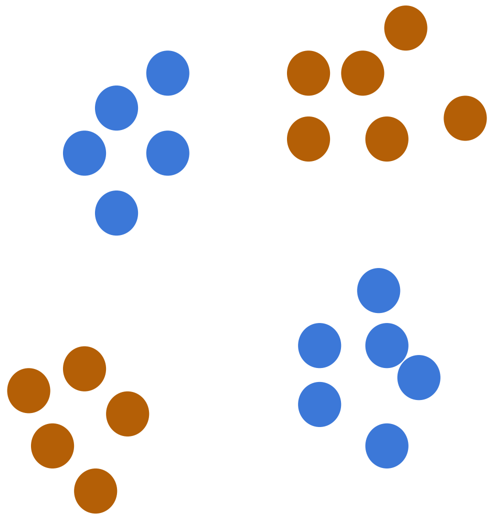
Federated Nearest Neighbor Classification with a Colony of Fruit-Flies

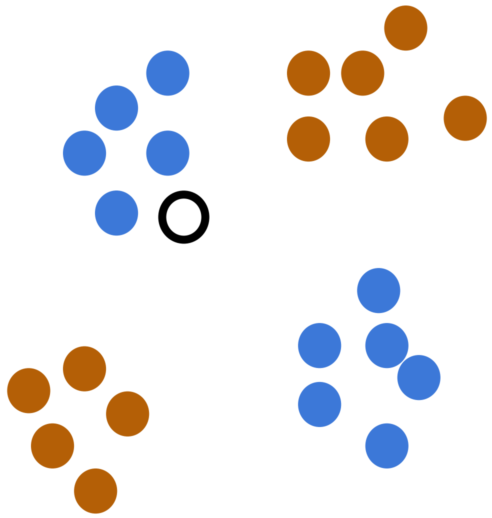
Parikshit Ram (**IBM Research AI**), Kaushik Sinha (**Wichita State University** & **IFML**)

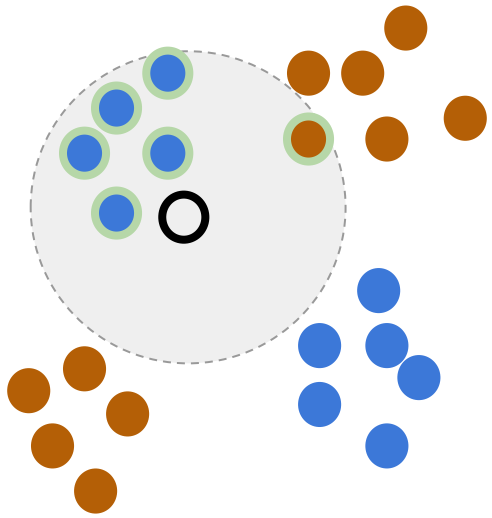
Institute for Foundations of MACHINE LEARNING

