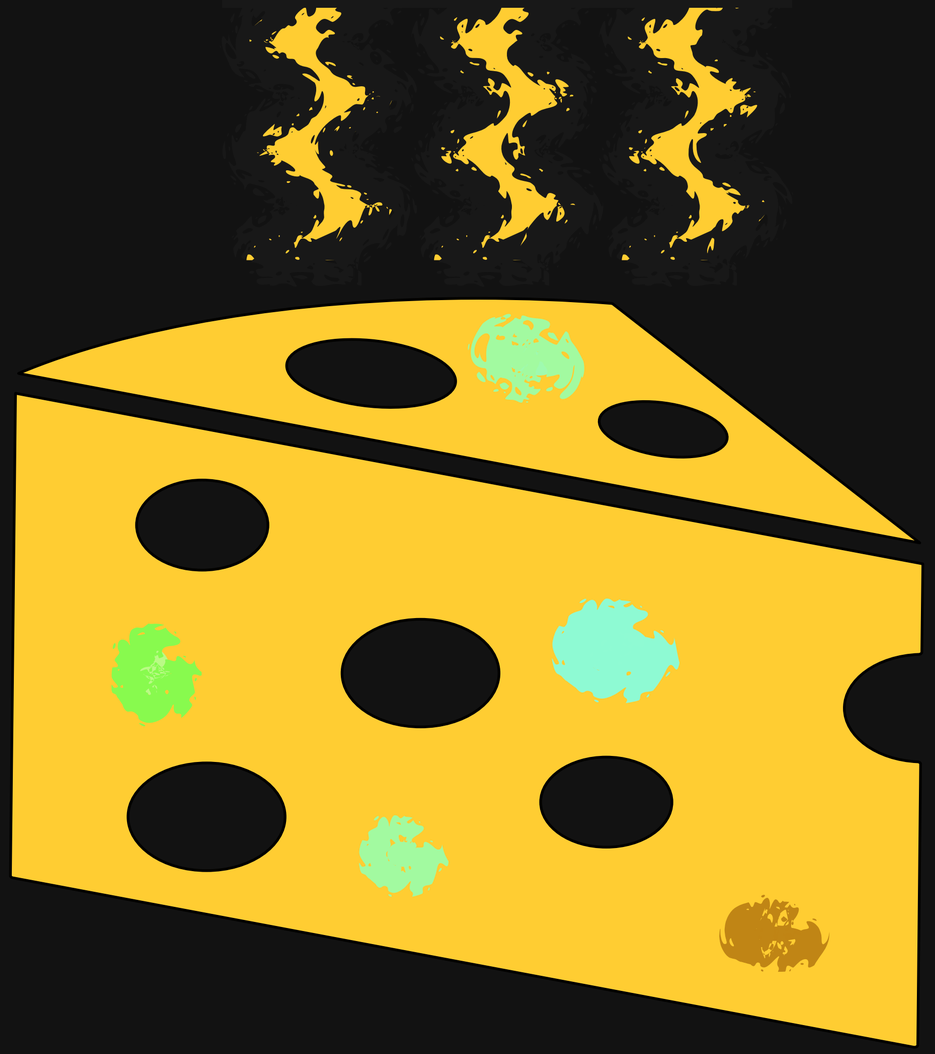
# Machine Learned Model Quality Monitoring in Fast Data and Streaming Applications

Emre Velipasaoglu

Lightbend

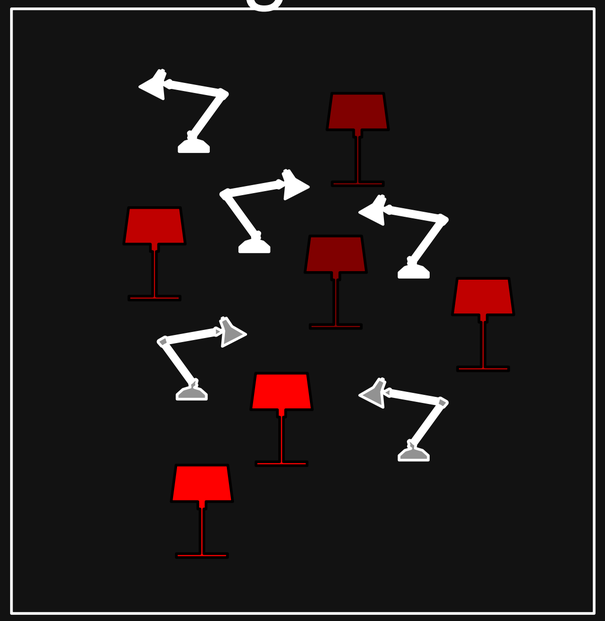
Strata
DATA CONFERENCE


model
quality

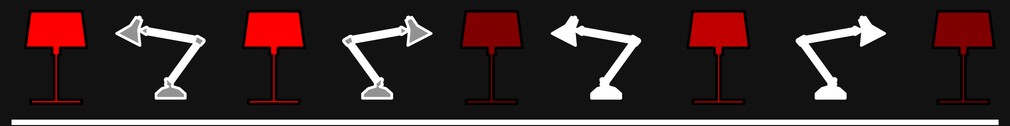


# core problem

training set



$\stackrel{*}{=}$

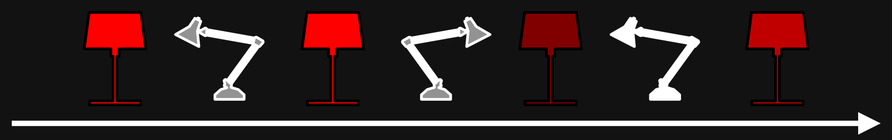


operation

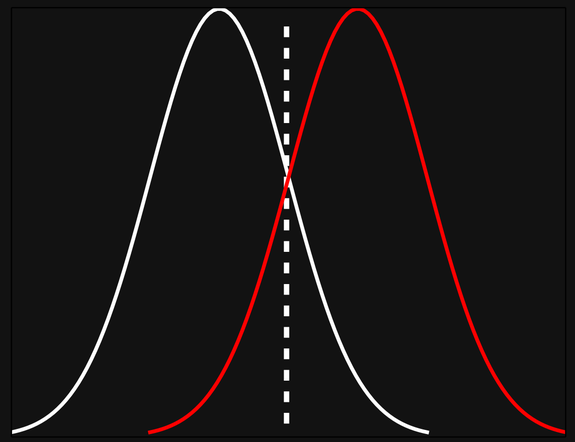
* same data generating distribution

(Some algorithms tolerate violation of this to a certain degree.)

