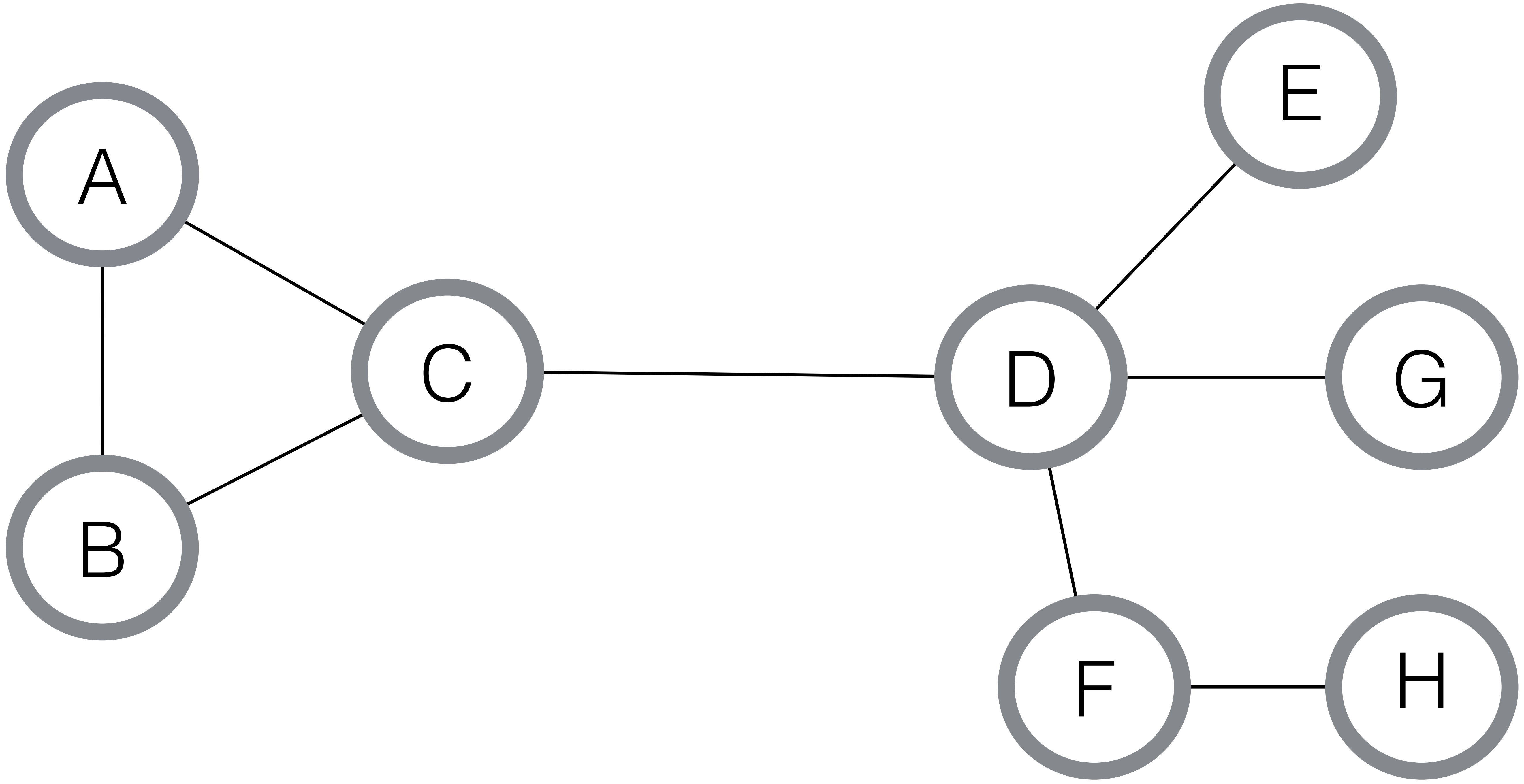
Lecture 2: Spatial Graph Convolution and its Theoretical Performance on Simple Random Data, Part 1

