Machine Learned Model Quality Monitoring in Fast Data and Streaming Applications

Emre Velipasaoglu

Lightbend

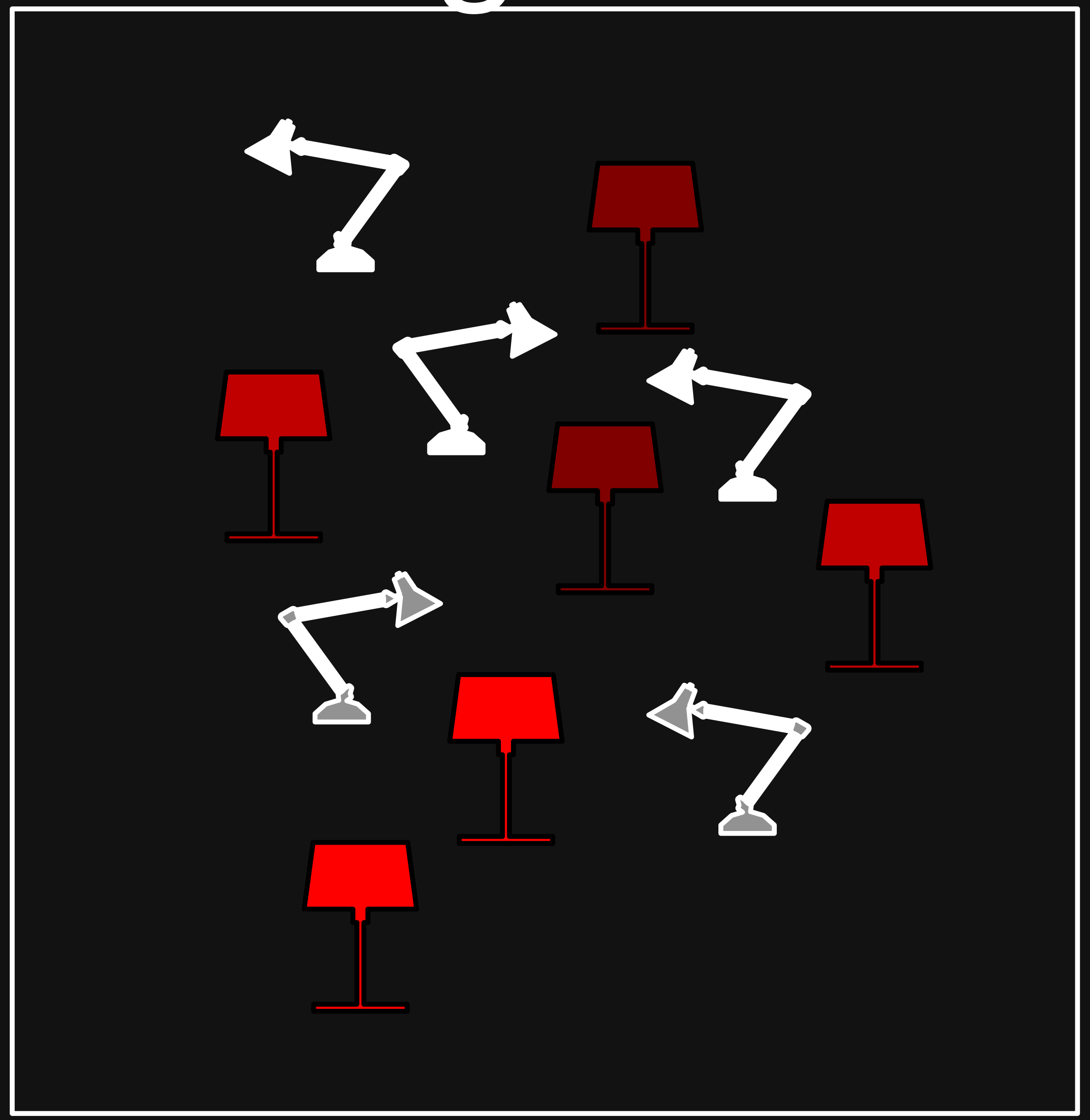
Strata

DATA CONFERENCE

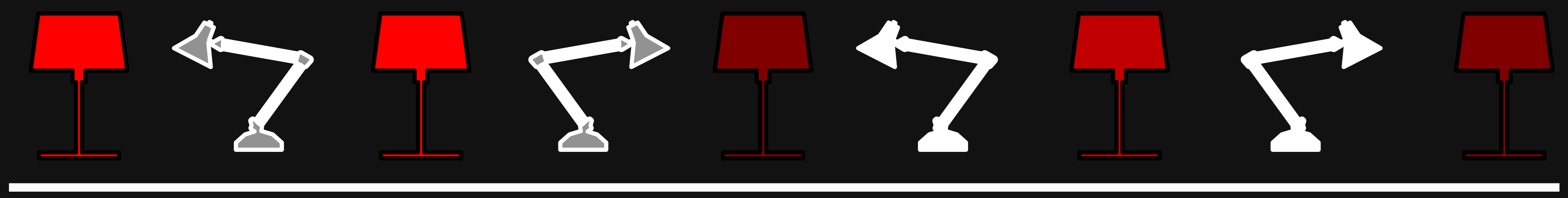
model quality

core problem

training set



$\stackrel{*}{=}$

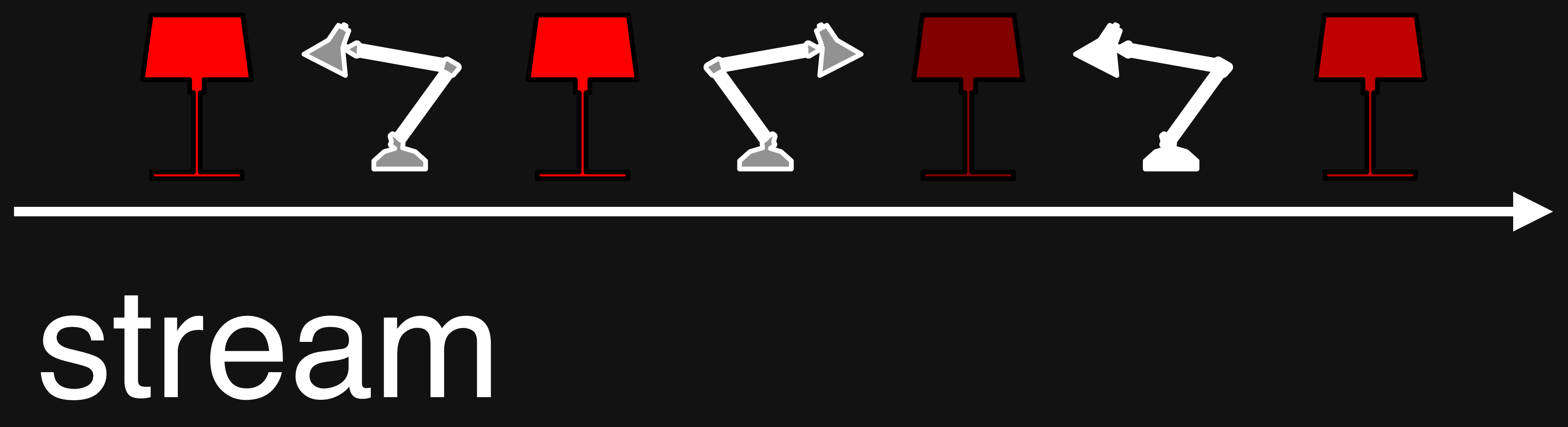


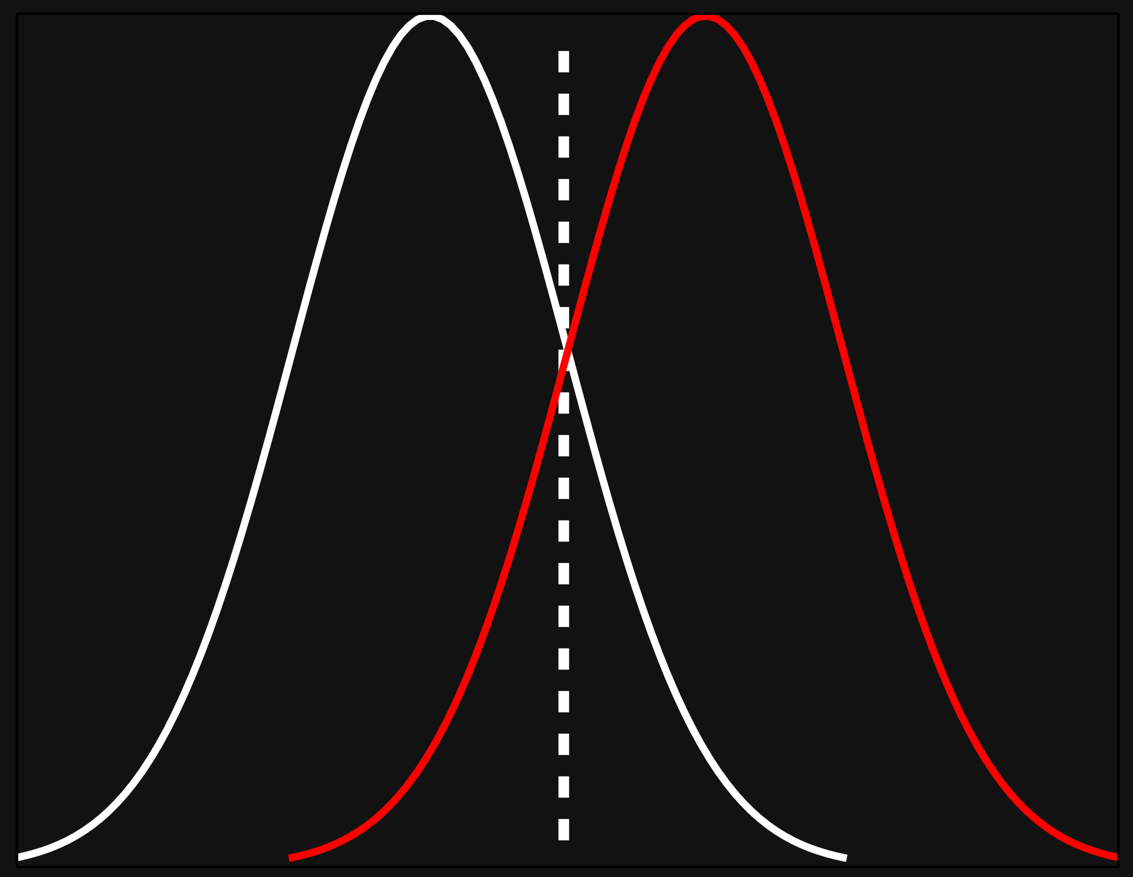
operation

* same data generating distribution

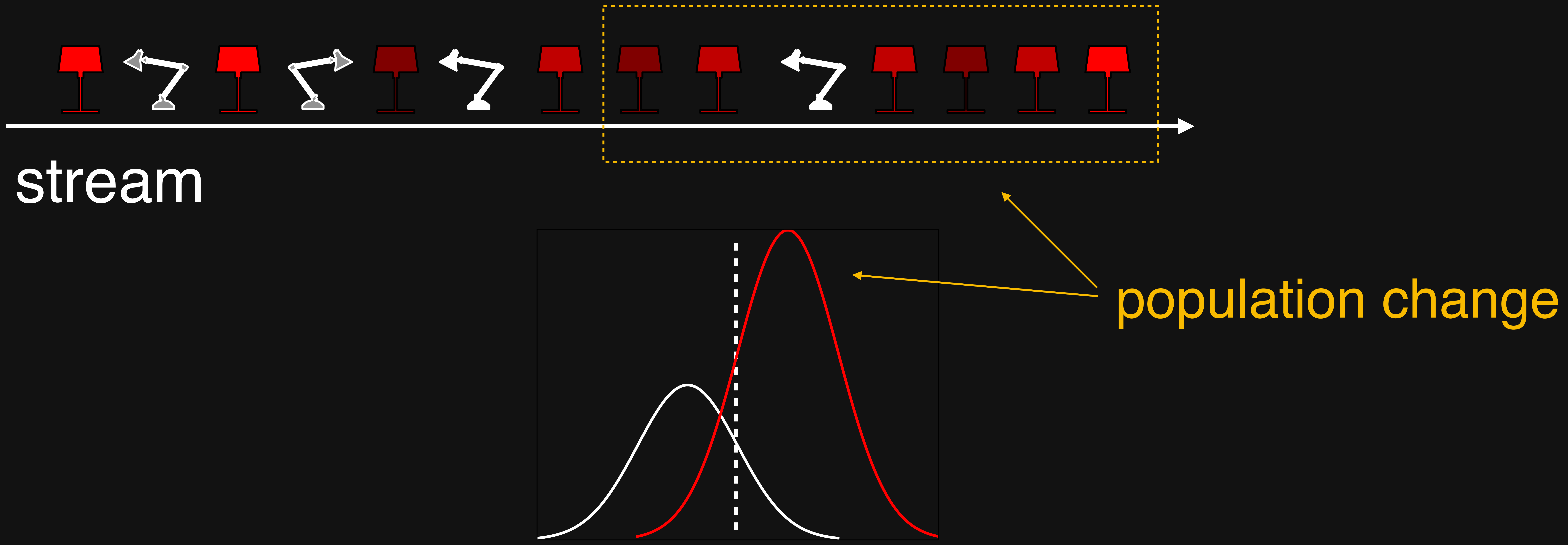
(Some algorithms tolerate violation of this to a certain degree.)