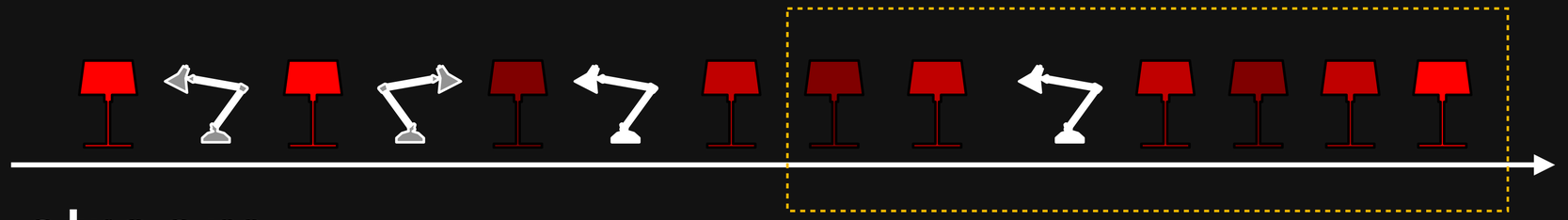
# core problem



stream

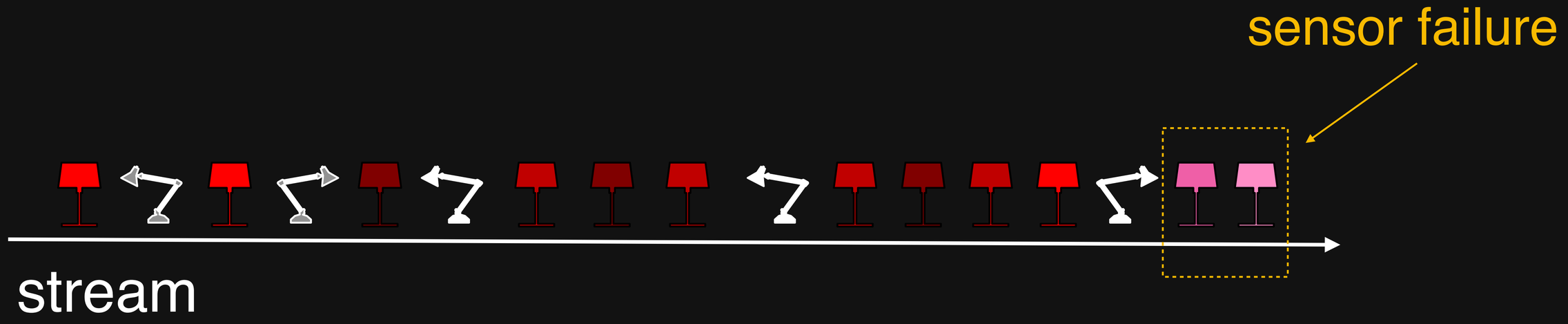


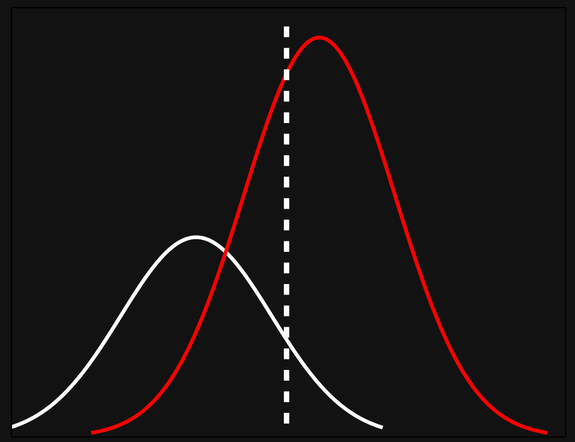
# core problem



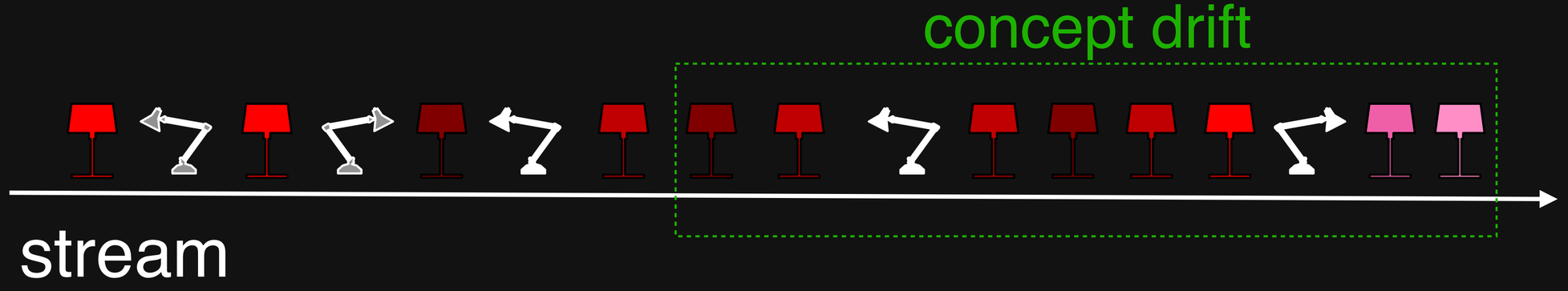

population change

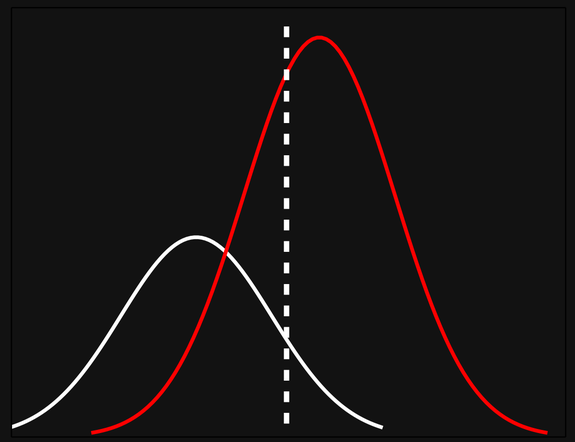
# core problem



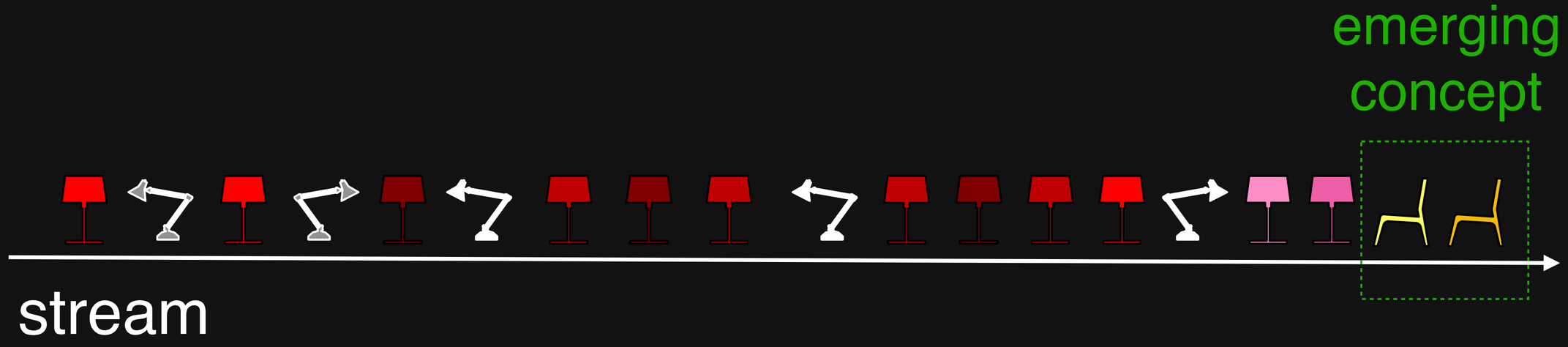



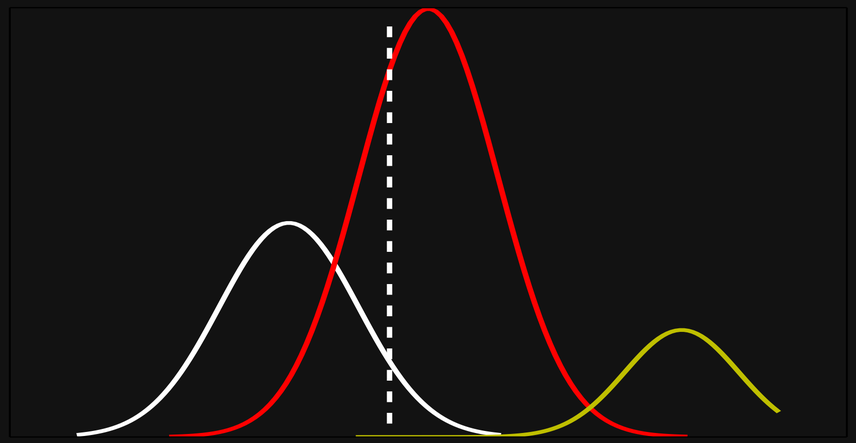
# core problem



# core problem



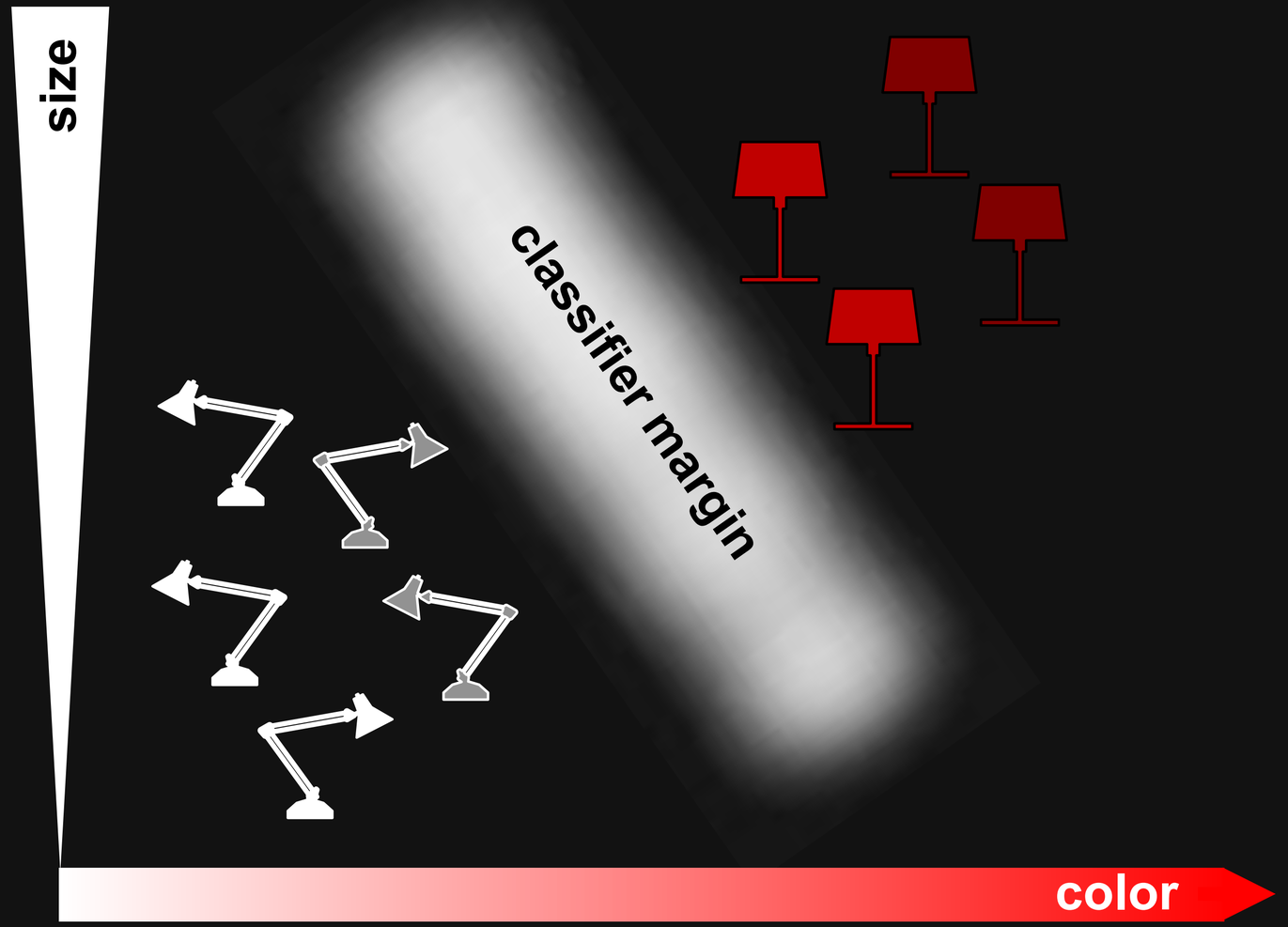
common 

solution

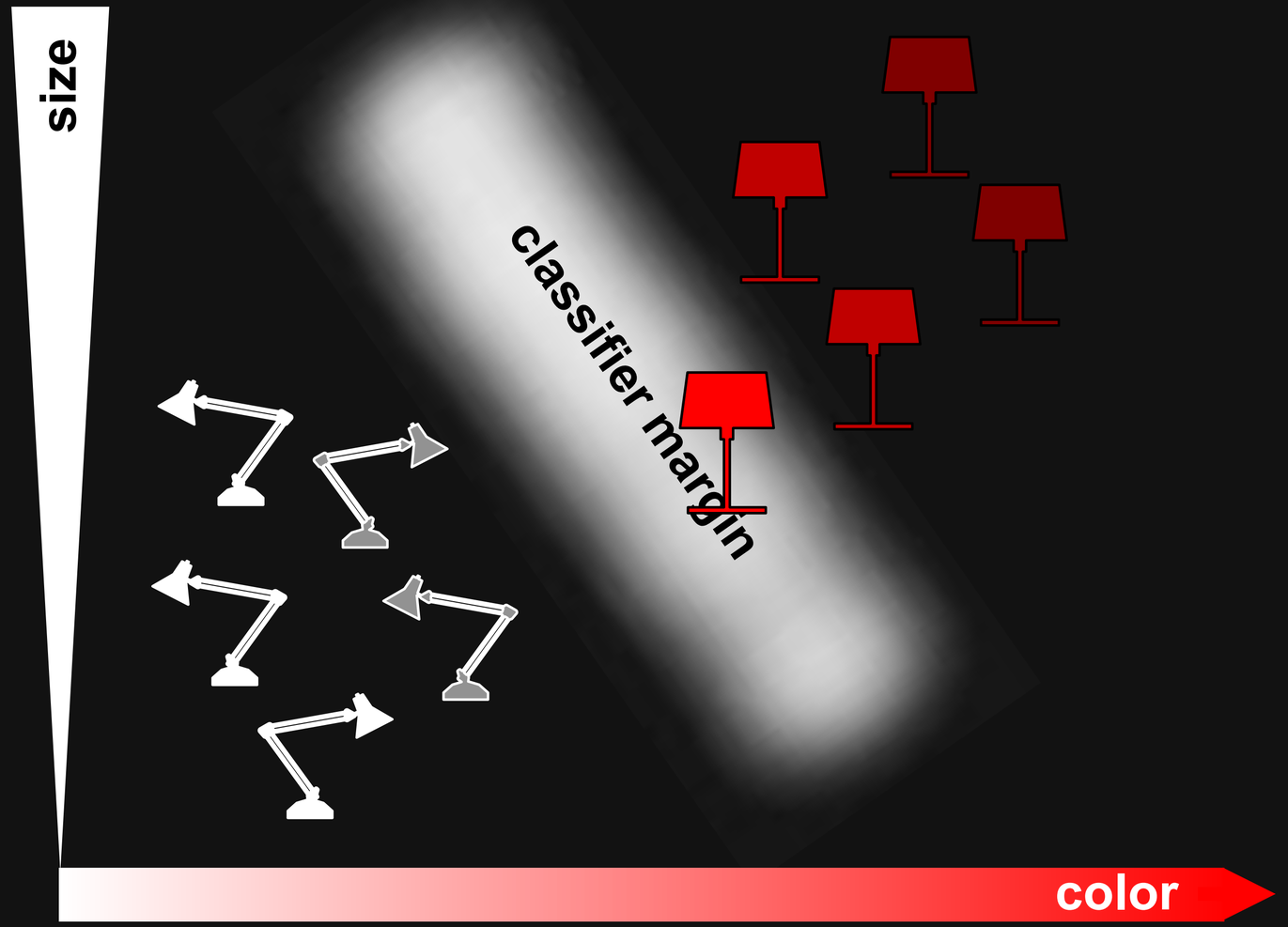
model

batch

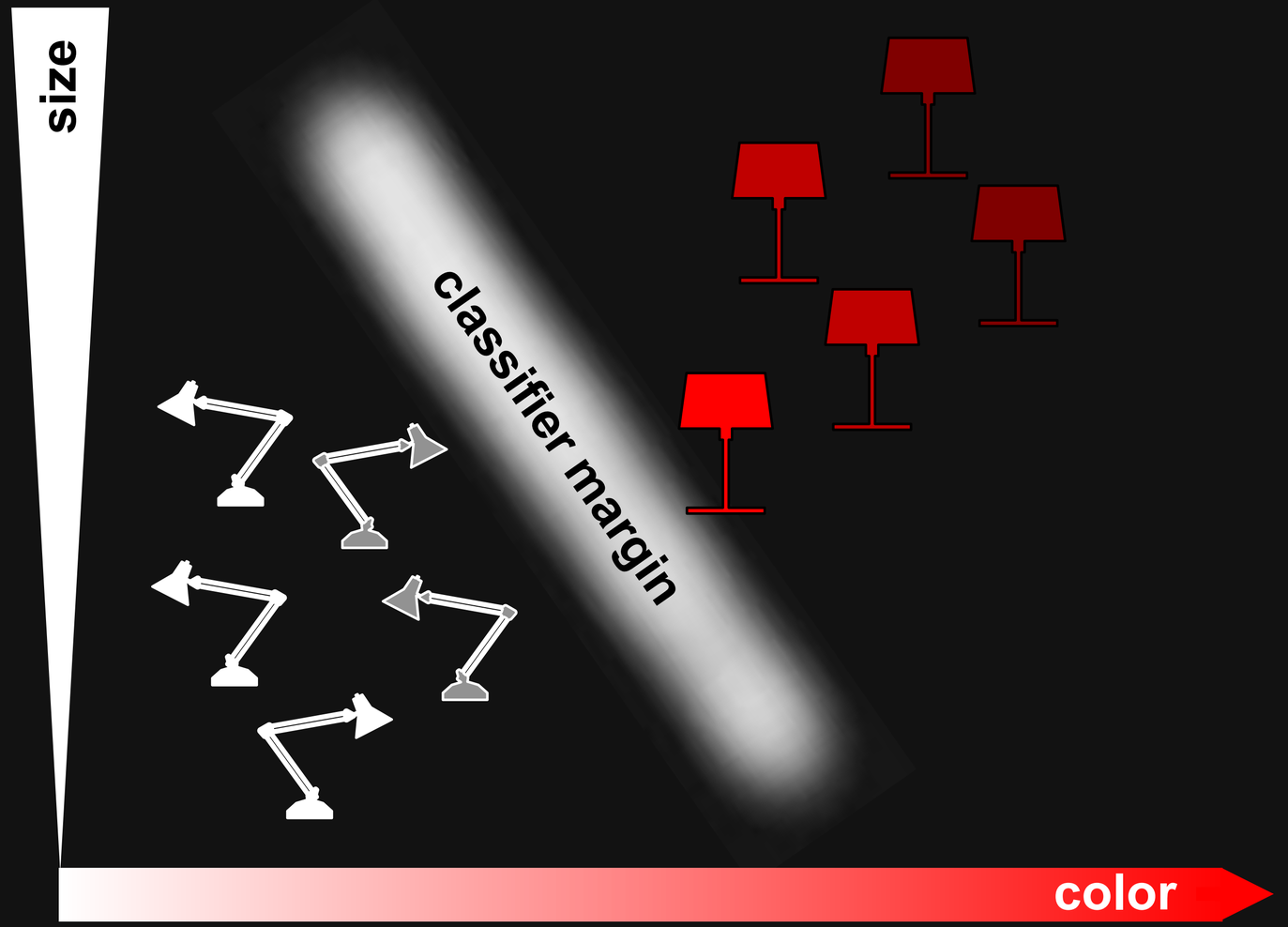
# can active learning help?



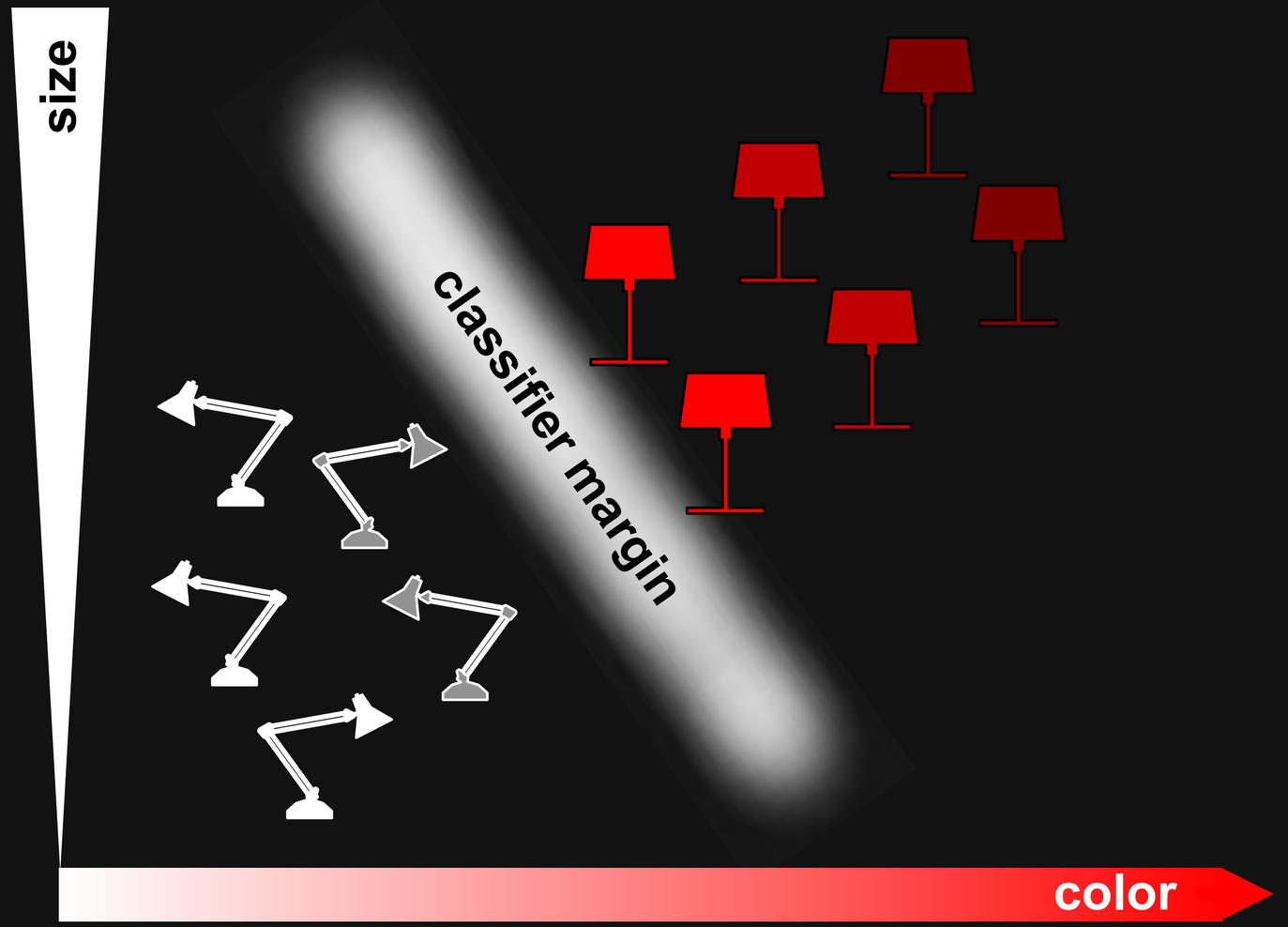
# can active learning help?



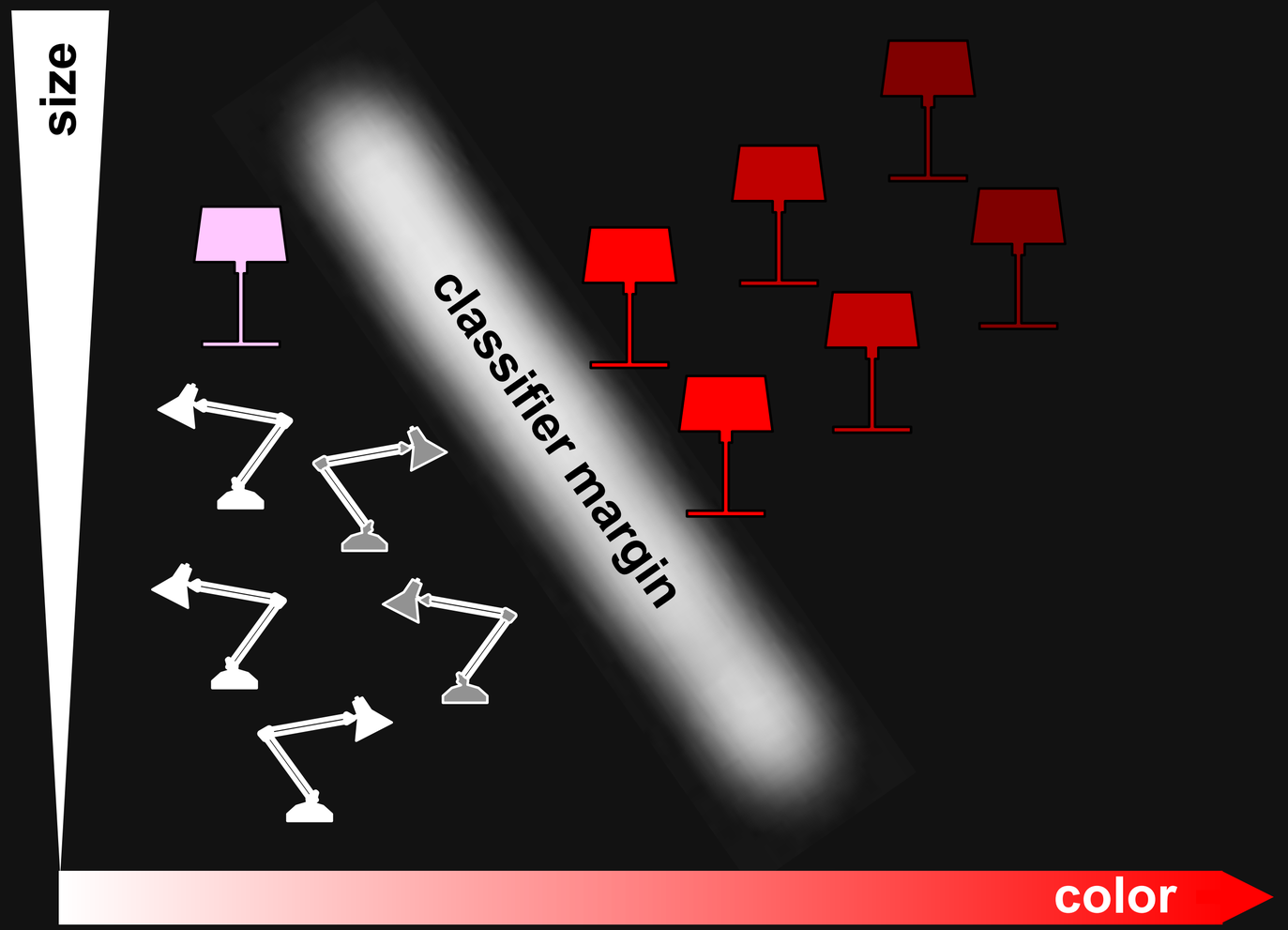
# can active learning help?