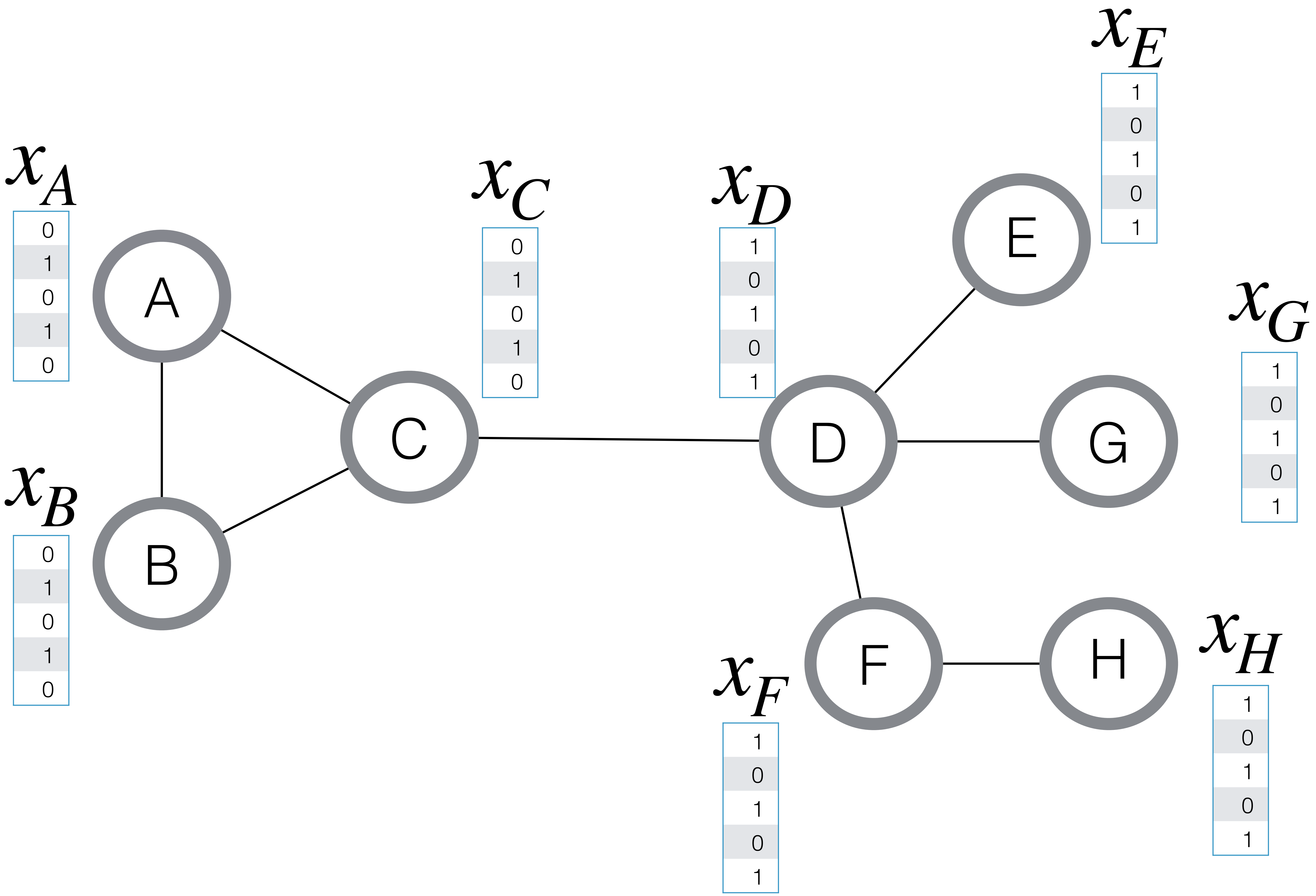
Kimon Fountoulakis

UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Graphs

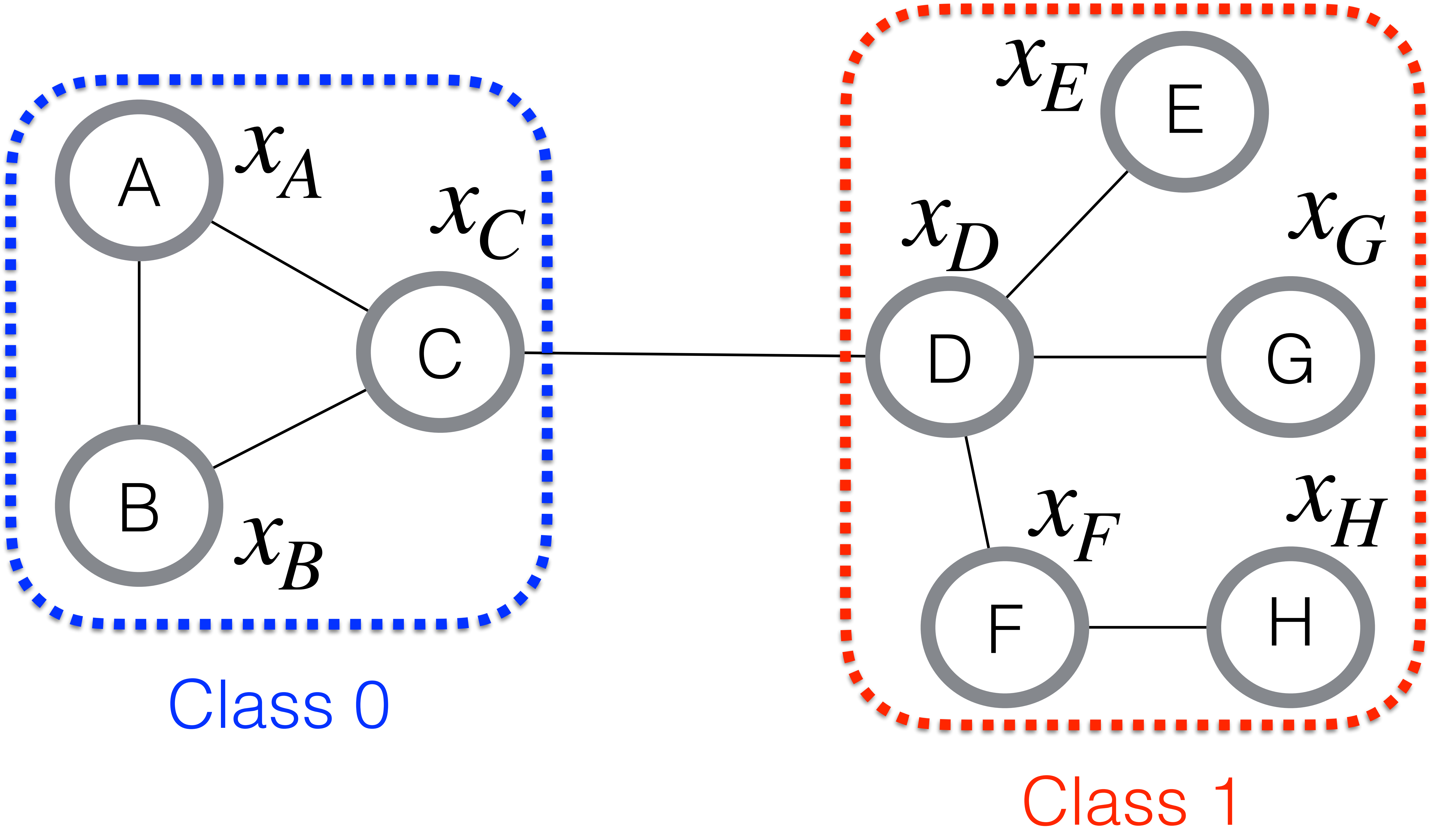


Graphs + features



- x_i is the feature vector for node i

Node classification

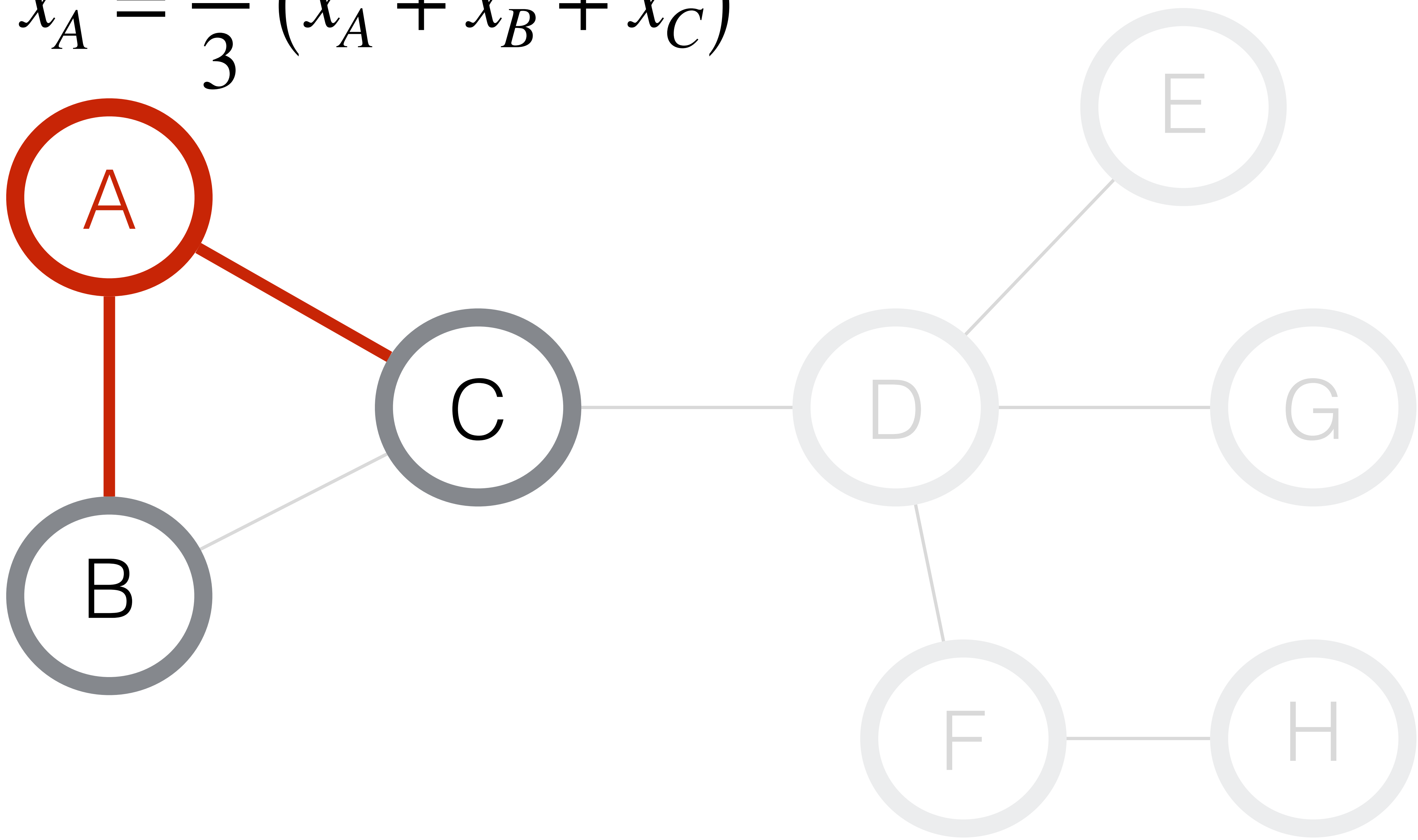


