# Efficiently Sampling Multiplicative Attribute Graphs Using a Ball-Dropping Process

Hyokun Yun
(work with S.V.N. Vishwanathan)

Departments of Statistics, Purdue University

August 6, 2012

## Overview

**Question**

How to *efficiently* sample graphs from the Multiplicative Attribute Graph Model?

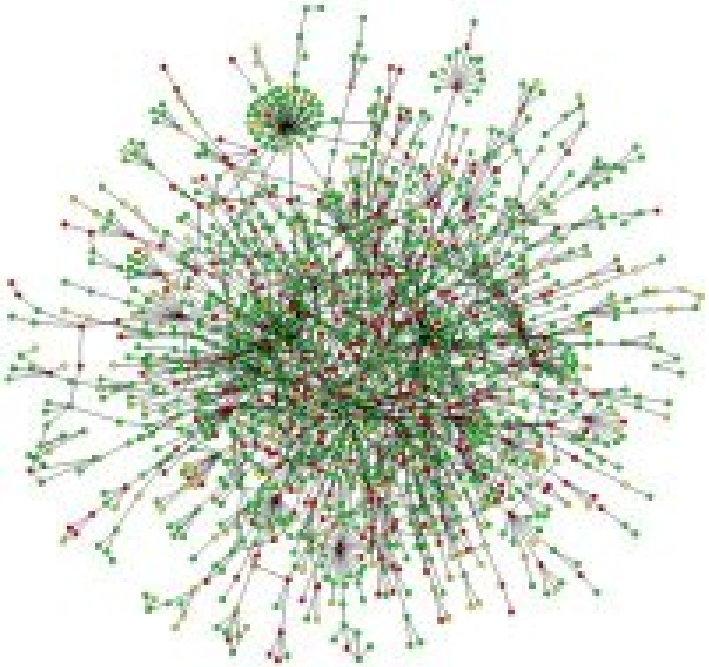
- We introduce the first sub-quadratic sampling algorithm for sampling Multiplicative Attribute Graphs
- **Time complexity:** $O\left(\left(\log_2(n)\right)^3 |E|\right)$ under some mild conditions
  - $n$ : the number of *nodes* in the graph
  - $|E|$ : number of *edges*
- Exploit the close connection between Kronecker Product Graph Model (KPGM) and Multiplicative Attribute Graph Model (MAGM)
- Can sample a graph with 8 million nodes and 20 billion edges in under 6 hours (naïve algorithm will take 93 days)

# Outline

# Outline

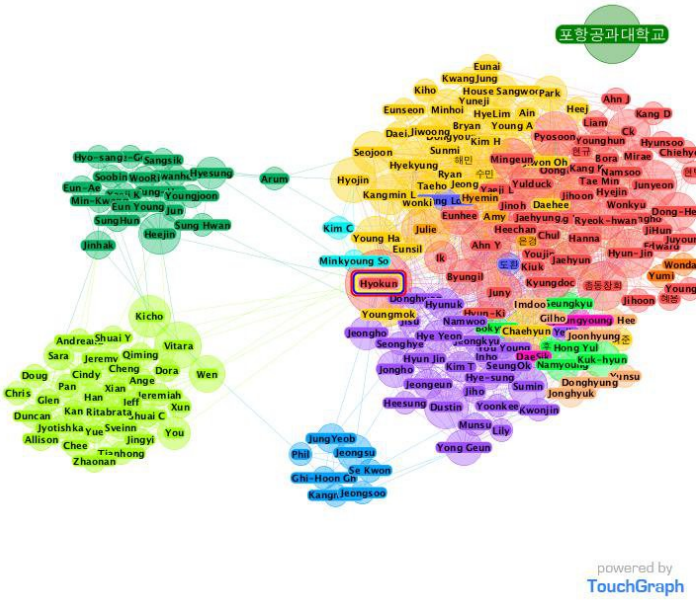
# Motivation

Protein-Protein Interaction Network



H. Jeong, et al., 2001

Social Network (Facebook)



# Why Statistical Models?

- A class of probability distributions which describes
  - the stochastic process that could have generated data
  - *uncertainty* in data
- Traditionally:
  - Continuous Data: Normal Distribution
  - Count Data: Poisson Distribution
- Now, we have Graph Data: ????