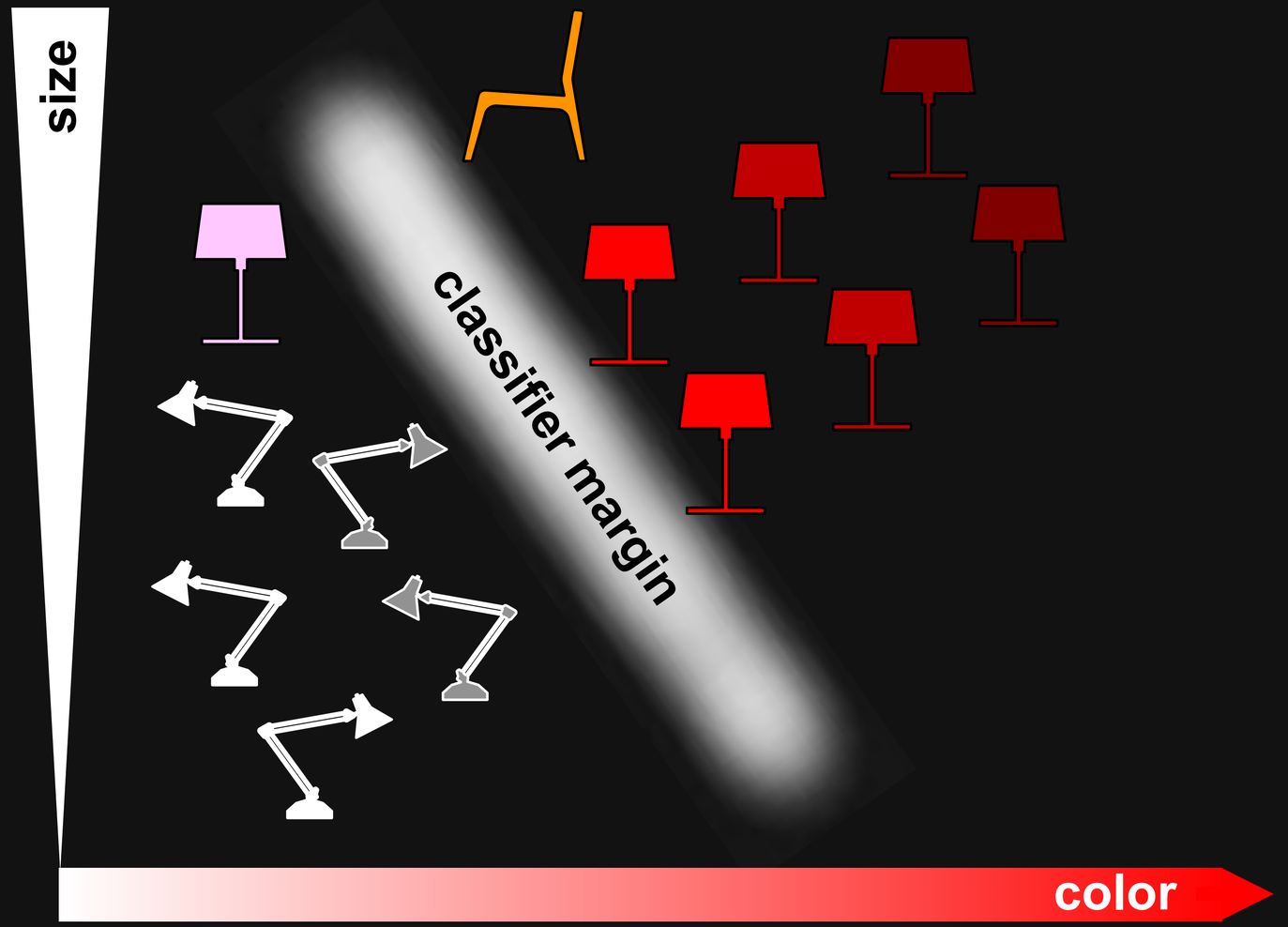
# can active learning help?



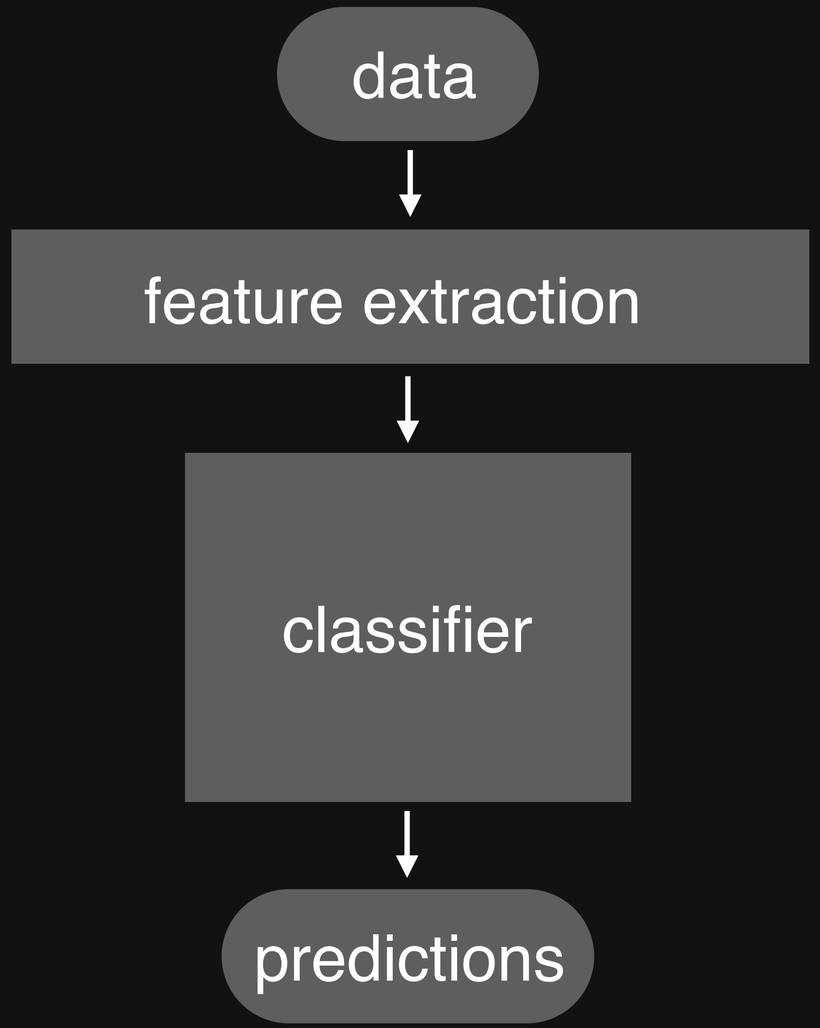
# can active learning help?



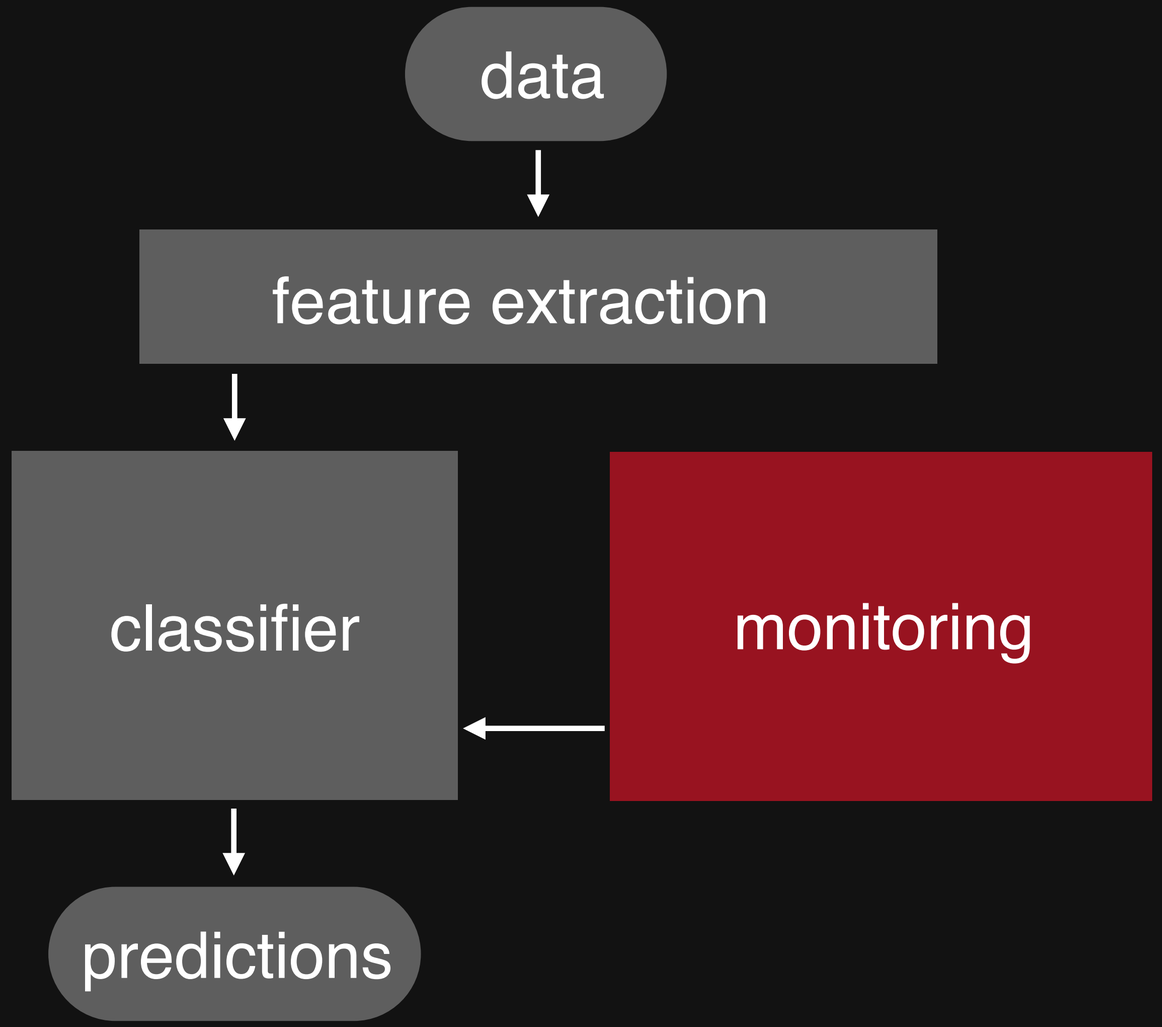
# can active learning help?