core problem

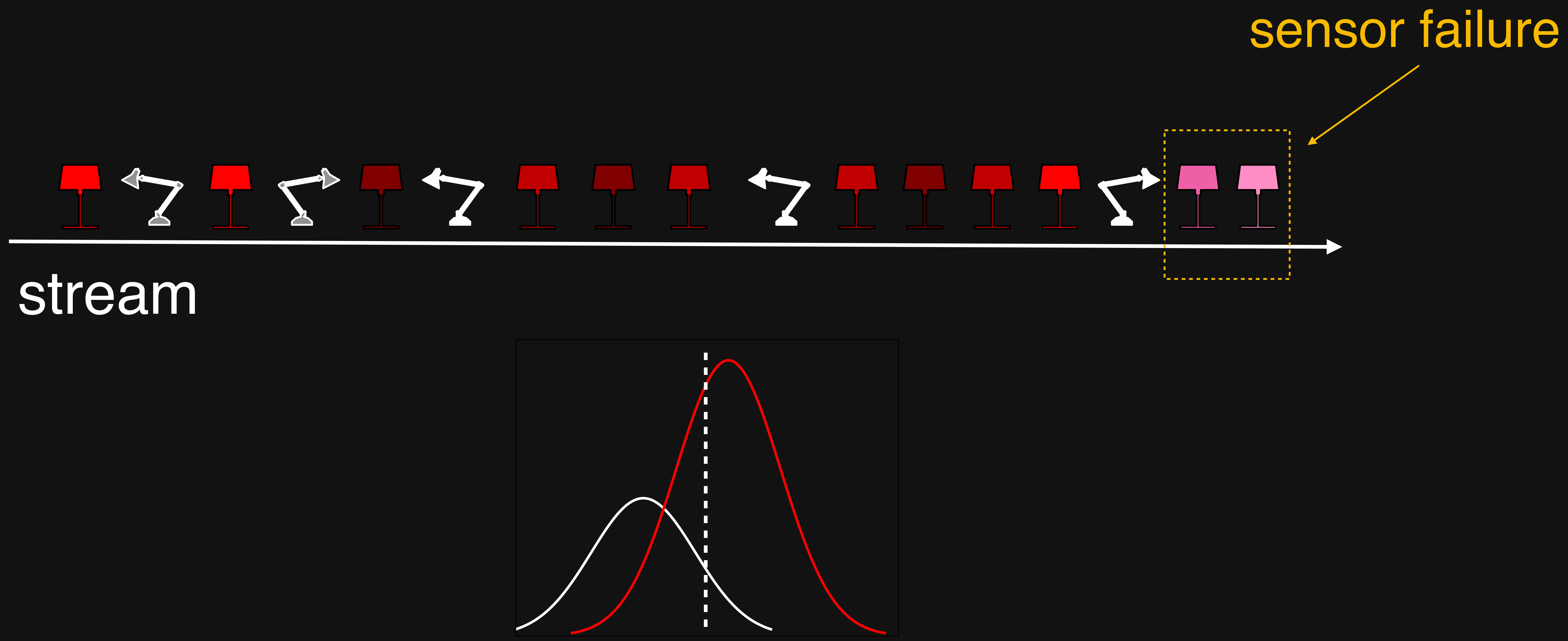




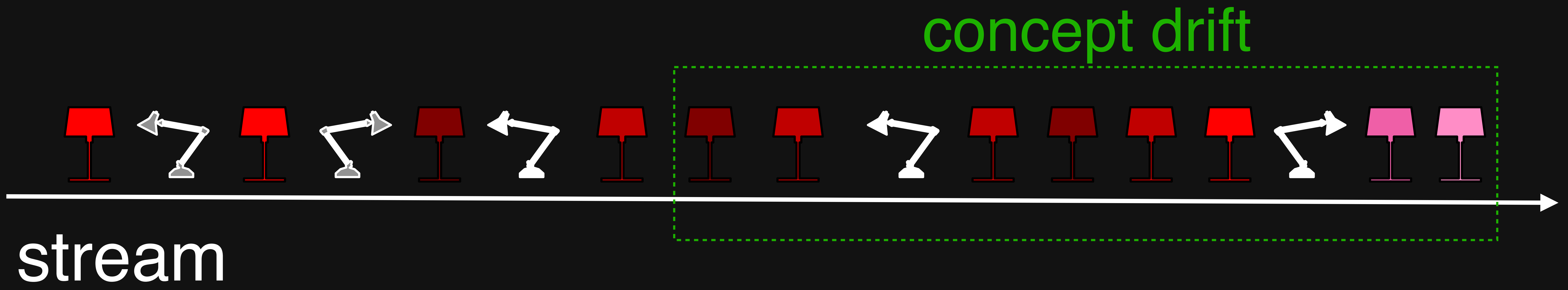
core problem

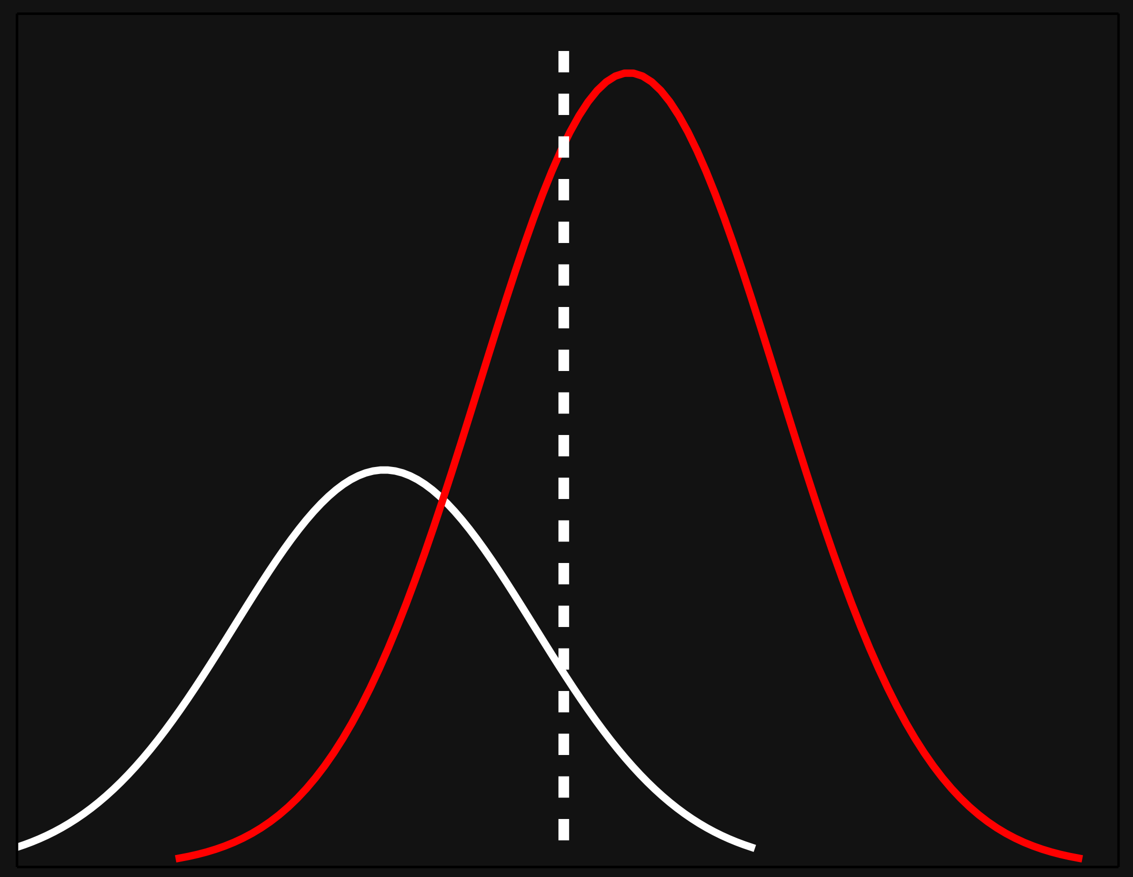


core problem

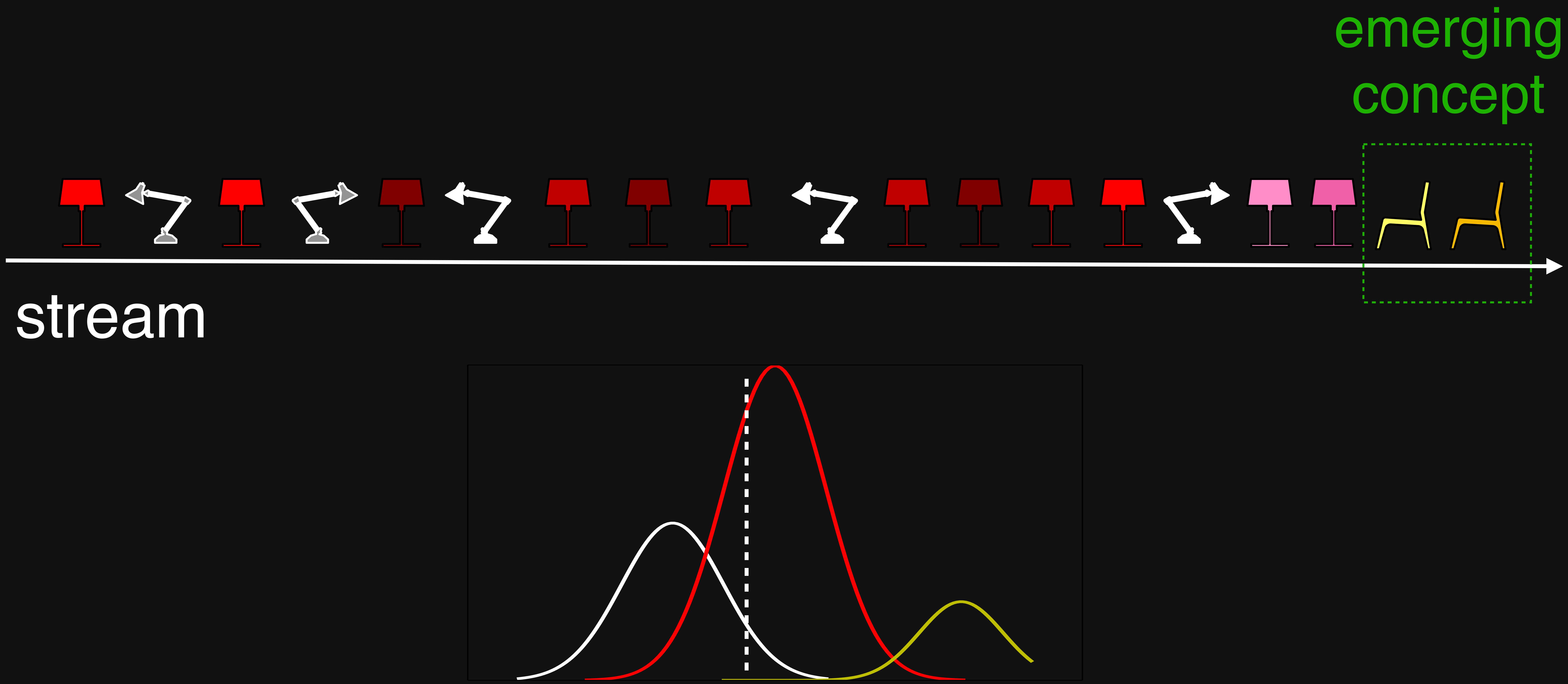


core problem

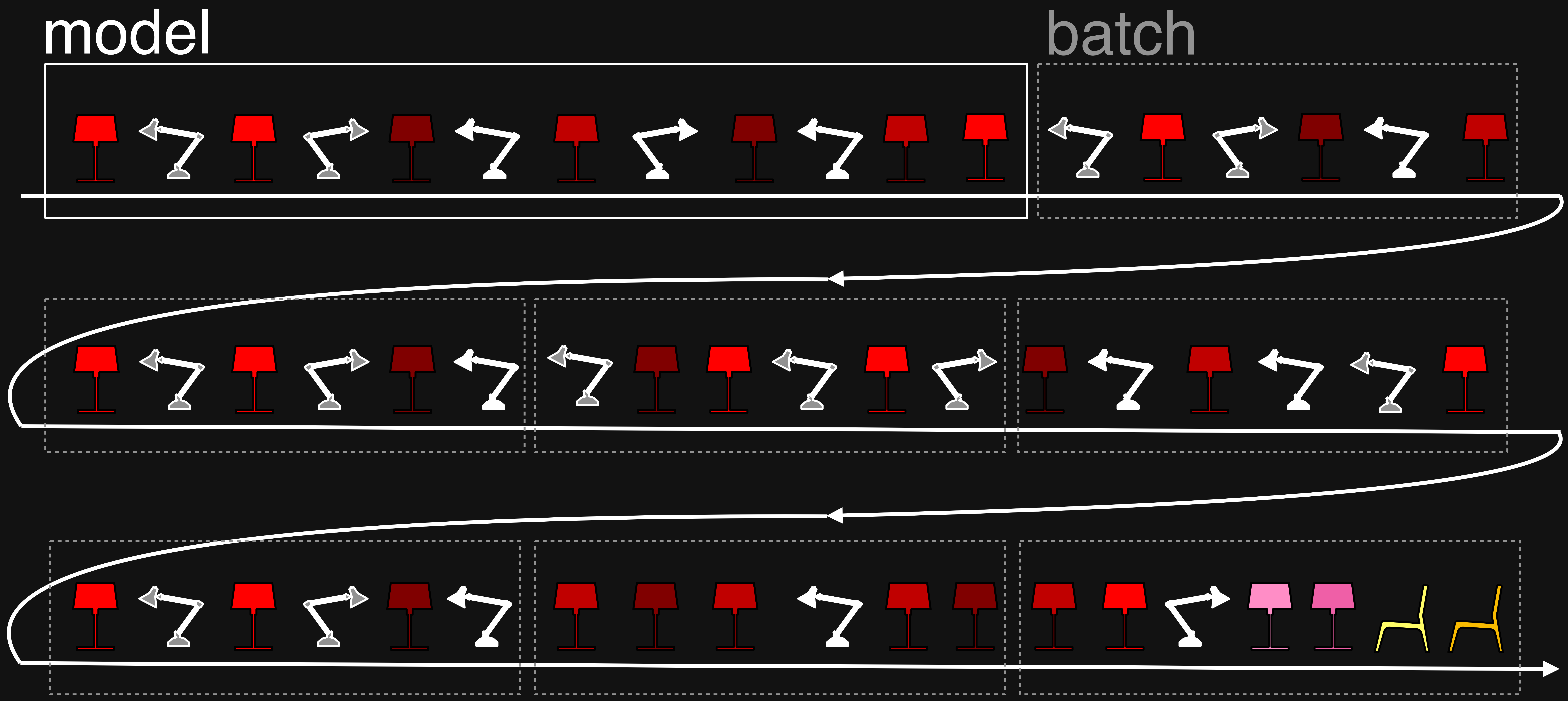




