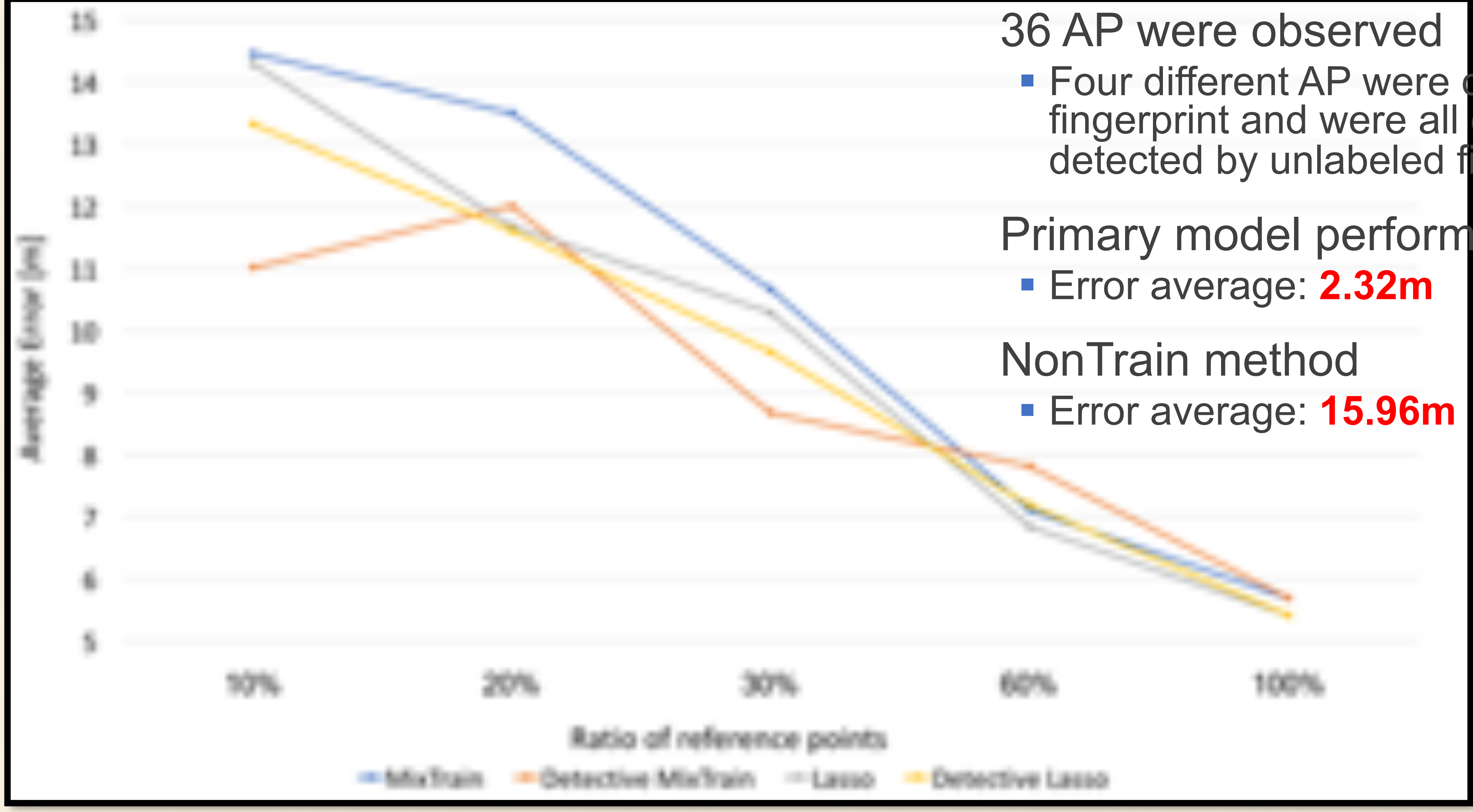
[1]: http://www.umechikanavi.jp/

Underground Dataset

- Details:
  - Set 39 reference points at 8m intervals covering 348m^2 (71m×65m)
    - Underground district where infamous for murderous conge on of peop
  - 2,693 labeled, 764 unlabeled fingerprint in total
    - Six scans per point with Nexus5 every two weeks
  - Retrain model every two weeks
    - Sliding amount of reference points
      - 10%, 20%, 30%, 60%, 100%
    - Apply No-sweat Detective to conventional methods
      - MixTrain method
      - Lasso method
    - Validate average error of final model
      - NonTrain method (never trained) as base-line