We need a probability distribution on the space of graphs!

# Statistical Questions

- Formulate parametric family of distributions $\mathcal{P} = \{P_\theta\}$
- Given $G$ and $\theta$, evaluate the likelihood $L(\theta) = P_\theta(G)$
- Given $G$, find the $\theta$ that maximzes $L(\theta)$
- Given $\theta$, sample $G$ from the model $P_\theta$
  - To assess the goodness-of-fit, you need to generate samples from the estimated parameter $\theta$
  - You may want to generate graphs similar to $G$

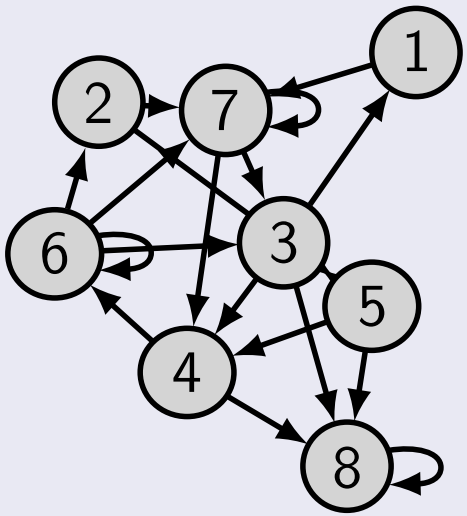
# Statistical Questions

- Formulate parametric family of distributions $\mathcal{P} = \{P_\theta\}$
- Given $G$ and $\theta$, evaluate the likelihood $L(\theta) = P_\theta(G)$
- Given $G$, find the $\theta$ that maximzes $L(\theta)$
- Given $\theta$, sample $G$ from the model $P_\theta$
  - To assess the goodness-of-fit, you need to generate samples from the estimated parameter $\theta$
  - You may want to generate graphs similar to $G$

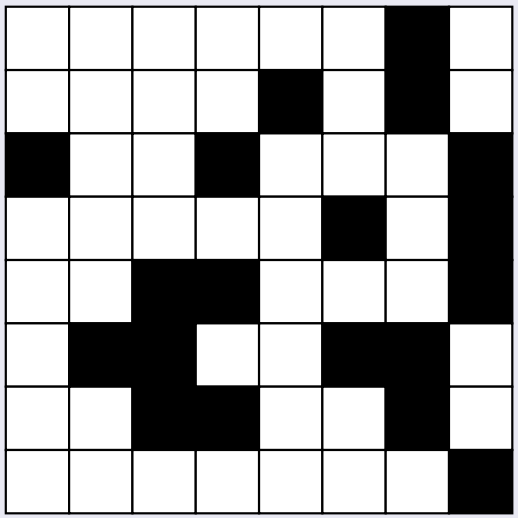
# Notation

- $V = \{1, 2, \ldots, n\}$ denote nodes in the graph
- $E \subset V \times V$ denote directed edges in the graph
  ex: $(3, 4) \in E$ denotes there is a link from node 3 to 4
- Graph $G = (V, E)$

## Graph $G$




## Adjacency Matrix $A$



# Erdös-Rényi model (1960)

- Every edge in the graph is an independently and identically distributed Bernoulli random variable with parameter $\theta$
- $P_\theta(G) = \prod_{(i,j)\in E} \theta \prod_{(i,j)\notin E} (1-\theta) = \theta^{|E|}(1-\theta)^{n^2-|E|}$
- Simple and efficient; however not realistic model

# Exponential Random Graph Model (ERGM)

- Let $\phi(G)$ denote a vector of graph statistics; number of edges, number of cycles and so on. Then,

$$P_\theta(G) = \frac{1}{Z(\theta)} \exp\left(\langle \theta, \phi(G) \rangle\right)$$

- Key problem: $Z(\theta)$ extremely hard to calculate
- Not scalable to graphs with millions of nodes

# Aldous-Hoover Theorem

## Aldous-Hoover Theorem

Every (exchangeable) graph model can be written:

$$P(G \mid \theta, \xi) = \prod_{(i,j)\in E} f_\theta(\xi_i, \xi_j) \prod_{(i,j)\notin E} (1 - f_\theta(\xi_i, \xi_j)),$$

where $\xi = (\xi_1, \ldots, \xi_n)$ and $\theta, \xi_i$'s are all independent random variables.