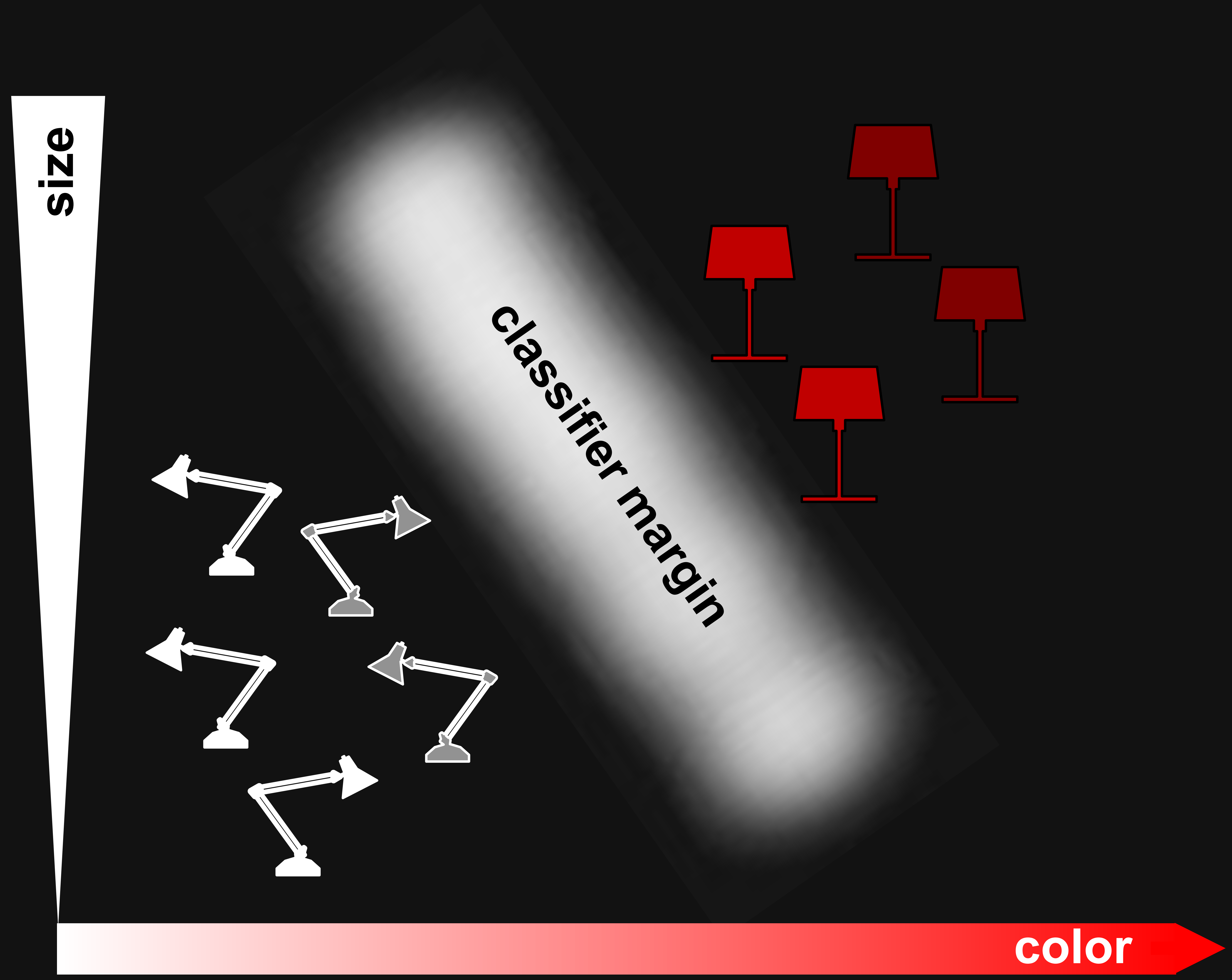
core problem



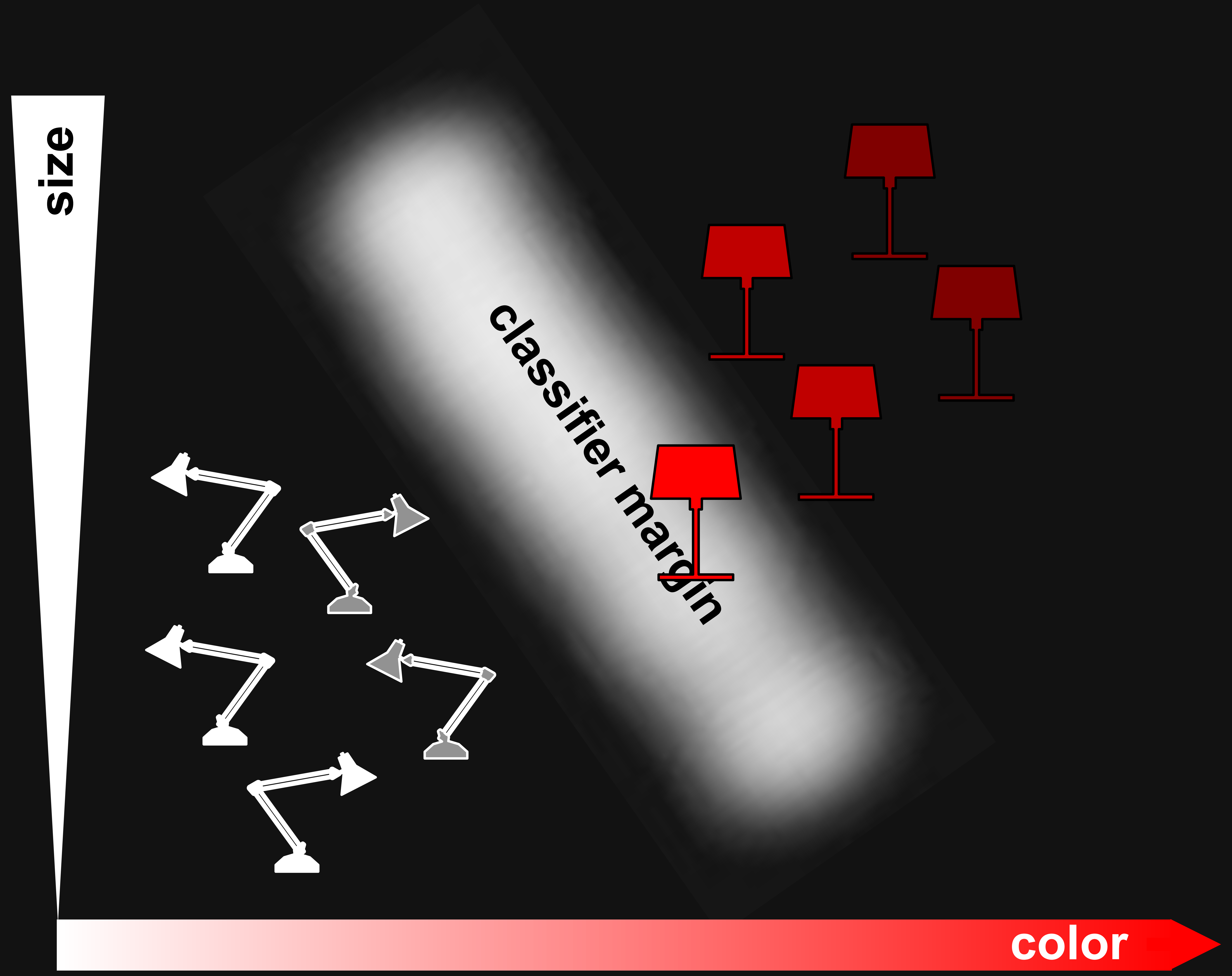
common poor solution



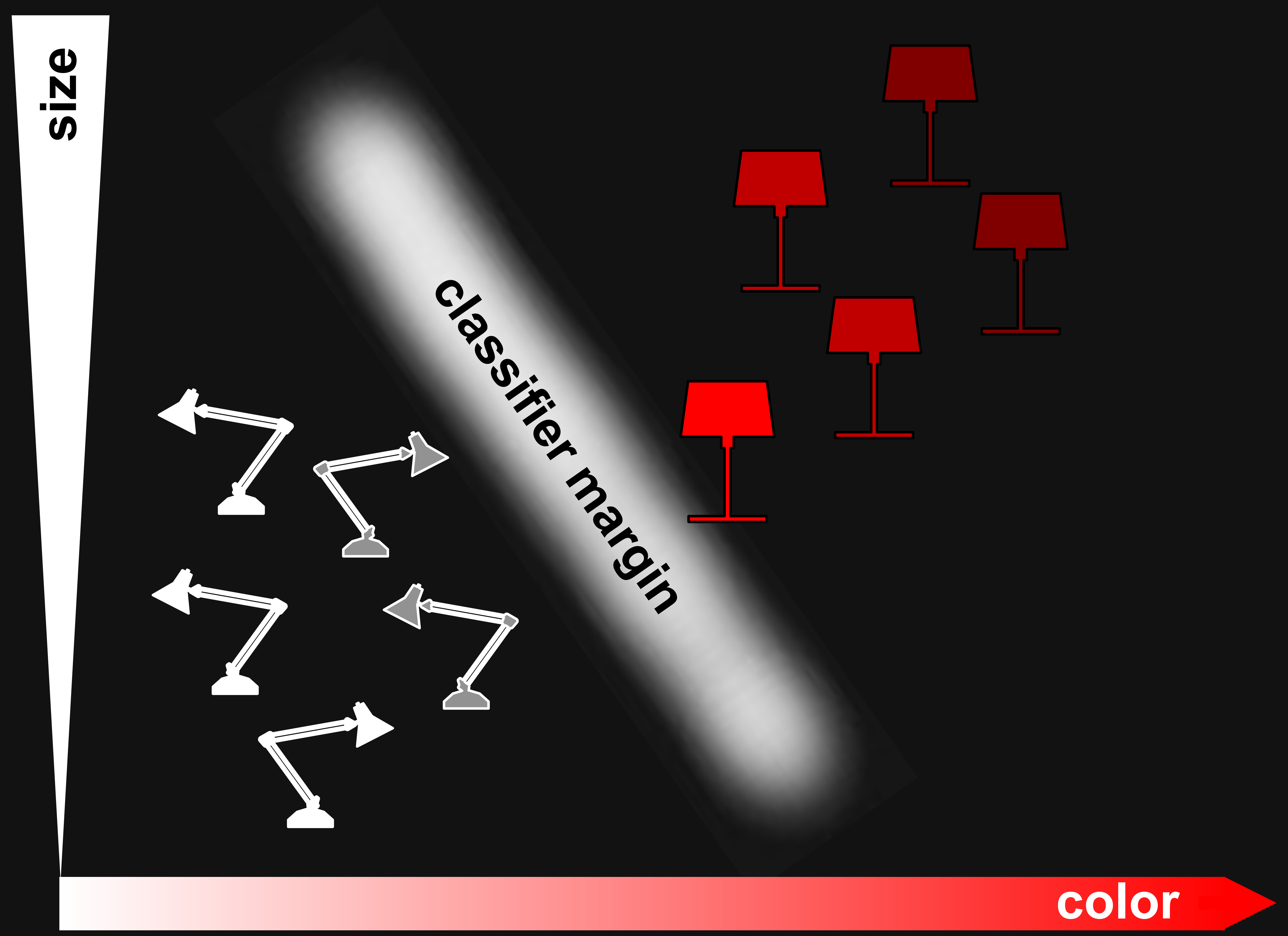
can active learning help?



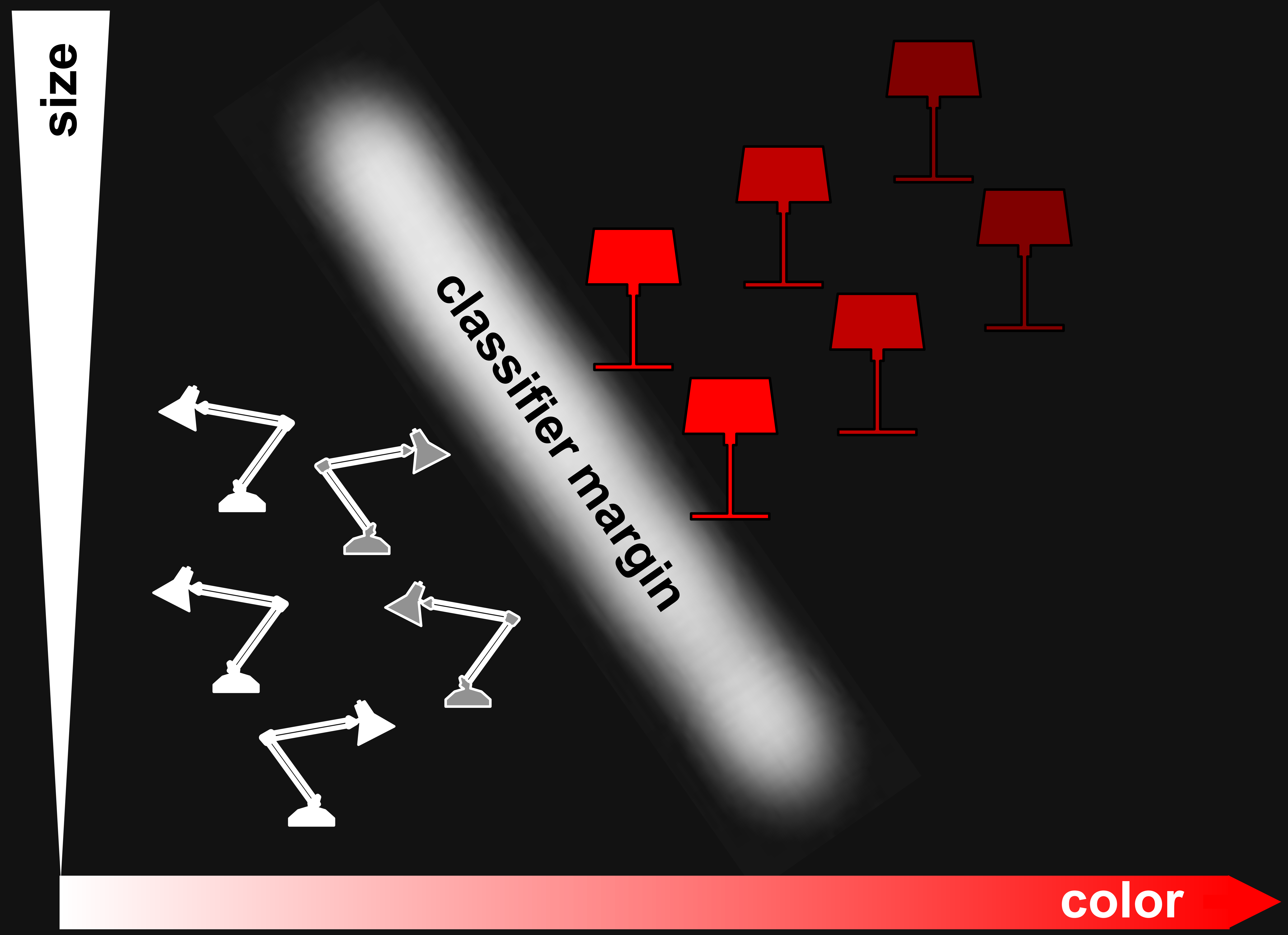
can active learning help?