- x_i is the feature vector for node i

Vanilla Graph Convolution Network (GCN): aggregation function

$$x'_A = \frac{1}{3}\left(x_A + x_B + x_C\right)$$

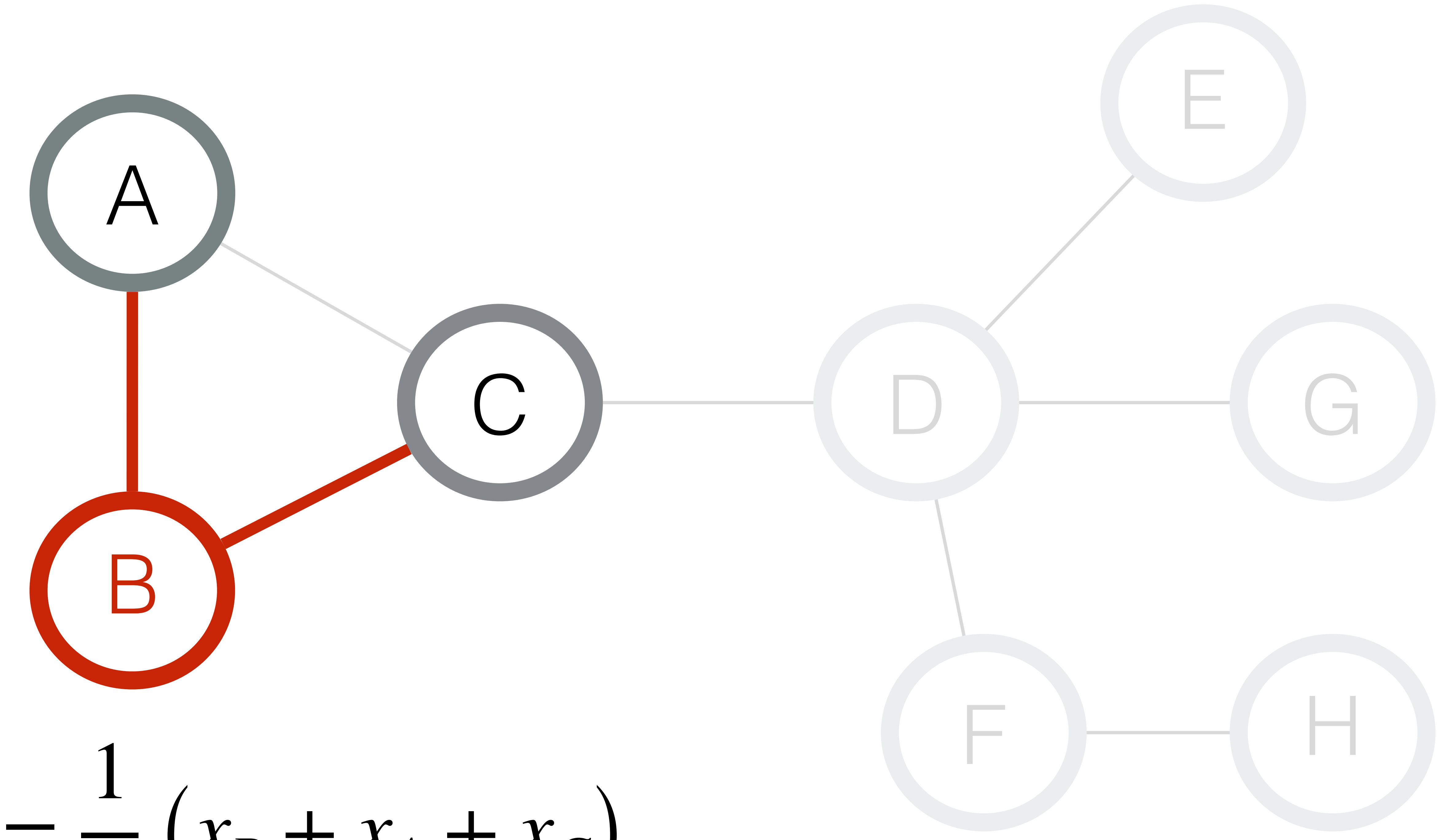
Node A gets messages x_B and x_C and combines it with its own x_A uniformly.