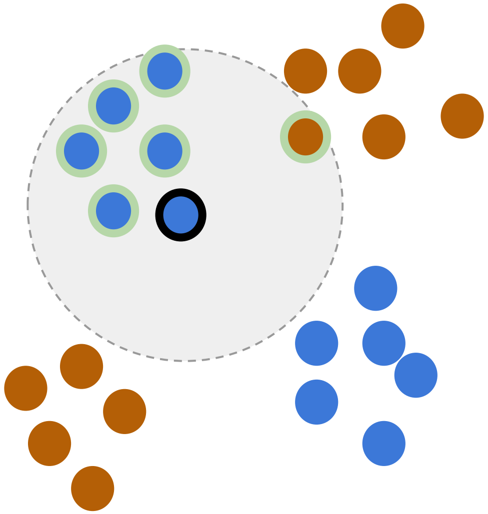
WICHITA STATE UNIVERSITY

Nearest Neighbor Classifier

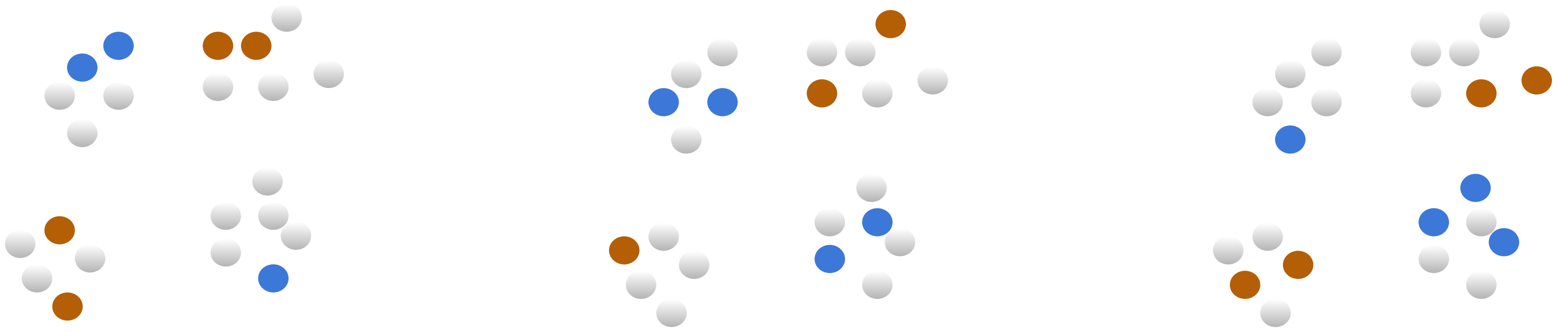


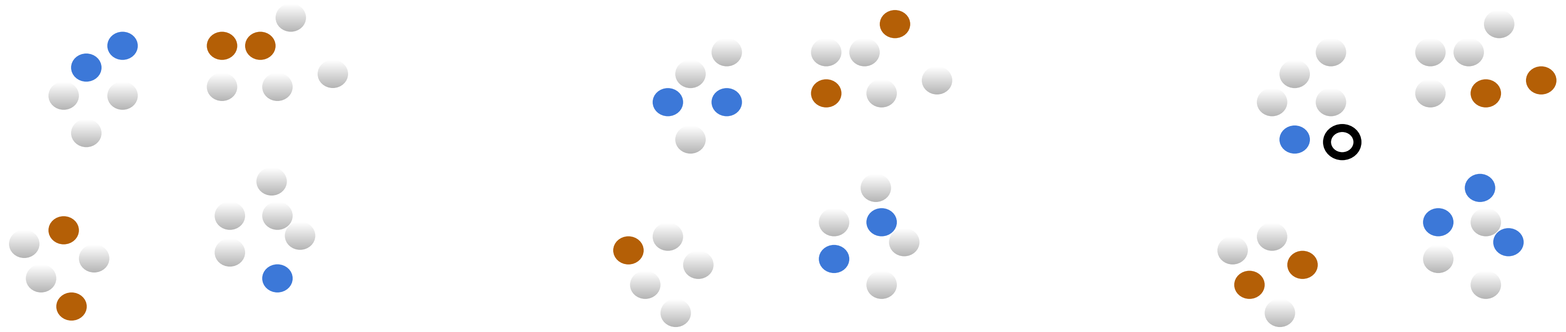


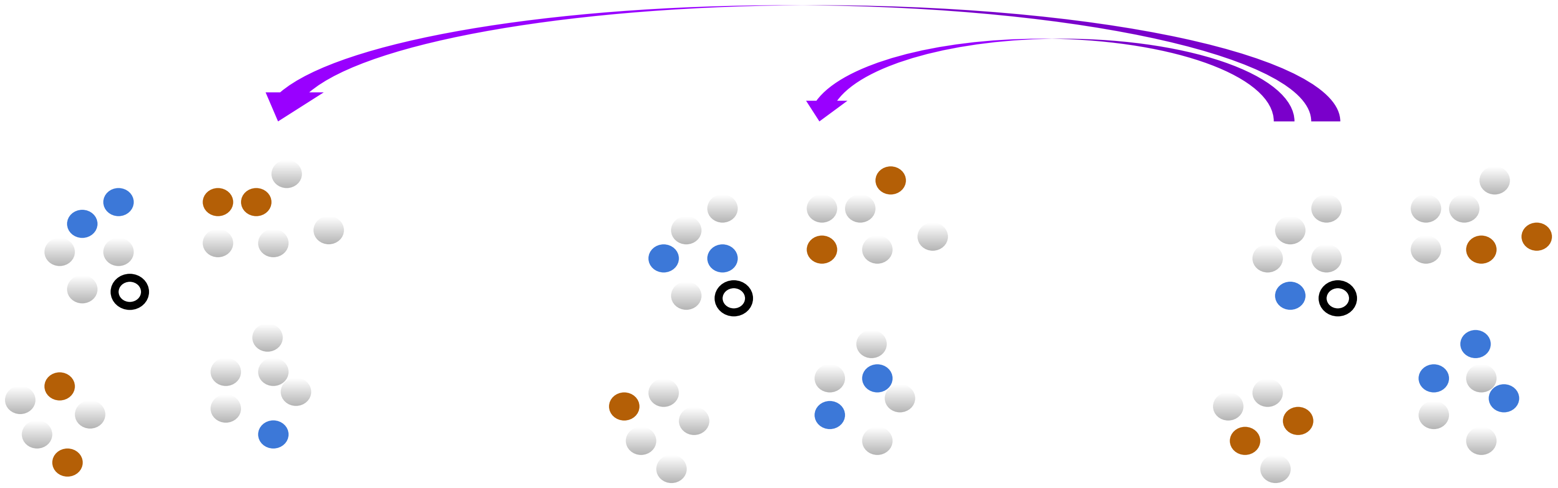


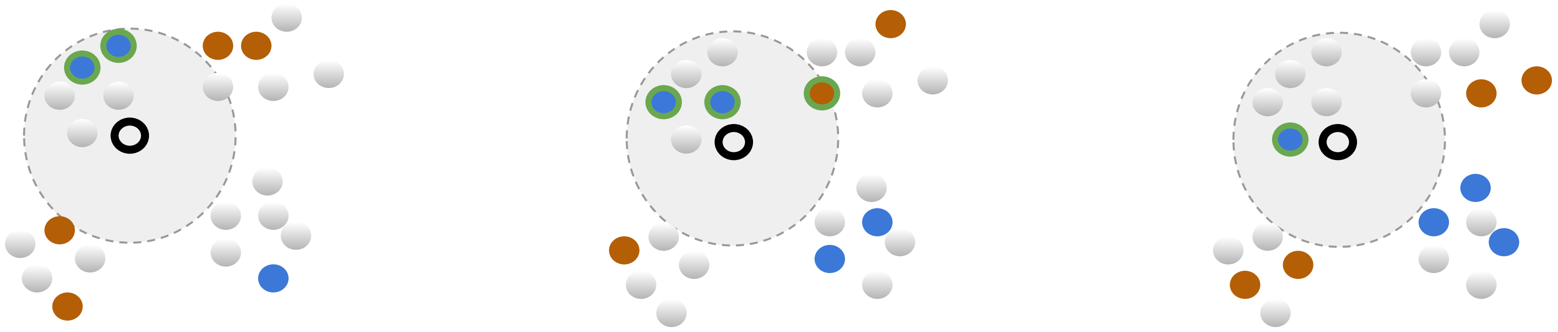


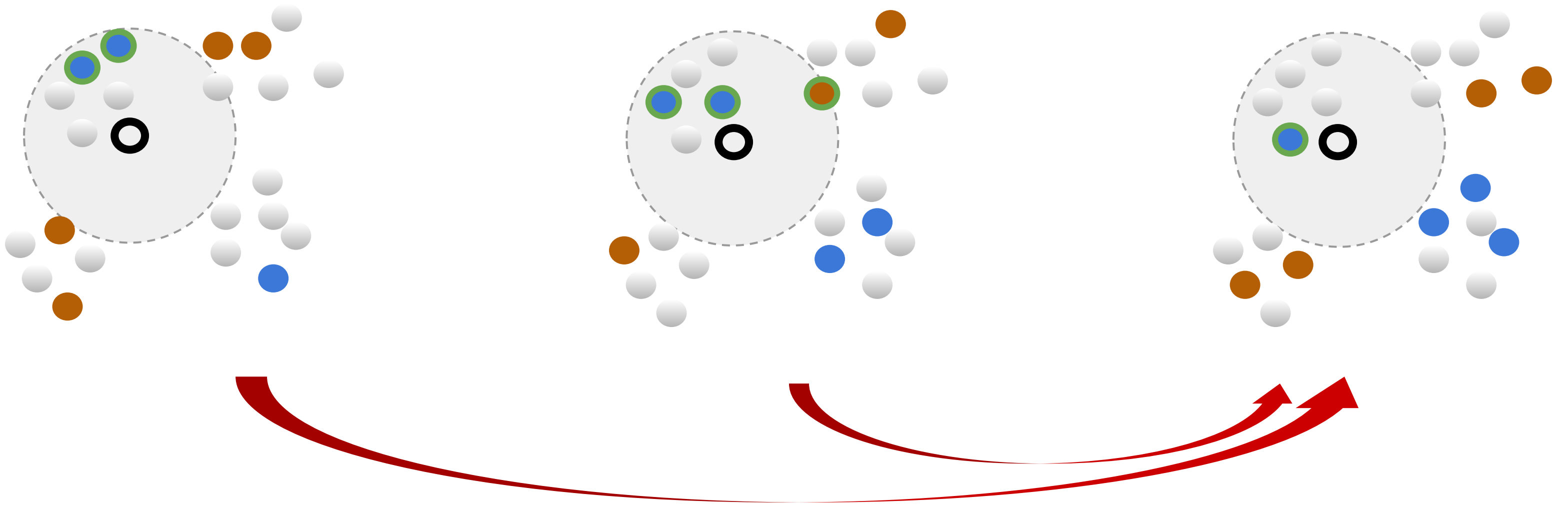
Distributed
Nearest Neighbor
Classifier

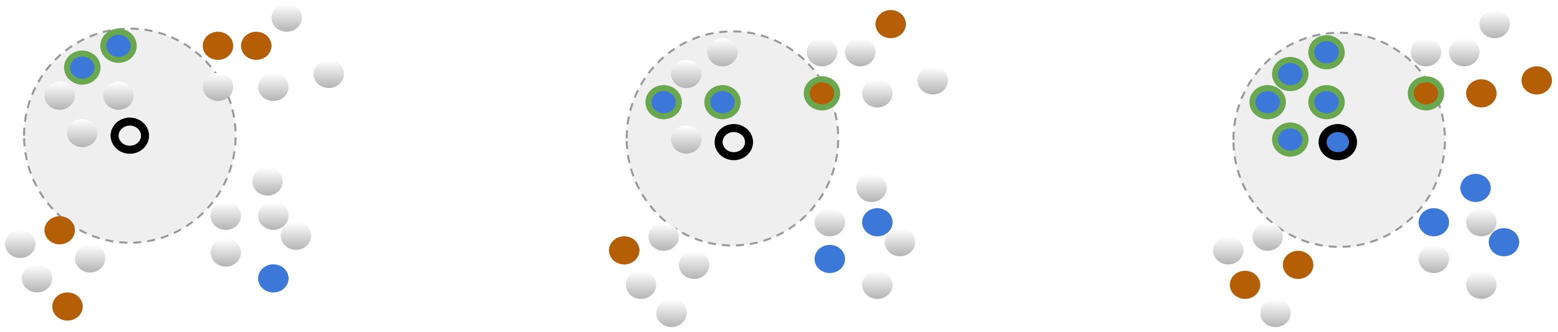




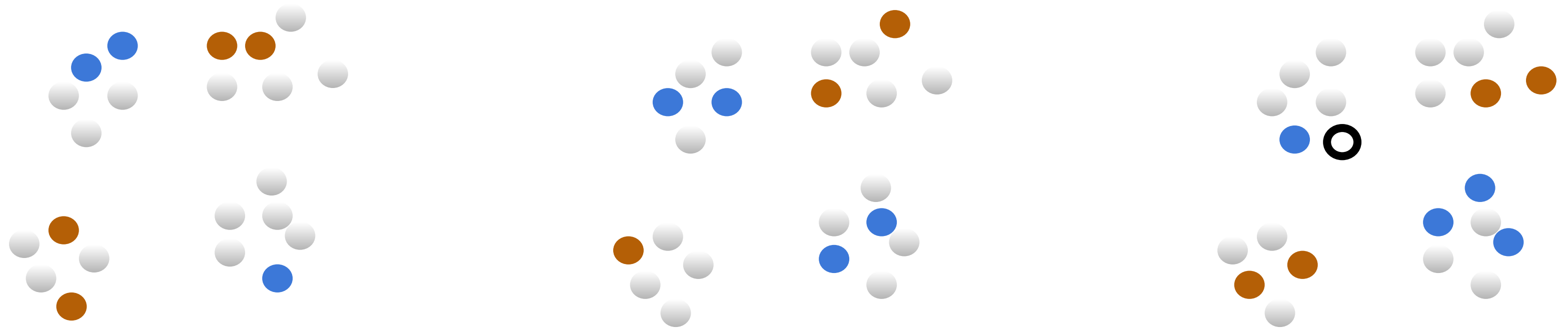


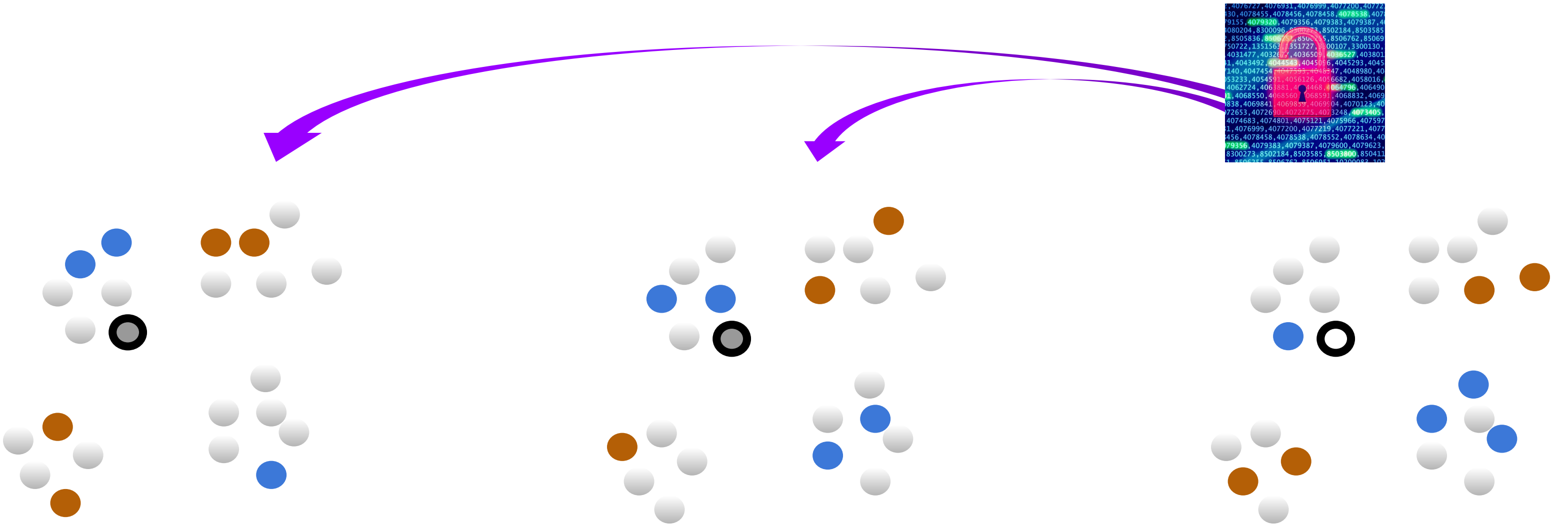


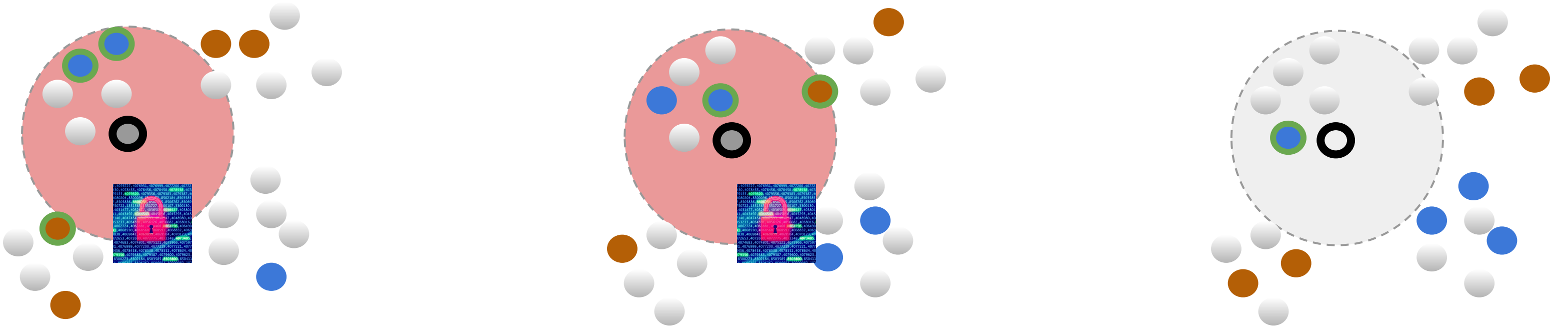


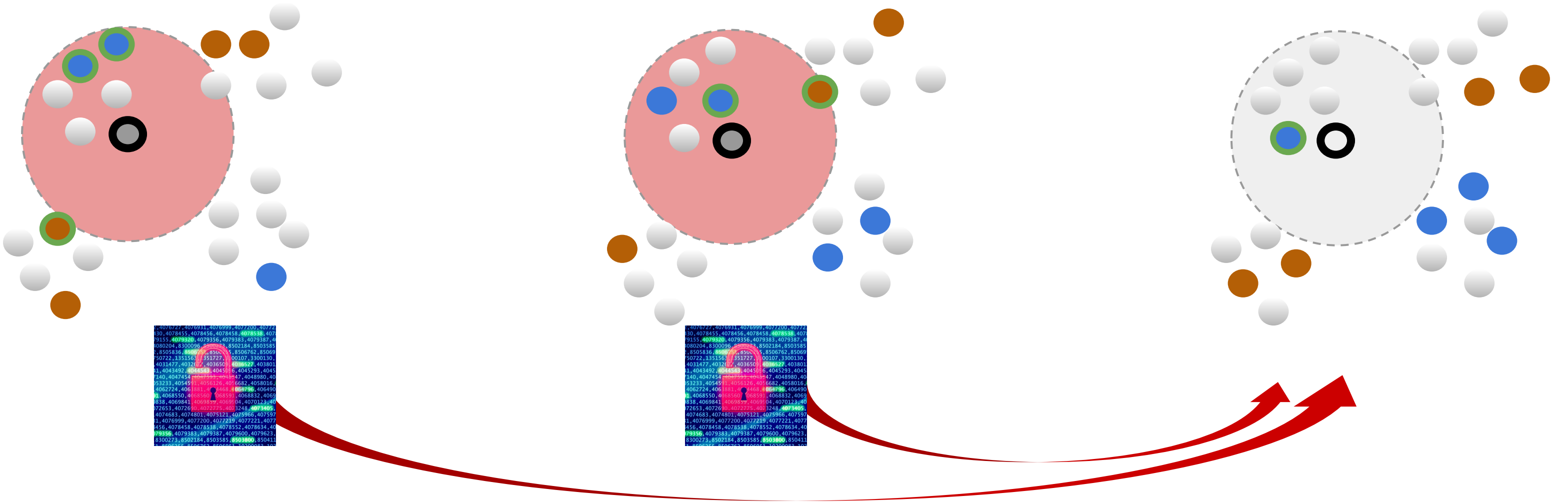


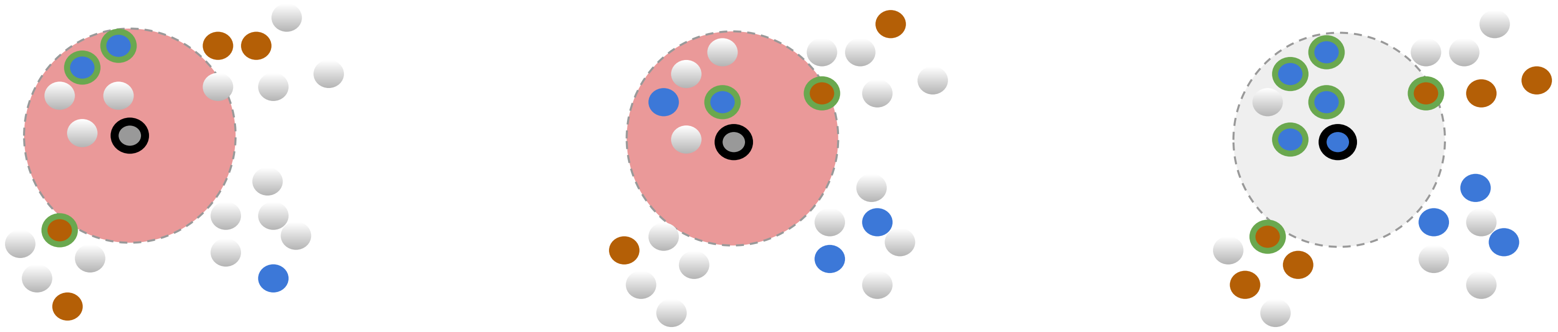
Federated
Nearest Neighbor
Classifier



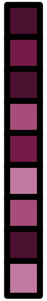


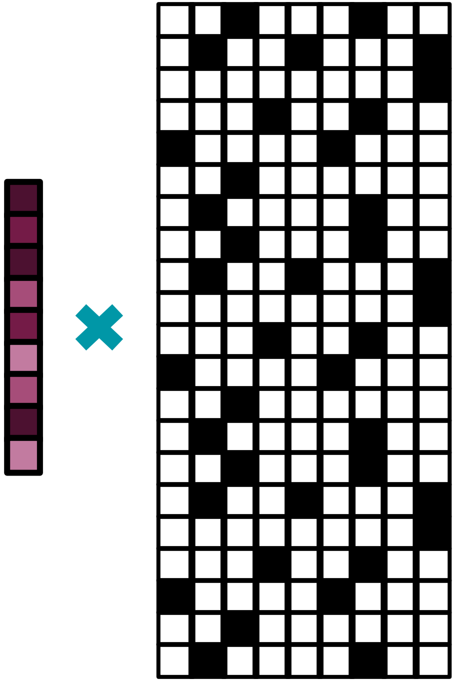


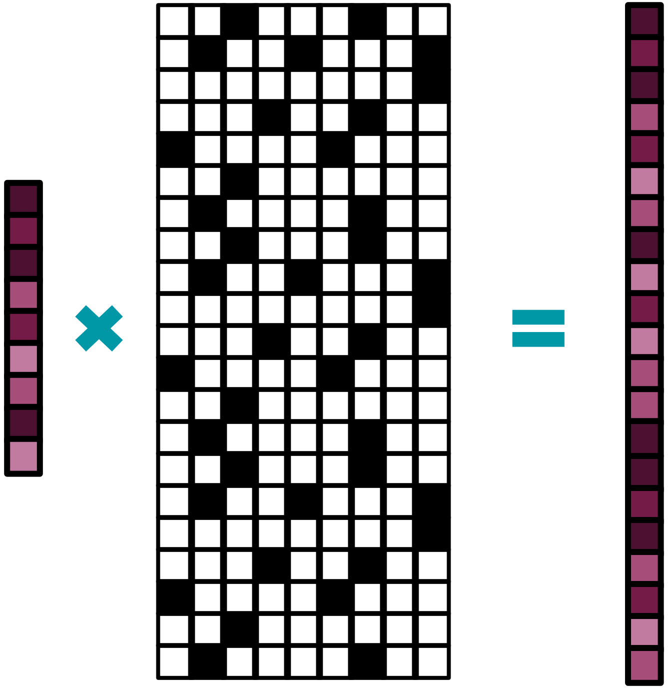


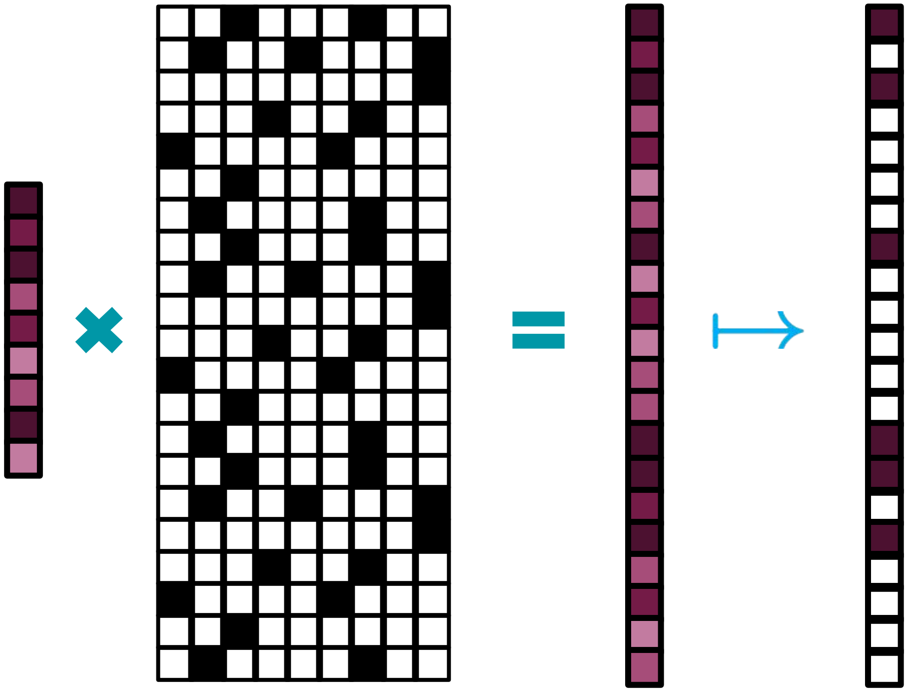


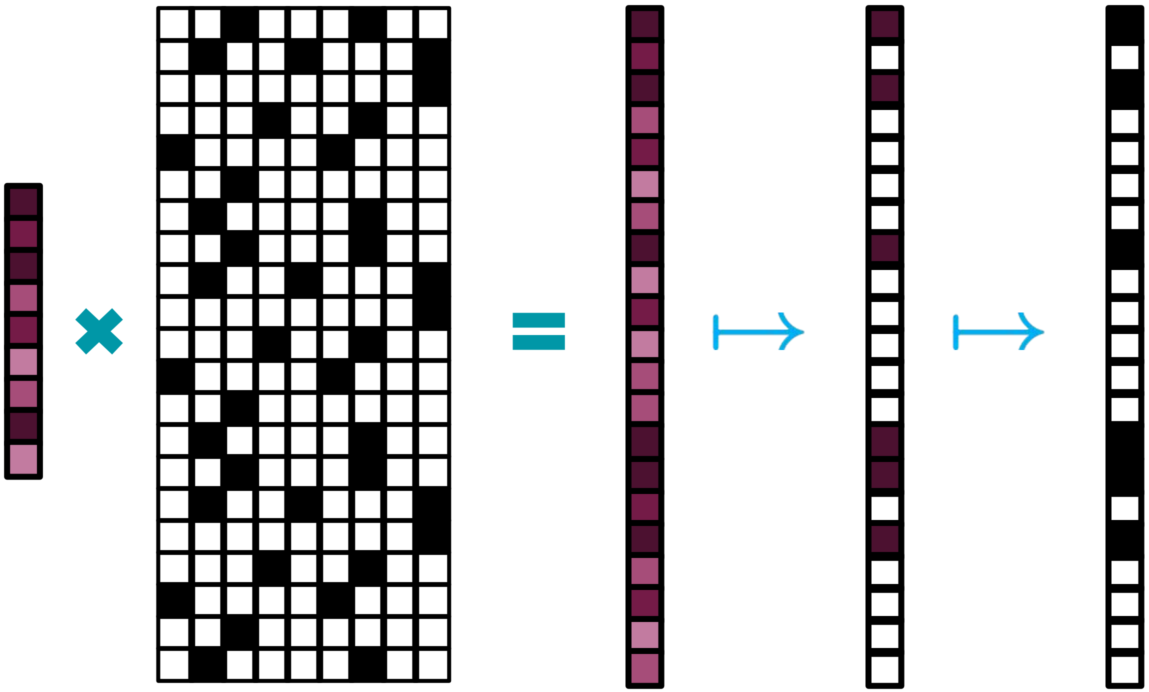
Fruit-Fly Inspired Data-Structures











REPORT

A neural algorithm for a fundamental computing problem

SANJOY DASGUPTA, CHARLES F. STEVENS, AND , SAKET NAVLAKHA Authors Info & Affiliations

SCIENCE • 10 Nov 2017 • Vol 358, Issue 6364 • pp. 793-796 • DOI: 10.1126/science.aam9868

REPORT

A neural algorithm for a fundamental computing problem

SANJOY DASGUPTA, CHARLES F. STEVENS, AND, SAKET NAVLAKHA Authors Info & Affiliations

SCIENCE · 10 Nov 2017 · Vol 358, Issue 6364 · pp. 793-796 · DOI: 10.1126/science.aam9868

PNAS

A neural data structure for novelty detection

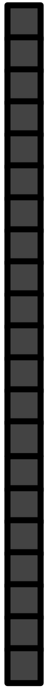
Sanjoy Dasgupta[a], Timothy C. Sheehan[b], Charles F. Stevens[c,d,1], and Saket Navlakha[e,1]

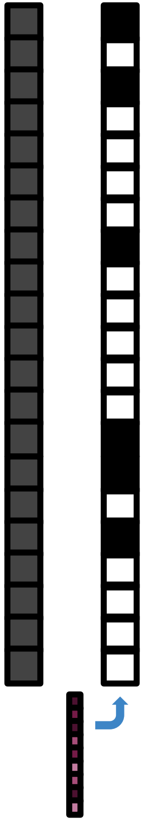
[a]Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093; [b]Neurosciences Graduate Program, University of California, San Diego, La Jolla, CA 92093; [c]Kavli Institute for Brain & Mind, University of California, San Diego, La Jolla, CA 92093; [d]Molecular Neurobiology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037; and [e]Integrative Biology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037