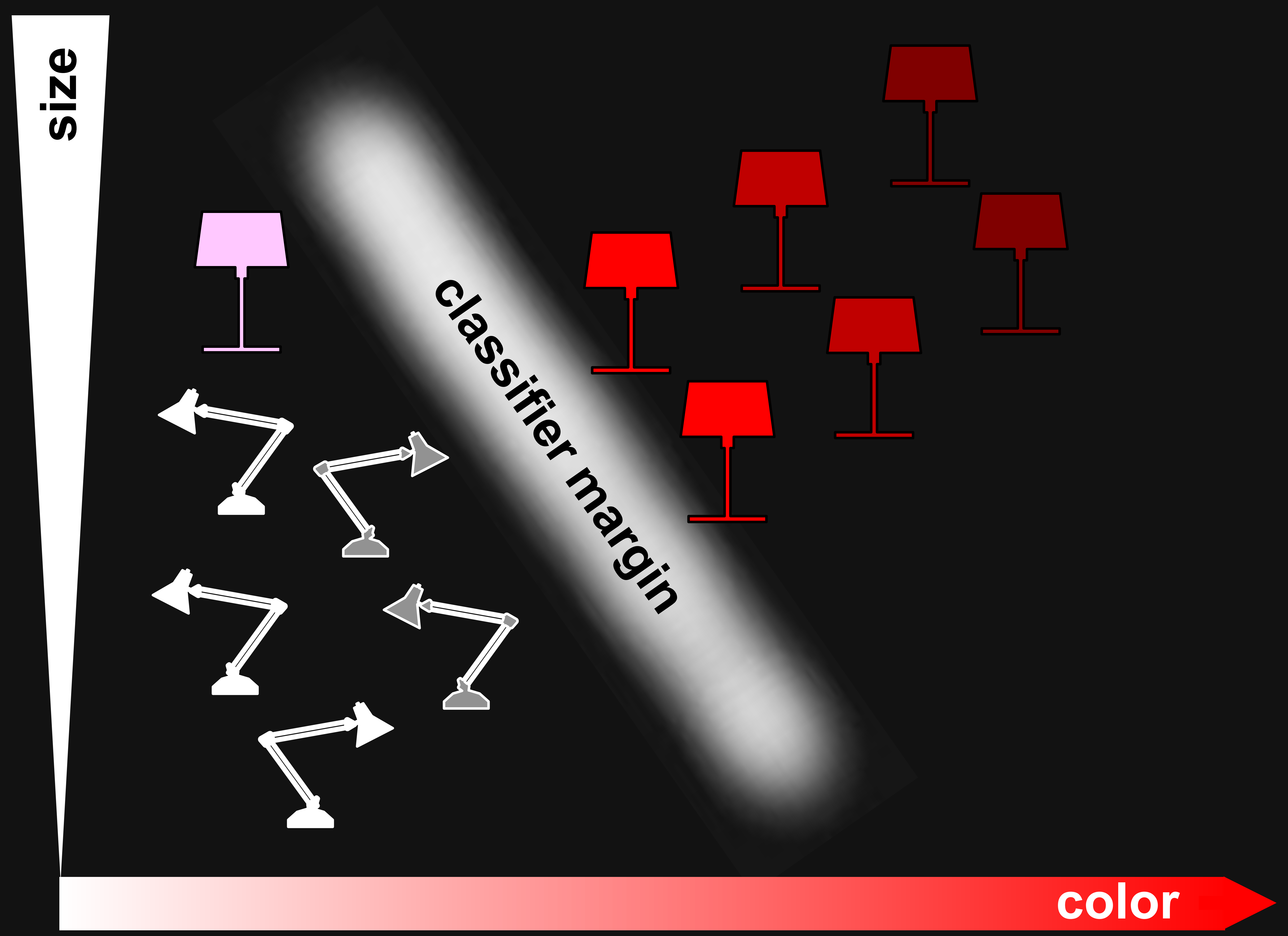
can active learning help?



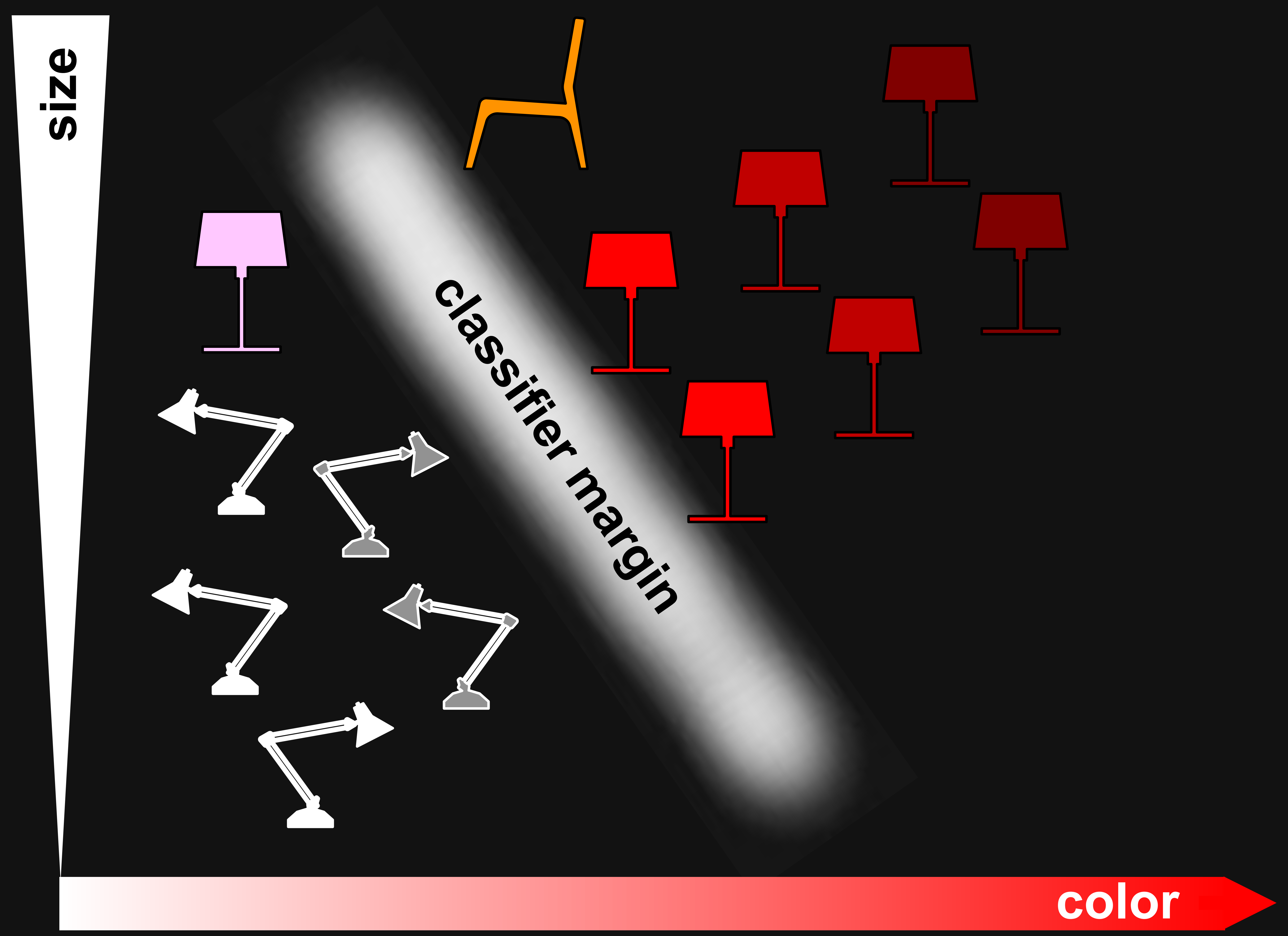
can active learning help?



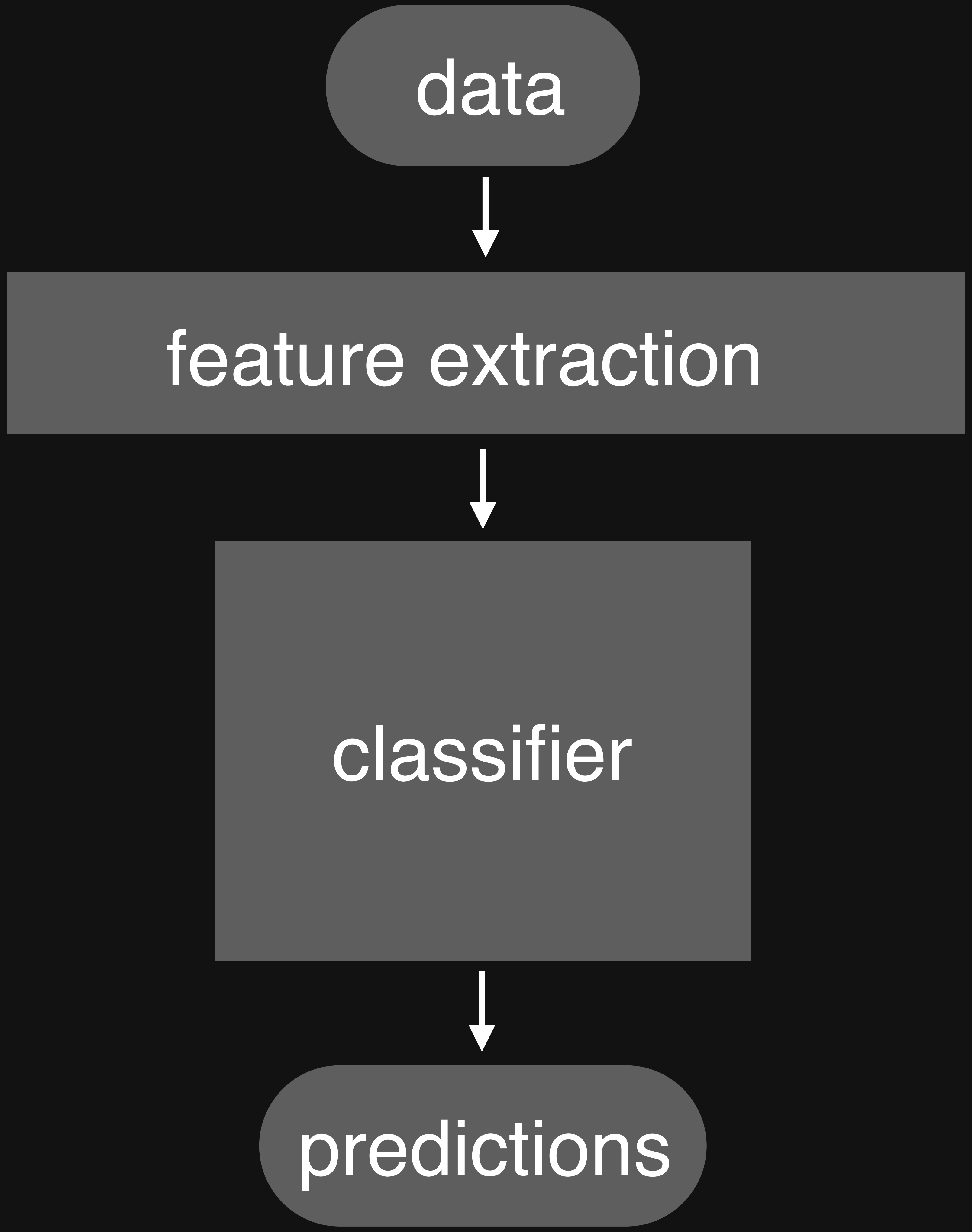
can active learning help?



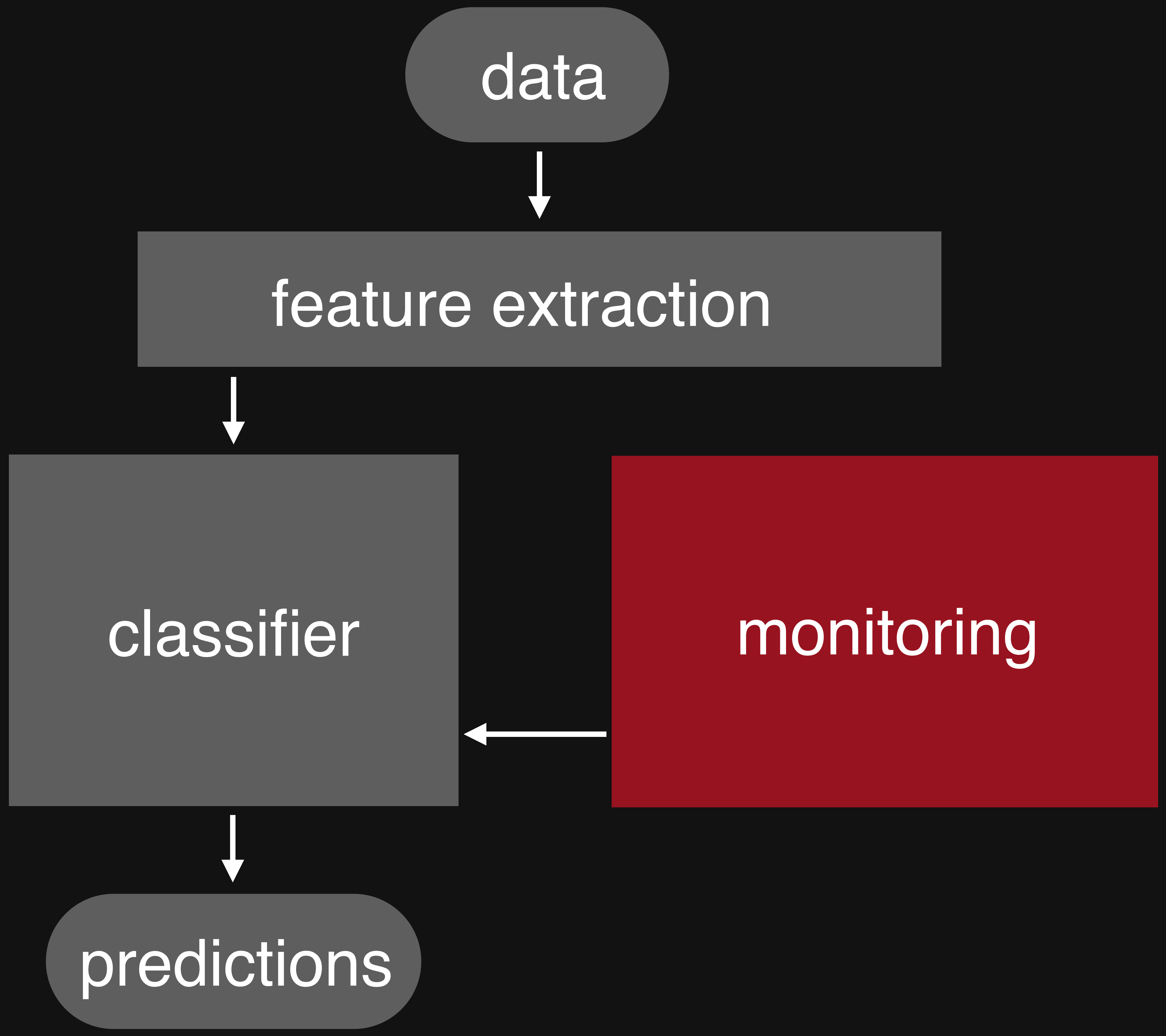
can active learning help?