Results – Underground Dataset 2/3



36 AP were observed

- Four different AP were detected by labeled fingerprint and were all covered by 10 AP detected by unlabeled fingerprint

Primary model performance

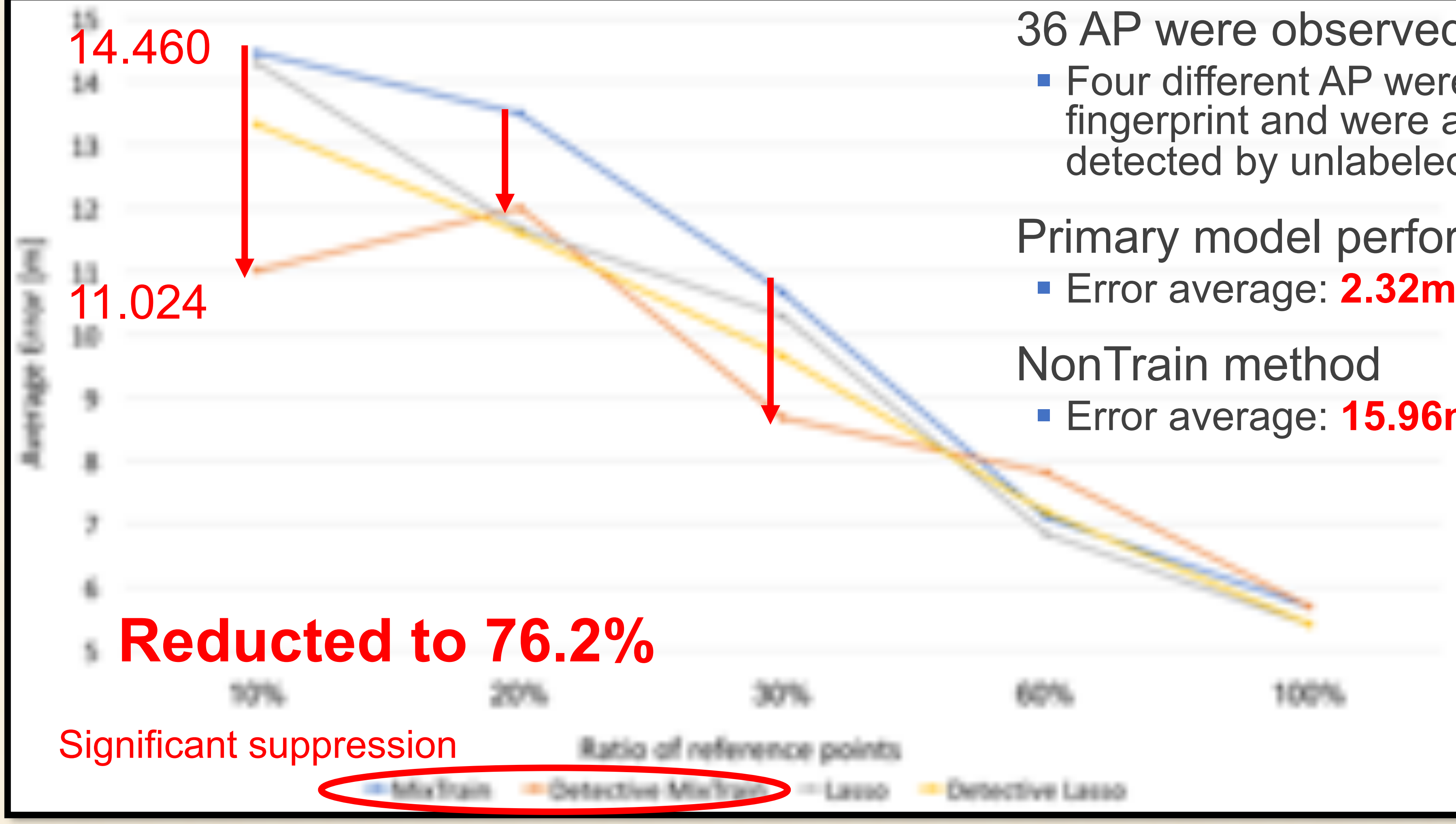
- Error average: **2.32m**

NonTrain method

- Error average: **15.96m**

Average error varying in amount of reference points

Results – Underground Dataset 2/3



36 AP were observed

- Four different AP were detected by labeled fingerprint and were all covered by 10 AP detected by unlabeled fingerprint