- This justifies conditional independence assumption
- The game is how to define $f_\theta(\cdot)$?

# The Hunt for Scalability

- Latent Space Model (Hoff et al, 2002)
  - Embed nodes into latent Euclidean space
  - $\Omega(n^2)$ computation
- Kronecker Product Graph Model (KPGM, Leskovec et al., 2010)
  - Parameter Estimation: $O\left(n \log_2\left(n\right)\right)$ for each MCMC step
  - Sampling: $O\left(\log_2\left(n\right)|E|\right)$
- Multiplicative Attribute Graph Model (MAGM, Kim and Leskovec, 2010)
  - Generalization of KPGM
  - Parameter Estimation: $O\left(\left(\log_2\left(n\right)\right)^2|E|\right)$
  - Sampling: !?!?

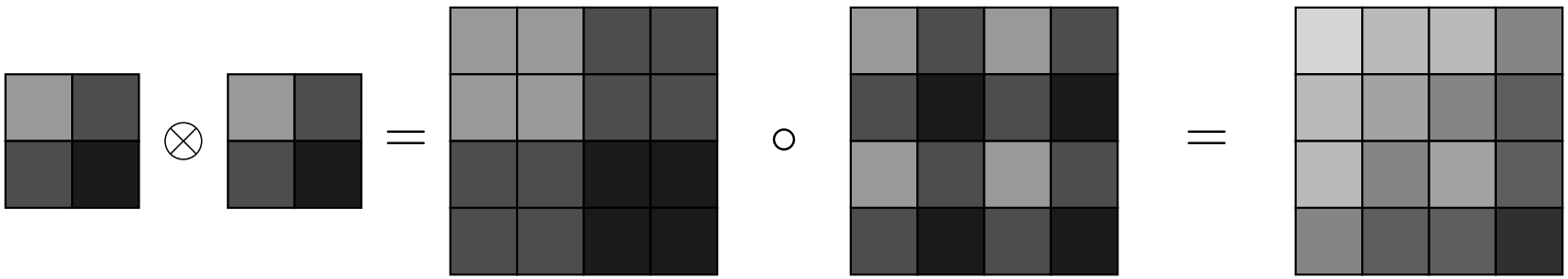
# Outline

# Kronecker Multiplication

Suppose you are given a $2 \times 2$ matrix $\Theta$.

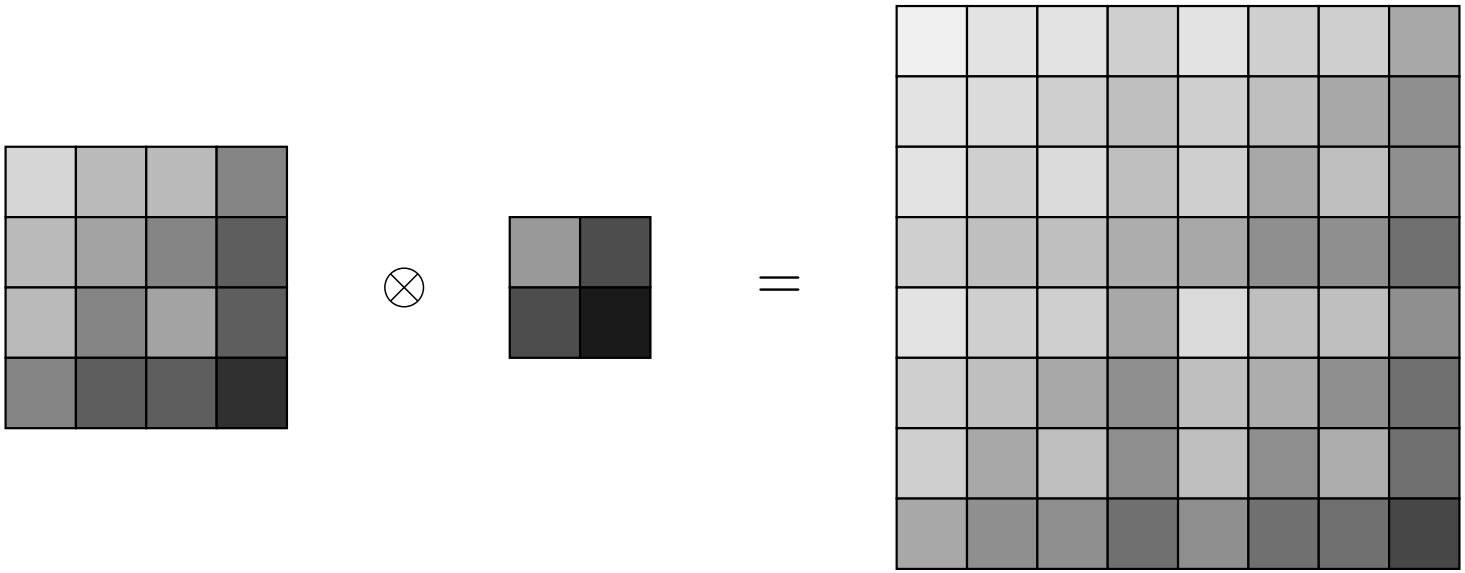
$$\Theta = \begin{bmatrix} 0.4 & 0.7 \\ 0.7 & 0.9 \end{bmatrix} \quad \Theta \otimes \Theta = \begin{bmatrix} 0.16 & 0.28 & 0.28 & 0.49 \\ 0.28 & 0.36 & 0.49 & 0.63 \\ 0.28 & 0.49 & 0.36 & 0.63 \\ 0.49 & 0.63 & 0.63 & 0.81 \end{bmatrix}$$