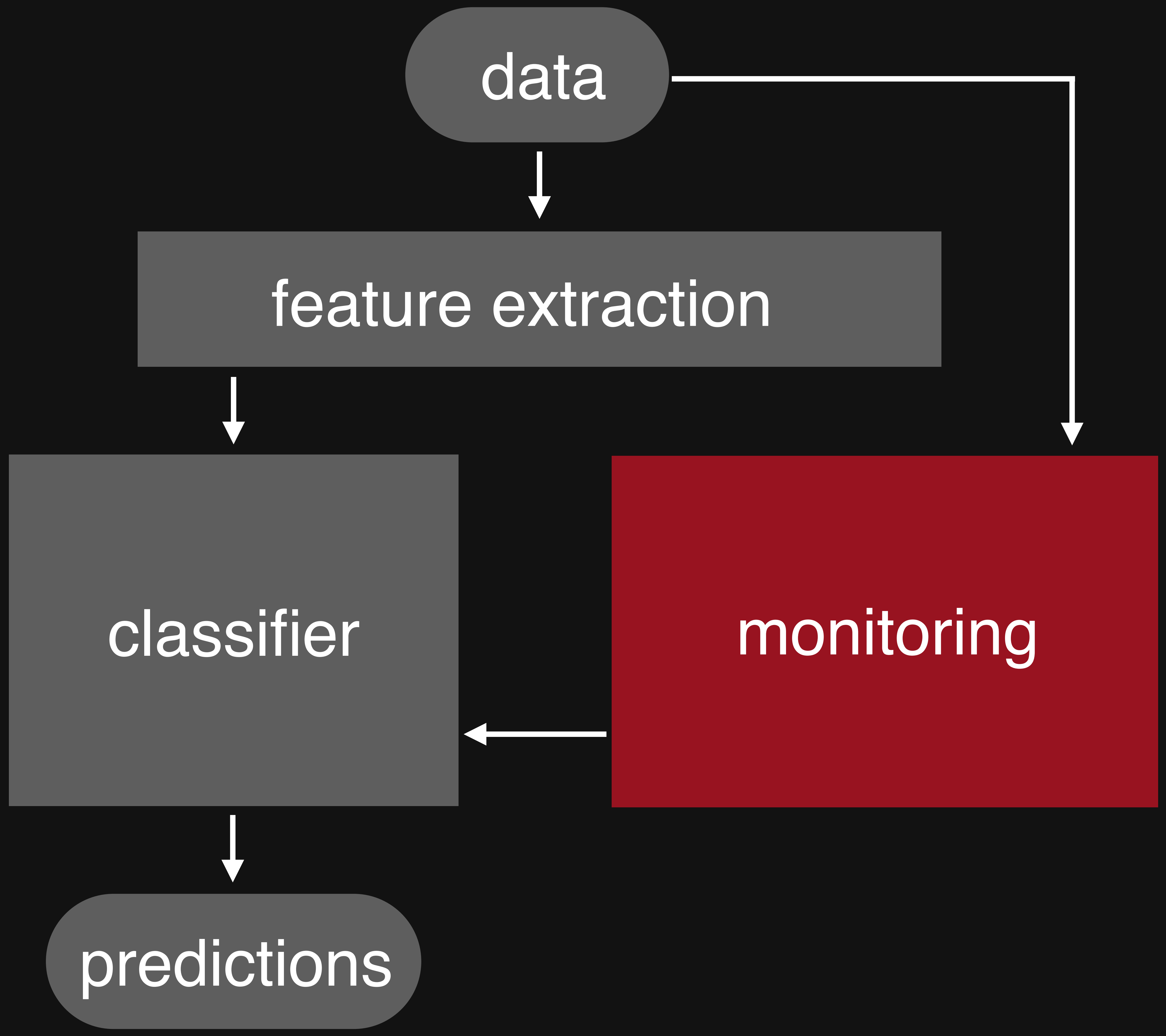
a better solution



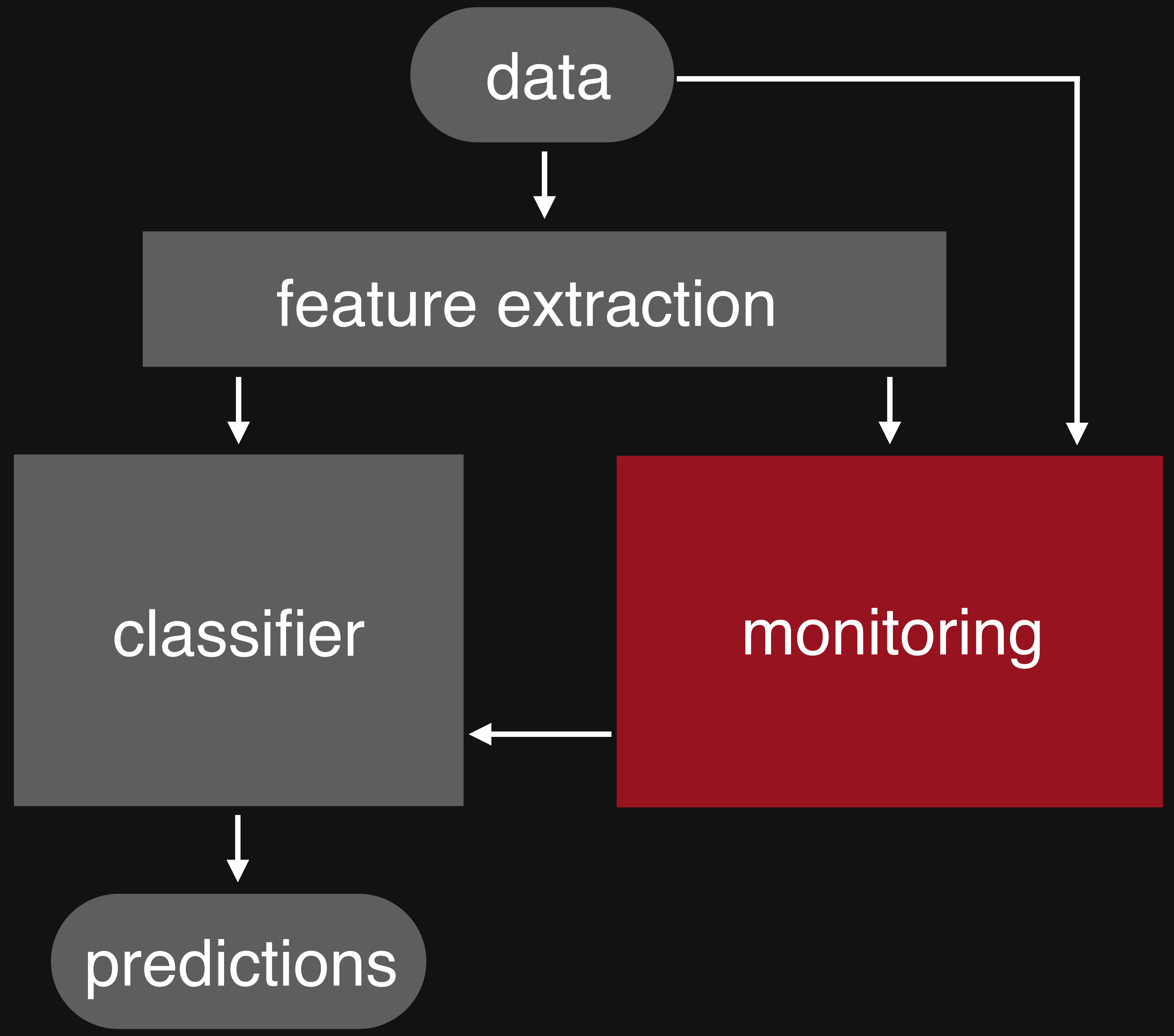
a better solution



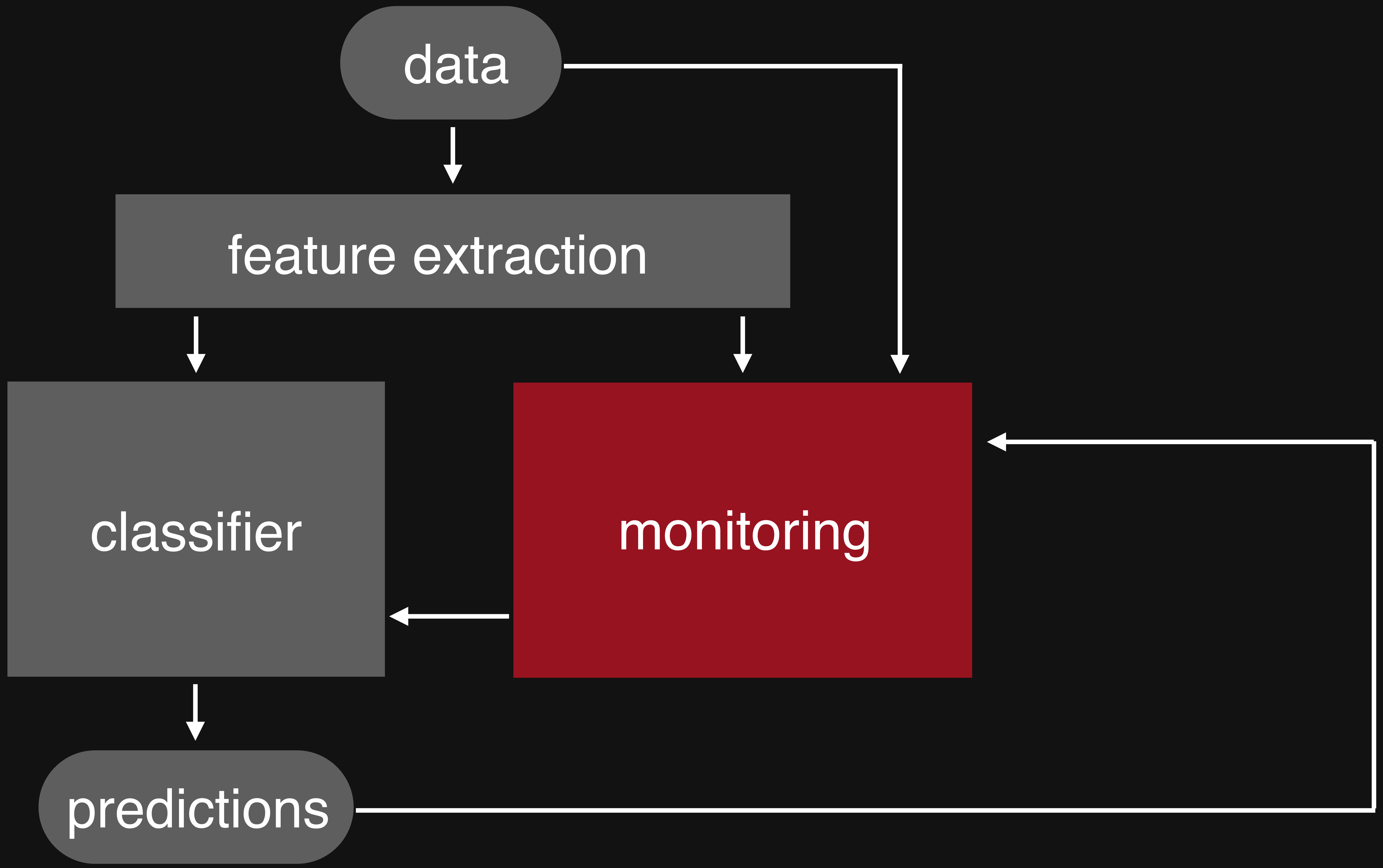
a better solution



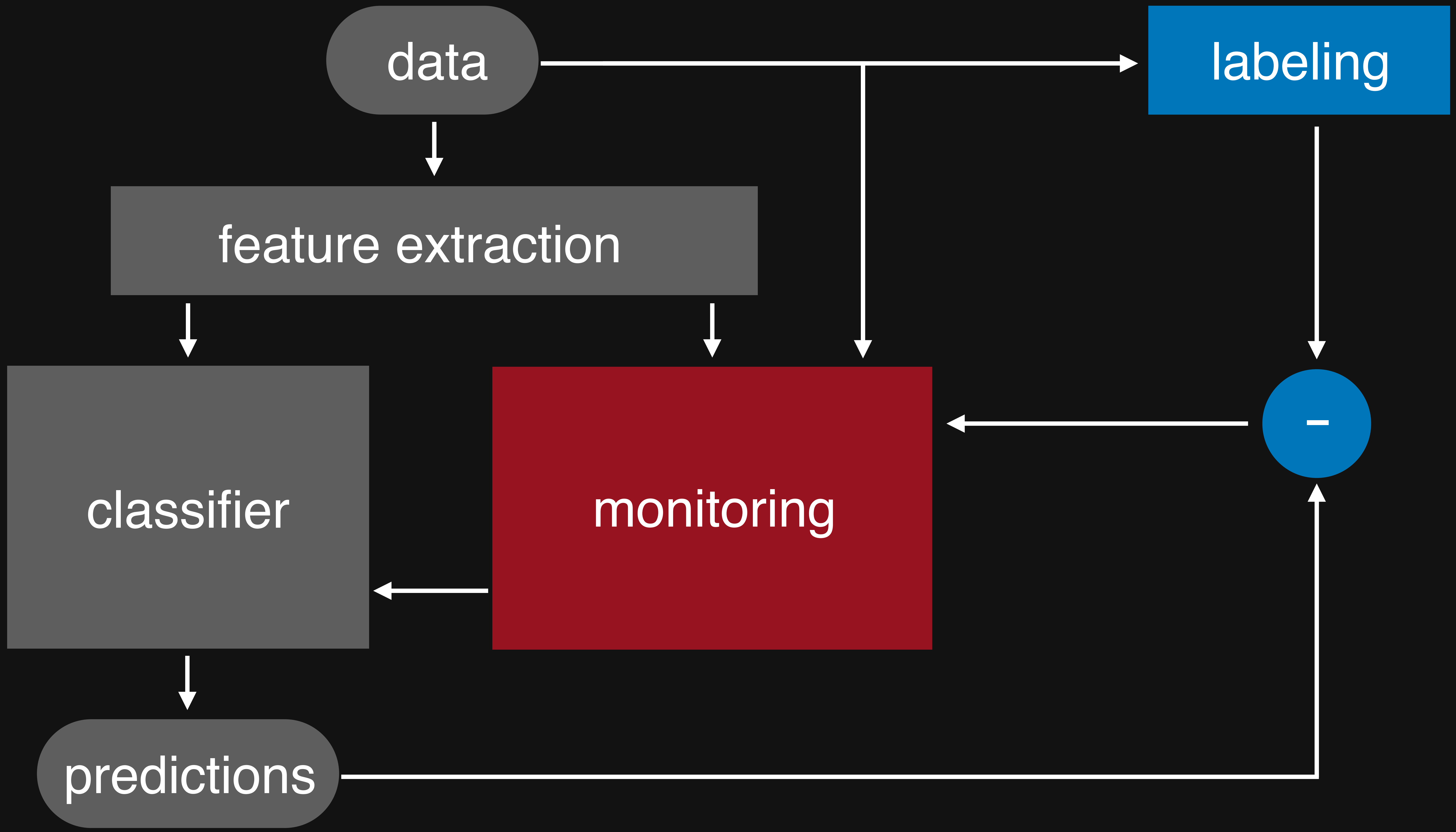
a better solution



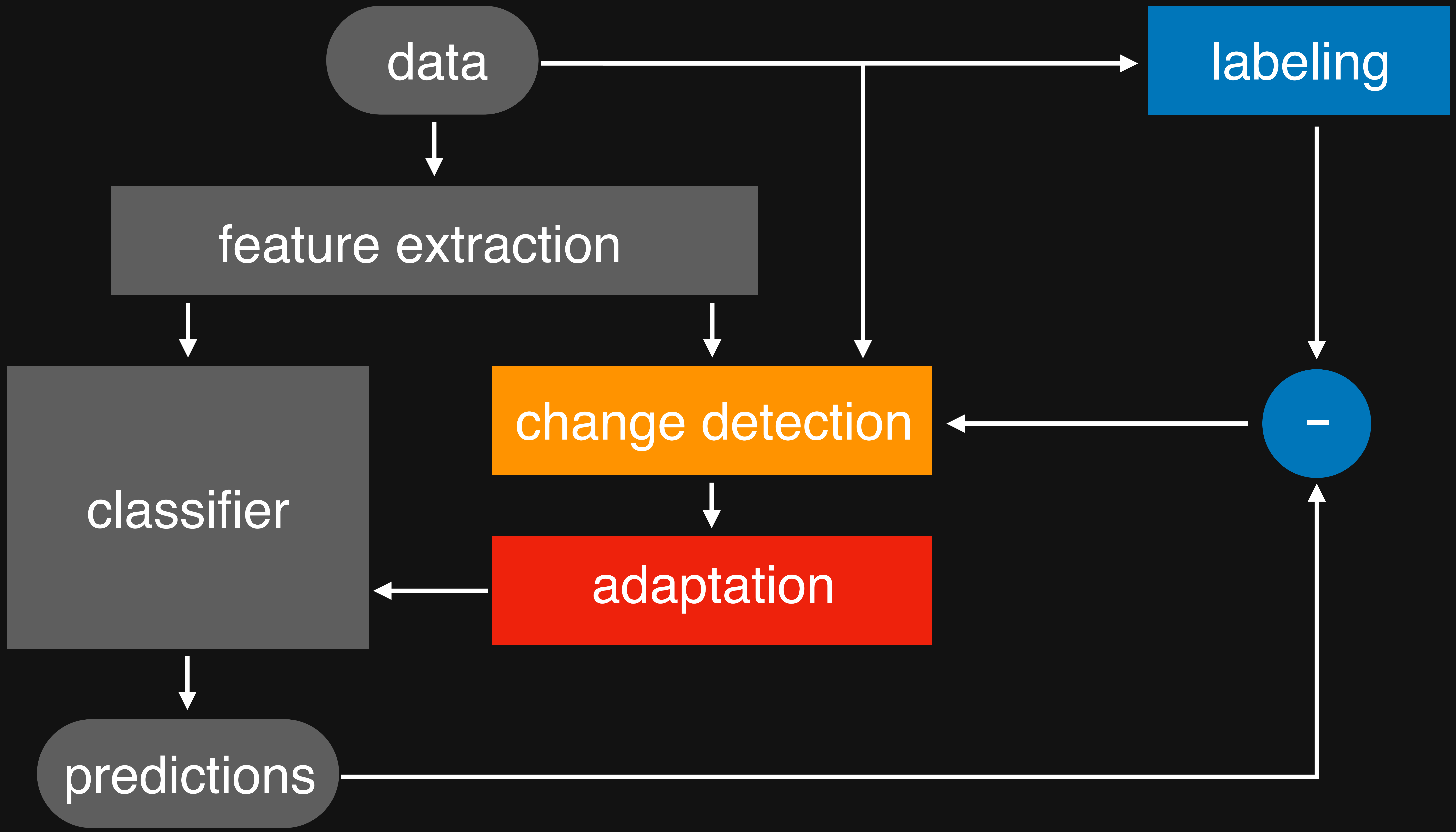
a better solution



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

Predicted \ True	0	1
0	TN	FN
1	FP	TP

sequential analysis

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

True / Predicted	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

True / Predicted	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

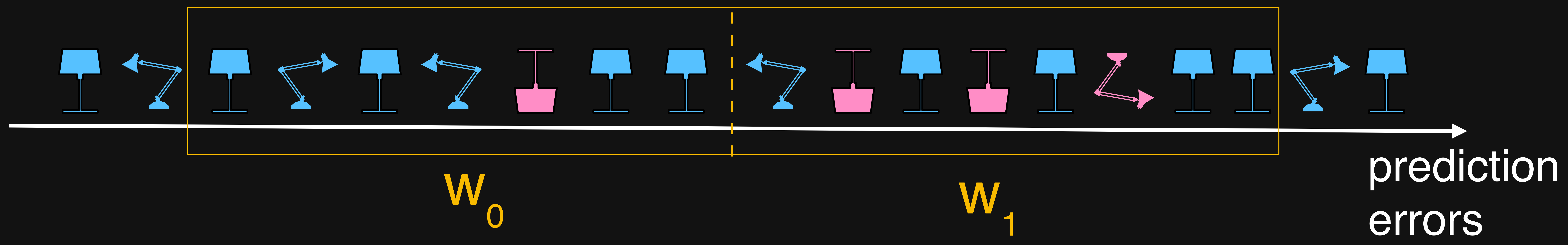