Primary model performance

- Error average: **2.32m**

NonTrain method

- Error average: **15.96m**

Average error varying in amount of reference points

Results – Underground Dataset 2/3

36 AP were observed

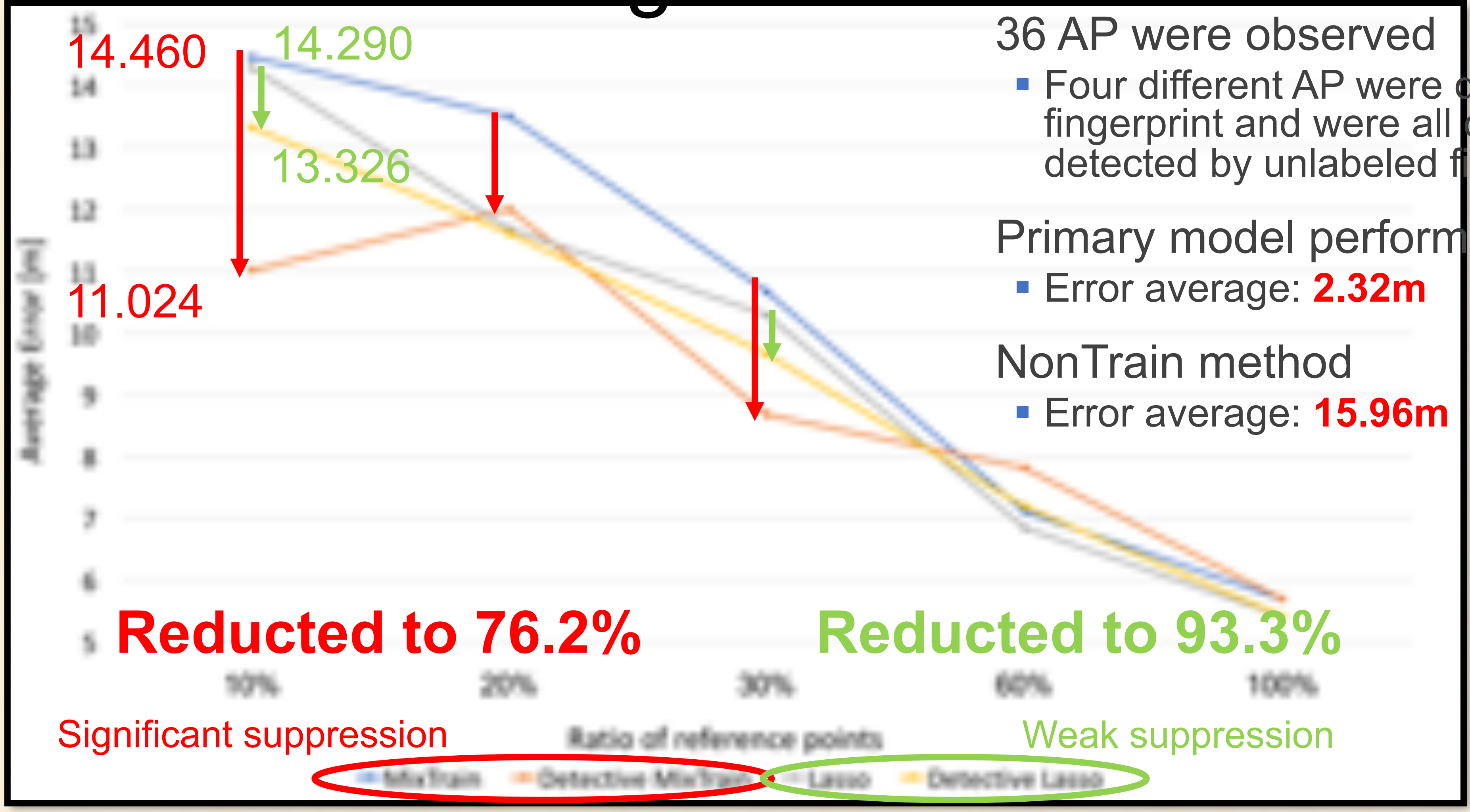
- Four different AP were detected by labeled fingerprint and were all covered by 10 AP detected by unlabeled fingerprint