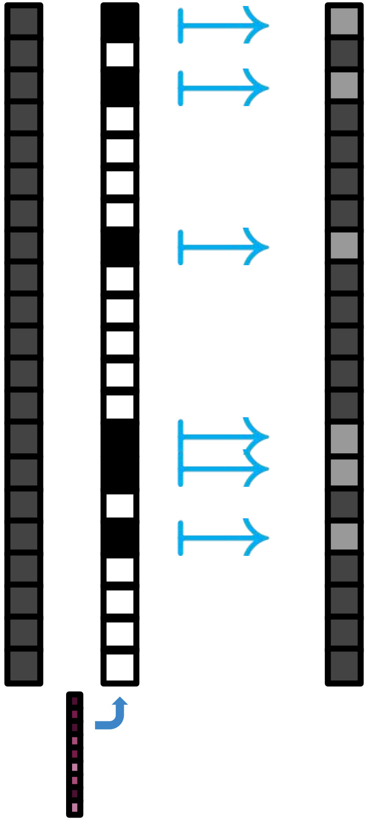
Contributed by Charles F. Stevens, October 30, 2018 (sent for review August 22, 2018; reviewed by Piotr Indyk and Glenn Turner)

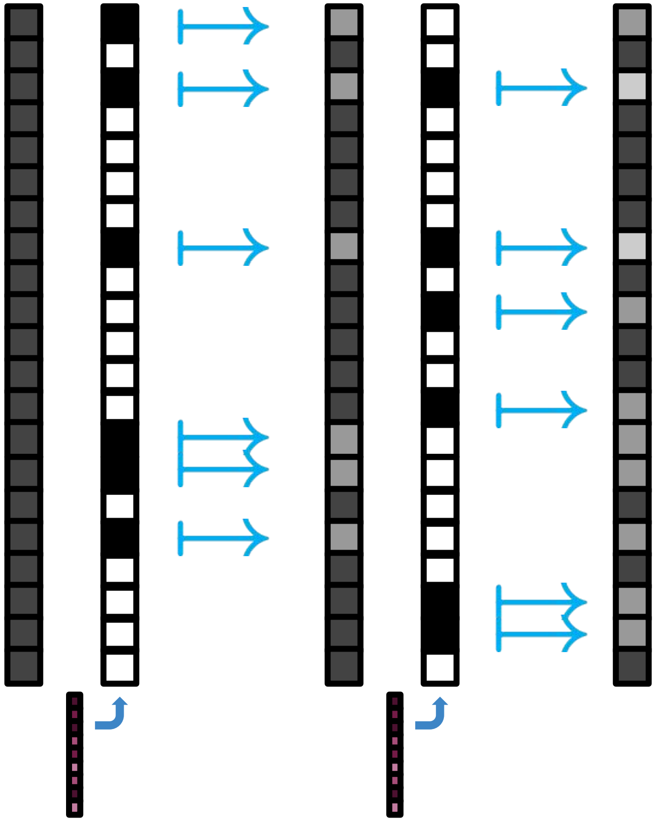
Novelty detection is a fundamental biological problem that organisms must solve to determine whether a given stimulus departs from those previously experienced. In computer science, this prob-

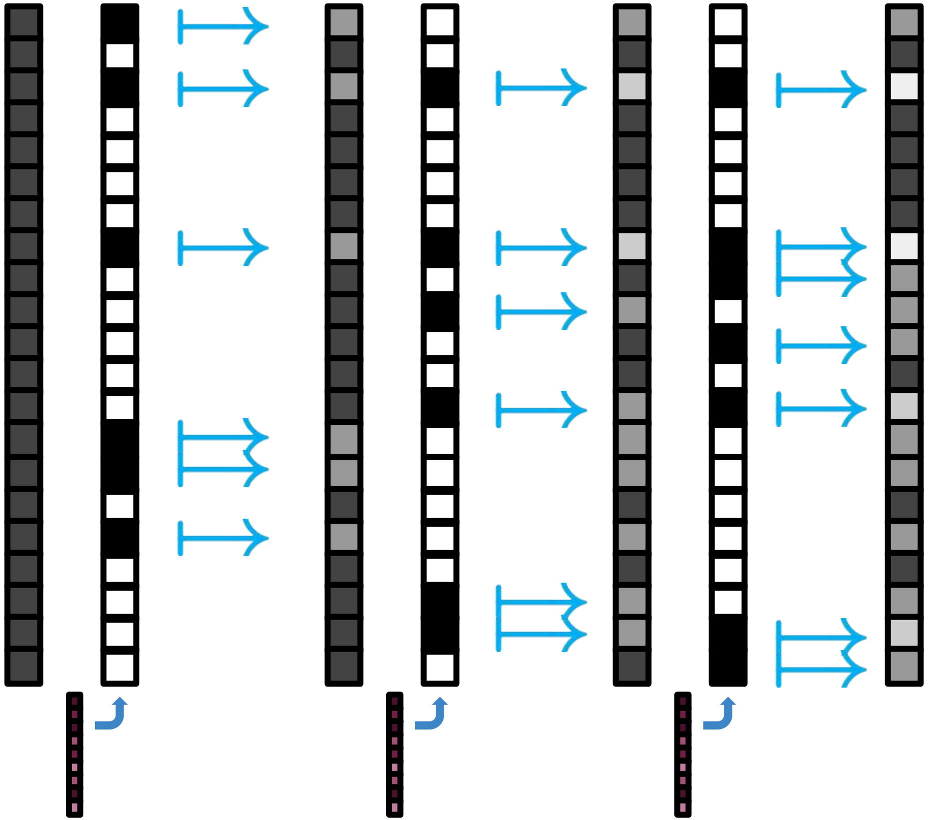
by 34 mushroom body output neurons (MBONs) that receive input from the KCs and perform many different functions (11–13). Here we are concerned with one such MBON, called