clustering / novelty detection

feature distribution monitoring

model-dependent monitoring

error distribution monitoring

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



  - Drop the last element if any

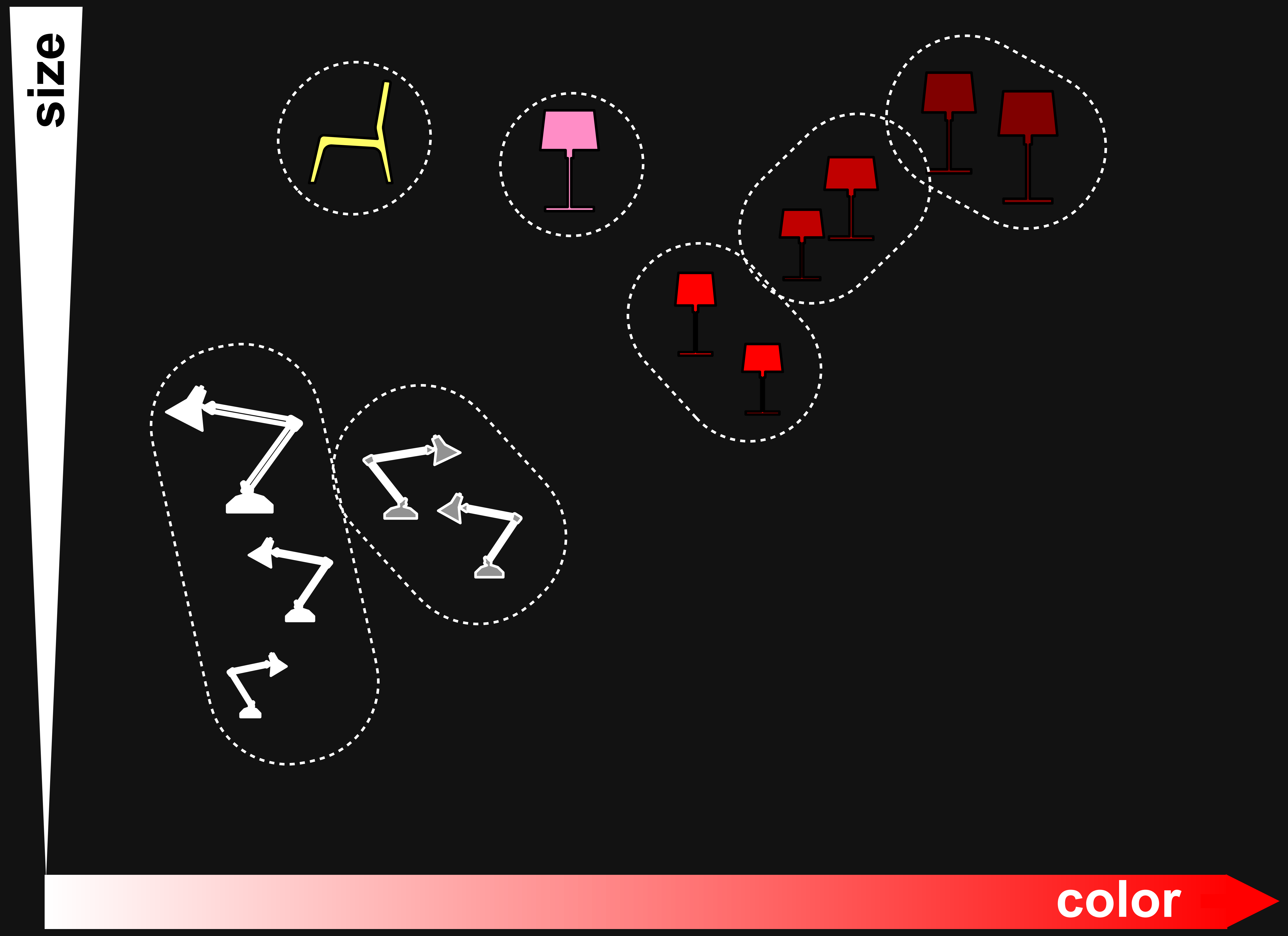
$$|\mu_0 - \mu_1| > \theta_{Hoeffding}$$

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

monitor how?

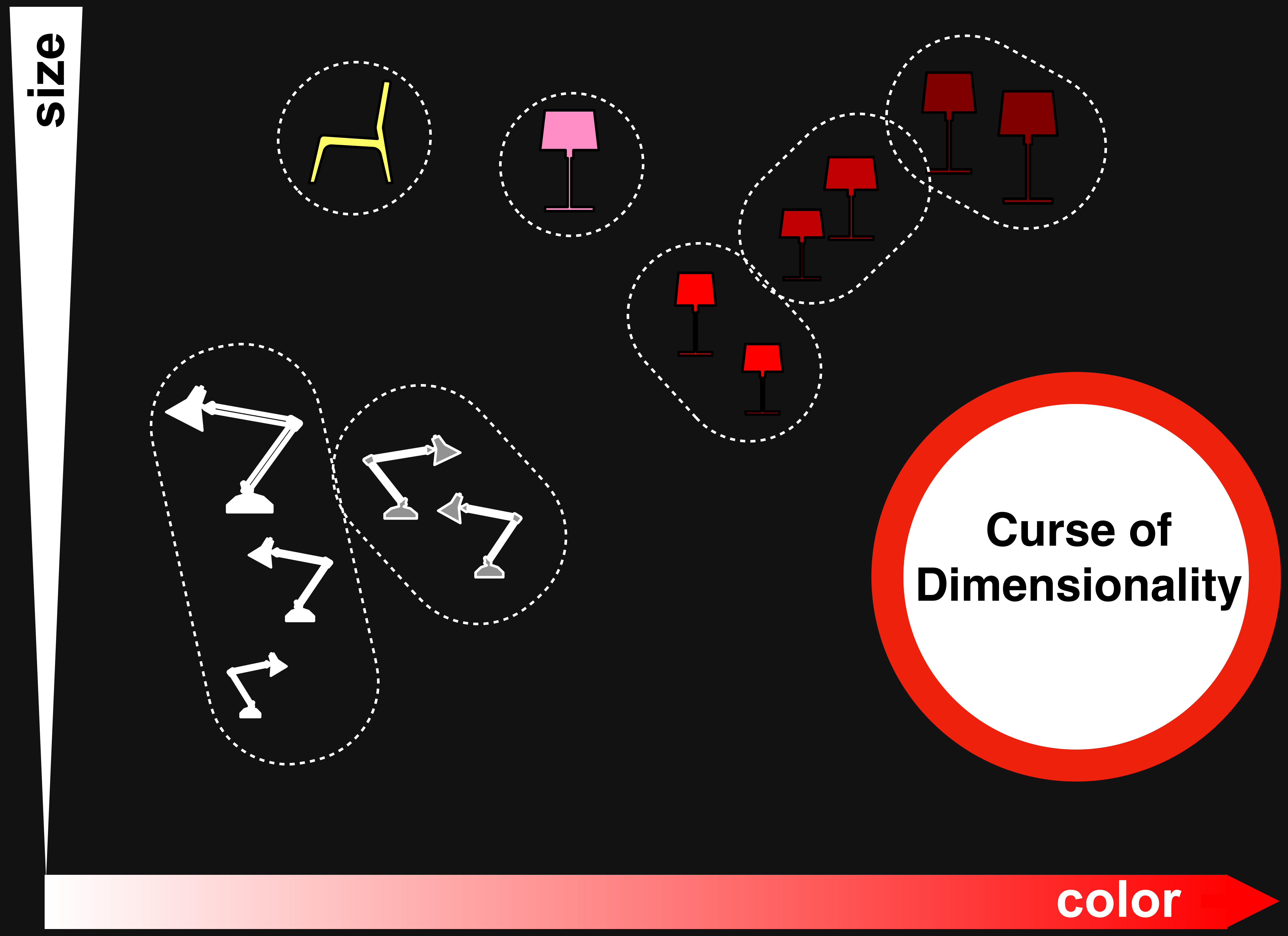
- supervised
  - statistical process control
  - sequential analysis
  - error distribution monitoring
- unsupervised
  - clustering / novelty detection
  - feature distribution monitoring
  - model-dependent monitoring