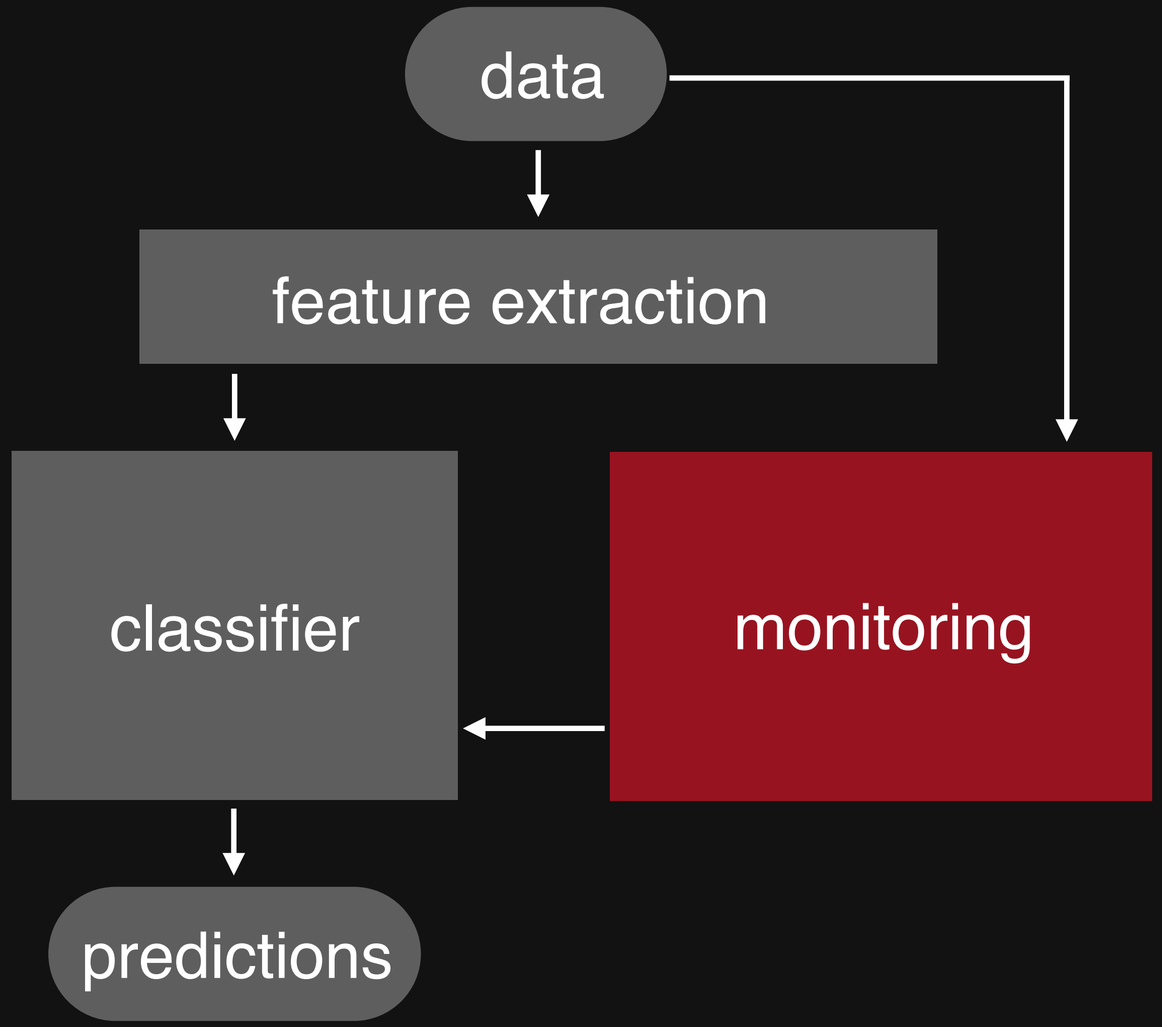
# a better solution



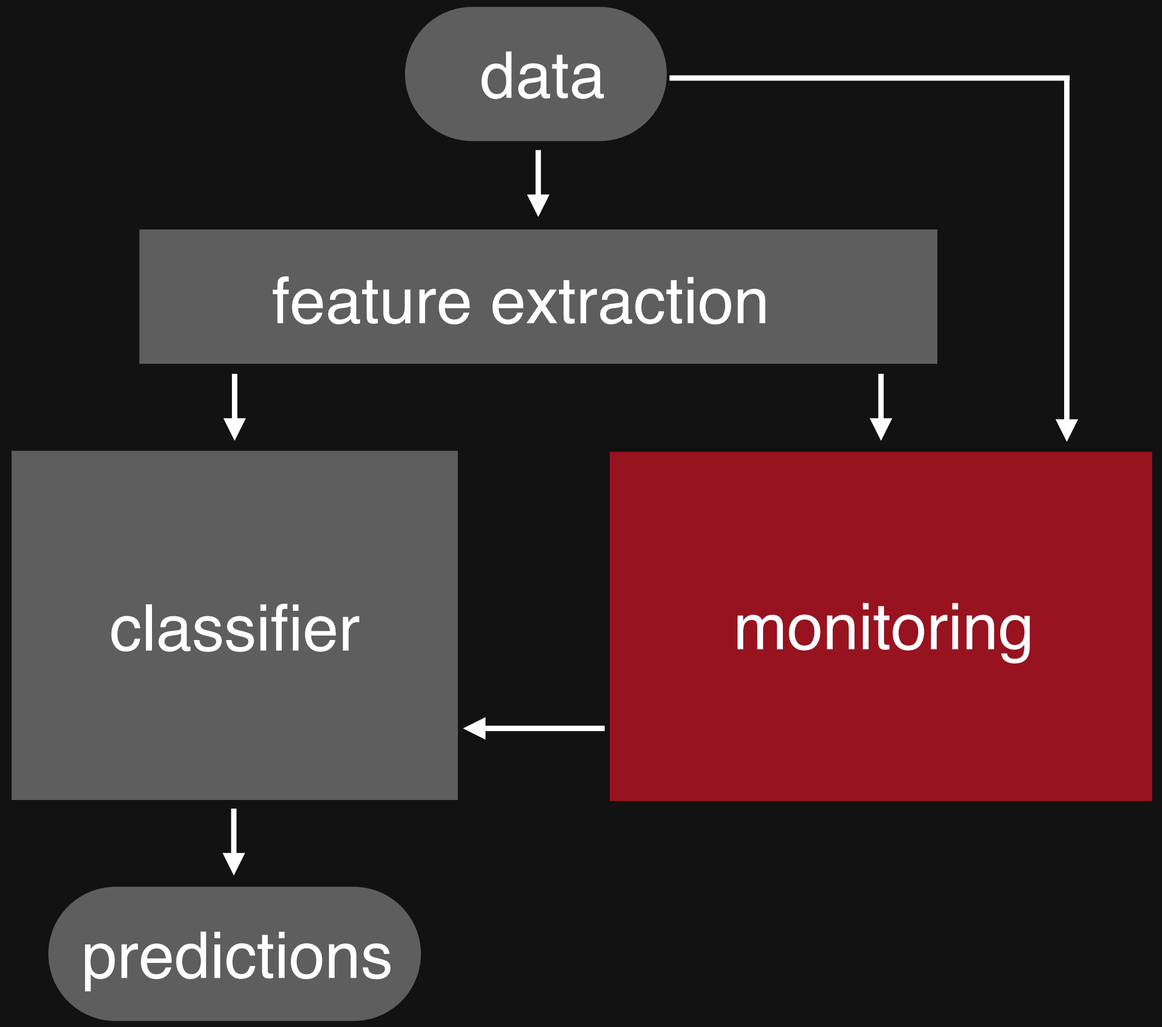
# a better solution



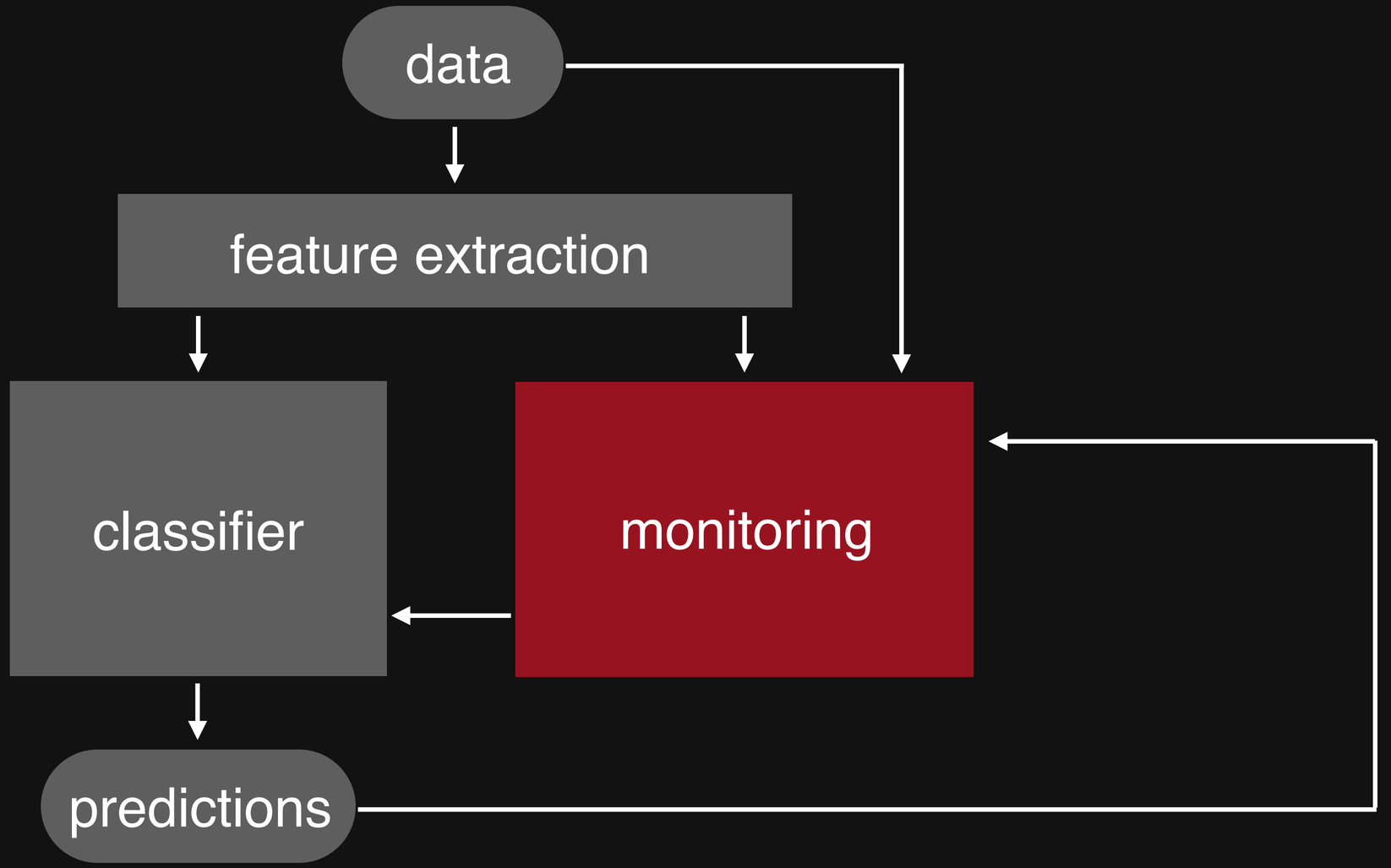
# a better solution



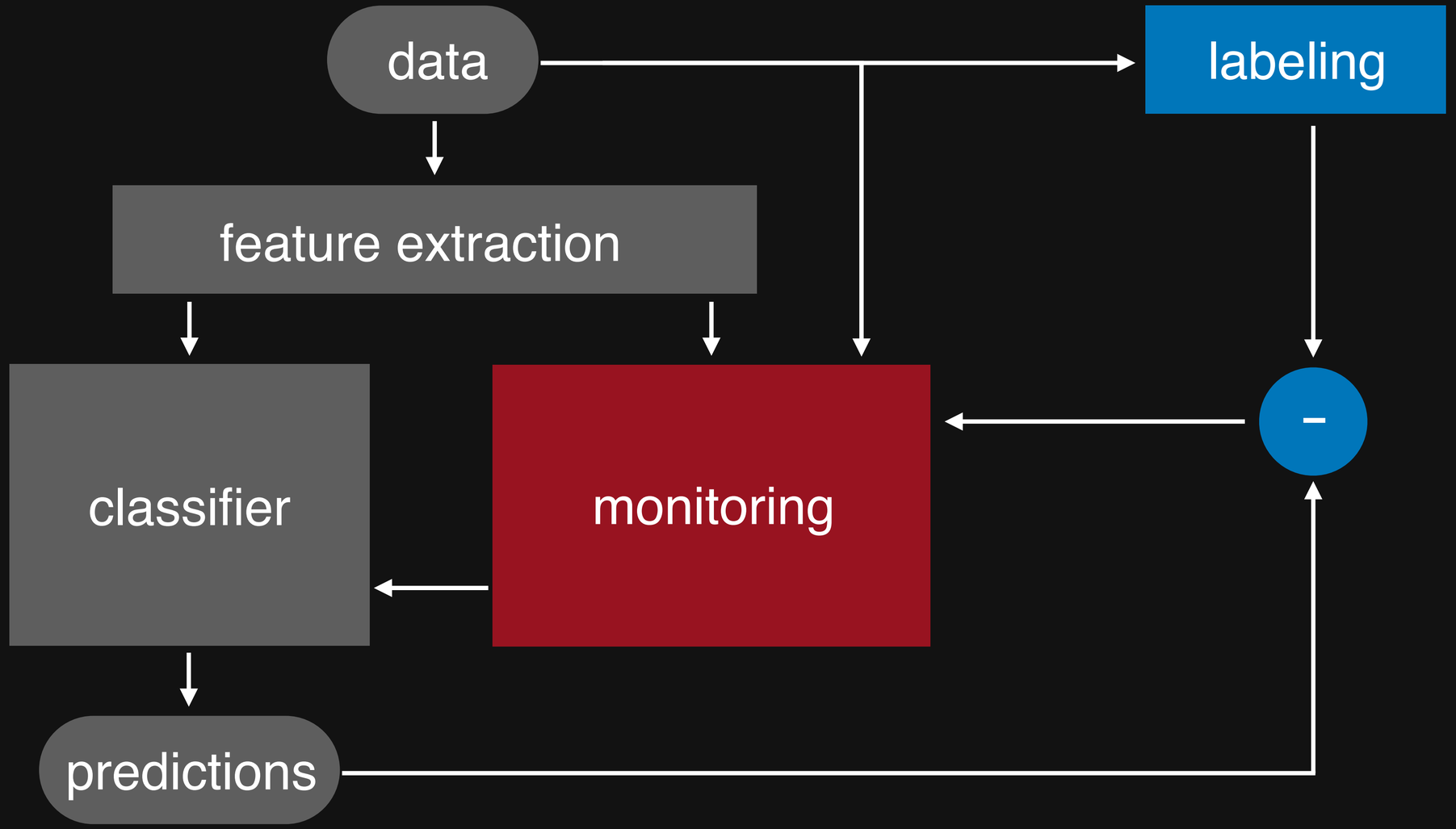
# a better solution



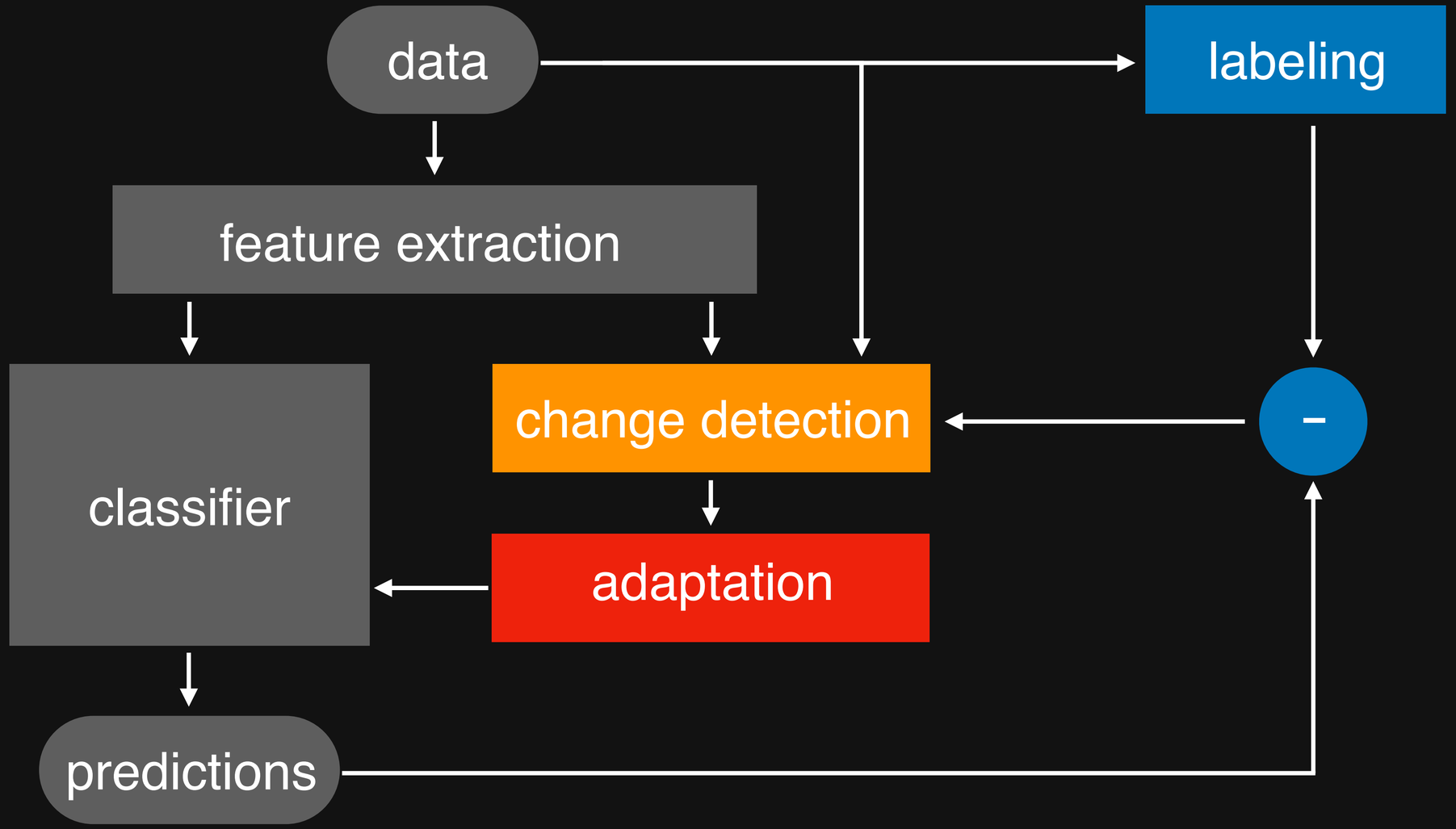
# a better solution

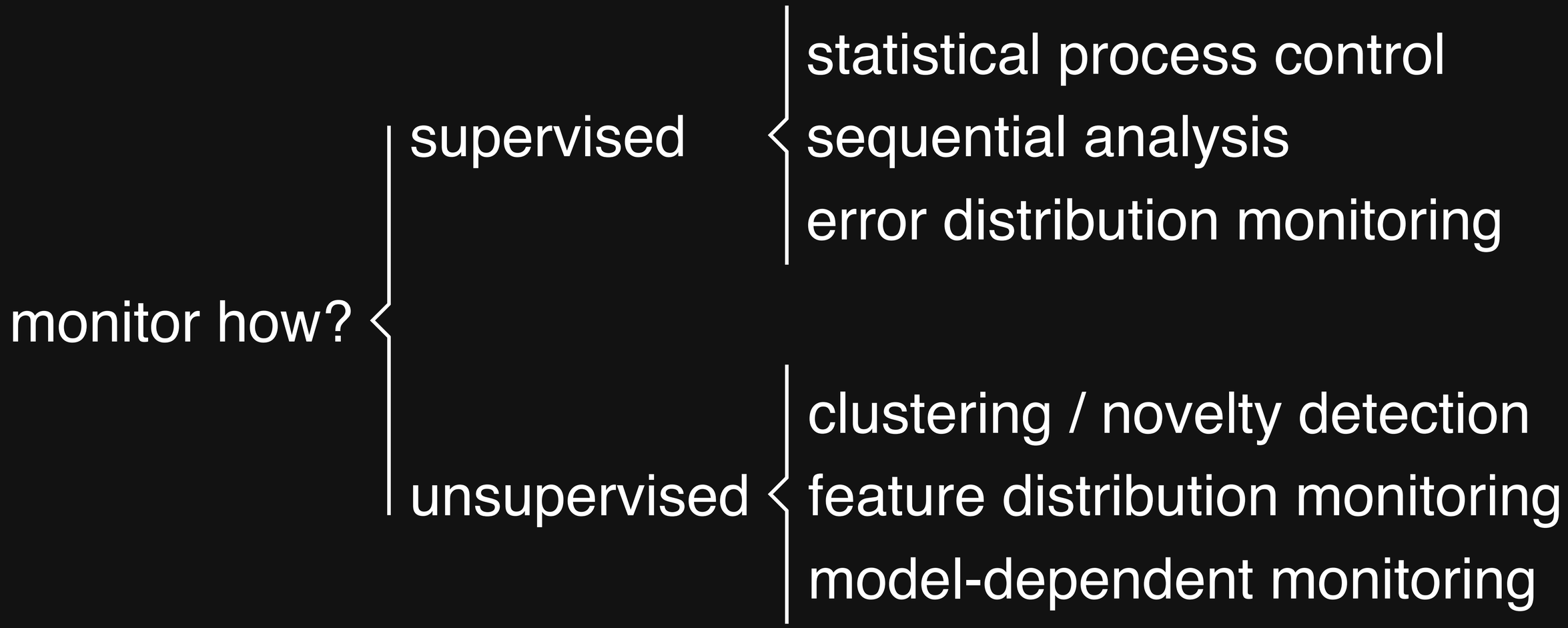


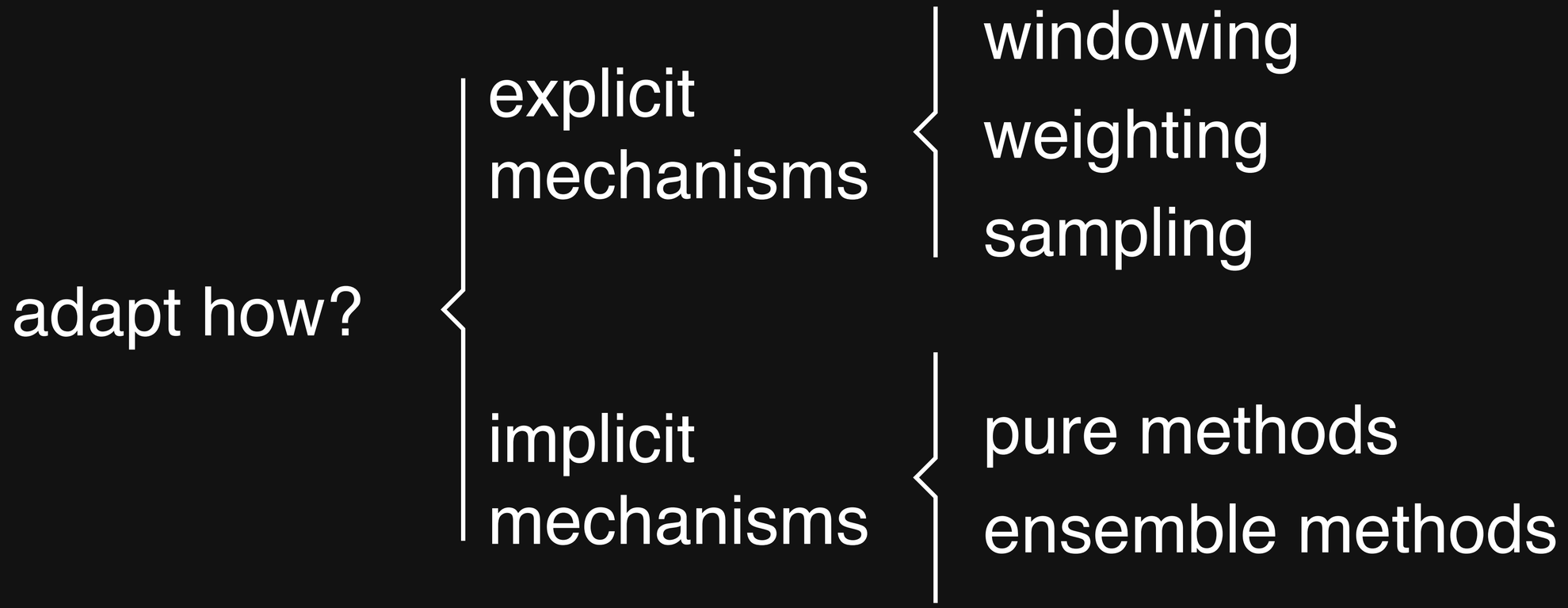
# a better solution



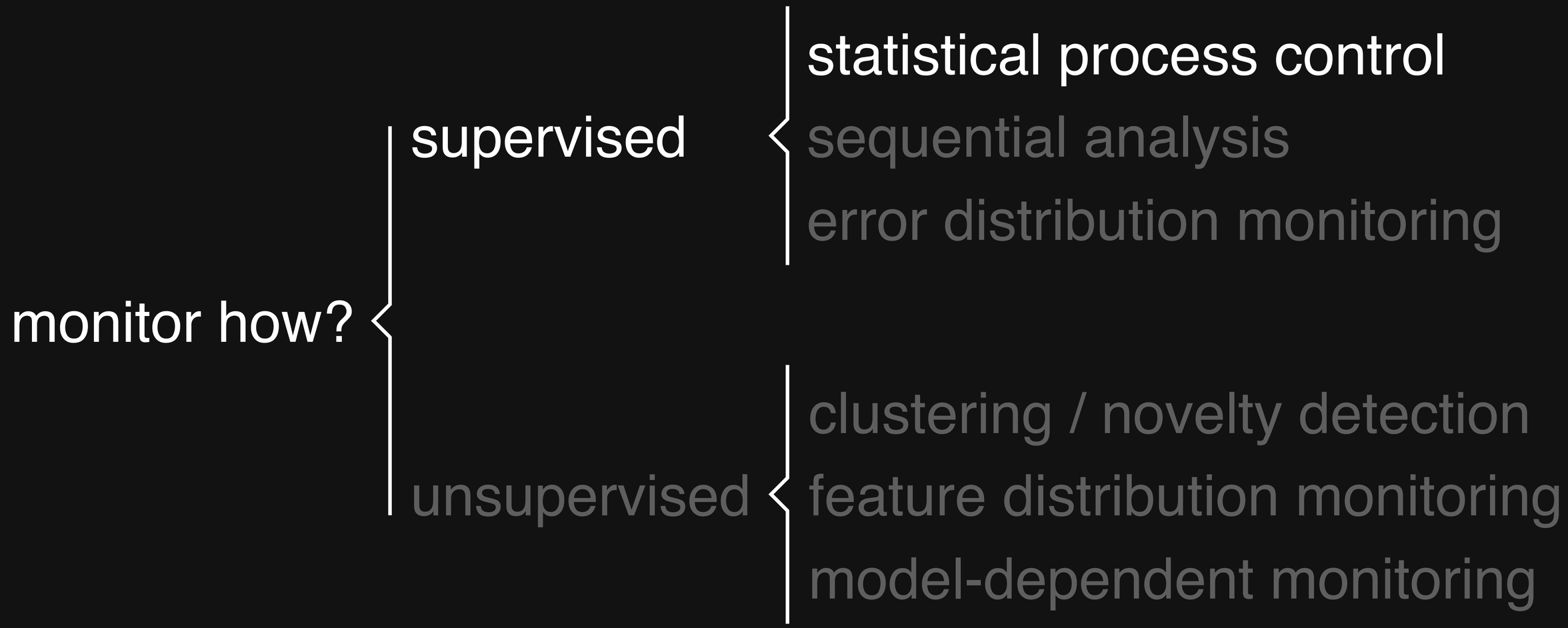
# a better solution

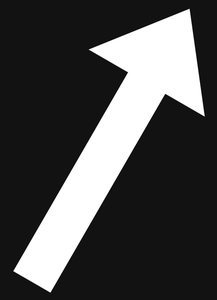



monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring


adapt how?
explicit mechanisms
windowing
weighting
sampling
implicit mechanisms
pure methods
ensemble methods

which method?


monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring

ML theory: samples  errors 

# statistical process control

- Drift Detection Method [DDM]
  - # of errors is Binomial:

$$\mu = np_t$$

$$\sigma = \sqrt{\frac{p_t(1-p_t)}{n}}$$

  - alert:

$$p_t + \sigma_t \geq p_{min} + 3\sigma_{min}$$

# statistical process control

- Drift Detection Method [DDM]
  - # of errors is Binomial:

$$\mu = np_t$$

$$\sigma = \sqrt{\frac{p_t(1-p_t)}{n}}$$

  - alert:

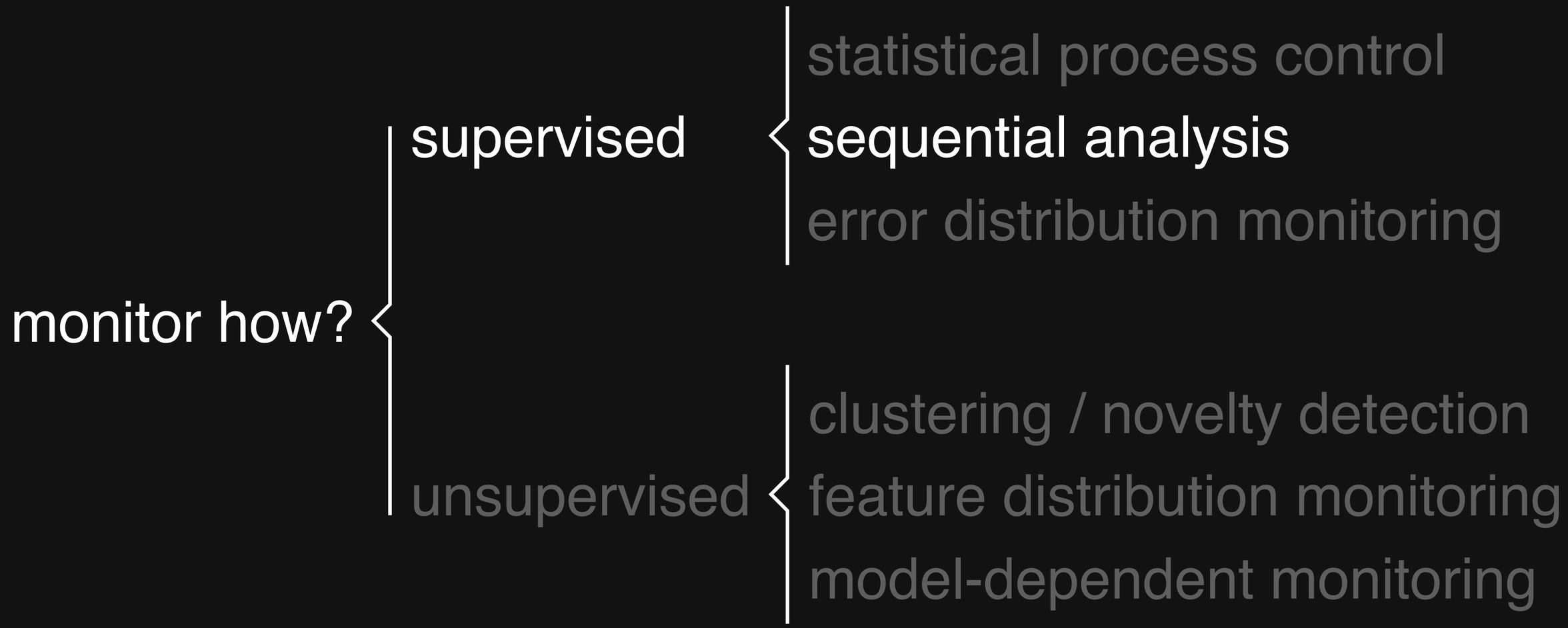
$$p_t + \sigma_t \geq p_{min} + 3\sigma_{min}$$

- Early Drift Detection Method [EDDM]
  - distance between errors
    better for gradual drift
  - warn & start caching:

$$\frac{p_t + 2\sigma_t}{p_{max} + 2\sigma_{max}} < 0.95$$

  - alert and reset max:

$$\frac{p_t + 2\sigma_t}{p_{max} + 2\sigma_{max}} < 0.90$$


monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring

# sequential analysis

- Linear Four Rates [LFR]