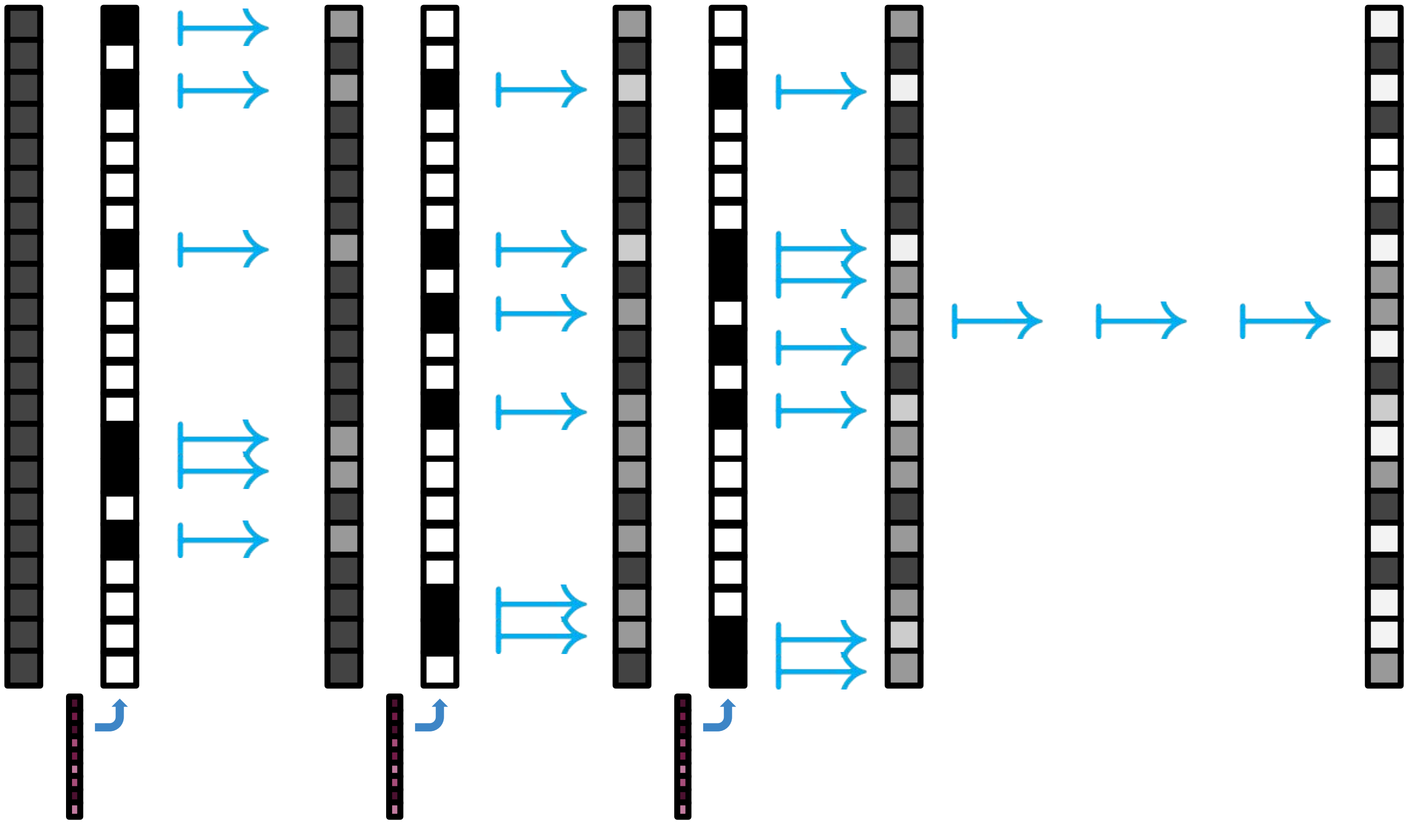


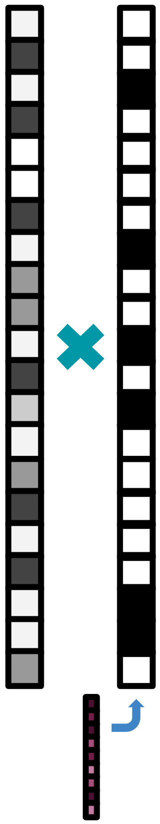


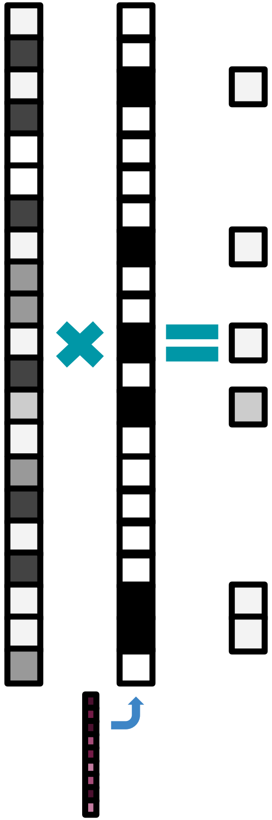


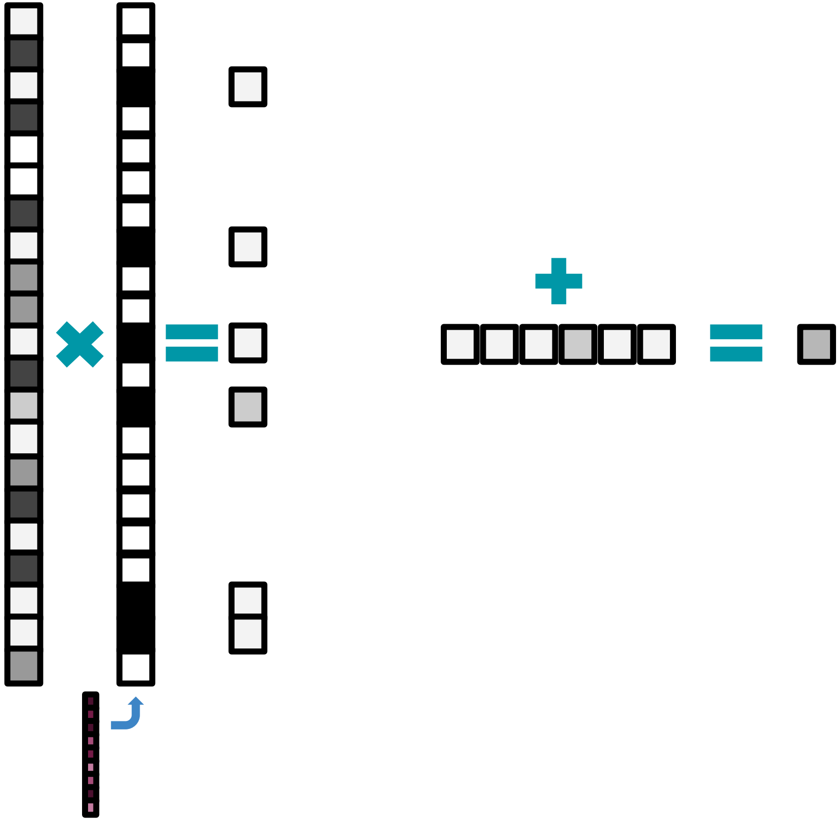


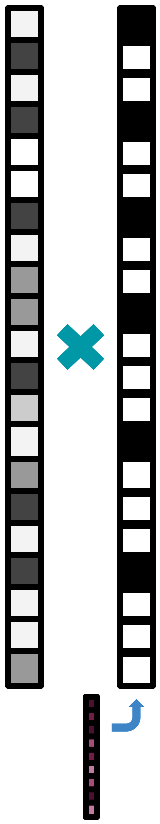


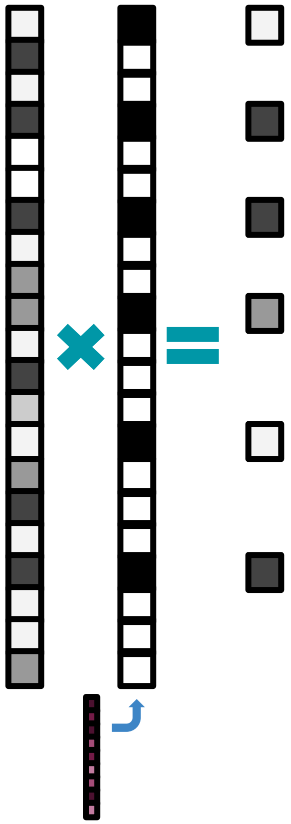


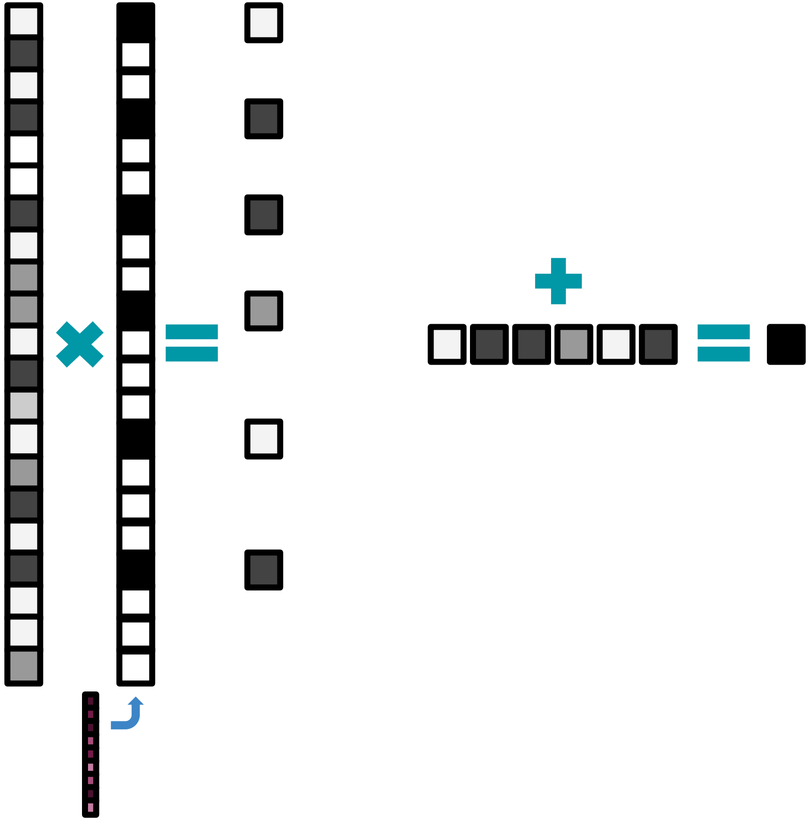




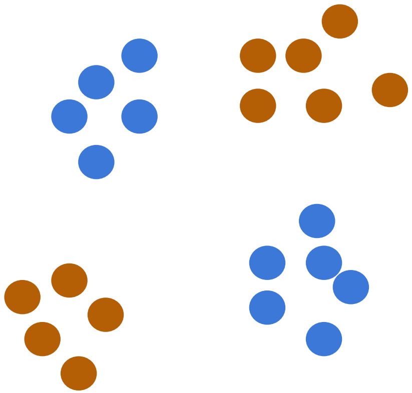


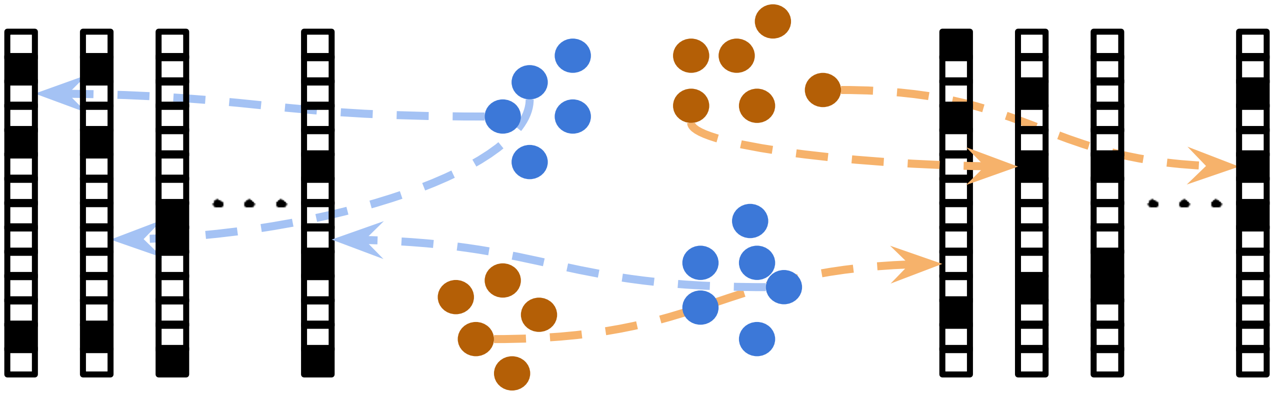


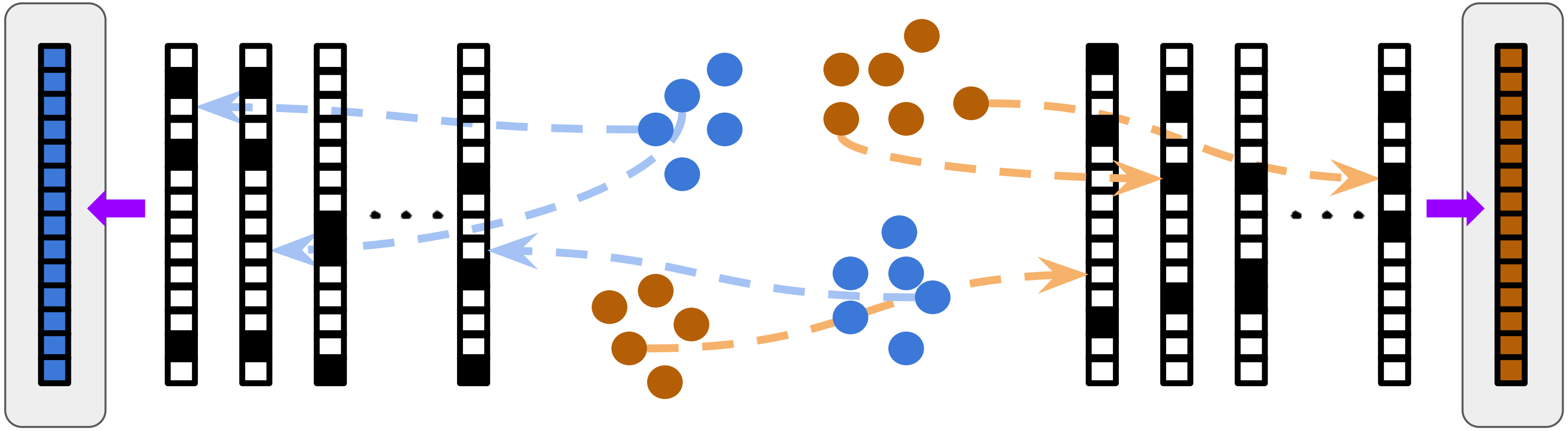


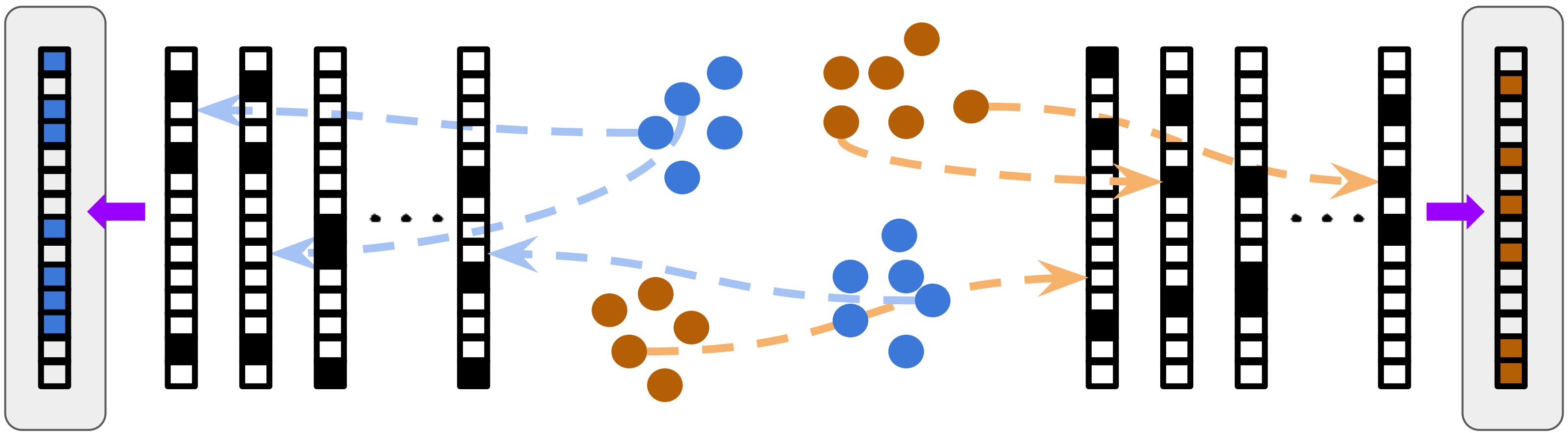


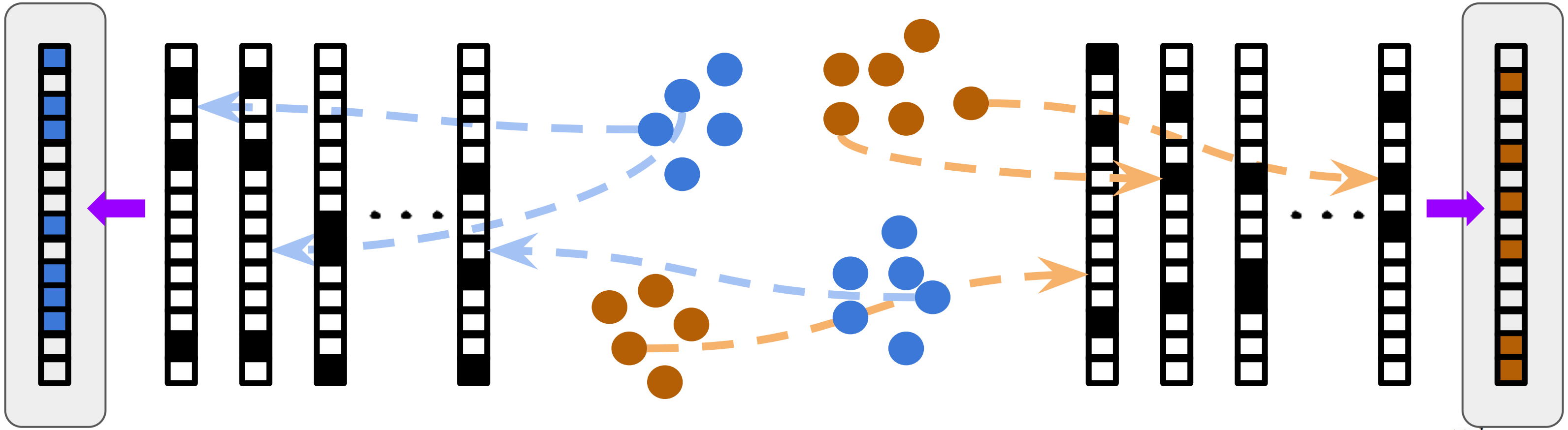
FlyNN Classifier











Sample $\mathsf{M} \in \{0,1\}^{m \times d}$ with seed R

Initialize $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_L \leftarrow \mathbf{1}_m \in (0,1)^m$

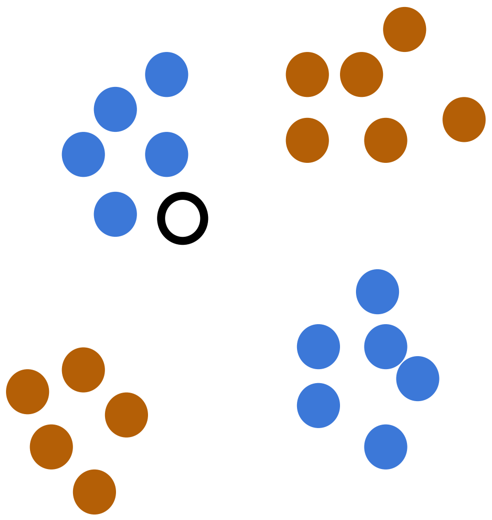
for $(\boldsymbol{x}, y) \in S$ **do**

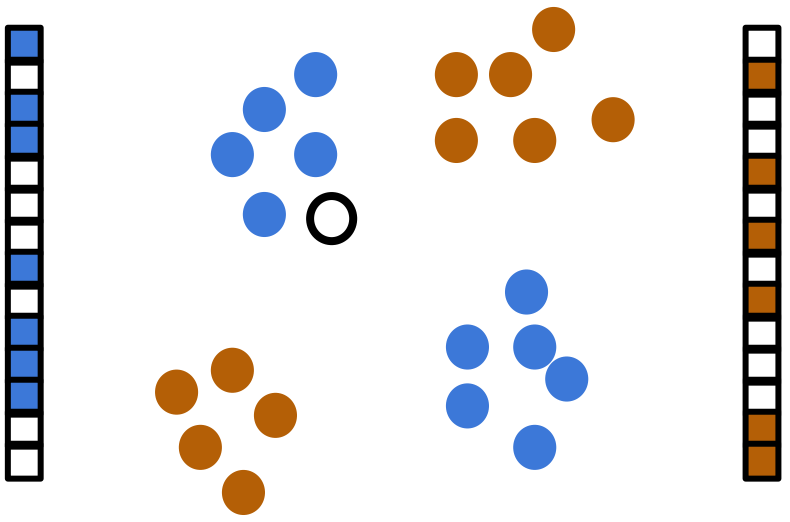
  $\mathbf{h} \leftarrow \Gamma_\rho(\mathsf{M}\boldsymbol{x})$

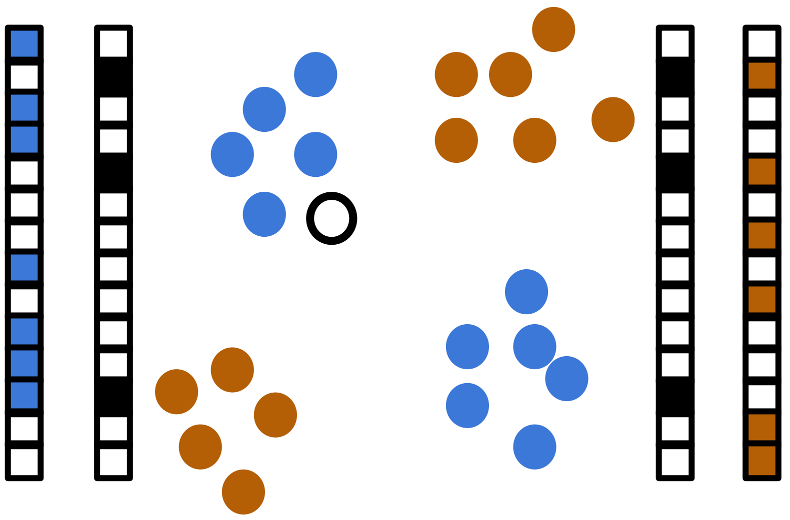
  $\boldsymbol{w}_y[i] \leftarrow \gamma \cdot \boldsymbol{w}_y[i]\ \forall i \in [m] : \mathbf{h}[i] = 1$