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

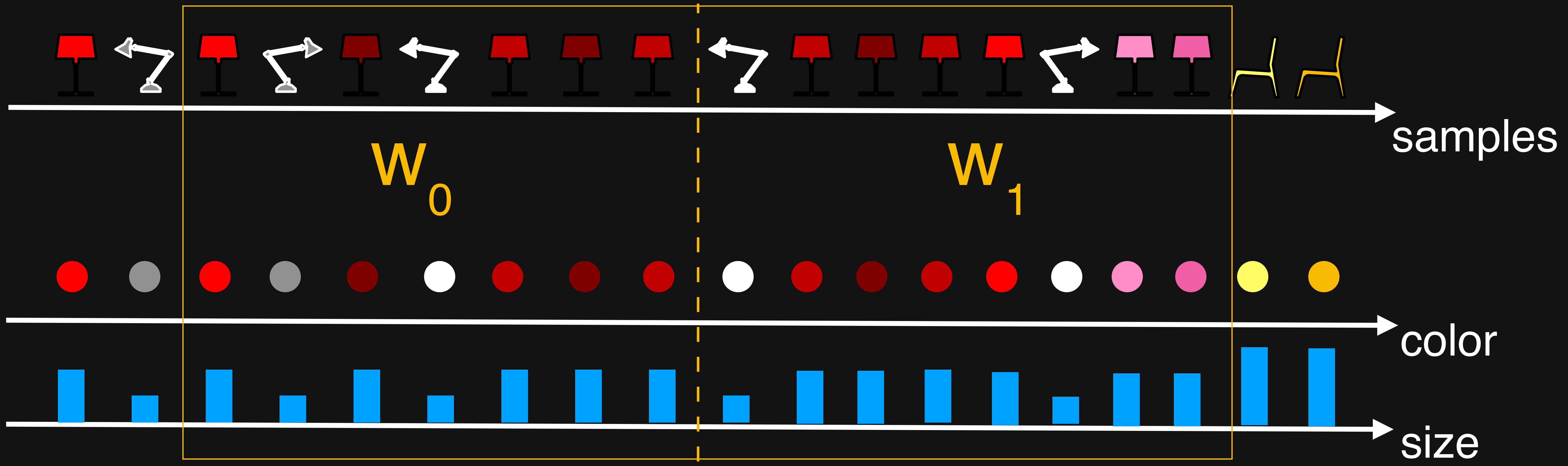


- OLINDDA: K-means, periodically merge unknown to known or flag
- MINAS: micro-clusters, incremental stream clustering
- DETECTNOD: Discrete Cosine Transform to estimate distances efficiently
- Woo-ensemble: Treat outliers as potential emerging class centroids
- ECSMiner: Store and use cluster summary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

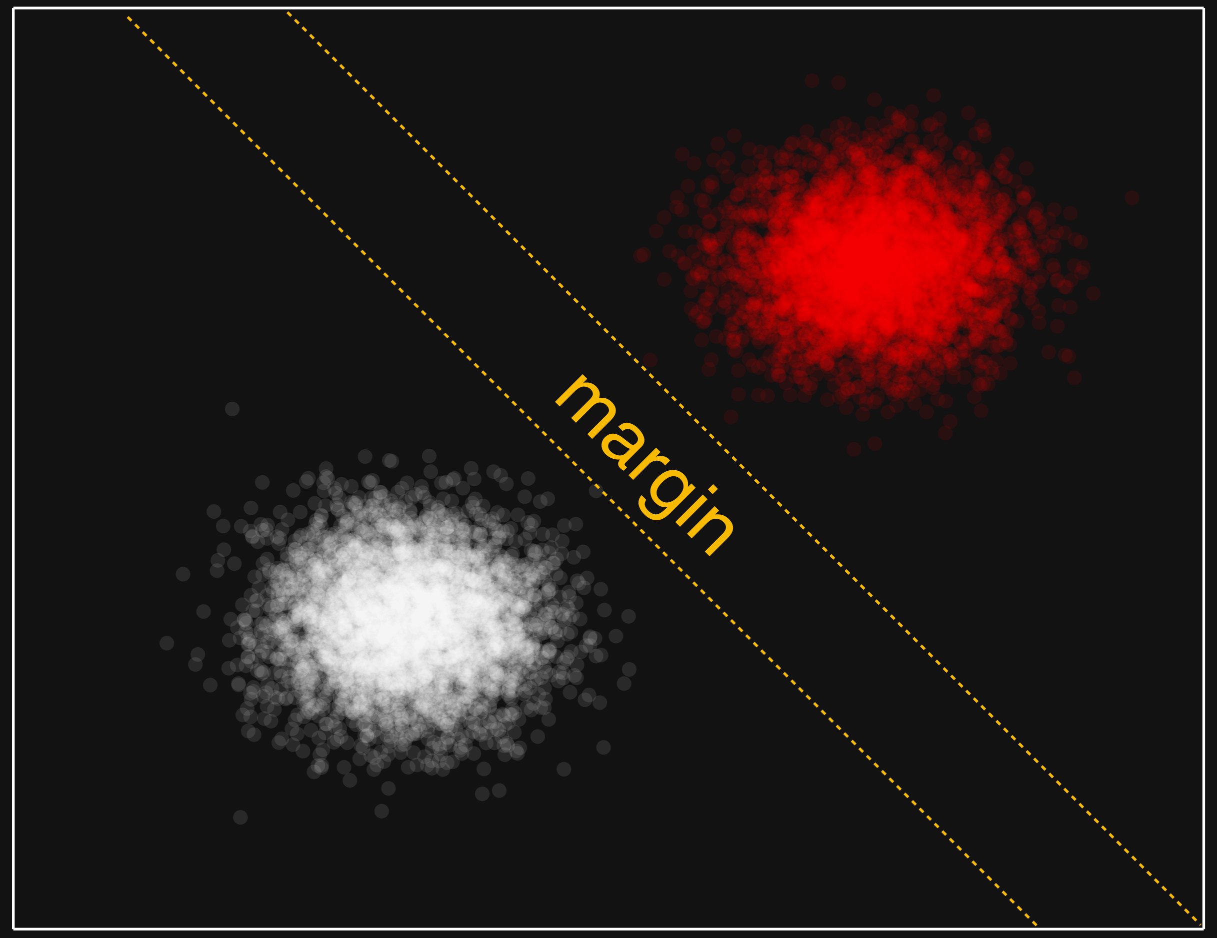
unsupervised
- clustering / novelty detection
- feature distribution monitoring
- model-dependent monitoring

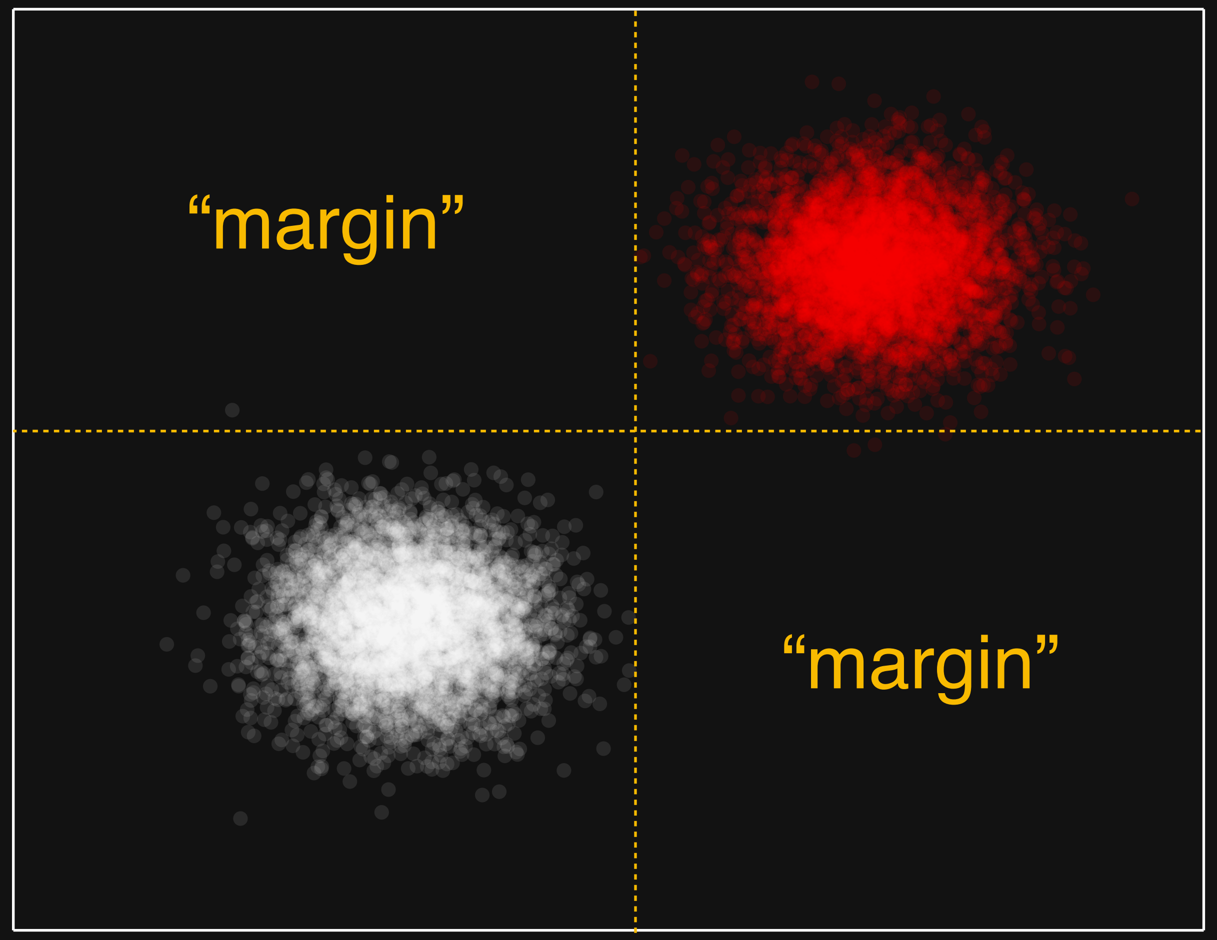
model-dependent monitoring

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

model-dependent monitoring

- Not all changes matter
- Posterior probability estimate
  - Use [A-distance] ~ generalized KS distance
    - designed to be less sensitive to irrelevant changes
- Margin distribution
  - Compare average [Margin]s of 1-norm SVM
  - Generalized margin [MD3]:
    - Embed base classifier in a Random Feature Bagged Ensemble
    - Margin == high disagreement region of the ensemble





adapt how?