Primary model performance

- Error average: **2.32m**

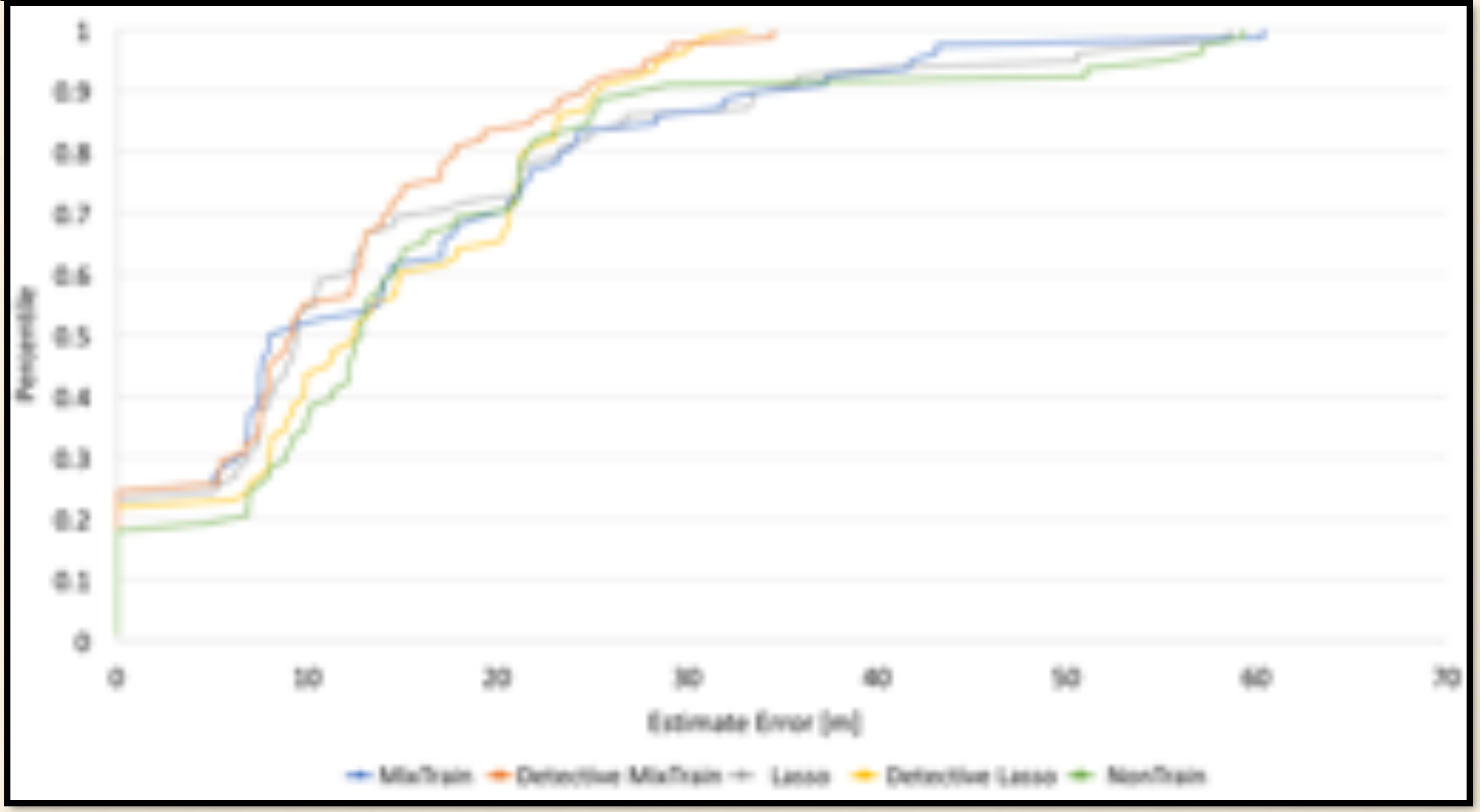
NonTrain method