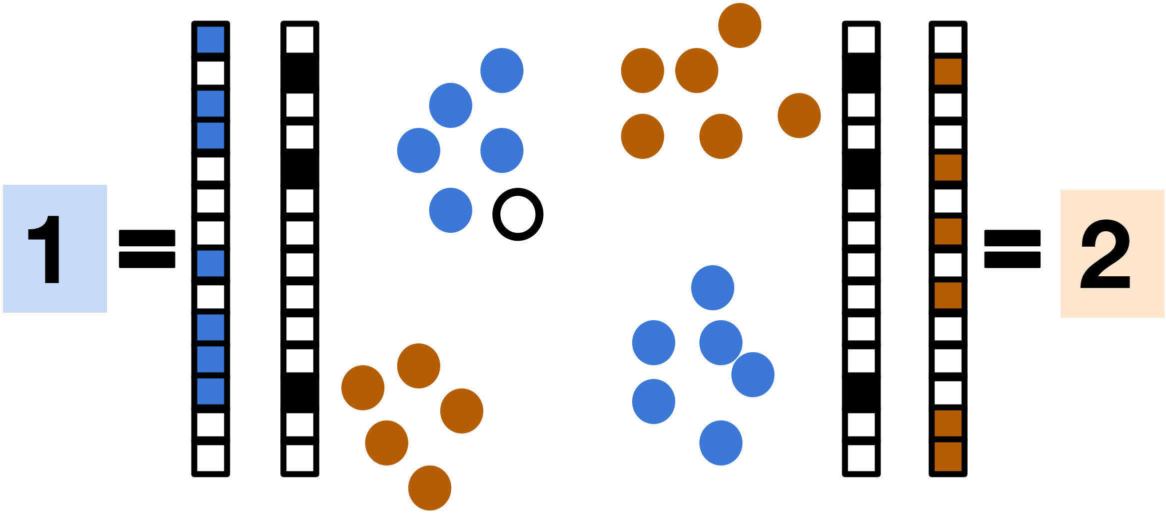
end

return $(\mathsf{M}, \{\boldsymbol{w}_l, l \in [L]\})$

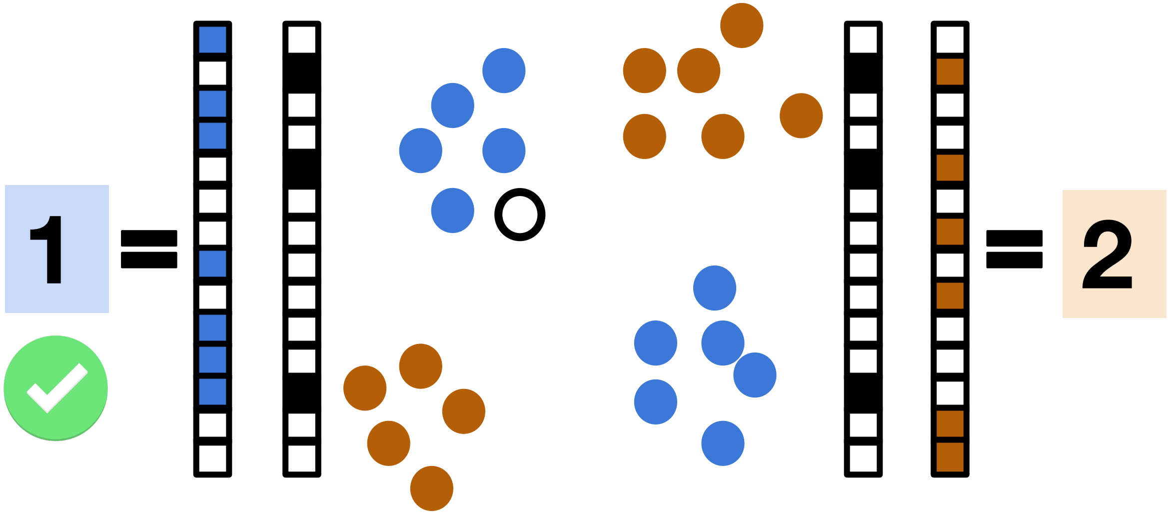






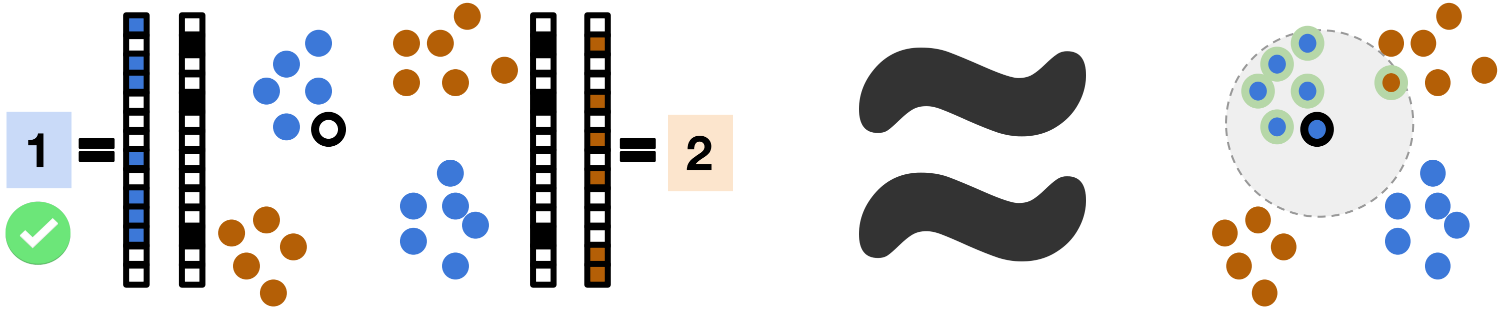


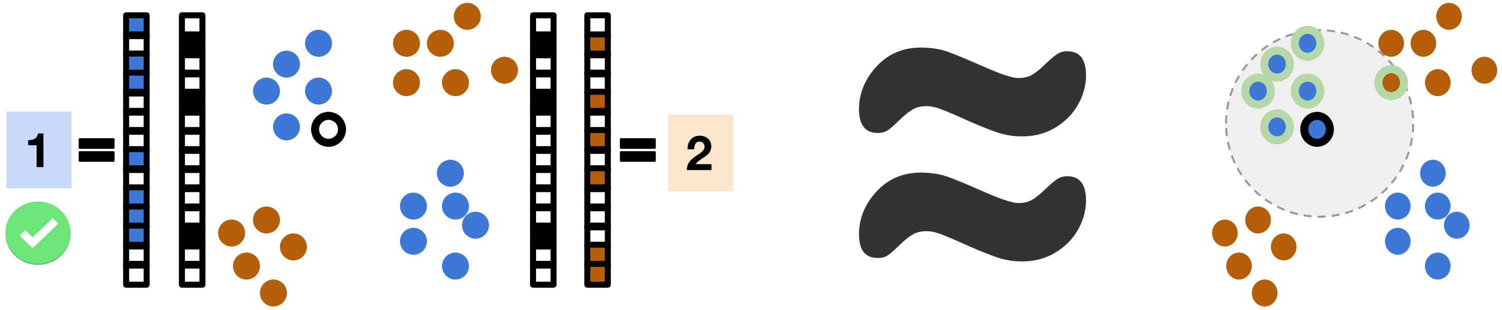
1 =
= 2



$\mathsf{h} \leftarrow \Gamma_\rho\left(\mathsf{M}\boldsymbol{x}\right)$

return $\operatorname{argmin}_{l\in[L]} \boldsymbol{w}_l^\top \mathsf{h}$

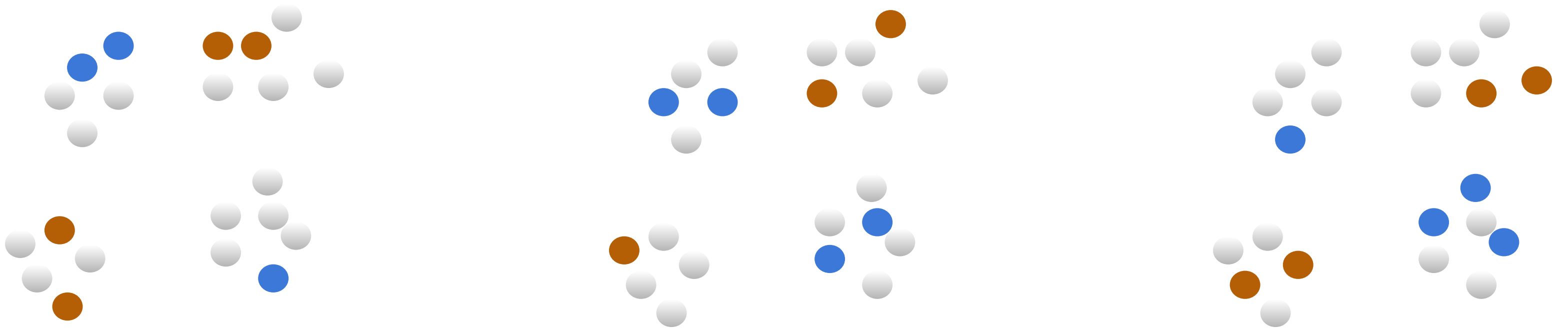

1
2

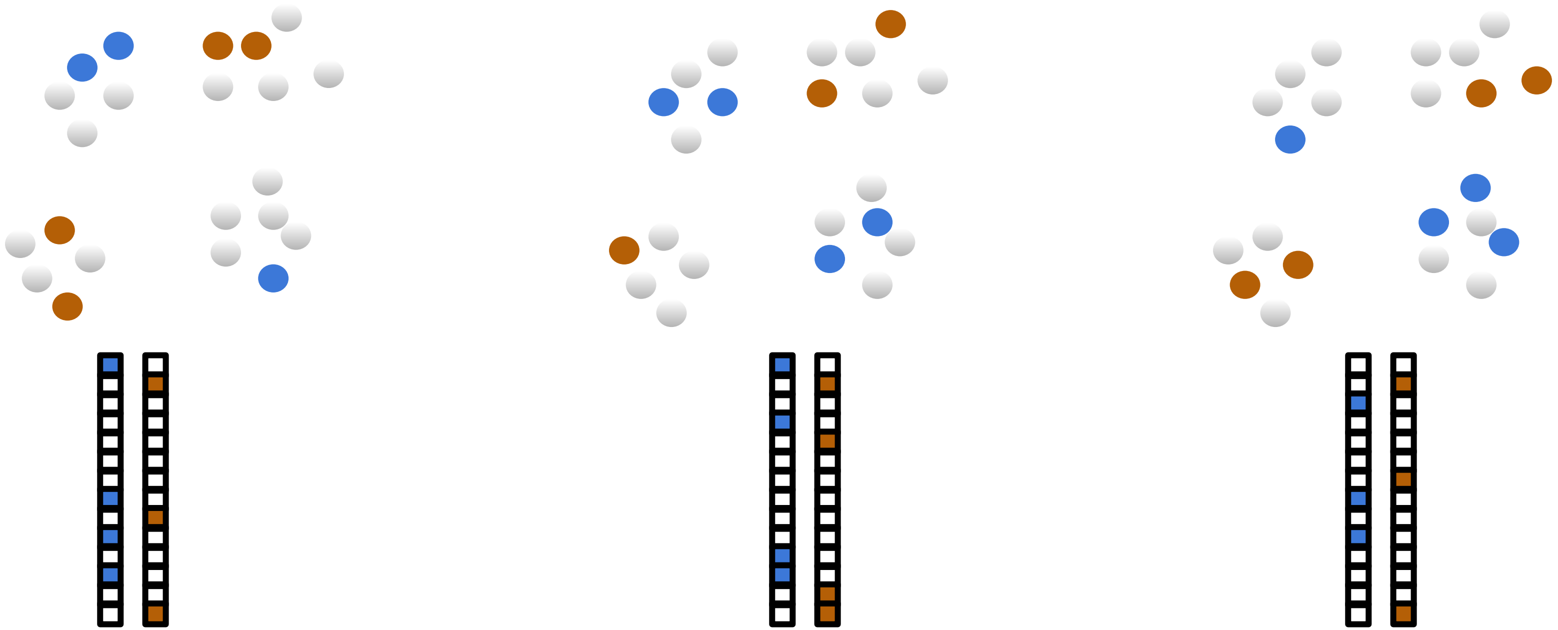


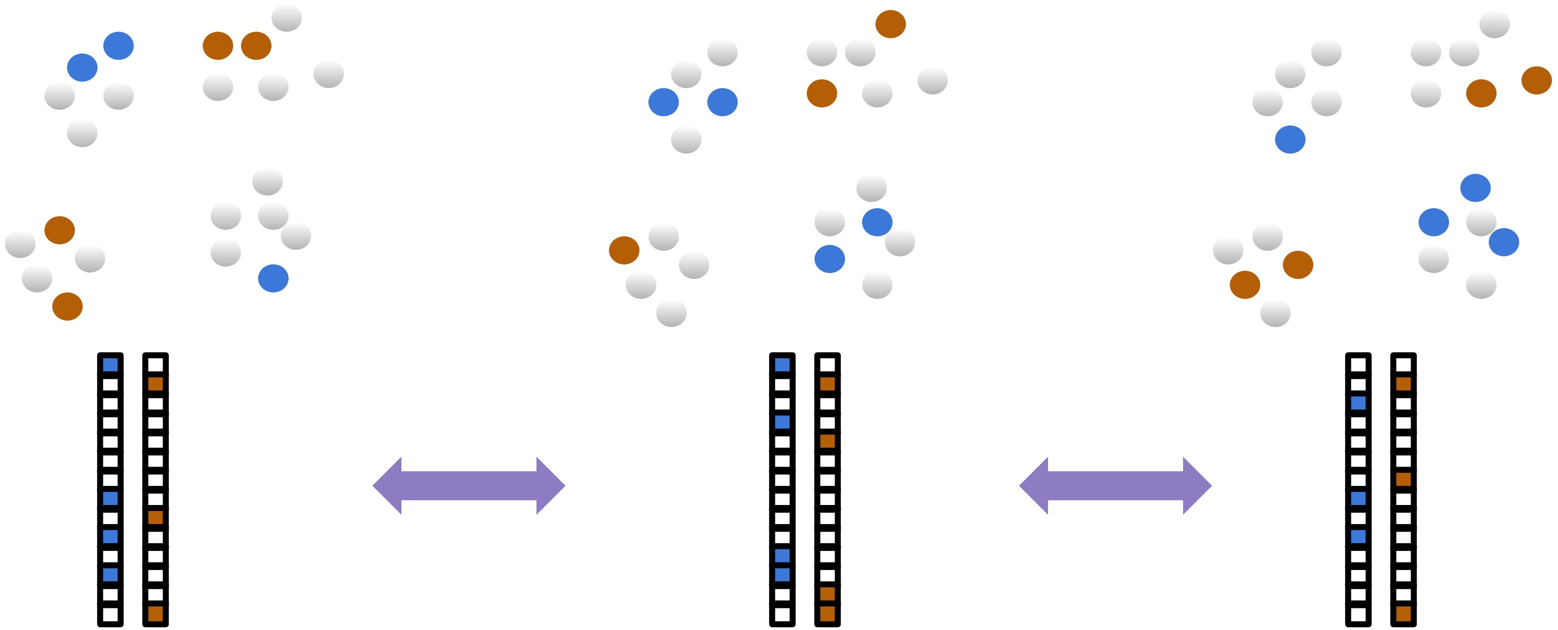
Theorem 3. *Fix s, ρ, m and k. Given a training set S of size n and a test example $\boldsymbol{x} \in \mathbb{R}^d$ sampled from a permutation invariant distribution, let $\boldsymbol{x}_*$ be its $(\lceil k+1/2 \rceil)^{th}$ nearest neighbor from S measured using ℓ_∞ metric. If $\|\boldsymbol{x} - \boldsymbol{x}_*\|_\infty \leq \min\{\eta/2, O(1/s)\}$ then, $\hat{y}_{\texttt{FlyNN}} = \hat{y}_{k\text{NNC}}$ with probability $\geq 1 - \left(O(\rho n/m) + e^{-O(\rho)}\right)$, where $\hat{y}_{\texttt{FlyNN}}$ and $\hat{y}_{k\text{NNC}}$ are respectively the predictions of* `FlyNN` *and* k*NNC.*

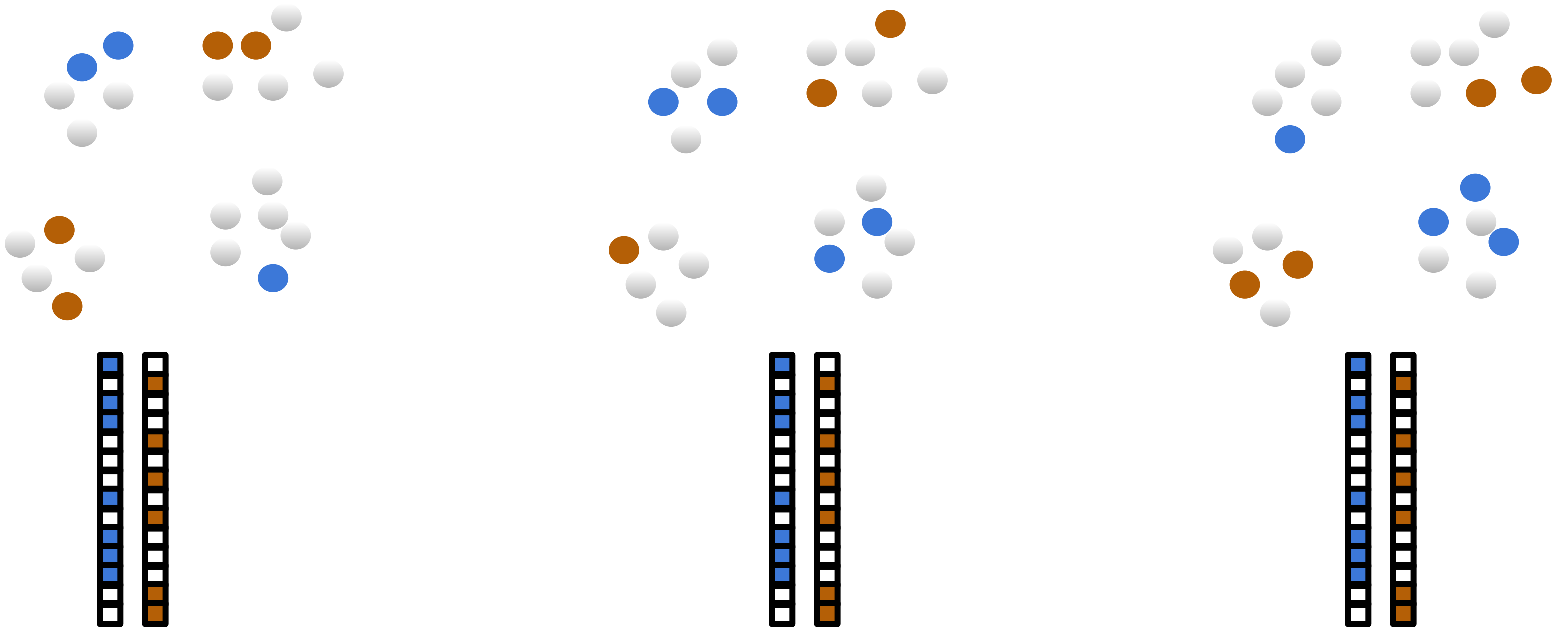
FlyNN-FL:

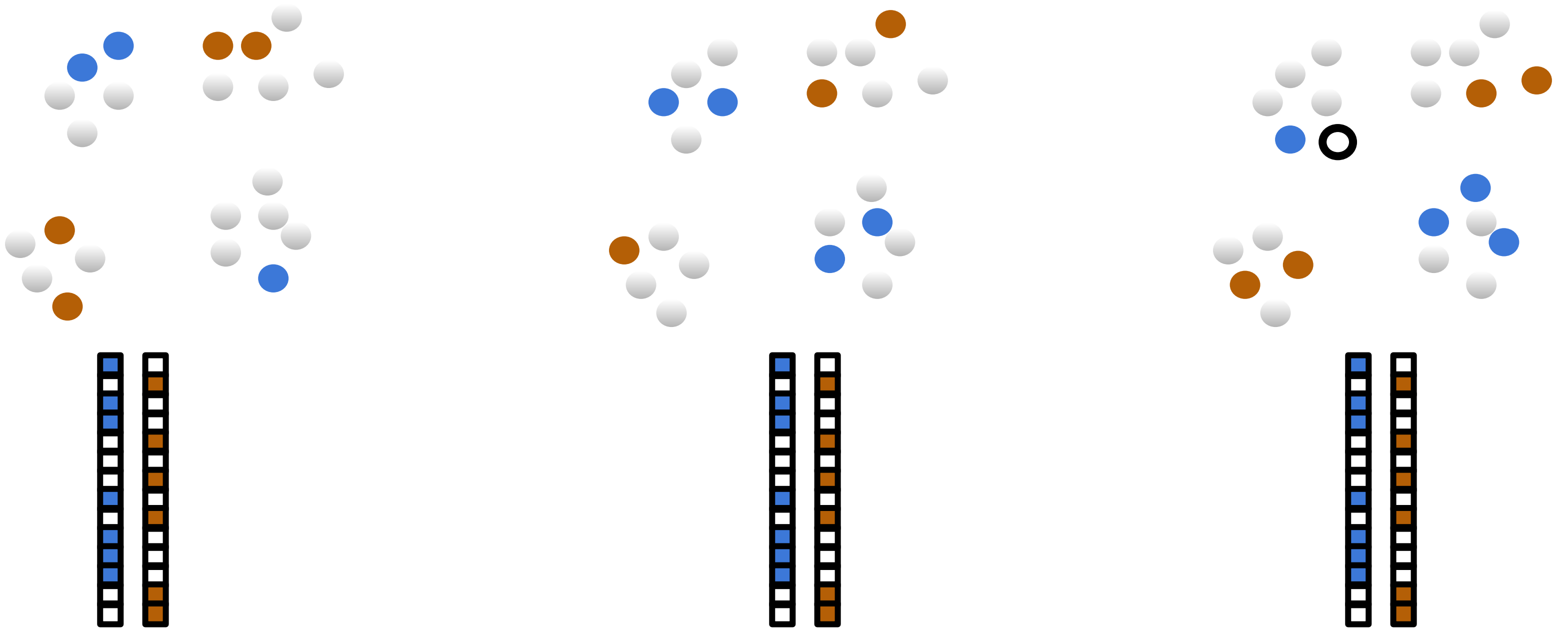
Federated Nearest Neighbor Classifier

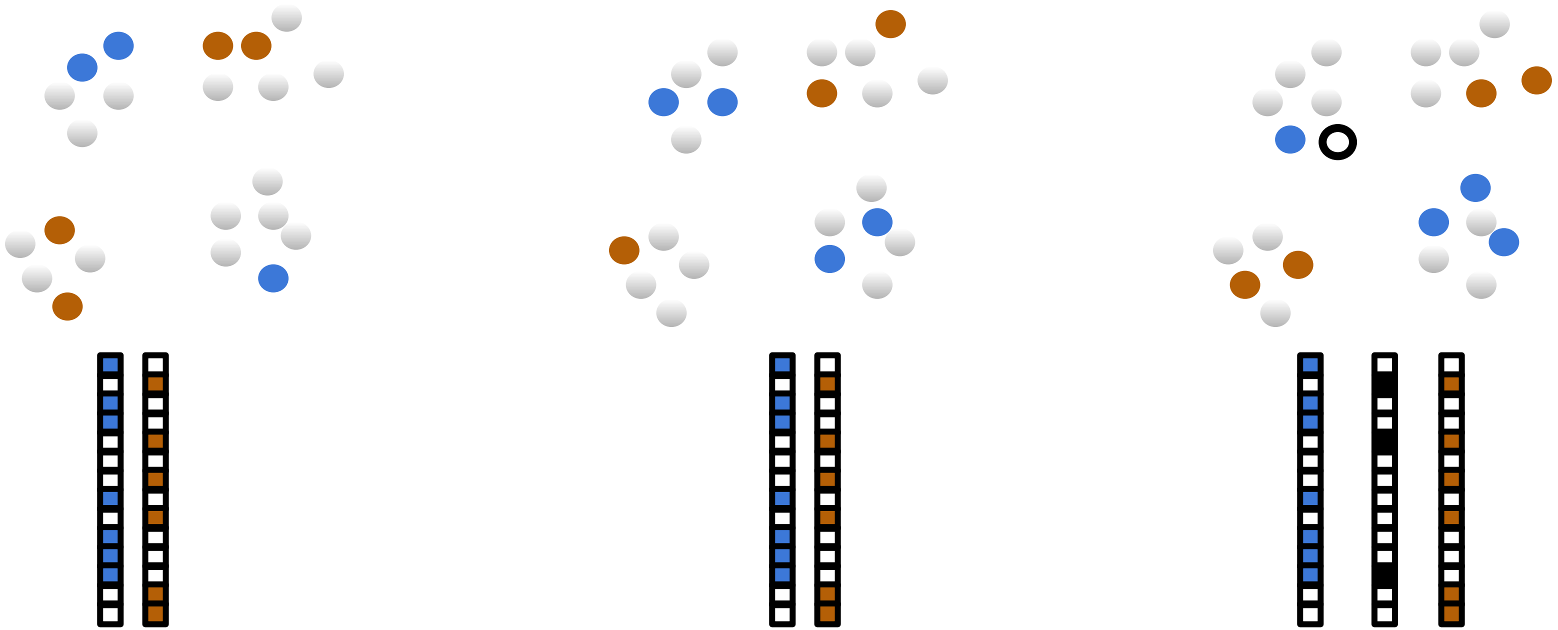


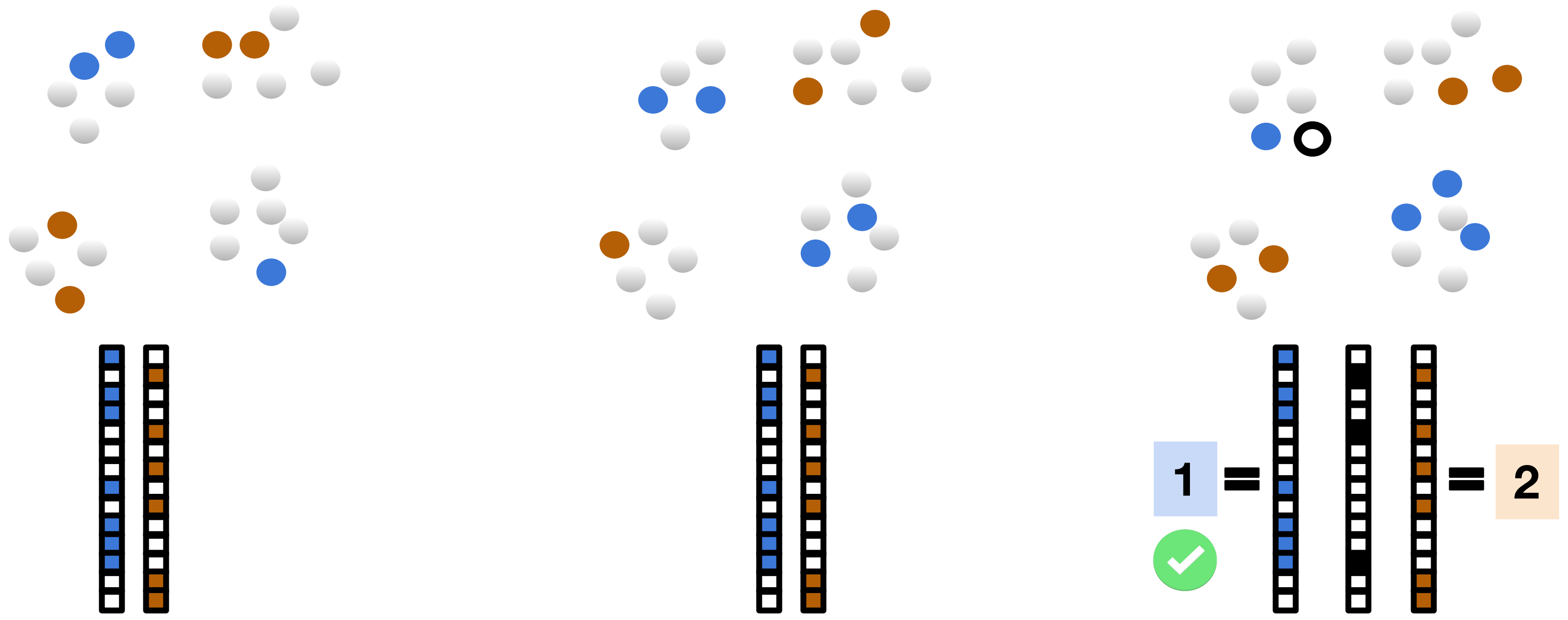


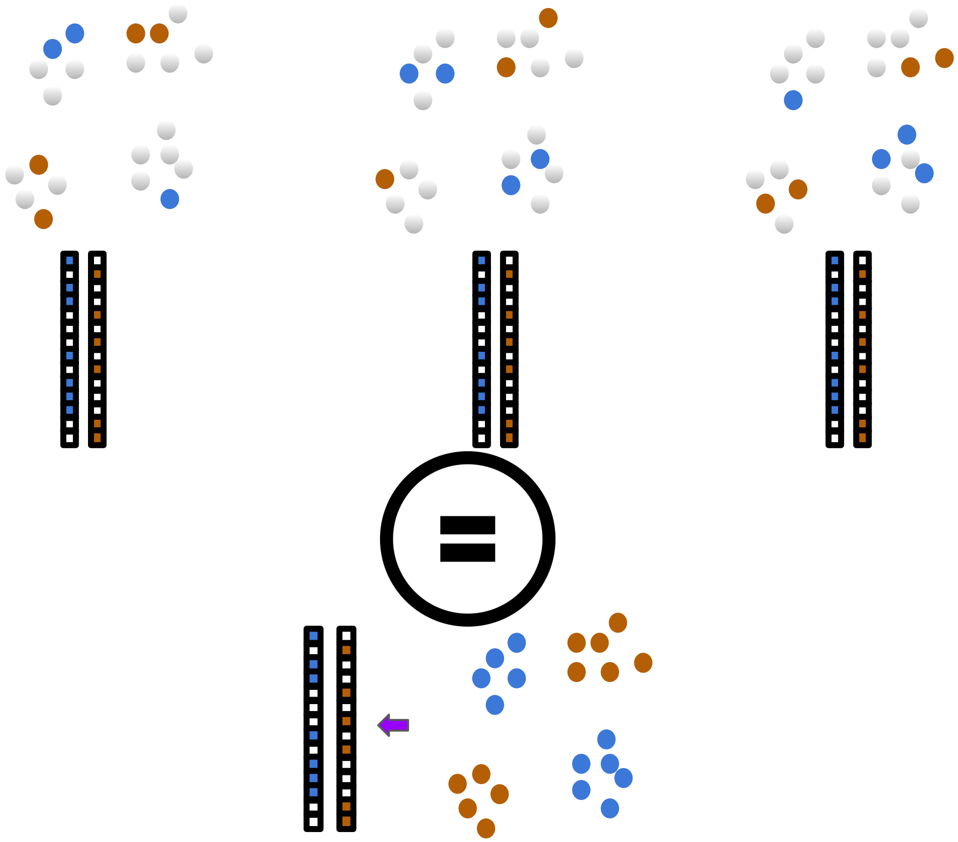


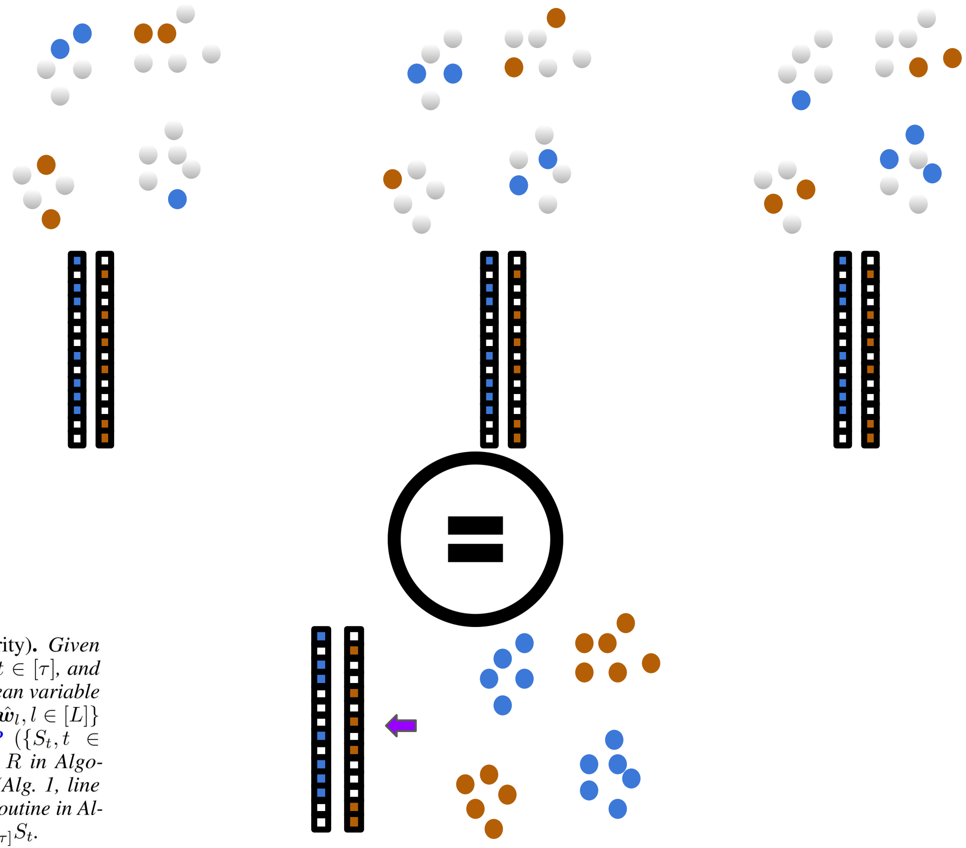




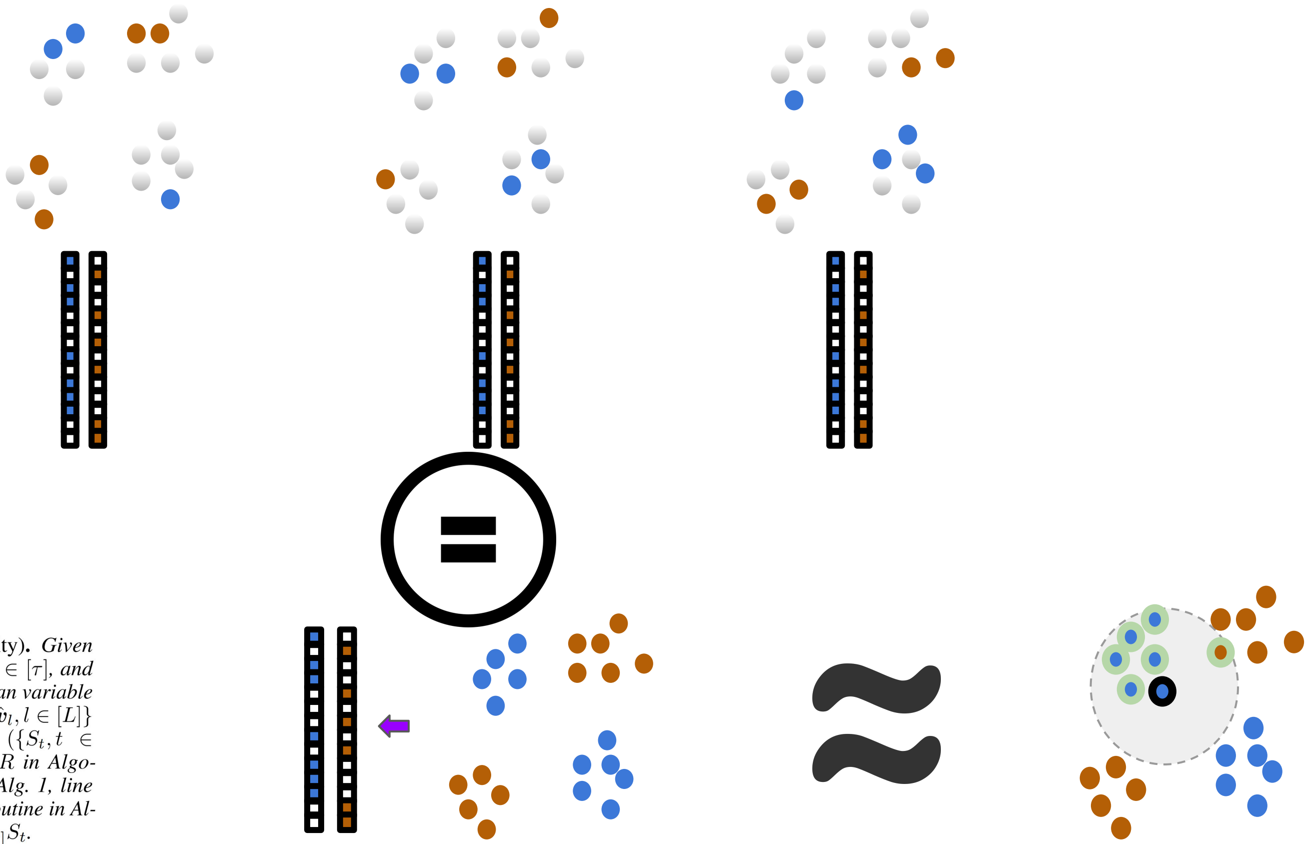



1
2

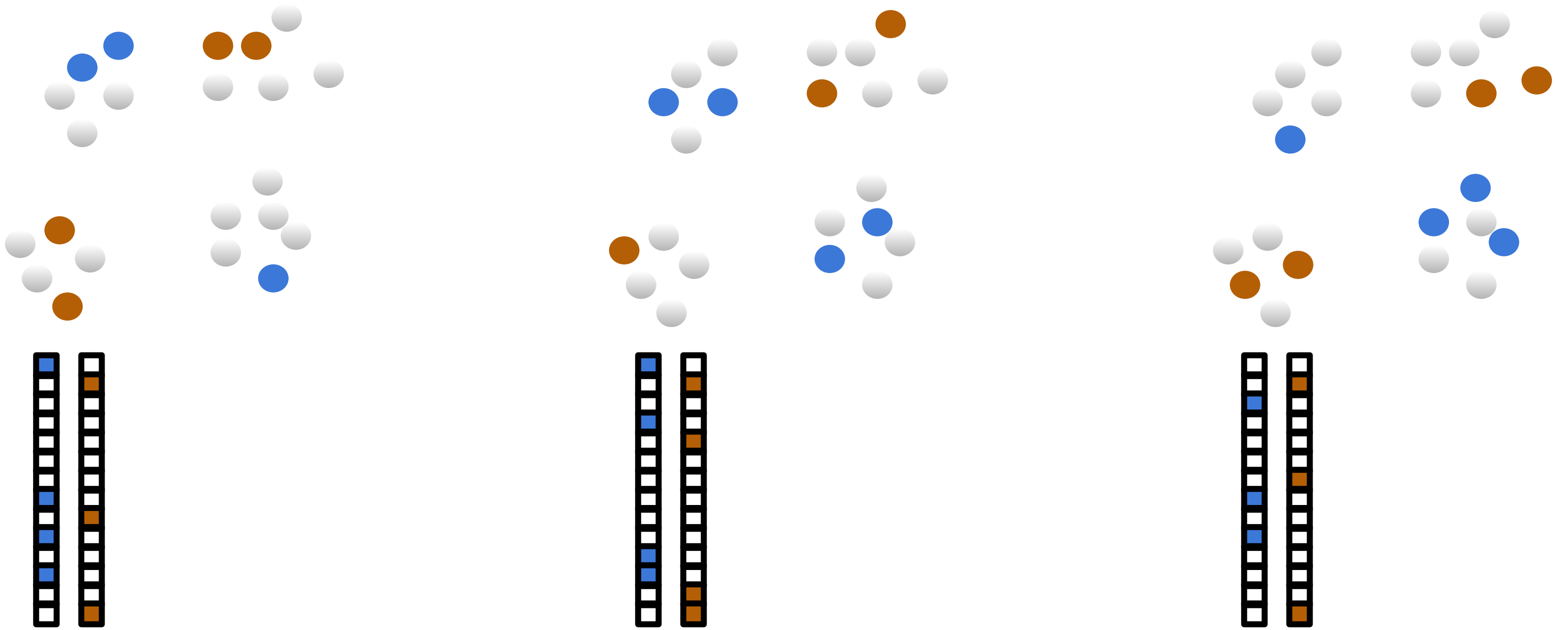


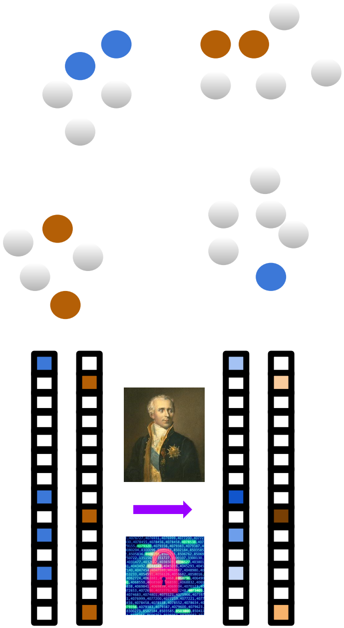


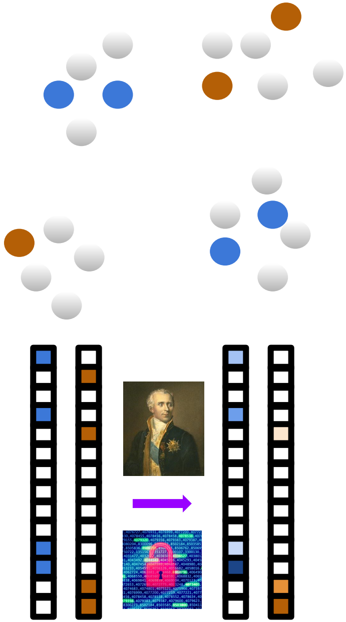
Theorem 4 (Non-private Federated training parity)**.** *Given training sets $S_t \subset \mathbb{R}^d \times [L]$ on each party $V_t, t \in [\tau]$, and a* $\texttt{FlyNN}$ *configured as in Lemma 1, if the boolean variable* $\texttt{IS\_DP}$ *is False, then the per-party final* $\texttt{FlyNN}$ $\{\hat{\boldsymbol{w}}_l, l \in [L]\}$ *(Alg. 2, line 9) output by Train*$\texttt{FlyNN}$*FLDP* $(\{S_t, t \in [\tau]\}, m, s, \rho, \gamma, \texttt{IS\_DP}, \epsilon, T)$ *with random seed R in Algorithm 2 is equal to the* $\texttt{FlyNN}$ $\{\boldsymbol{w}_l, l \in [L]\}$ *(Alg. 1, line 8) output by Train*$\texttt{FlyNN}$ (S, m, s, ρ, c, R) *subroutine in Algorithm 1 with the pooled training set $S = \cup_{t\in[\tau]} S_t$.*