T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): aggregation function

Node B gets messages x_A and x_C and combines it with its own x_B uniformly

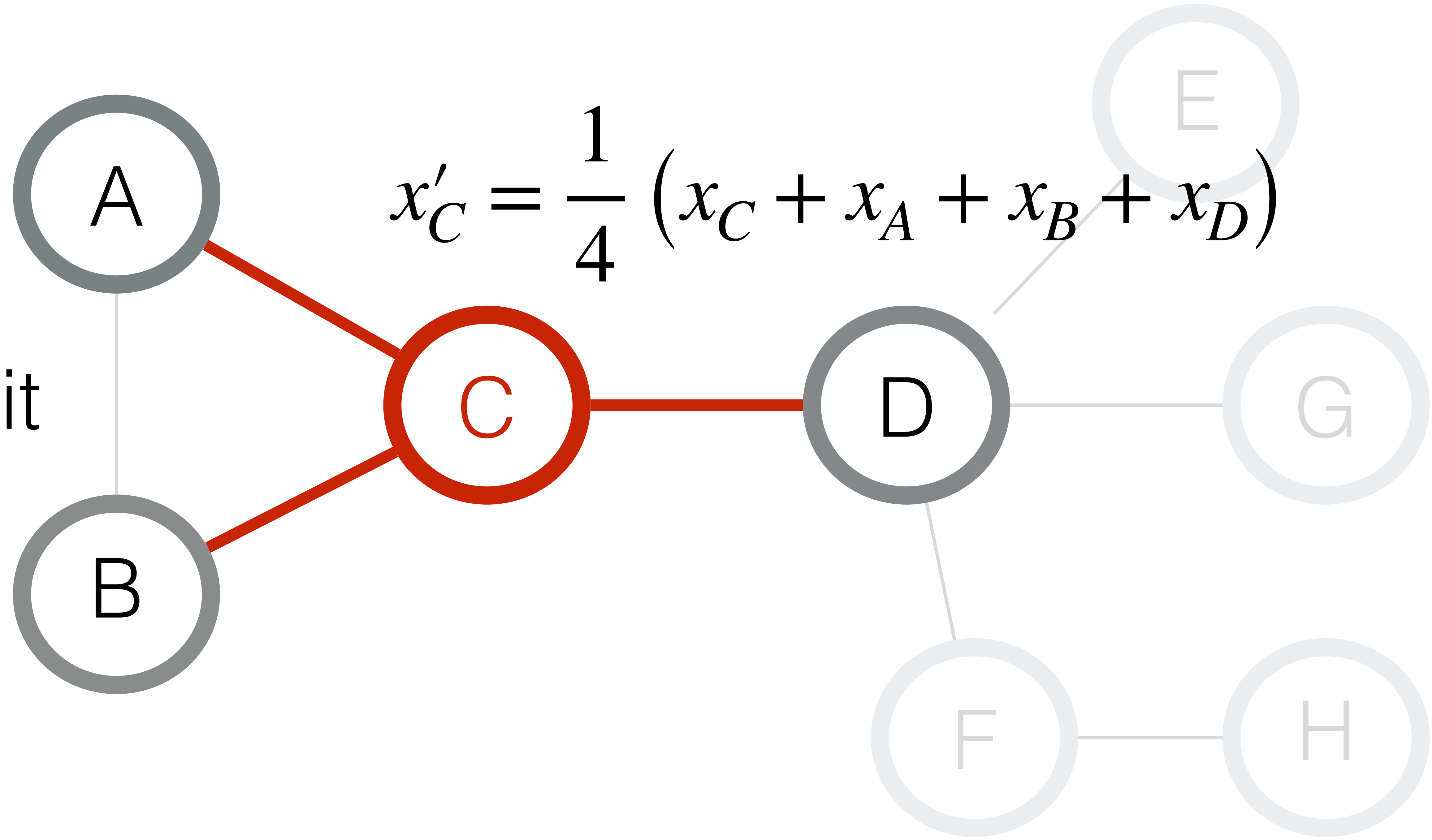


$$x'_B = \frac{1}{3}\left(x_B + x_A + x_C\right)$$

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): aggregation function

Node C gets messages x_A, x_B and x_D, and combines it with its own x_C uniformly



$$x'_C = \frac{1}{4}\left(x_C + x_A + x_B + x_D\right)$$

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): aggregation function

$$X'_i := \frac{1}{D_{ii}} \sum_{i=1}^{n} A_{ij} X_j$$

Convolved data for node i

Degree of node i

Adjacency matrix

Data For node j

- A component of A is equal to 1 if two nodes are connected with an edge
- D is a diagonal matrix where each component shows the number of neighbors of a node

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): aggregation function in matrix form

$$X' := D^{-1}AX$$

Convolved data Degree matrix Adjacency matrix

- A component of A is equal to 1 if two nodes are connected with an edge
- D is a diagonal matrix where each component shows the number of neighbors of a node

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): learning parameters

$$X'W := D^{-1}AXW$$

- Learning matrix W. It's value are decided by minimizing a loss function.

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): activation

$$\sigma(X'W) := \sigma(D^{-1}AXW)$$

-Activation function σ. Examples include $\sigma(y) := \max(y, 0)$ or $\sigma(y) := \text{sigmoid}(y) = 1/(1 + e^{-y})$ which squeezes values in [0,1].

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Vanilla Graph Convolution Network (GCN): multiple layers

Example: 3-layer GCN

$$X' := \underbrace{\sigma_3(D^{-1}A\,\underbrace{\sigma_2(D^{-1}A\,\underbrace{\sigma_1(D^{-1}AXW_1)}_{\text{layer 1}}\,W_2)}_{\text{layer 2}}\,W_3)}_{\text{layer 3}}$$