  - stationary data => constant contingency table

| True / Predicted | 0 | 1 |
|---|---|---|
| 0 | TN | FN |
| 1 | FP | TP |

# sequential analysis

- Linear Four Rates [LFR]
  - stationary data => constant contingency table
  - calculate four rates

| Predicted \ True | 0 | 1 |
|---|---|---|
| 0 | TN | FN |
| 1 | FP | TP |

$$P_{npv} = \frac{TN}{TN + FN}$$

$$P_{ppv/precision} = \frac{TP}{TP + FP}$$

$$P_{tnr/specificity} = \frac{TN}{TN + FP} \qquad P_{tpr/recall} = \frac{TP}{TP + FN}$$

# sequential analysis

- Linear Four Rates [LFR]
  - stationary data => constant contingency table
  - calculate four rates
  - incremental updates

| Predicted \ True | 0 | 1 |
|---|---|---|
| 0 | TN | FN |
| 1 | FP | TP |

$$P_{npv} = \frac{TN}{TN + FN}$$

$$P_{ppv/precision} = \frac{TP}{TP + FP}$$

$$P_{tnr/specificity} = \frac{TN}{TN + FP} \qquad P_{tpr/recall} = \frac{TP}{TP + FN}$$

$$P_*{}^t \leftarrow \eta_* P_*{}^{t-1} + (1 - \eta_*) I_{y_t = \hat{y}_t}$$

# sequential analysis

- Linear Four Rates [LFR]
  - stationary data => constant contingency table
  - calculate four rates
  - incremental updates
  - test for change
    - Monte Carlo sampling for significance level
    - Bonferoni correction for correlated tests
  - O(1)
  - Better than (E)DDM for class imbalance

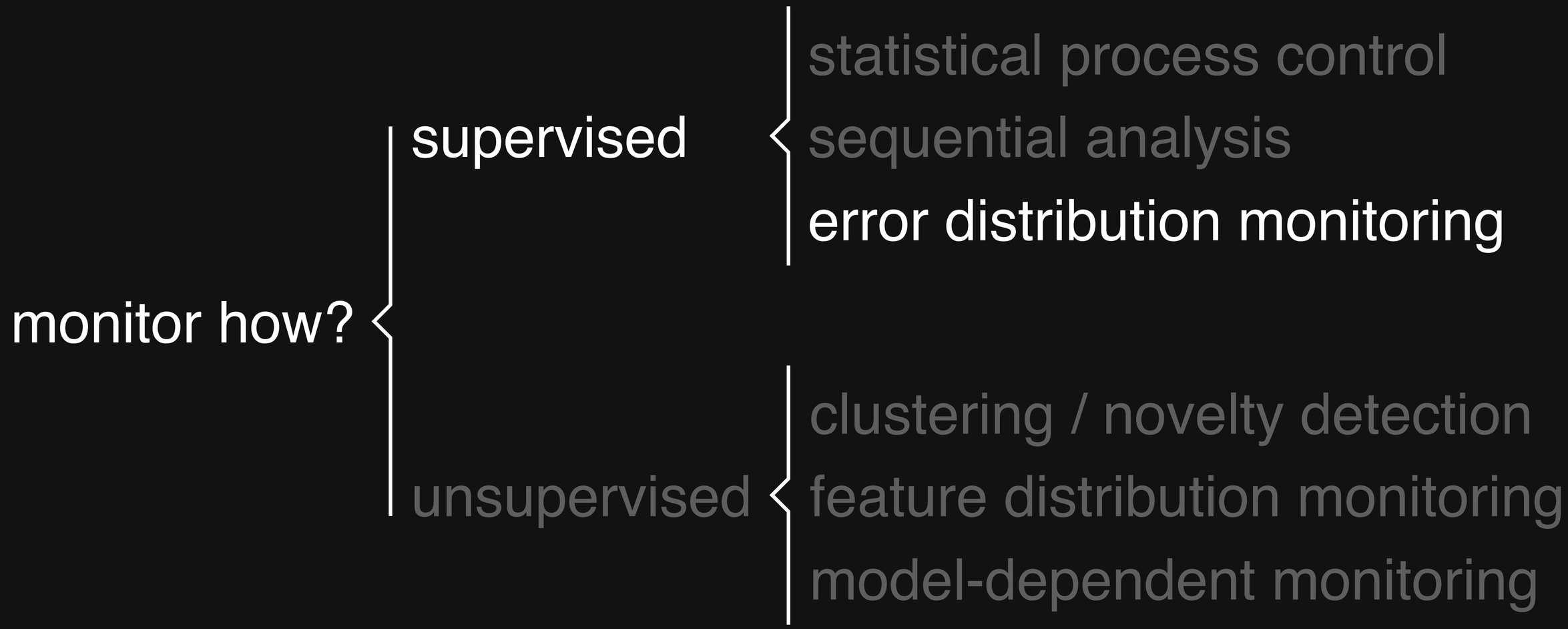
| Predicted \ True | 0 | 1 |
|---|---|---|
| 0 | TN | FN |
| 1 | FP | TP |

$$P_{npv} = \frac{TN}{TN + FN}$$

$$P_{ppv/precision} = \frac{TP}{TP + FP}$$

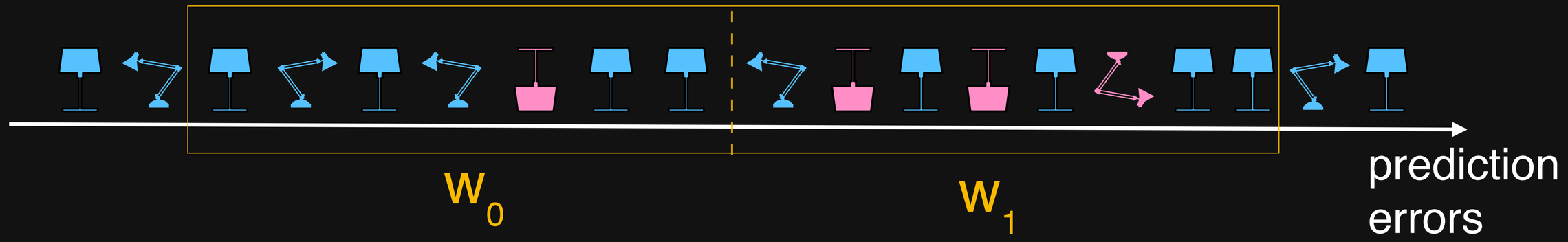
$$P_{tnr/specificity} = \frac{TN}{TN + FP} \qquad P_{tpr/recall} = \frac{TP}{TP + FN}$$