Theorem 4 (Non-private Federated training parity)**.** *Given training sets $S_t \subset \mathbb{R}^d \times [L]$ on each party $V_t, t \in [\tau]$, and a* `FlyNN` *configured as in Lemma 1, if the boolean variable* `IS_DP` *is False, then the per-party final* `FlyNN` $\{\hat{\boldsymbol{w}}_l, l \in [L]\}$ *(Alg. 2, line 9) output by Train*`FlyNN`*FLDP* $(\{S_t, t \in [\tau]\}, m, s, \rho, \gamma, \texttt{IS\_DP}, \epsilon, T)$ *with random seed R in Algorithm 2 is equal to the* `FlyNN` $\{\boldsymbol{w}_l, l \in [L]\}$ *(Alg. 1, line 8) output by Train*`FlyNN` (S, m, s, ρ, c, R) *subroutine in Algorithm 1 with the pooled training set $S = \cup_{t \in [\tau]} S_t$.*







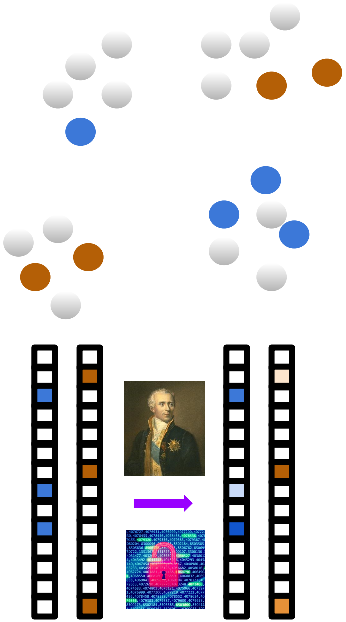


Image Source: Wikimedia Commons

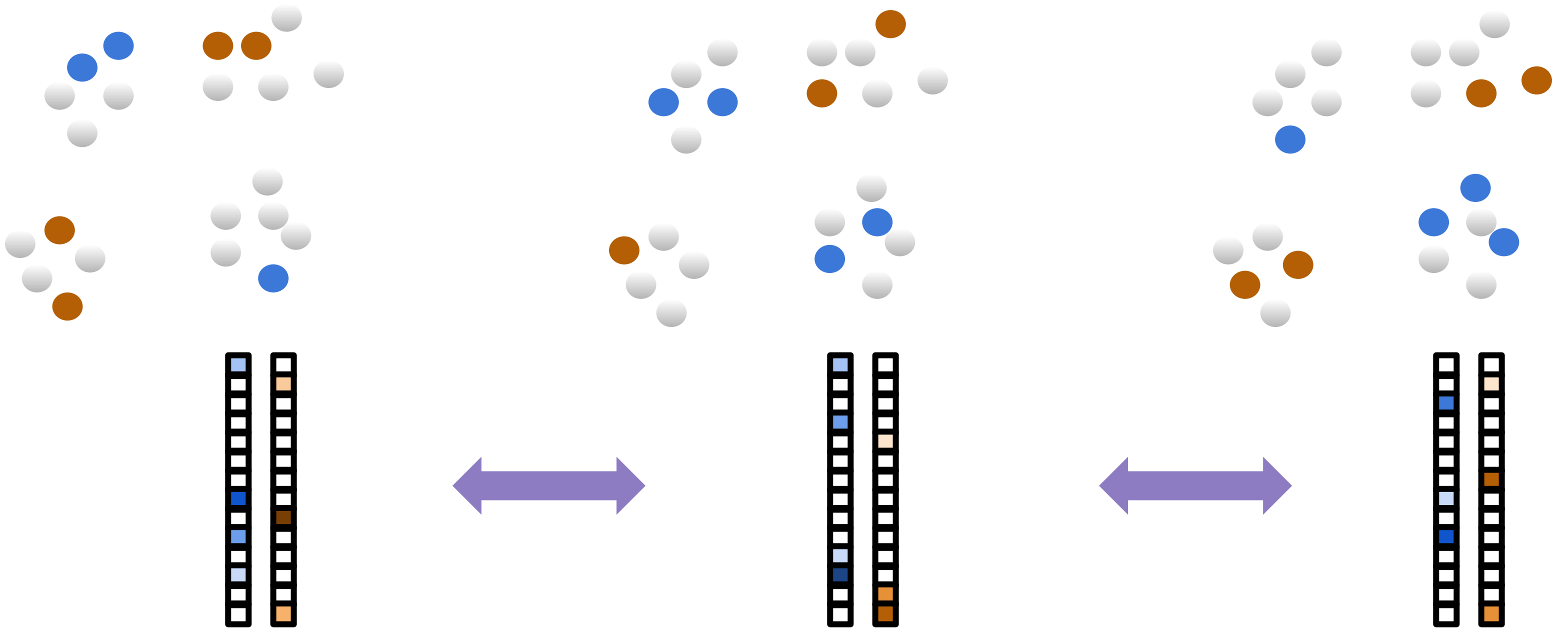


Image Source: Wikimedia Commons

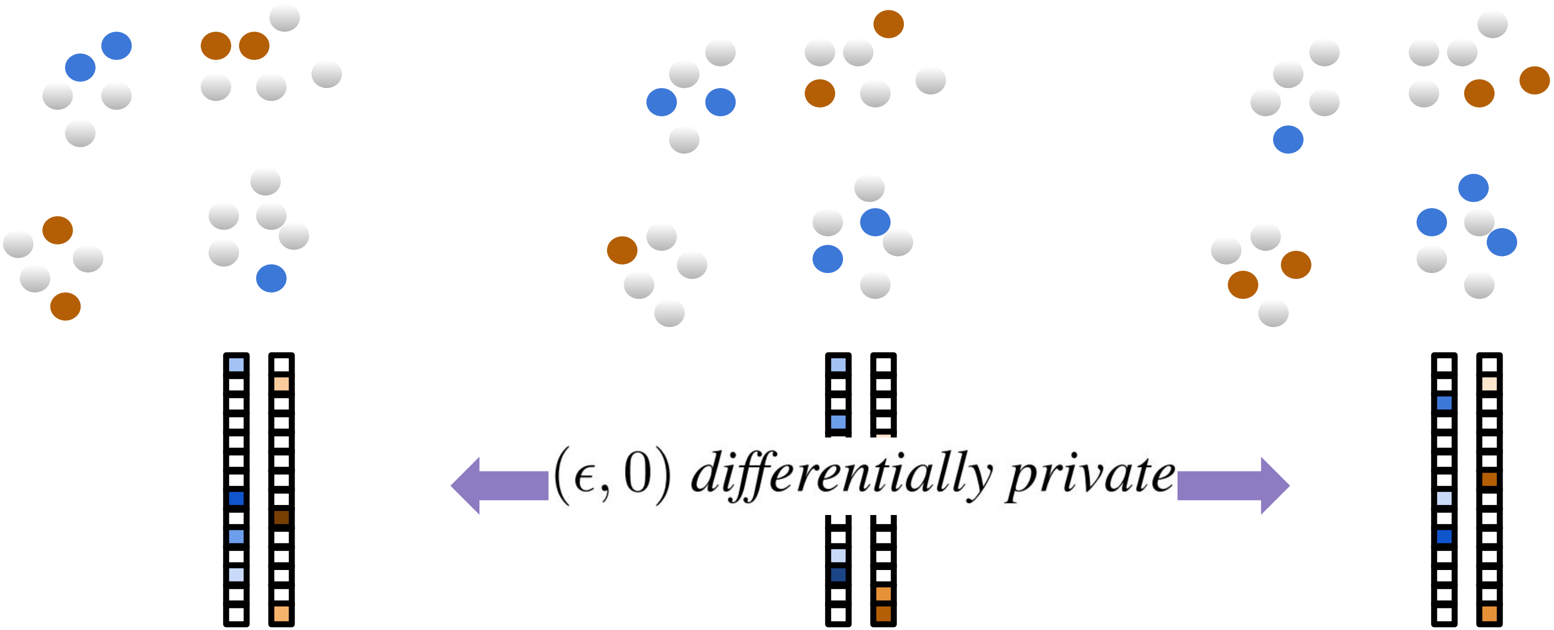
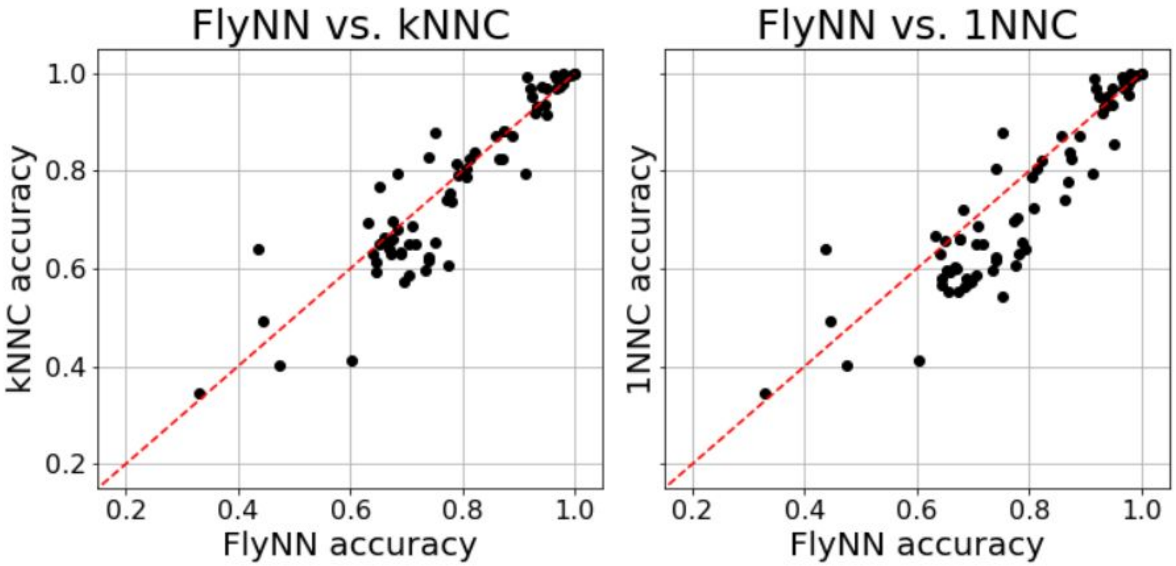
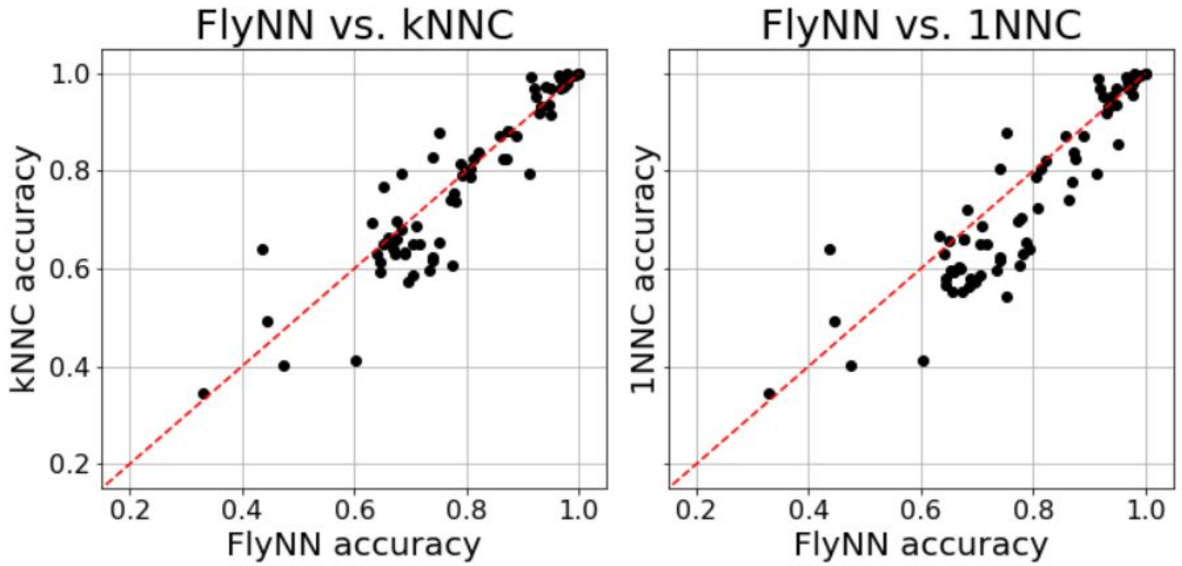