T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks, ICLR 2017

Let's keep things simple

$$x' = D^{-1}AXw$$

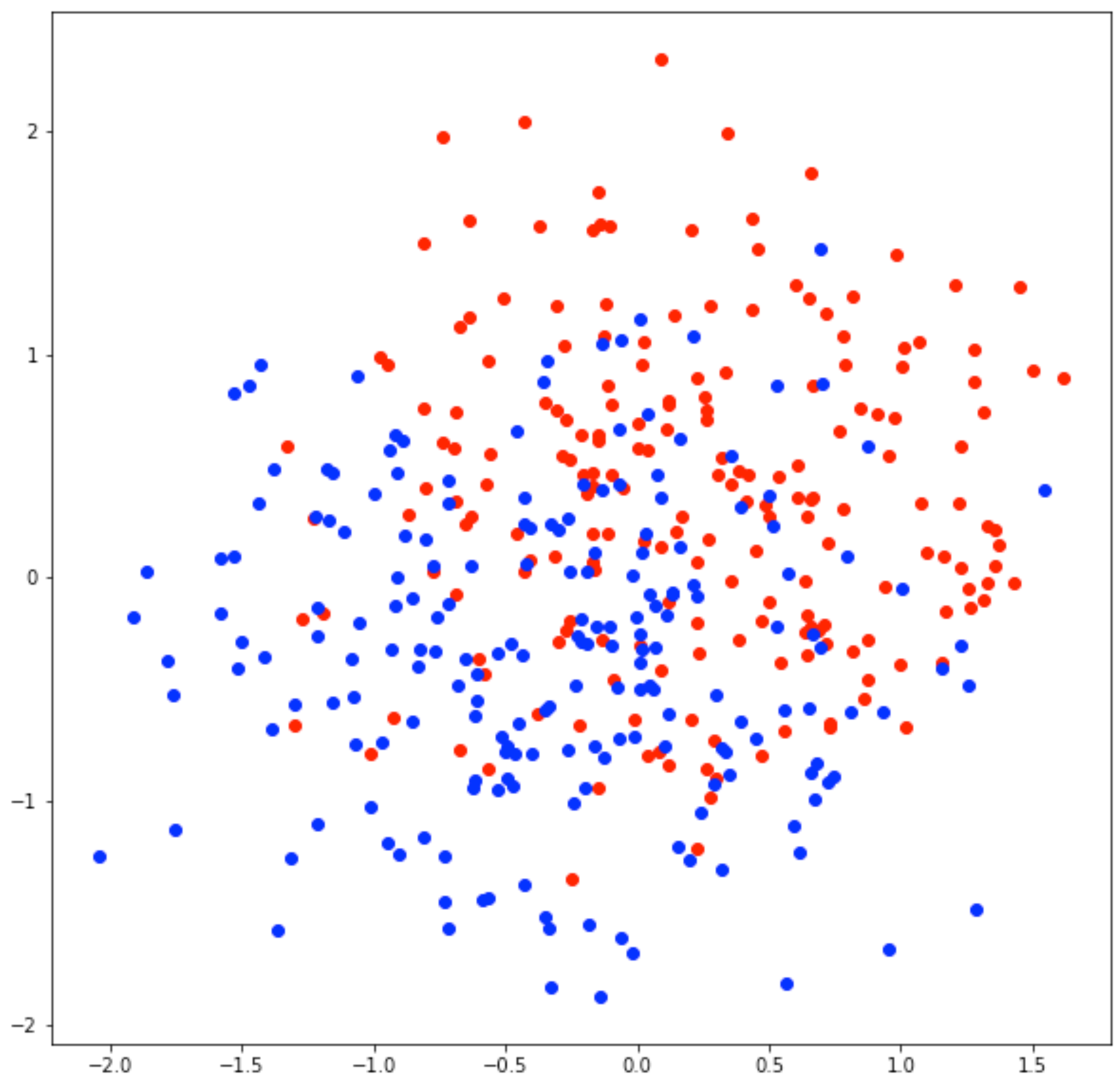
- w is a vector of length equal to the number of features

- x' is a vector of length equal to the number of nodes, indicates predicted class membership

We want to answer the following for linear classifiers

- Does graph convolution of the data help generalization?
- Which graphs are good graphs and which are not?
- What if we test on a graph that comes from a distribution with different parameters?

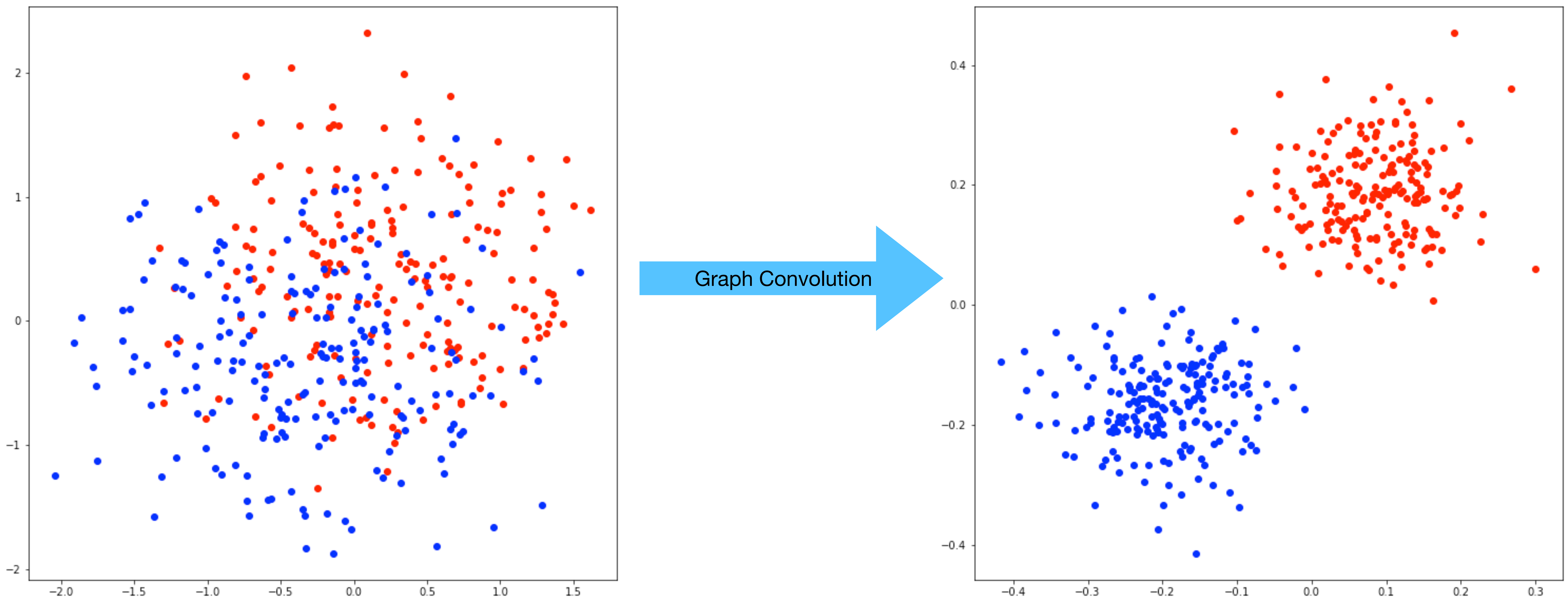
What can graph convolution do?



Original Data

- Consider distributions with 2D features
- We cannot separate this data linearly
- Can graph convolution help?