$$P_*{}^t \leftarrow \eta_* P_*{}^{t-1} + (1 - \eta_*) I_{y_t = \hat{y}_t}$$


monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring

# error distribution monitoring

- ADaptive WINdowing [ADWIN]
  - Consider all partitions of a window



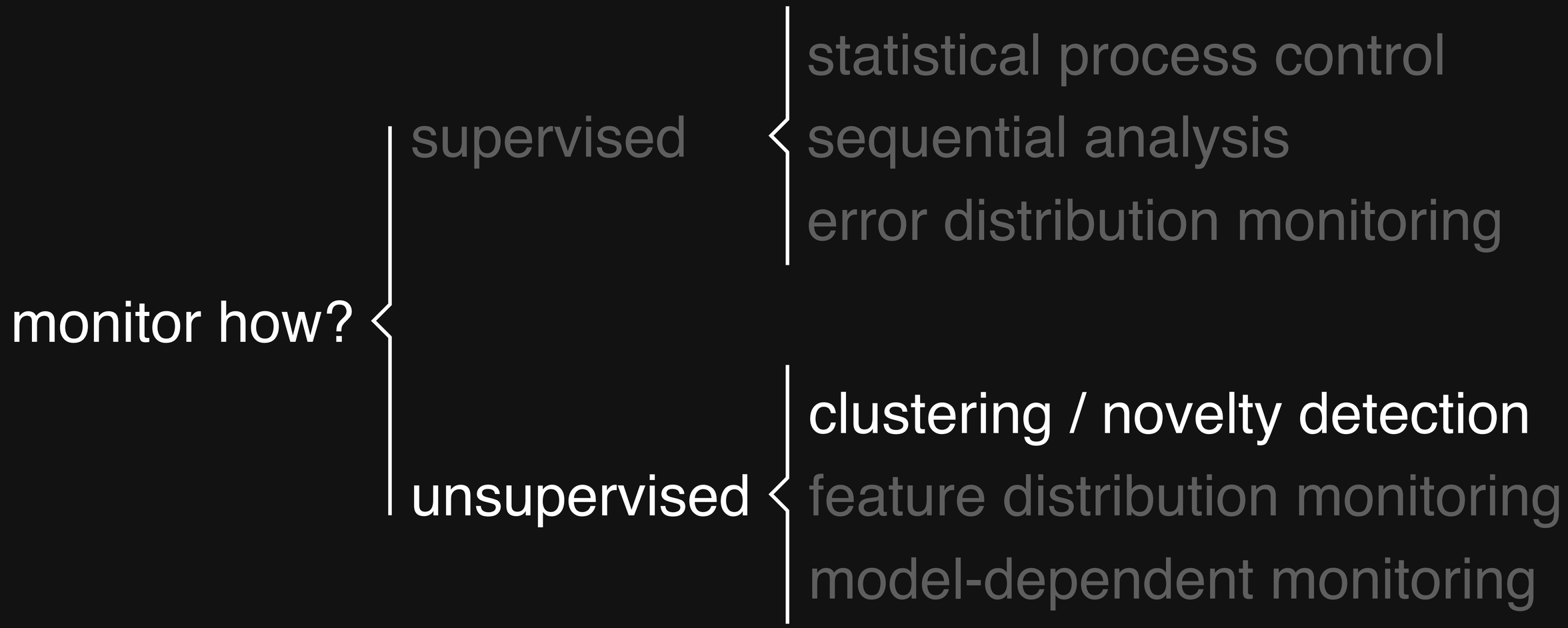

  - Drop the last element if any

$$|\mu_0 - \mu_1| > \theta_{Hoeffding}$$

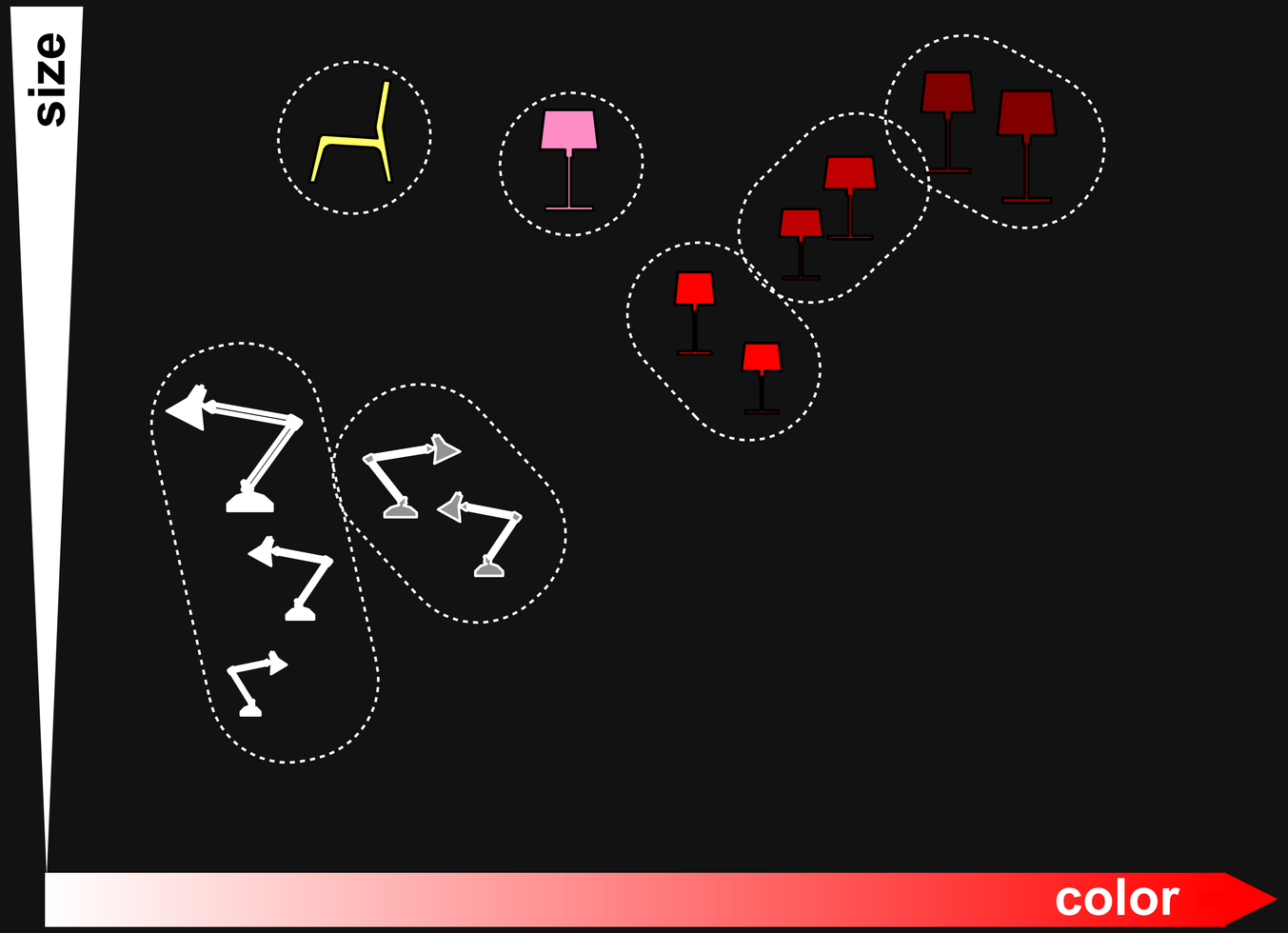
  - Efficient version O(log W)
    - Data structure for windows ~ exponential histograms
    - Drop last window rather than last element

# resampling

- Prediction loss over random permutations vs. ordered training data
- Parallel permutation test version available
- Still expensive
- Only method directly applicable to regression setting
- Side note: Even with finite training set, drift could be problematic if model is developed naively.

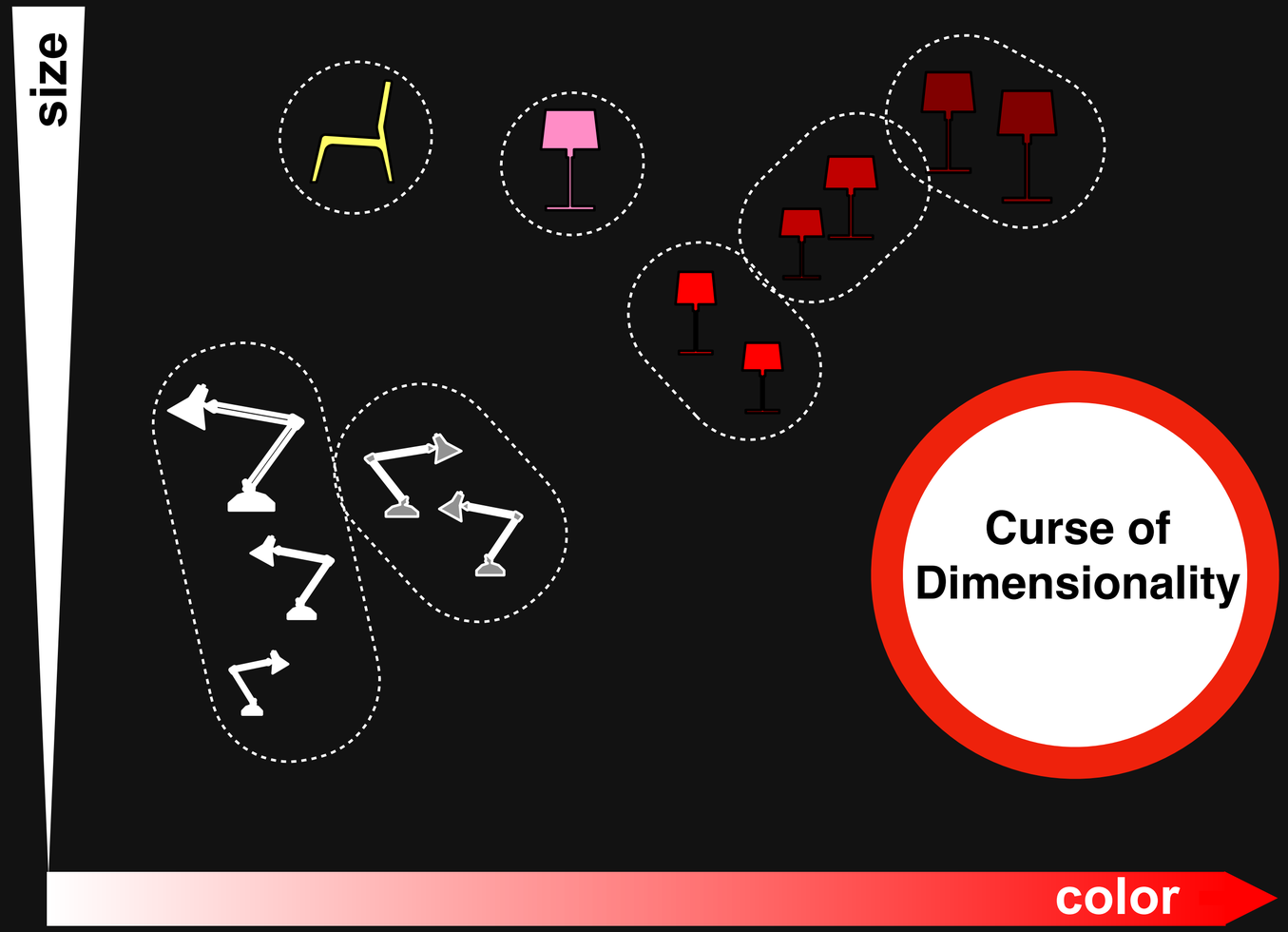

monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring

# clustering / novelty detection

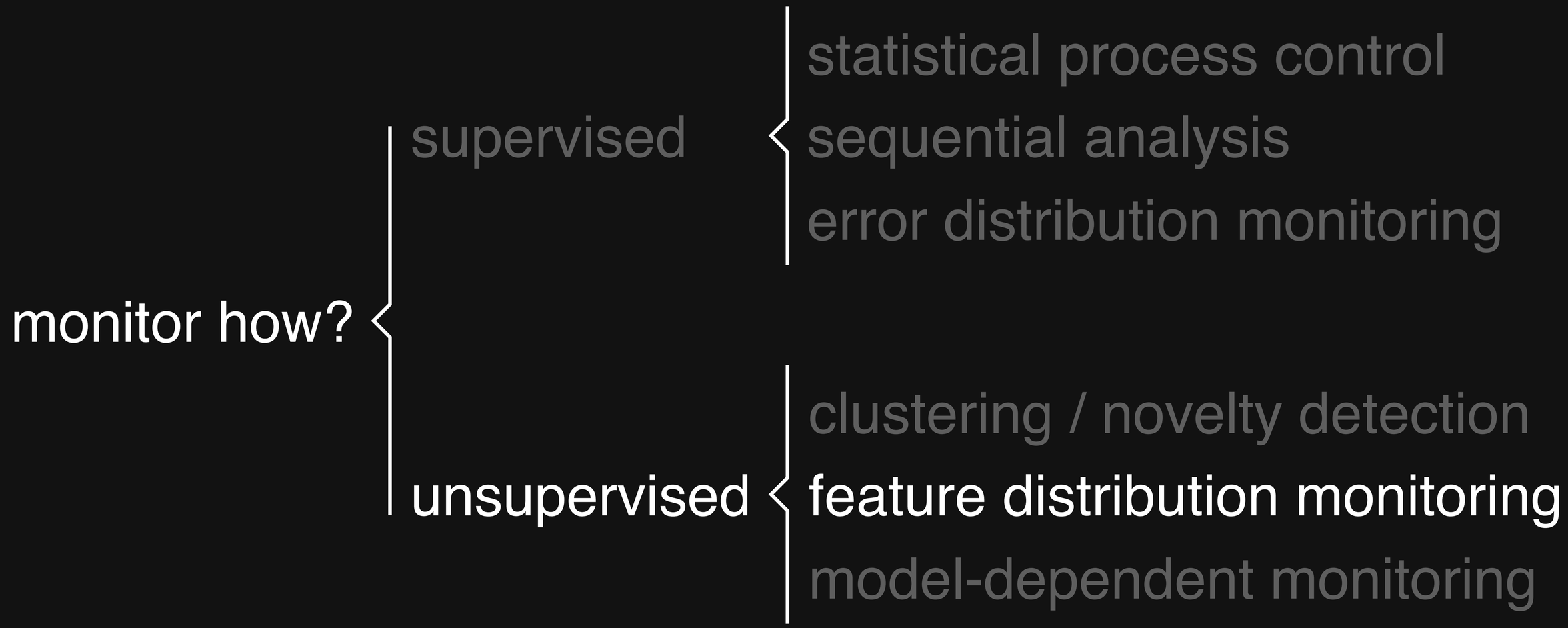



- OLINDDA: K-means, periodically merge unknown to known or flag
- MINAS: micro-clusters, incremental stream clustering
- DETECTNOD: Discrete Cosine Transform to estimate distances efficiently
- Woo-ensemble: Treat outliers as potential emerging class centroids
- ECSMiner: Store and use cluster summary efficiently
- GC3: Grid based clustering