Image Source: Wikimedia Commons

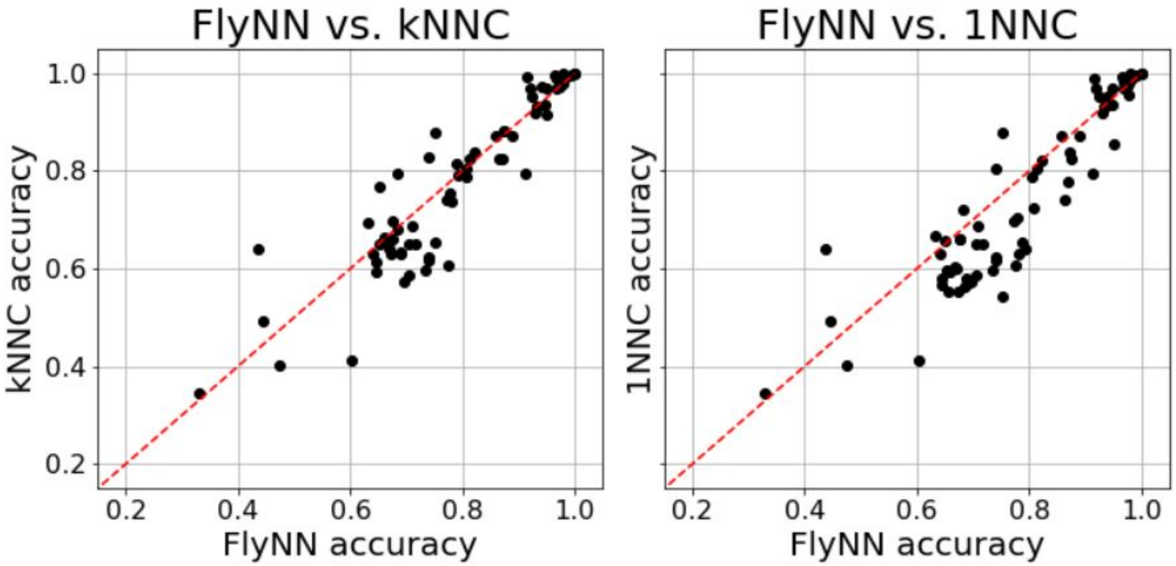
Experiments



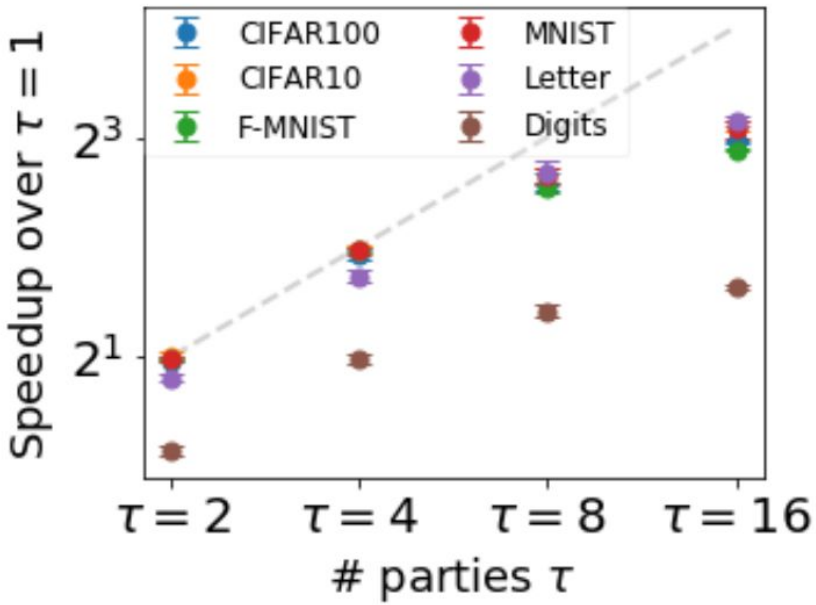
METHOD	(i) FRAC.	(ii) W/T/L	(iii) IMP.	(iv) TT	(v) WSRT
kNNC	0.55	39/2/30	0.35%	5.30E-2	7.63E-2
1NNC	0.66	47/2/22	2.36%	1.55E-5	2.81E-5



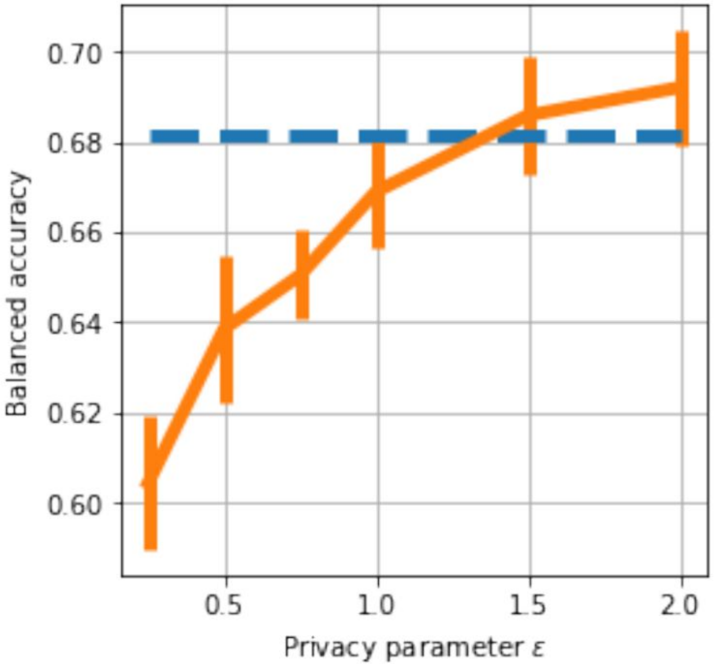
METHOD	(i) FRAC.	(ii) W/T/L	(iii) IMP.	(iv) TT	(v) WSRT
kNNC	0.55	39/2/30	0.35%	5.30E-2	7.63E-2
1NNC	0.66	47/2/22	2.36%	1.55E-5	2.81E-5



METHOD	(i) FRAC.	(ii) W/T/L	(iii) IMP.	(iv) TT	(v) WSRT
kNNC	0.55	39/2/30	0.35%	5.30E-2	7.63E-2
1NNC	0.66	47/2/22	2.36%	1.55E-5	2.81E-5

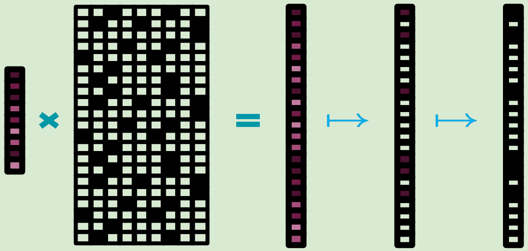


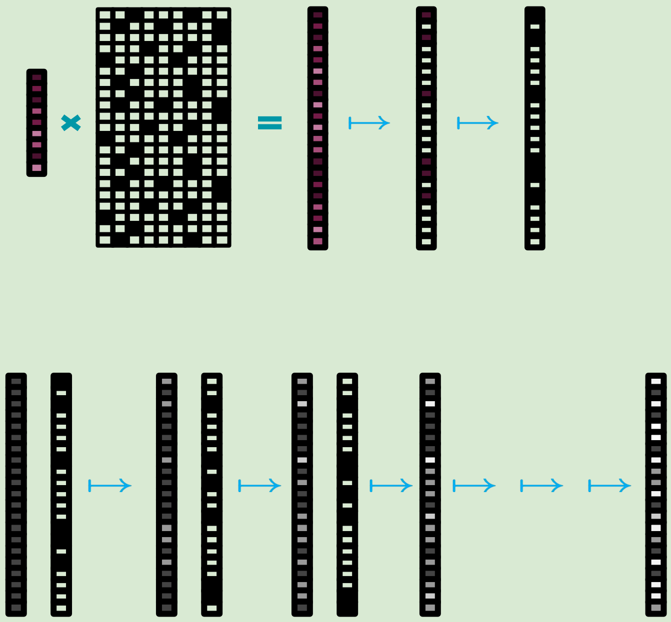
CIFAR100
CIFAR10
F-MNIST
MNIST
Letter
Digits
2^3
2^1
Speedup over $\tau = 1$
$\tau = 2$
$\tau = 4$
$\tau = 8$
$\tau = 16$
parties τ

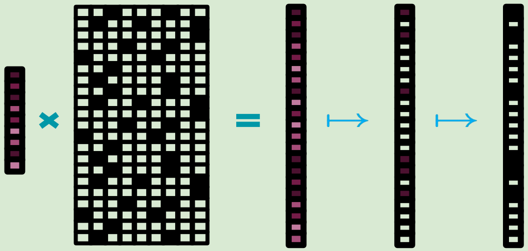

0.70
0.68
0.66
0.64
0.62
0.60
Balanced accuracy
0.5
1.0
1.5
2.0
Privacy parameter ε

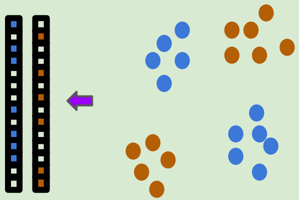
Additional results

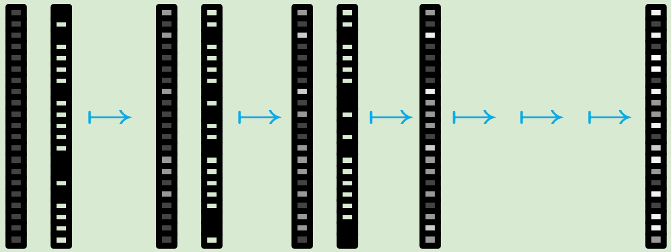
- ➔ Hyper-parameter dependence
- ➔ Ablation and comparison to other methods
- ➔ Extensive evaluation on synthetic data

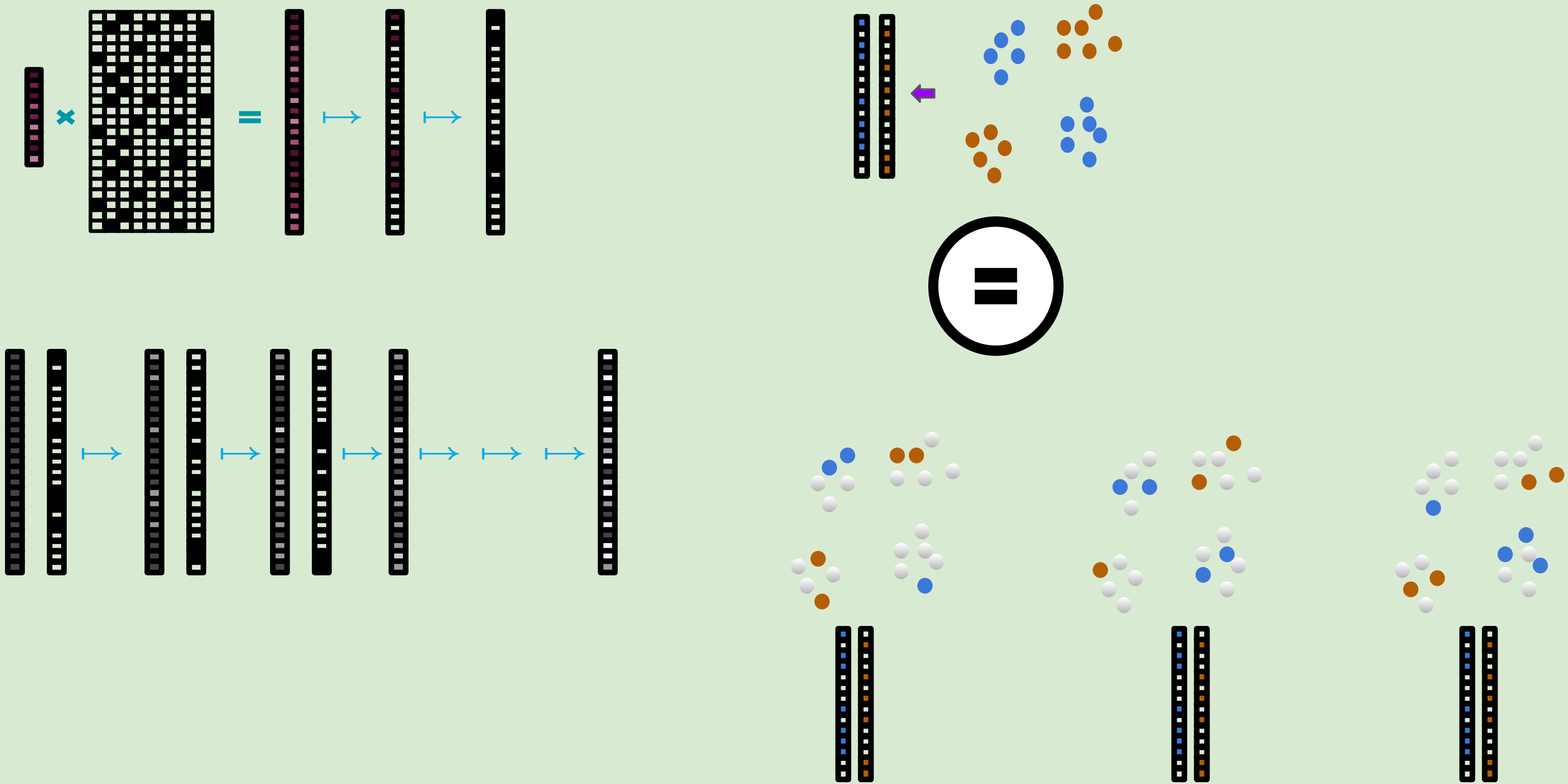


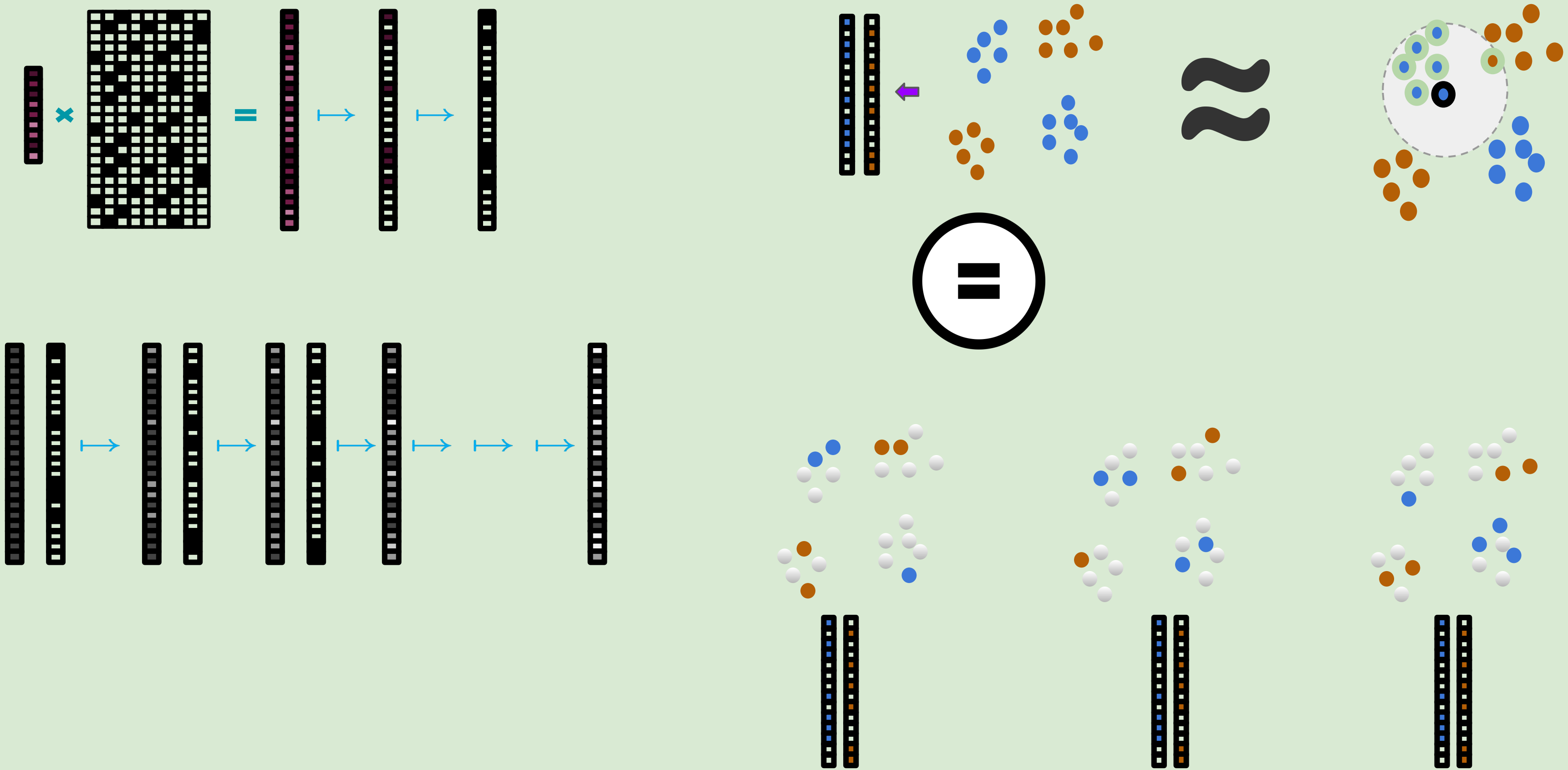












github.com/rithram/flynn

p.ram@acm.org


README.md

Code for "Federated Nearest Neighbor Classification with a Colony of Fruit-Flies"

Table of contents:

- Setting up environment
  - Prerequisites
  - Installing requirements
- Running experiments
  - Running experiments on hyper-parameter dependence
  - Running comparison to baselines with synthetic data
  - Running comparison to baselines with OpenML data
  - Scaling with number of parties
  - Differential privacy effect

Setting up environment

This section details the setup of the compute environment for executing the provided scripts.

Prerequisites

- `python3.8`
- `pip`
- `virtualenv`