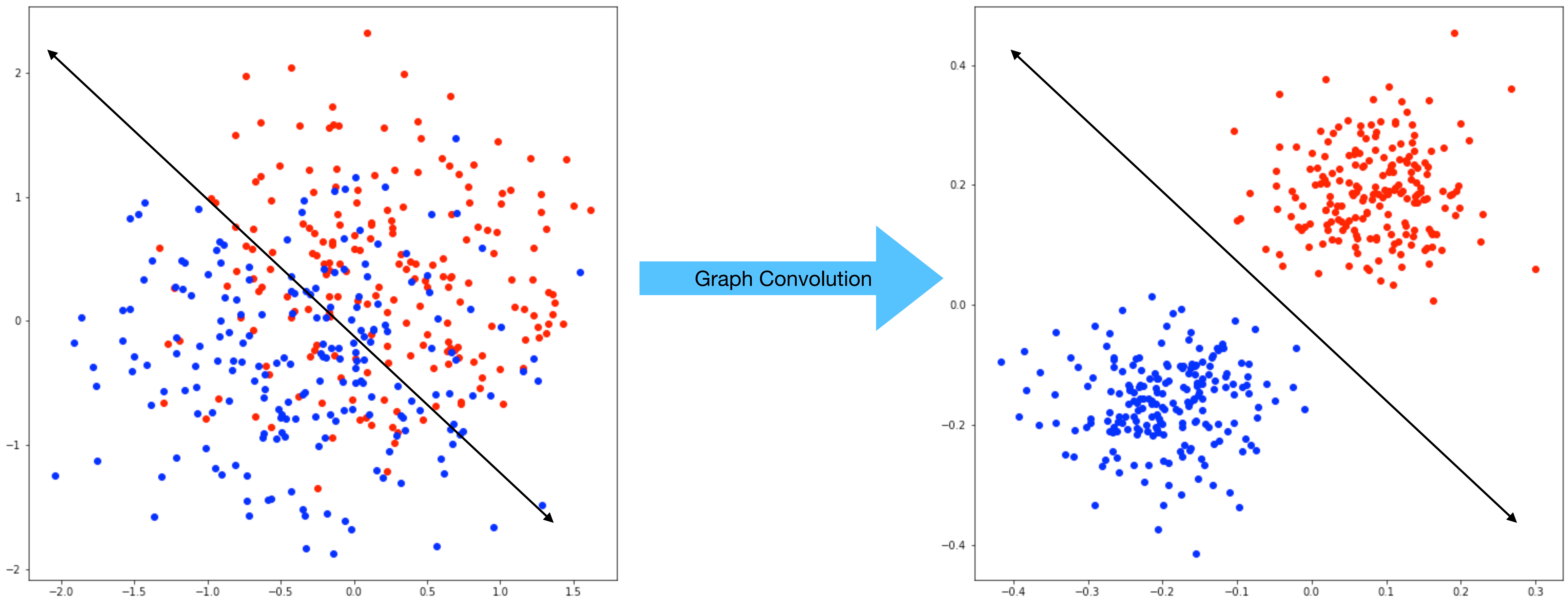
What can graph convolution do?



Original Data

After graph convolution

What can graph convolution do?



Original Data

After graph convolution

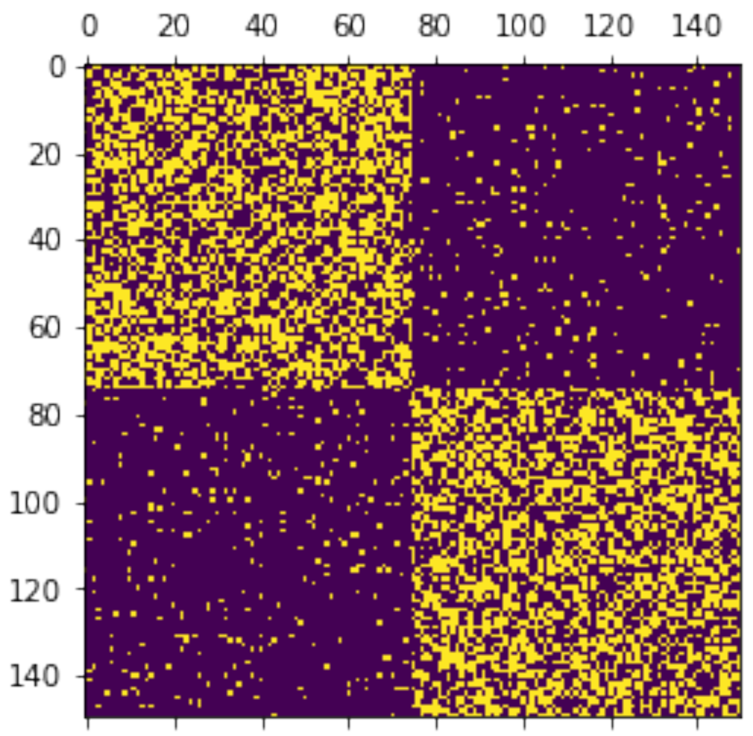
Graph convolution makes the data linearly separable

Data model: contextual stochastic block model

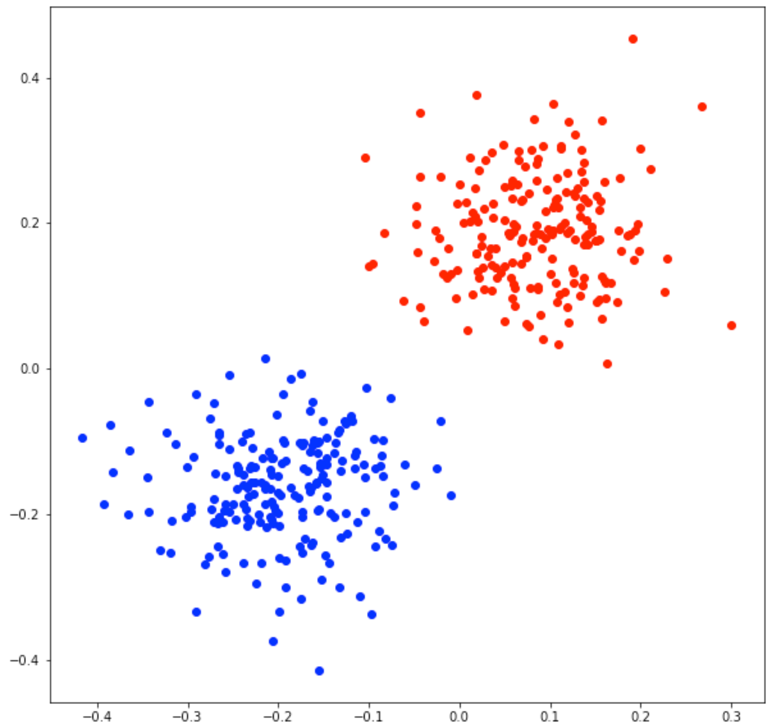
- Two-component balanced Gaussian Mixture Model (GMM) coupled with a Stochastic Block Model (SBM)

$$A \sim SBM(p, q)$$

$$\mathbb{P}(A_{ij} = 1) = \begin{cases} p & \text{if } i, j \text{ are in the same class} \\ q & \text{otherwise} \end{cases}$$

$A =$ 

$X_i \sim \mathcal{N}(\mu, \sigma^2 I)$ if $i \in C_0$
$X_i \sim \mathcal{N}(\nu, \sigma^2 I)$ if $i \in C_1$



Data alignment

- The classes in the features are aligned with the communities in the graph.