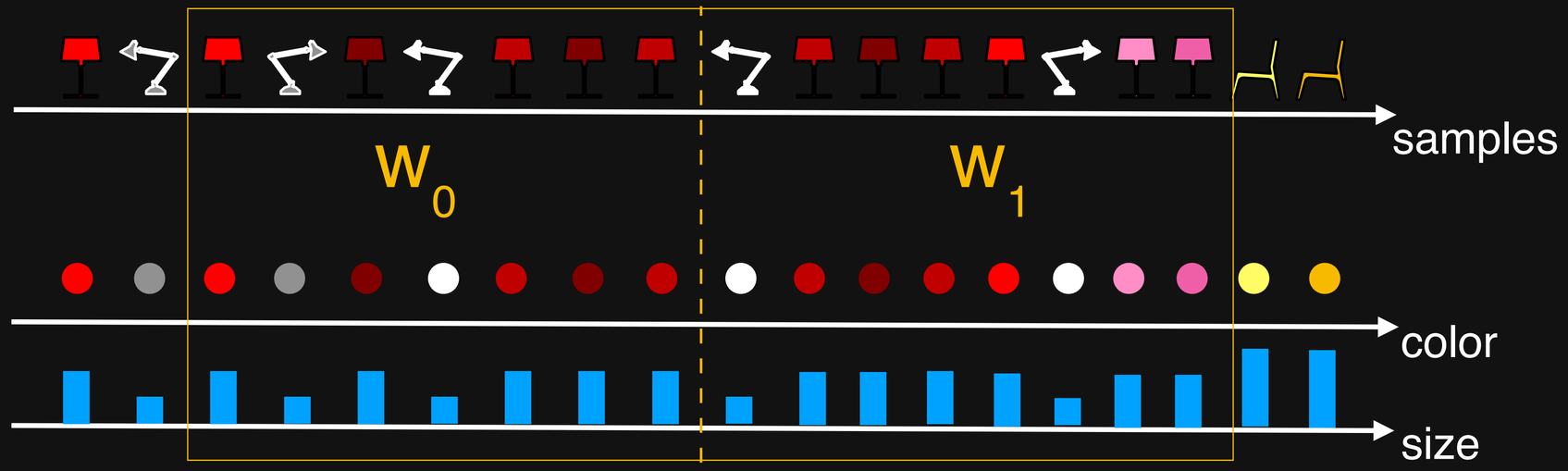
# clustering / novelty detection



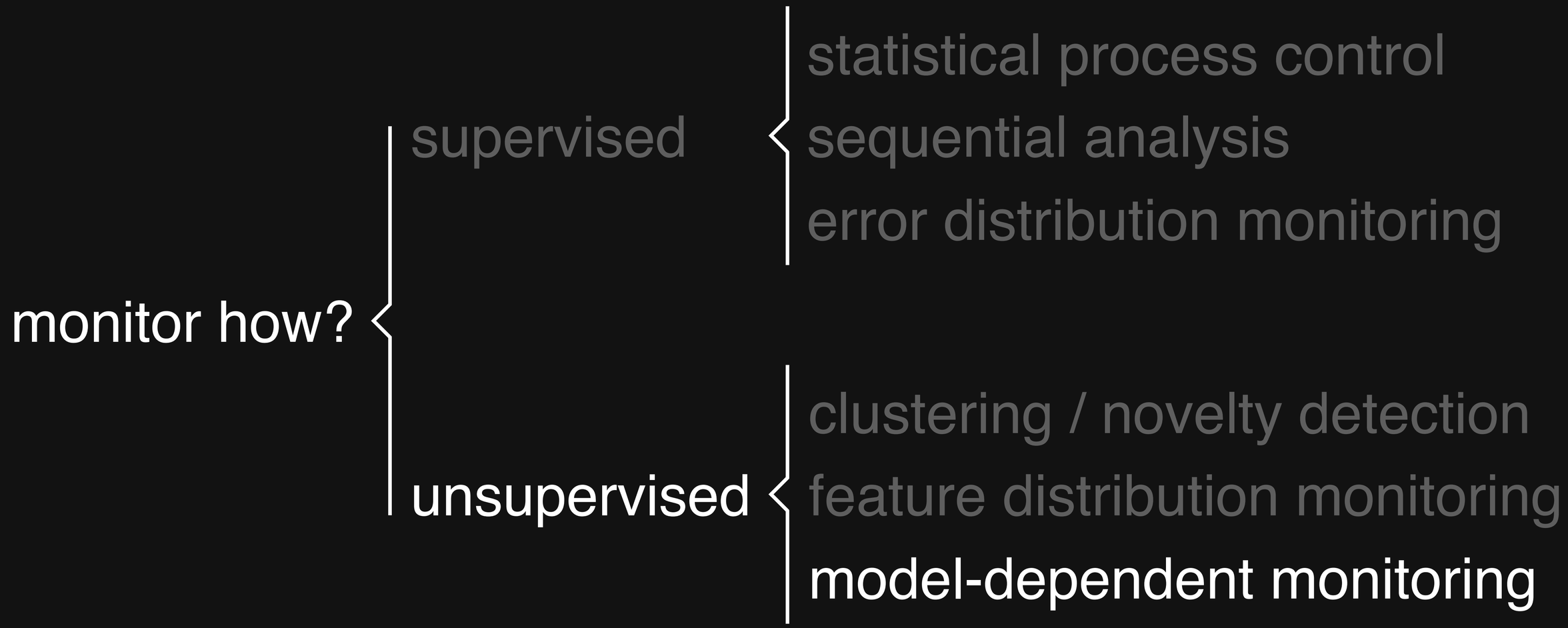

- OLINDDA: K-means, periodically merge unknown to known or flag
- MINAS: micro-clusters, incremental stream clustering
- DETECTNOD: Discrete Cosine Transform to estimate distances efficiently
- Woo-ensemble: Treat outliers as potential emerging class centroids
- ECSMiner: Store and use cluster summary efficiently
- GC3: Grid based clustering


monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring

# feature distribution monitoring



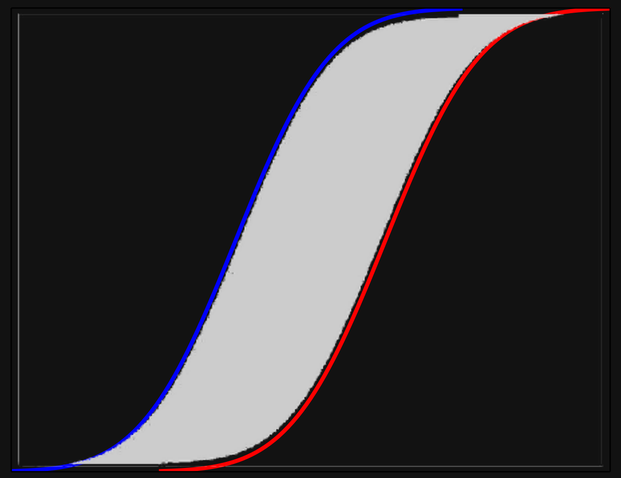

- Monitor individual features
- Many ways to compare:
  - Pearson correlation [Change of Concept - CoC]
  - Hellinger distance [HDDDM] ~ O(DB)
- Use PCA to reduce the number of features to track (top [PCA-1] or bottom [PCA-2] n%)


monitor how?
supervised
statistical process control
sequential analysis
error distribution monitoring
unsupervised
clustering / novelty detection
feature distribution monitoring
model-dependent monitoring

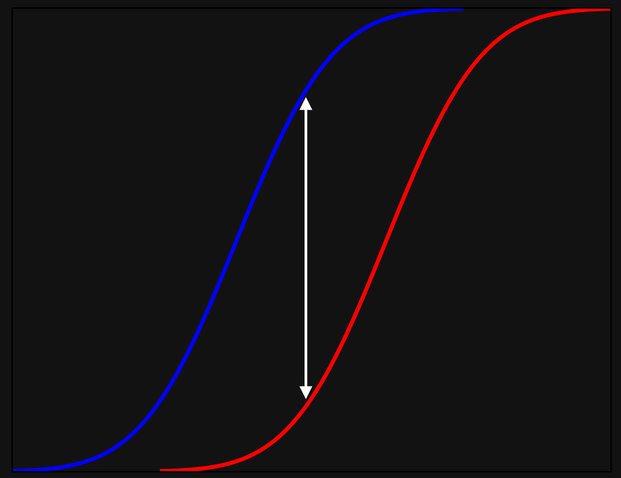
# model-dependent monitoring

- Not all changes matter
- Posterior probability estimate
  - Use [A-distance] ~ generalized KS distance
    - designed to be less sensitive to irrelevant changes

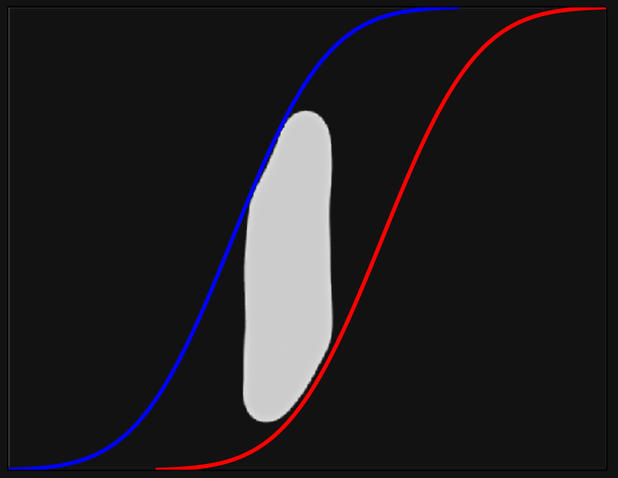
L1-distance



KS-distance

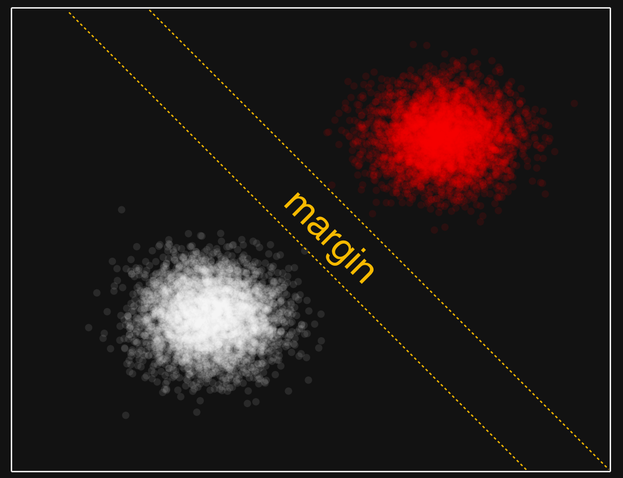


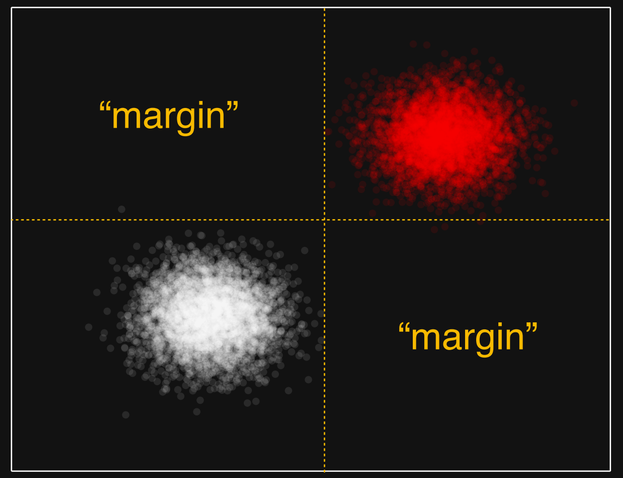
A-distance

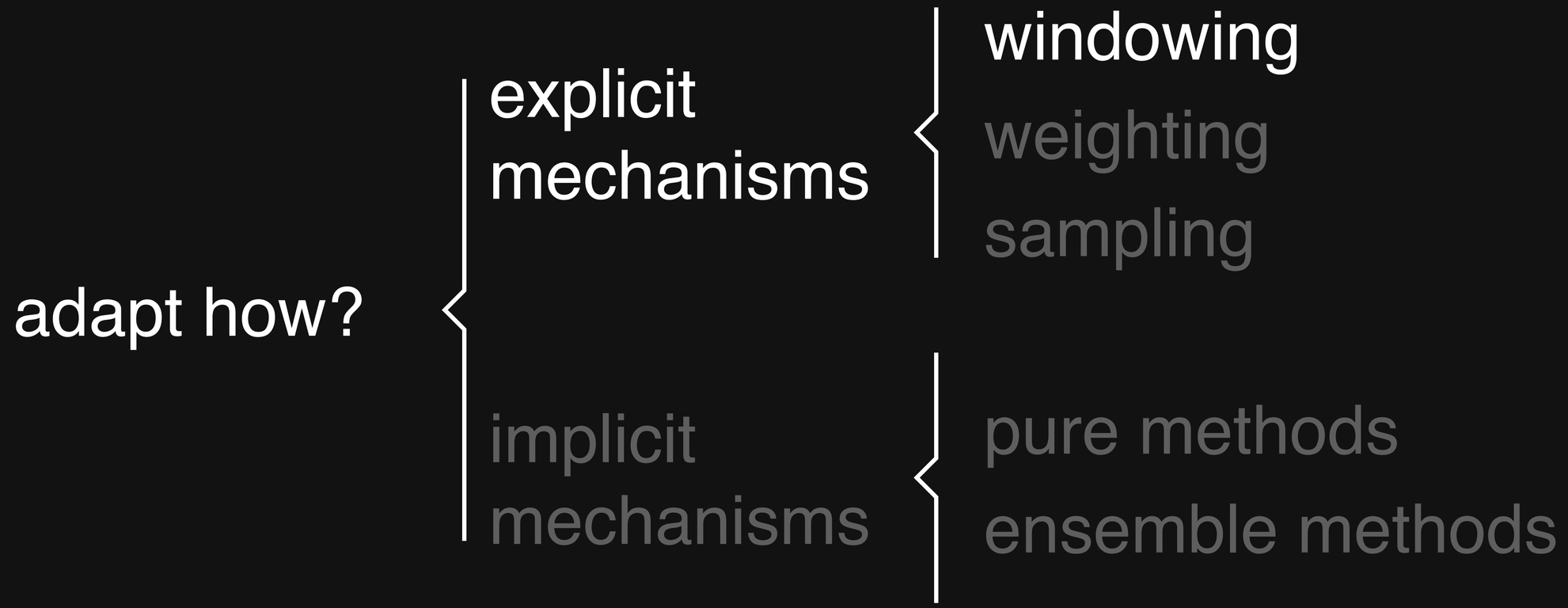


# model-dependent monitoring

- [Margin] distribution
  - rank statistic on density estimates for a binary representation of the data,
  - compare average margins of a linear classifier induced by the 1-norm SVM
  - based on the average zero-one or sigmoid error rate of an SVM classifier
- Generalized margin [MD3]:
  - Embed base classifier in a Random Feature Bagged Ensemble
  - Margin == high disagreement region of the ensemble

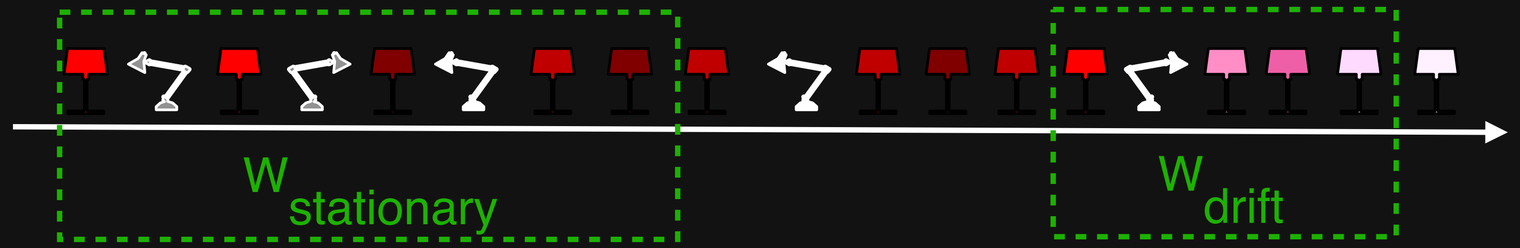



adapt how?
explicit mechanisms
windowing
weighting
sampling
implicit mechanisms
pure methods
ensemble methods