- Error average: **15.96m**



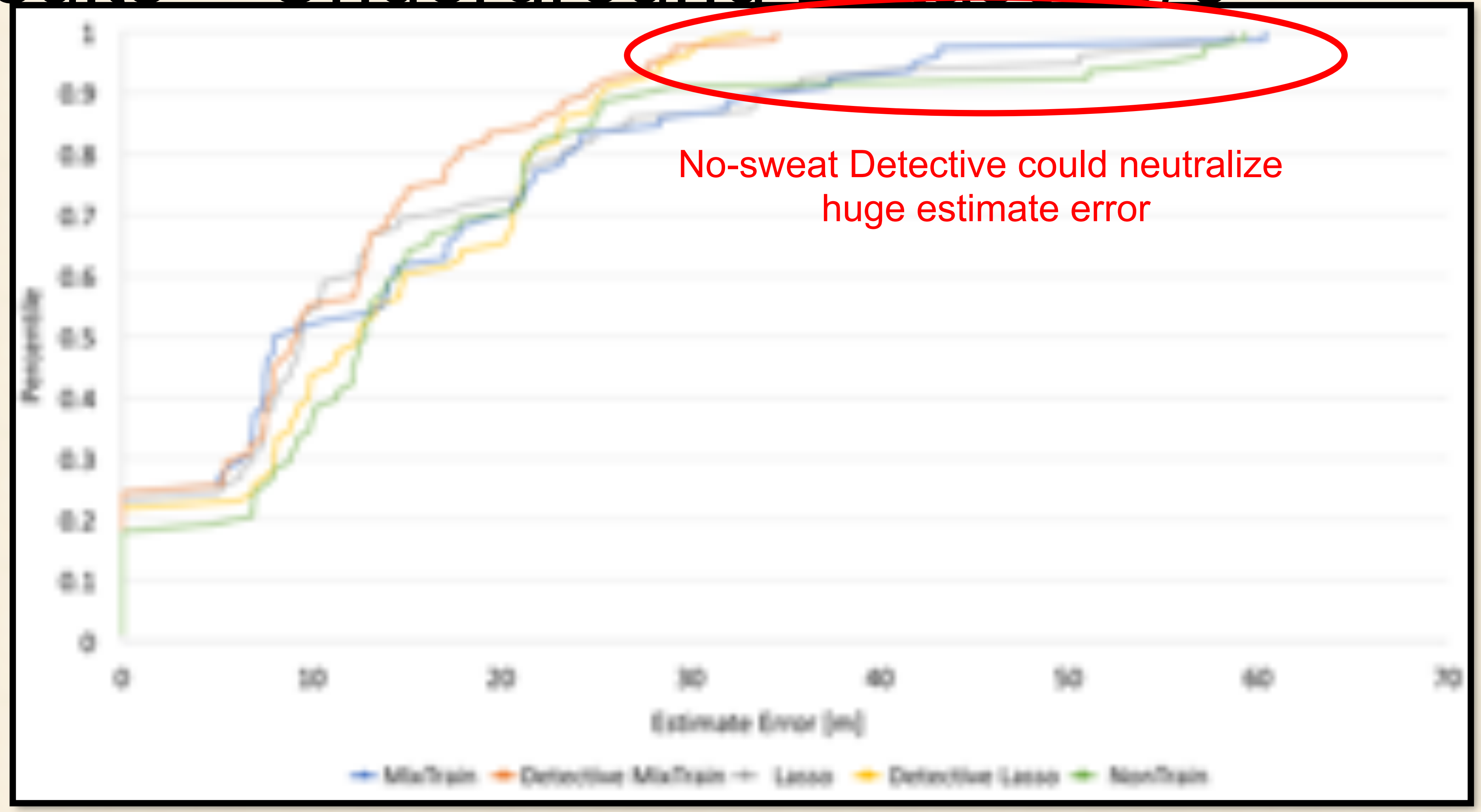
Average error varying in amount of reference points