Let's ignore the graph first and make some assumptions

- Let's assume that we are given the distributions of the data

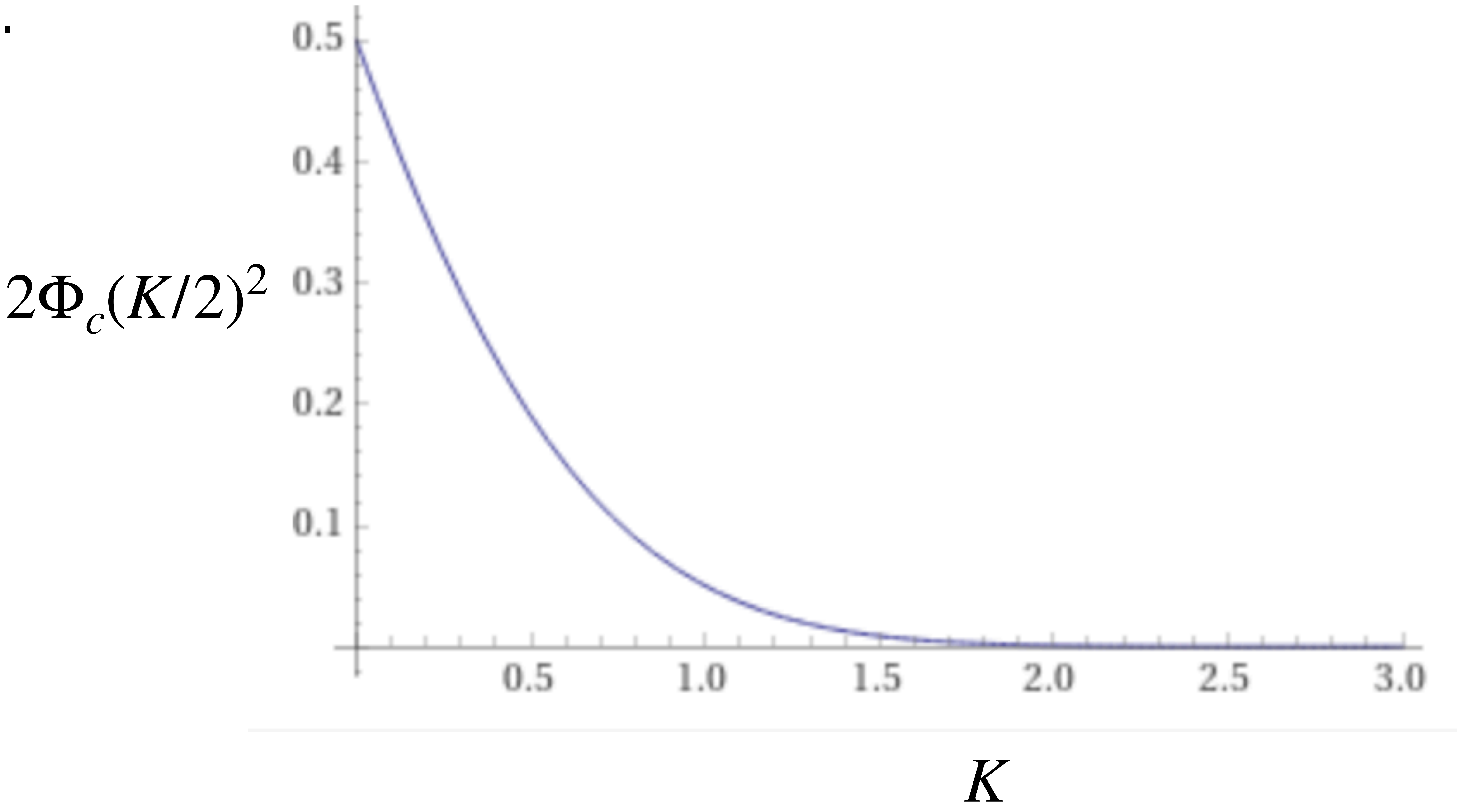
- This is very powerful knowledge because it assumes that we know the means μ, ν of the Gaussians and p, q.

- This allows to make prediction using the optimal Bayes classifier:

$$i^* = \text{argmax}_{i \in \{0,1\}} P(y = i \mid x)$$

What's the performance of the optimal Bayes classifier?

- If the distance between the means $||\mu - \nu|| \leq K\sigma$.
- Then a $2\Phi_c(K/2)^2$ fraction of all data points are misclassified by the Bayes classifier.



What's the performance of the optimal Bayes classifier?

-If the distance between the means $||\mu - \nu|| = \Omega(\sigma\sqrt{\log n})$

-Then there is a multi-layer perceptron (needs more than 2 layers) that classifies all data with high probability.