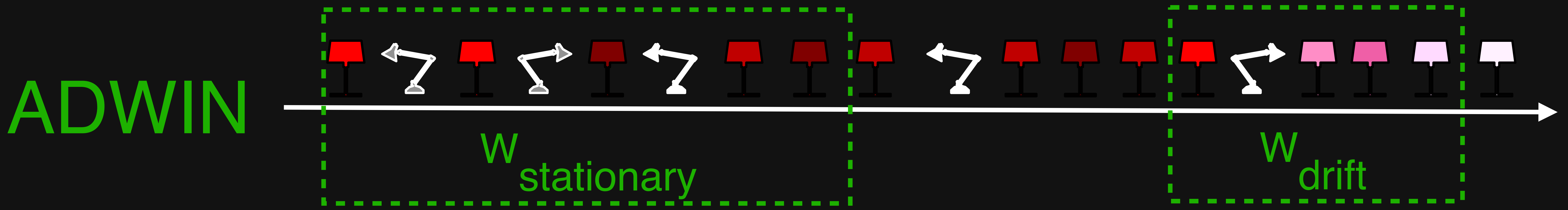
explicit mechanisms

- windowing
- weighting
- sampling

implicit mechanisms

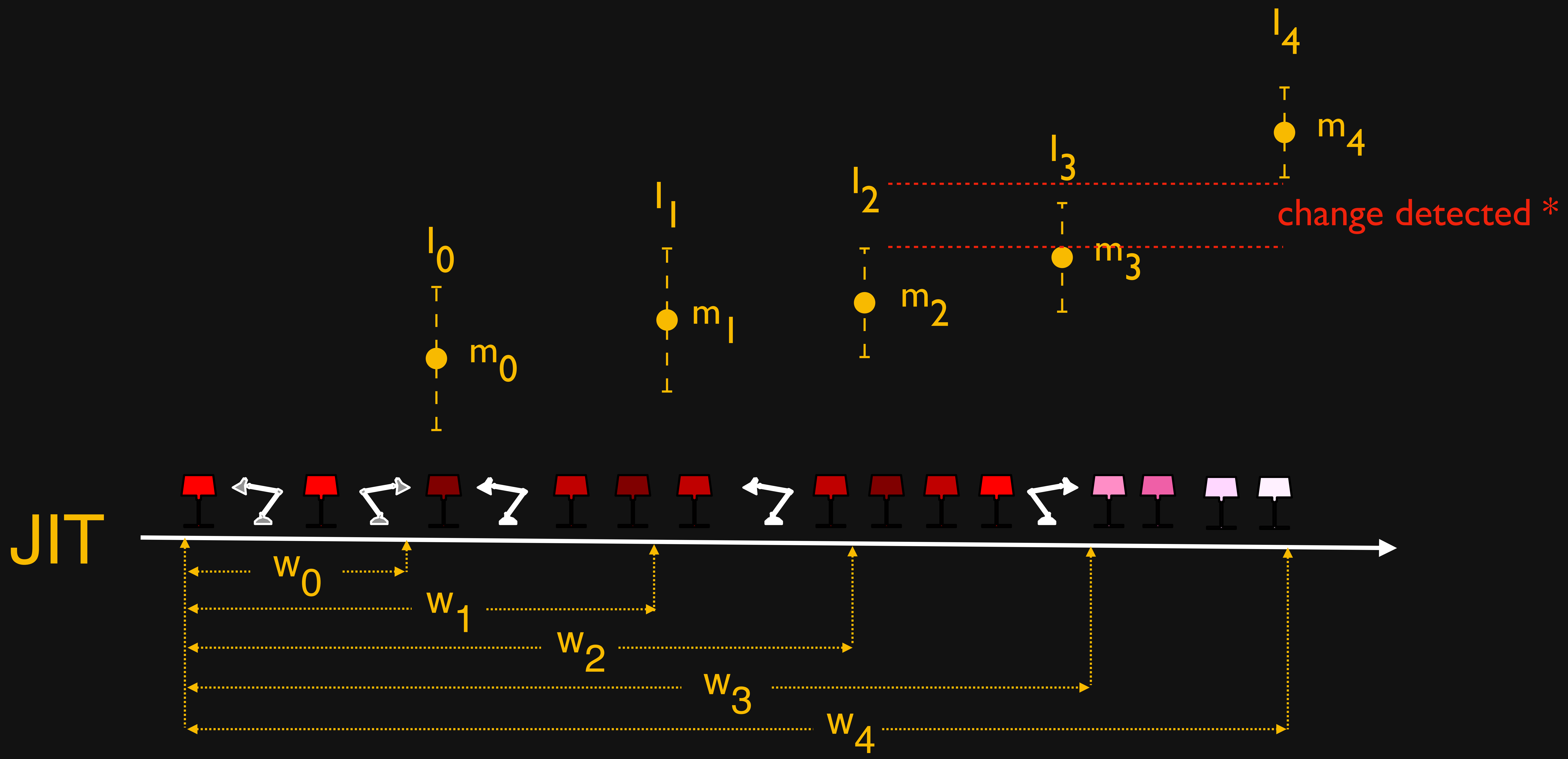
- pure methods
- ensemble methods

explicit mechanisms for adaptation



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

explicit mechanisms for adaptation



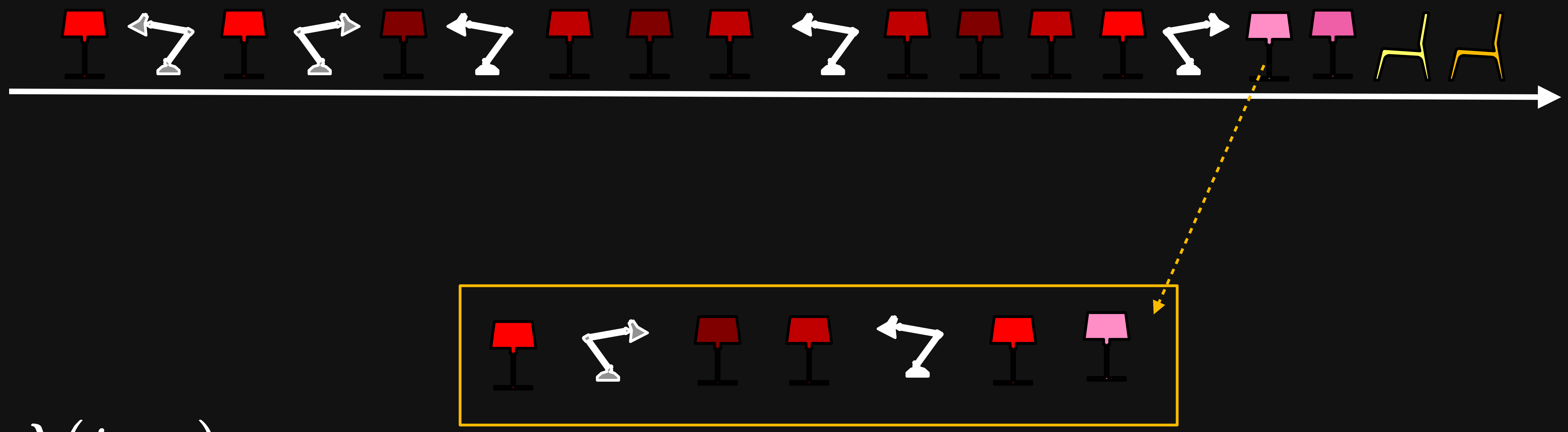
* Adaptation goes through a similar refinement process.

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

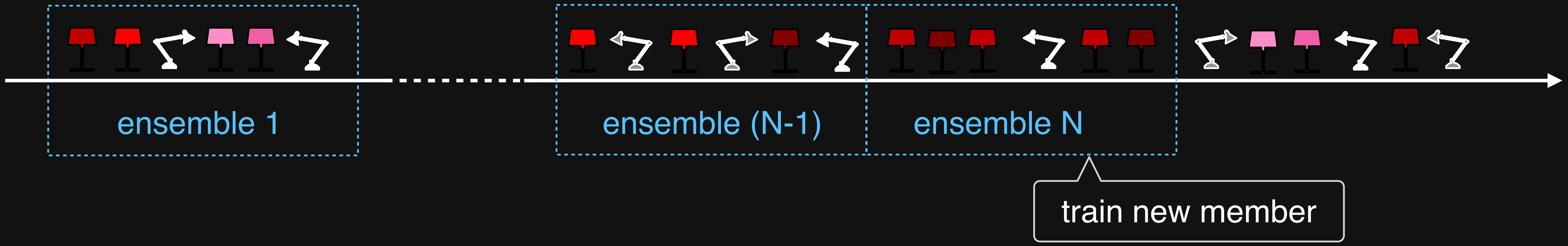
adapt how?

explicit mechanisms
- windowing
- weighting
- sampling

implicit mechanisms
- pure methods
- ensemble methods

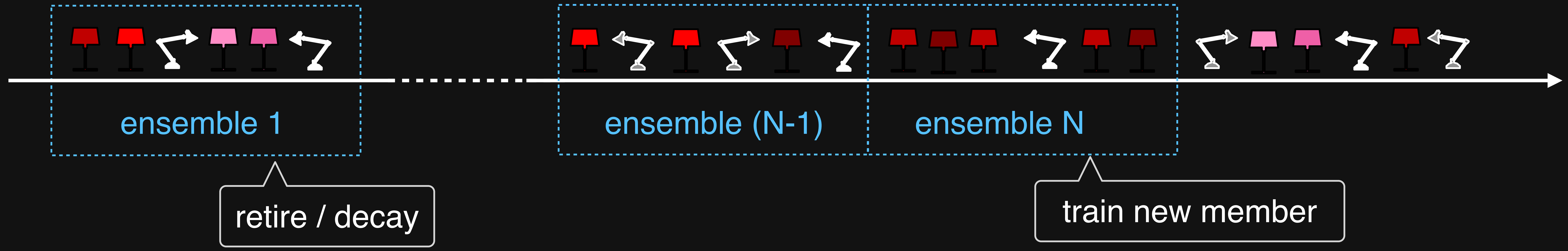
implicit mechanisms for adaptation

Ensemble Based Adaptation



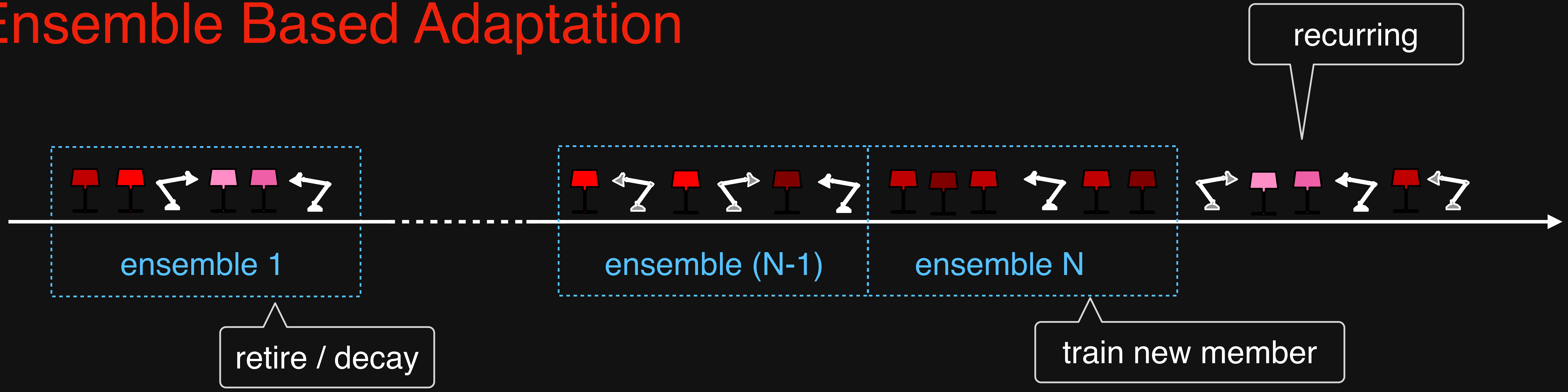
implicit mechanisms for adaptation

Ensemble Based Adaptation



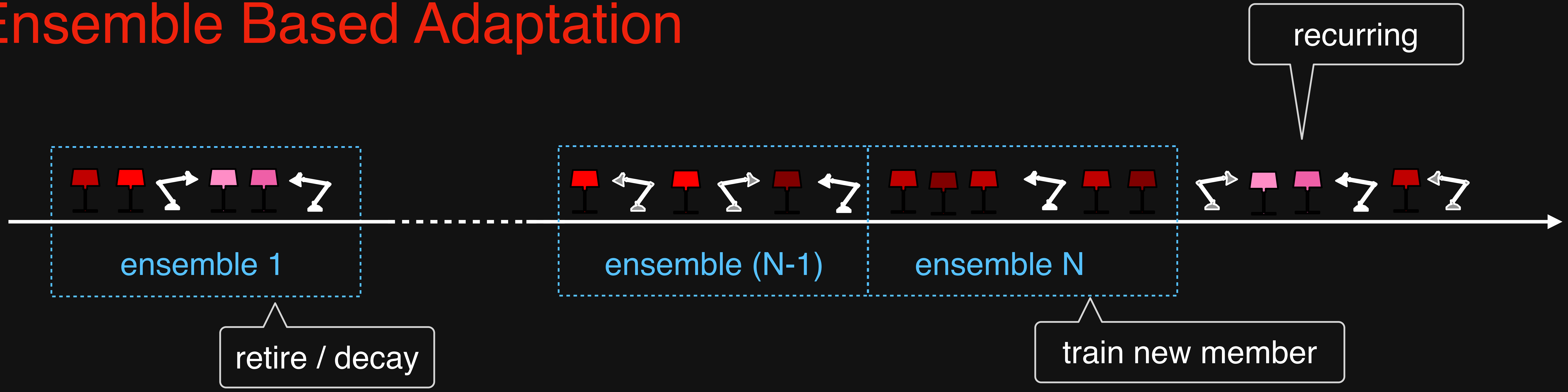
implicit mechanisms for adaptation

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 false alarms	sampling necessary in case of fast data, 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

which method?

Method	Efficiency	Pros	Cons	Notes
ECSMiner / GC3	$O(W^2 / k)$ $O(G \log C)$	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	

Appendix

- [LFR] : Wang, H., Abraham, Z., 2015. Concept drift detection for stream- ing data. In: International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–9.
- [DDM] : Gama, J., Medas, P., Castillo, G., Rodrigues, P., 2004. Learning with drift detection. In: Advances in artificial intelligence–SBIA 2004. Springer, pp. 286–295.
- [EDDM] : Baena-Garcıa, M., del Campo-A´vila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R., 2006. Early drift detection method. In: Fourth international workshop on knowledge discov- ery from data streams. Vol. 6. pp. 77–86.
- [ADWIN] : Bifet, A., Gavalda, R., 2007. Learning from time-changing data with adaptive windowing. In: SDM. Vol. 7. SIAM

Appendix

- [OLINDDA] : Spinosa, E. J., de Leon F de Carvalho, A. P., Gama, J., 2007. Olindda: A cluster-based approach for detecting novelty and concept drift in data streams. In: Proceedings of the 2007 ACM symposium on Applied computing. ACM, pp. 448–452.
- [MINAS] : Faria, E. R., Gama, J., Carvalho, A. C., 2013. Novelty detection algorithm for data streams multi-class problems. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, pp. 795–800.
- [DETECTNOD] : Hayat, M. Z., Hashemi, M. R., 2010. A dct based approach for detecting novelty and concept drift in data streams. In: International Conference of Soft Computing and Pattern Recognition (SoCPaR). IEEE, pp. 373–378.
- [Woo-ensemble] : Ryu, J. W., Kantardzic, M. M., Kim, M.-W., Khil, A. R., 2012. An efficient method of building an ensemble of classifiers in streaming data. In: Big data analytics. Springer, pp. 122–133.
- [ECSMiner] : Masud, M. M., Gao, J., Khan, L., Han, J., Thuraisingham, B., 2011. Classification and novel class detection in concept-drifting data streams under time constraints. IEEE TKDE 23 (6), 859–874.
- [GC3] : Sethi, T. S., Kantardzic, M., Hu, H., 2016b. A grid density based framework for classifying streaming data in the presence of concept drift. Journal of Intelligent Information Systems 46 (1), 179– 211.

Appendix

- [CoC] : Lee, J., Magoules, F., 2012. Detection of concept drift for learn- ing from stream data. In: IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS). IEEE, pp. 241–245.
- [HDDDM] : Ditzler, G., Polikar, R., 2011. Hellinger distance based drift detection for nonstationary environments. In: IEEE Symposium on Com- putational Intelligence in Dynamic and Uncertain Environments (CIDUE). IEEE, pp. 41–48.
- [PCA-1] : Kuncheva, L. I., Faithfull, W. J., 2014. Pca feature extraction for change detection in multidimensional unlabeled data. IEEE Transactions on Neural Networks and Learning Systems 25 (1), 69–80.
- [PCA-2] : Qahtan, A. A., Alharbi, B., Wang, S., Zhang, X., 2015. A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In: Proc. of the 21th ACM SIGKDD ICKDDM. ACM, pp. 935–944.

Appendix

- [A-distance]: D. Kifer, S. Ben-David, and J. Gehrke, “Detecting change in data streams,” in *Proc. 30th Int. Conf. Very Large Data Bases*, 2004, vol. 30, pp. 180–191.
- [margin] : Dries, A., Ruckert, U., 2009. Adaptive concept drift detection. Statistical Analysis and Data Mining 2 (5-6), 311–327.
- [MD3]: Sethi, T. S., Kantardzic, M., 2017. On the reliable detection of concept drift from streaming unlabeled data. Expert Syst. Appl. 82, C (October 2017), 77-99. DOI: https://doi.org/10.1016/j.eswa.2017.04.008
- [sampling]: C. C. Aggarwal, “On biased reservoir sampling in the presence of stream evolution,” in Proc. 32nd Int. Conf. Very Large Data Bases, 2006, pp. 607–618.

Appendix

- [JIT]: C. Alippi, G. Boracchi, and M. Roveri, “A just-in-time adaptive classification system based on the intersection of confidence intervals rule,” Neural Netw., vol. 24, no. 8, pp. 791–800, Oct. 2011.
- [ONSboost] : A. Pocock, P. Yiapanis, J. Singer, M. Lujan, and G. Brown, “Online nonstationary boosting,” in Proc. Int. Workshop Multiple Classifier Systems, 2010, pp. 205–214.
- [NSRF] : H. Abdulsalam, D. Skillicorn, and P. Martin, “Classification using streaming random forests,” IEEE Trans. Knowledge Data Eng., vol. 23, no. 1, pp. 22–36, Jan. 2011.
- [DWM] : J. Kolter and M. Maloof, “Dynamic weighted majority: An ensemble method for drifting concepts,” J. Mach. Learn. Res., vol. 8, pp. 2755–2790, Dec. 2007
- [Learn++.NSE] : R. Elwell and R. Polikar, “Incremental learning of concept drift in nonstationary environments,” IEEE Trans. Neural Netw., vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

thank you

emre.velipasaoglu@ Lightbend .com