Results – Underground Dataset 3/3



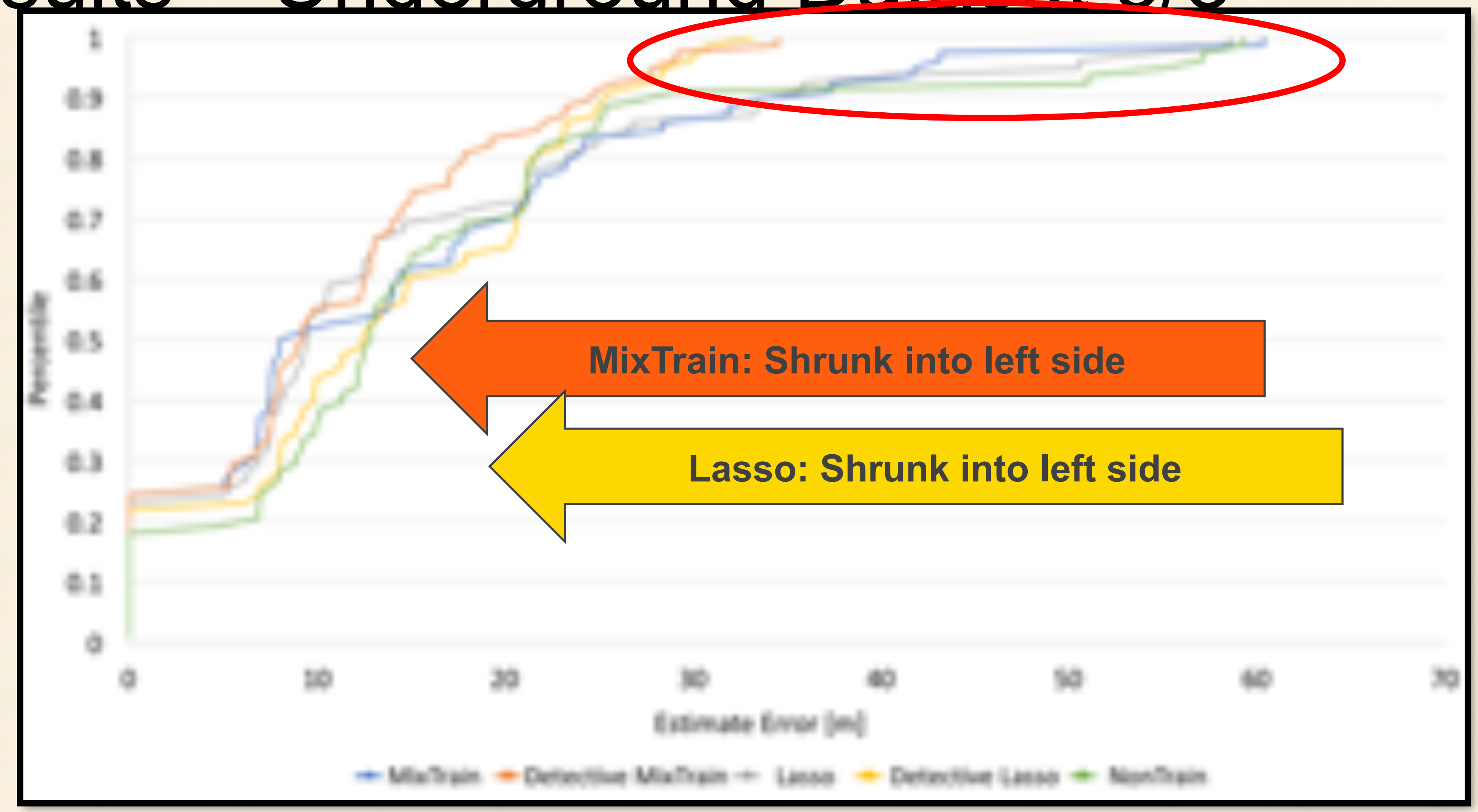
Cumulative distribution of estimate error