Graph convolution improves linear separability

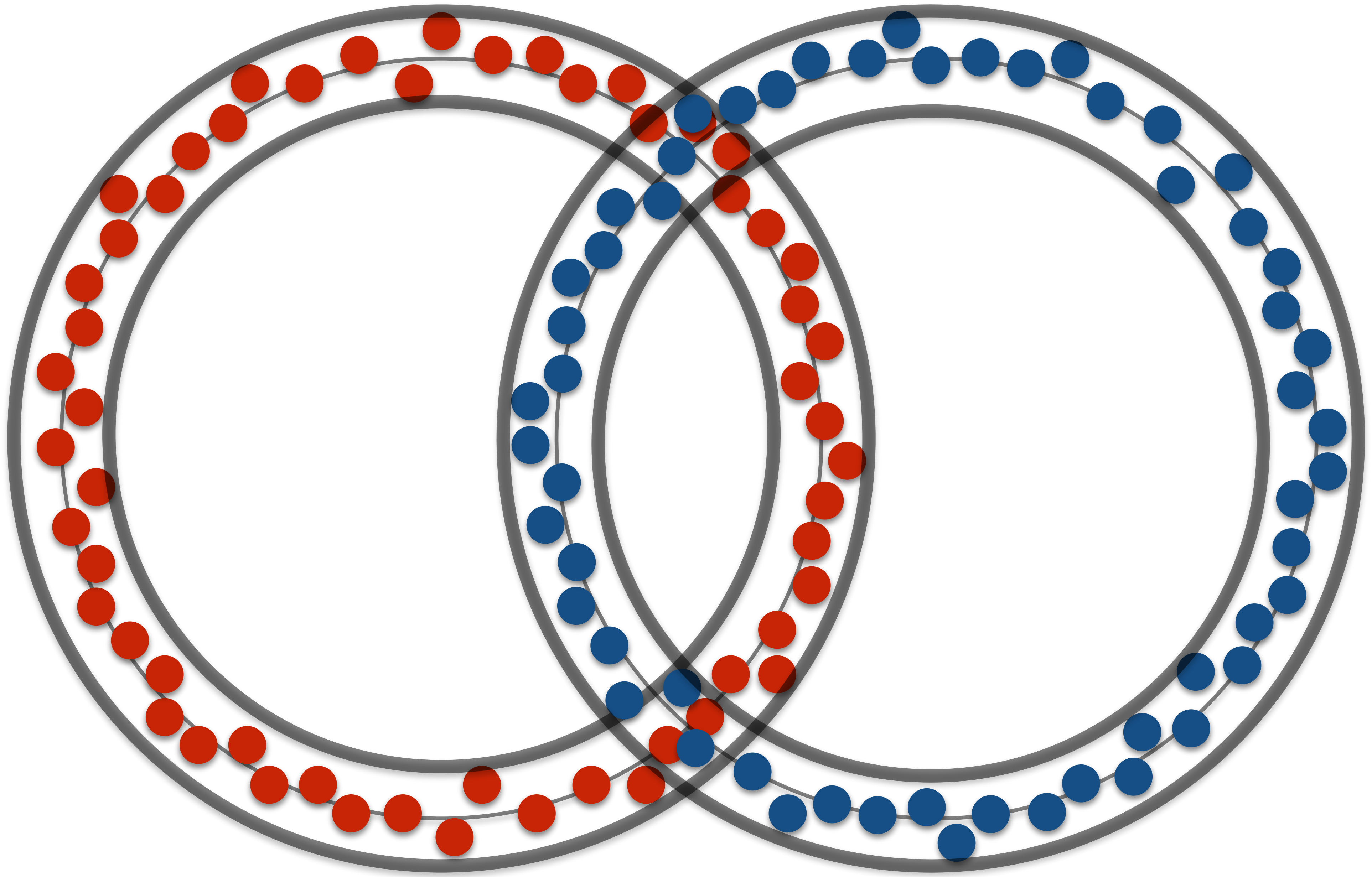
- Without the graph, no hyperplane can separate a binary GMM if means are $\mathcal{O}(\sigma)$ apart, i.e., $\|\mu - \nu\|_2 = \mathcal{O}(\sigma)$
- With graph convolution, this threshold changes to

$$\|\mu - \nu\| = \mathcal{O}\left(\frac{\sigma}{\sqrt{\mathbb{E}[D]}}\right)$$

Expected degree of a node

Proof sketch for no-convolution

Intuitive representation of Gaussian data in high dimensions



$$\|\mu - \nu\|_2 = \mathcal{O}(\sigma)$$

- We can actually show that there will be a constant fraction of misclassified data

Proof sketch for graph convolution

- After graph convolution the means move closer by a factor $\Gamma(p, q) = \dfrac{p-q}{p+q}$
- But the variance is reduced by $\mathbb{E}[D] = \mathcal{O}(n(p+q))$
- Thus the separability threshold changes from $\|\mu - \nu\|_2 = \mathcal{O}(\sigma)$ to

$$\|\mu - \nu\| = \mathcal{O}\left(\frac{\sigma}{\sqrt{\mathbb{E}[D]}}\right)$$

- Then we can show that the hyperplane that passes through the mid-point of the two means separates the data with high probability.

Bounds on training loss

- We use binary cross entropy loss to learn the classifier

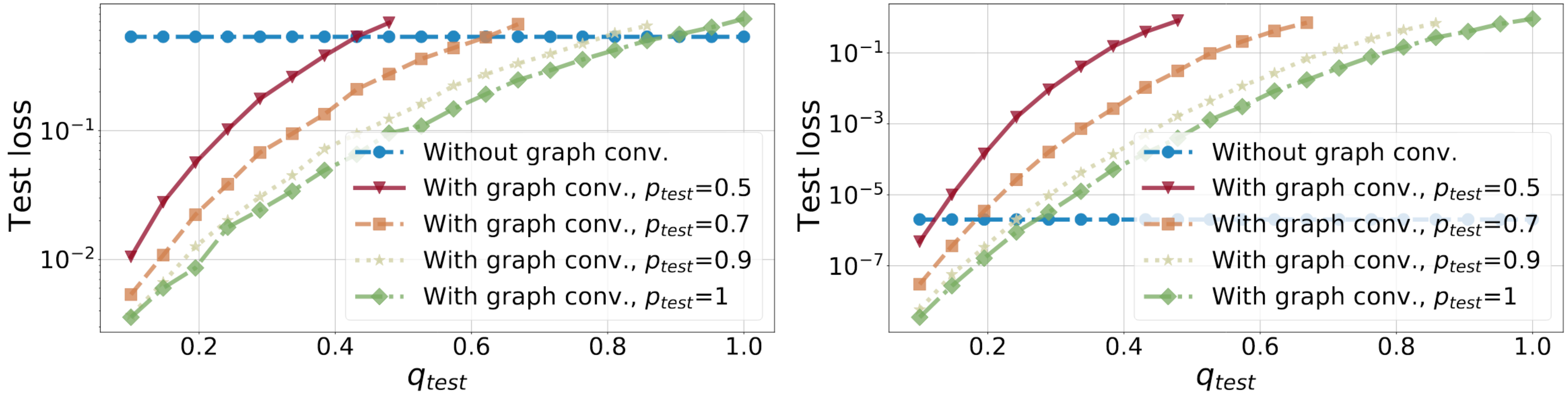
- Without graph convolution, if $||\mu - \nu||_2 = K\sigma$, then the loss is lower bounded by $(2\log 2)\Phi(-K/2)$

- In the regime where the convolved data is separable, the loss decays exponentially $Loss(A, X) \leq C \exp\left(-d||\mu - \nu||\Gamma(p, q)\right)$, where $\Gamma(p, q) = \dfrac{p - q}{p + q}$

Generalization

- If the graph is not sparse, then for any new dataset A, X with different n, p, q, the test loss is bounded above
  $Loss(A, X) \leq C \exp\left(-d\|\mu - \nu\|\Gamma(p, q)\right)$

- Loss increases with inter-class edge probability (noisy graph)



(a) $\|\mu_d - \nu_d\| = 2\sigma$

(b) $\|\mu_d - \nu_d\| = 16\sigma$