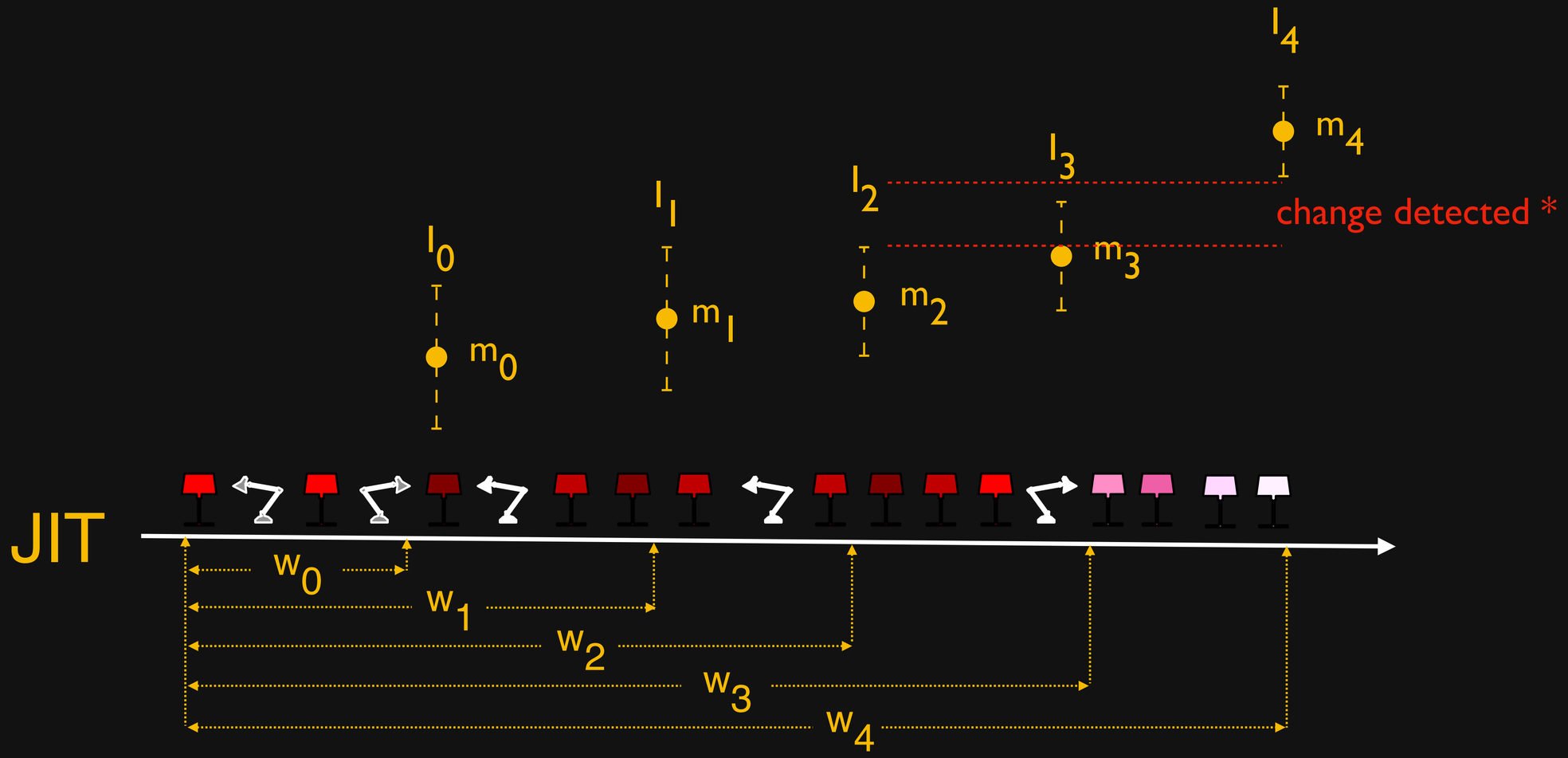
# explicit mechanisms for adaptation

ADWIN

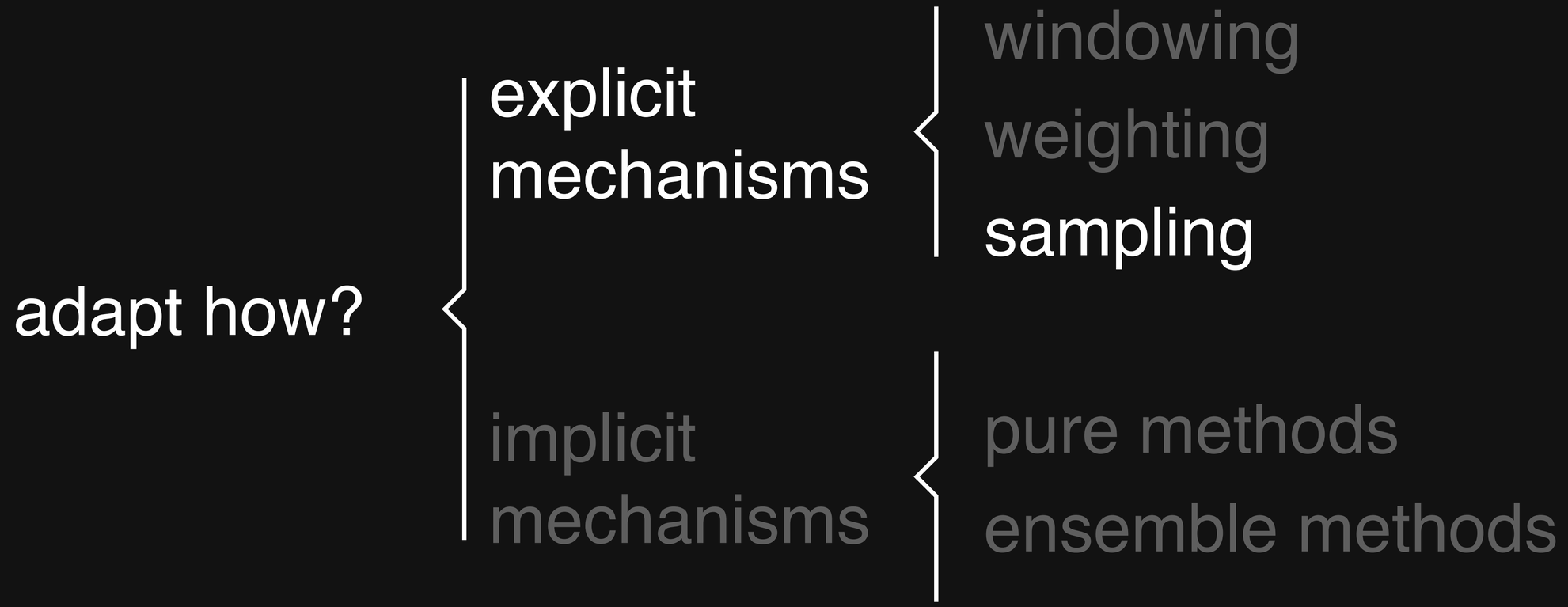



- Drop the last sub-window if threshold is exceeded. = Adaptively shrink window during drift.

# explicit mechanisms for adaptation

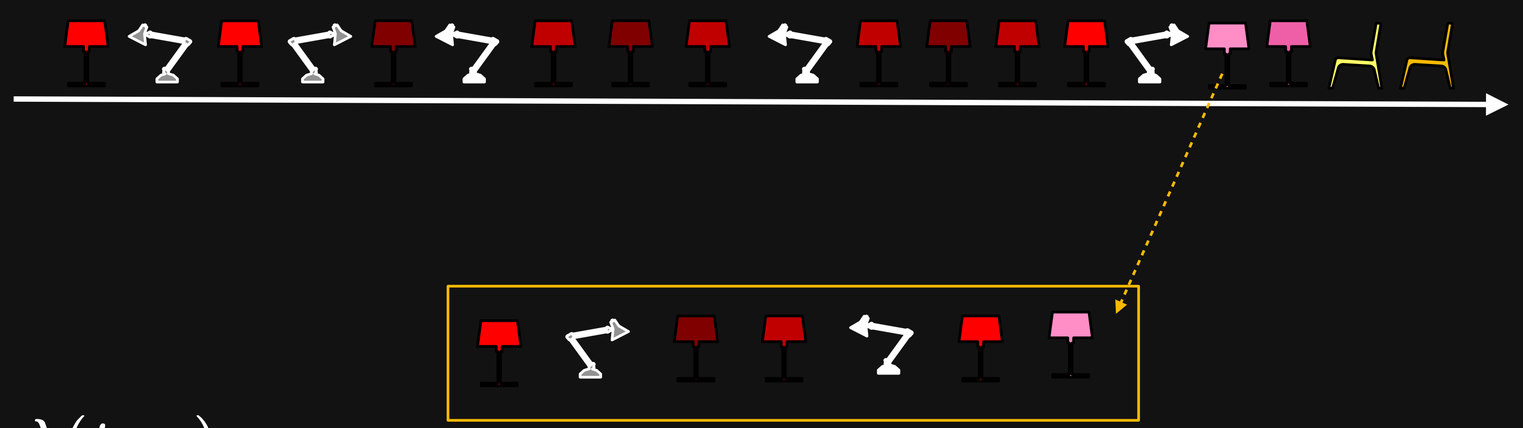



* Adaptation goes through a similar refinement process.


adapt how?
explicit mechanisms
windowing
weighting
sampling
implicit mechanisms
pure methods
ensemble methods

# explicit mechanisms for adaptation

Biased Reservoir Sampling



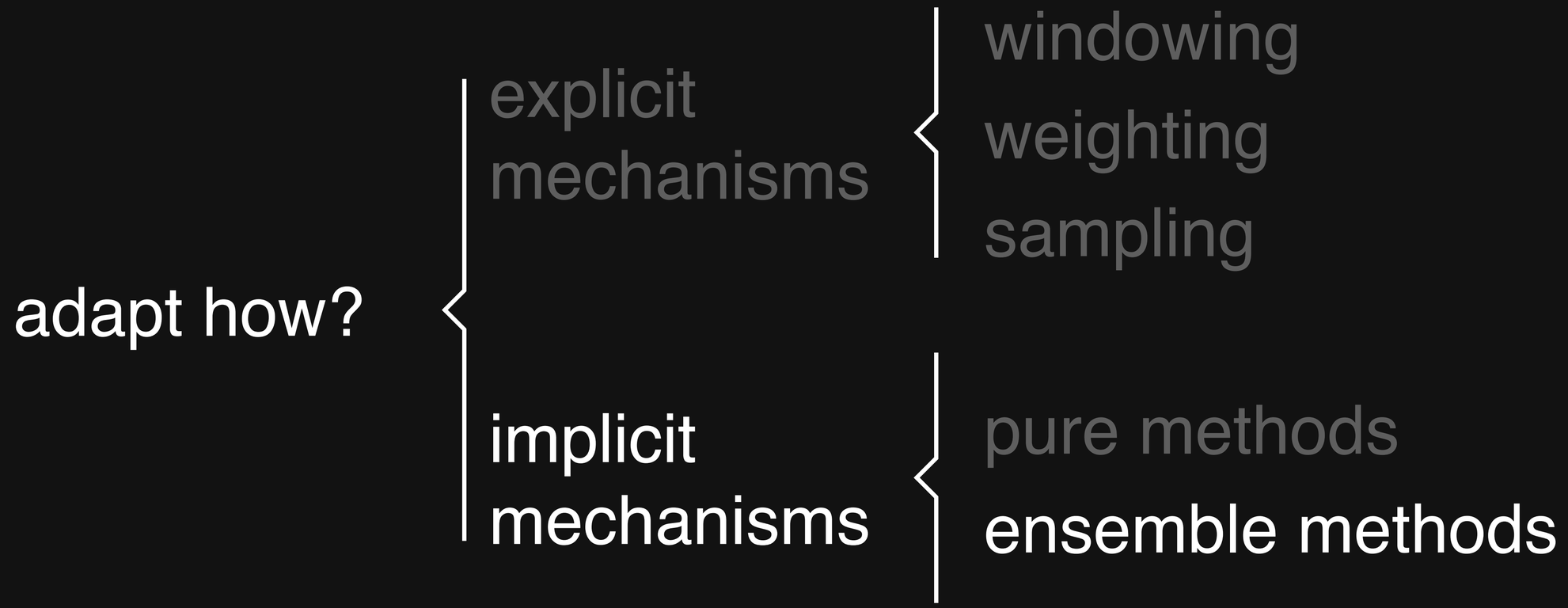
bias: $f(r,t) = e^{-\lambda(t-r)}$

capacity: $N = \frac{1}{\lambda}$

overwrite / exchange

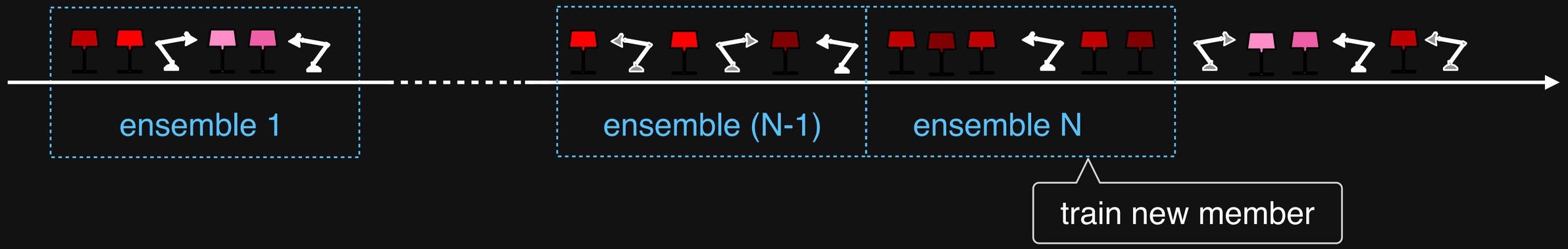
randomly w/ Prob{ $\%_{\text{full}}$ }

or append


adapt how?
explicit mechanisms
windowing
weighting
sampling
implicit mechanisms
pure methods
ensemble methods

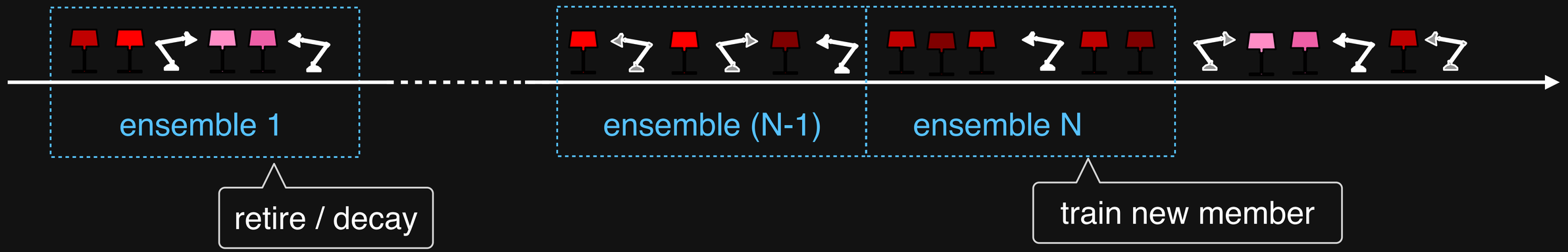
# implicit mechanisms for adaptation

## Ensemble Based Adaptation



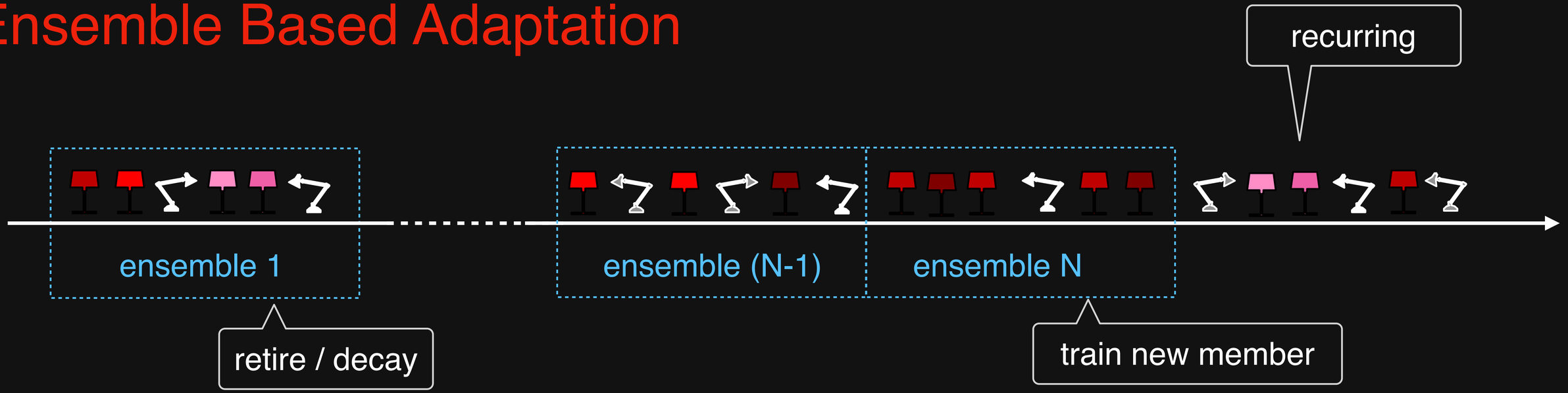
# implicit mechanisms for adaptation

## Ensemble Based Adaptation



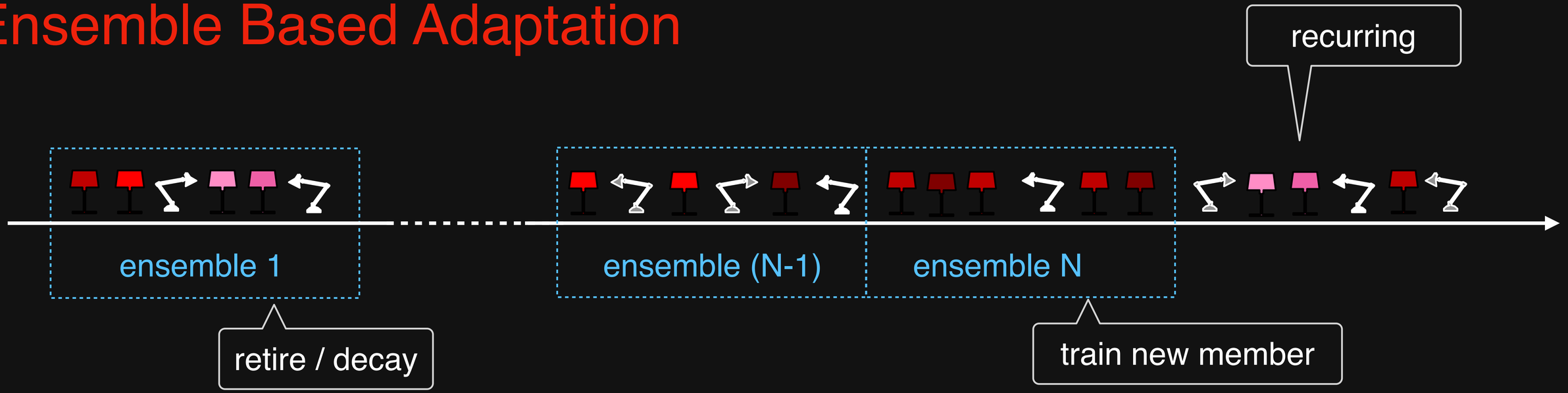
# implicit mechanisms for adaptation

## Ensemble Based Adaptation



# implicit mechanisms for adaptation

## Ensemble Based Adaptation




- Online NonStationary boosting [ONSboost]
- NonStationary Random Forests [NSRF]
- Dynamic Weighted Majority [DWM]
- Learn++ for NonStationary Environments [Learn++.NSE]

Lightbend

# which method?

| Method | Efficiency | Pros | Cons | Notes |
|---|---|---|---|---|
| DDM/EDDM | O(1) | no data stored | label cost<br>false alarms | sampling necessary in case of fast data,<br>microservices architecture ideal |
| LFR | O(1) | class imbalance OK | label cost | |
| ADWIN | O(log W) | better change localization | label cost | |
| JIT | O(log W) | no labels required | only for abrupt changes | best localization |

Lightbend

# which method?

| Method | Efficiency | Pros | Cons | Notes |
|---|---|---|---|---|
| ECSMiner / GC3 | $O(W^2 / k)$<br>$O(G \log C)$ | emerging concepts | *clusterable* drift only | use if emerging concepts expected |
| HDDDM | O(DB) | no labels | not for population drift or class imbalance | better when combined with PCA |
| A-distance | O(log W) | no labels | less false positives compared to HDDDM | good choice for unsupervised |
| Margin / MD3 | Learning, detection, adaptation bundled | reduced false alarms | must use feature bagged ensembles | best choice but must commit to using the specific machine learning algorithms |
| Ensemble methods | | recurring concepts | large batches | |

# References

https://gist.github.com/emrev12/0d75dc2d6c3e80012d10a82712b8ced0

# thank you

emre.velipasaoglu@  Lightbend .com