Results – Underground Dataset 3/3



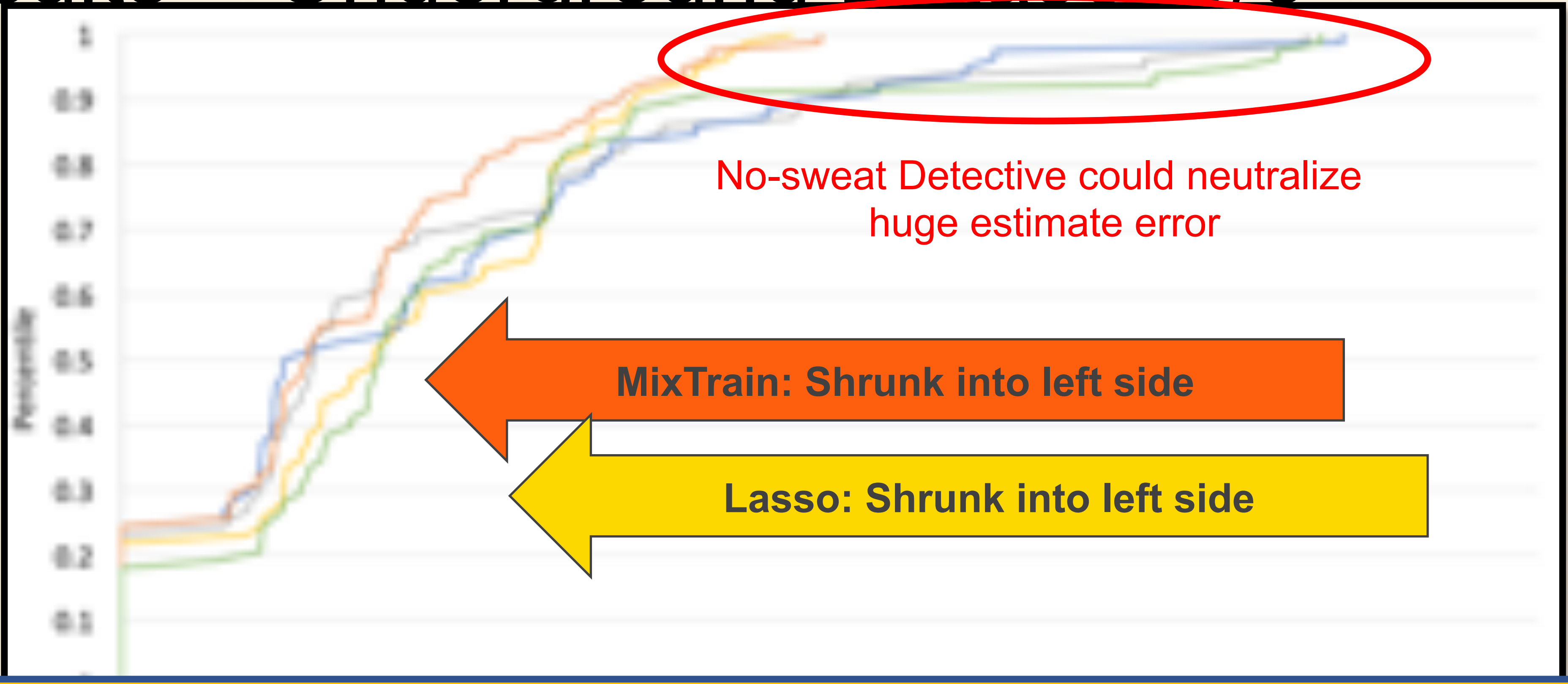
Cumulative distribution of estimate error

Results – Underground Dataset 3/3



Cumulative distribution of estimate error

Results – Underground Dataset 3/3



No-sweat Detective can be applied to existing transfer learning methods to maintain higher accuracy over long time operation

Cumulative distribution of estimate error

Summary and Outlook

- Fingerprinting localization mode deteriorates over time
- Transfer learning employed majorly
  - But additional dataset are randomly sampled
- Propose No-sweat Detective
  - To identify reference points where environmentally changed
  - With no effort using unlabeled fingerprint from user of location services
  - Could detect environmental distortion
  - Achieved higher recovery with same amount of additional dataset
    - With five-month-long observation
- Outlook
  - Validate performance recursively
  - Substitute unlabeled fingerprint as labeled fingerprint to let No-sweat Detective be autonomous