More visually



# Kronecker Multiplication

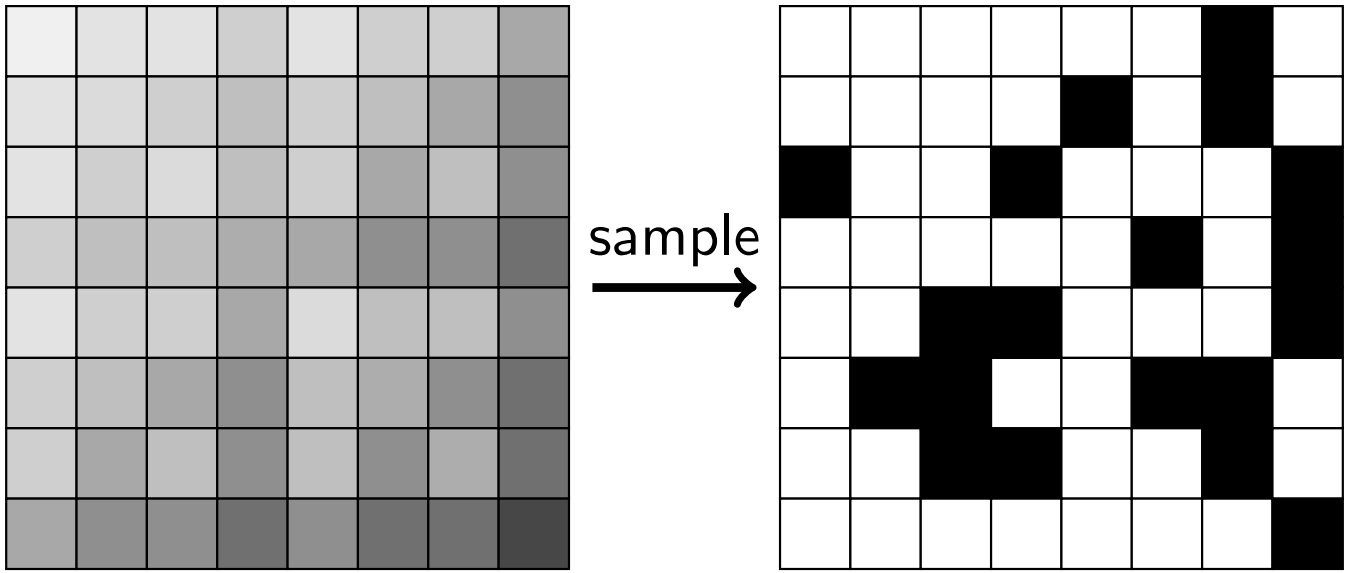
You can do it further!



# Generative Model

## Kronecker Product Graph Model (KPGM)

- The probability of an edge between nodes $(i, j)$ is given by $\left[\otimes^d \Theta\right]_{ij}$
- Each edge is sampled independently

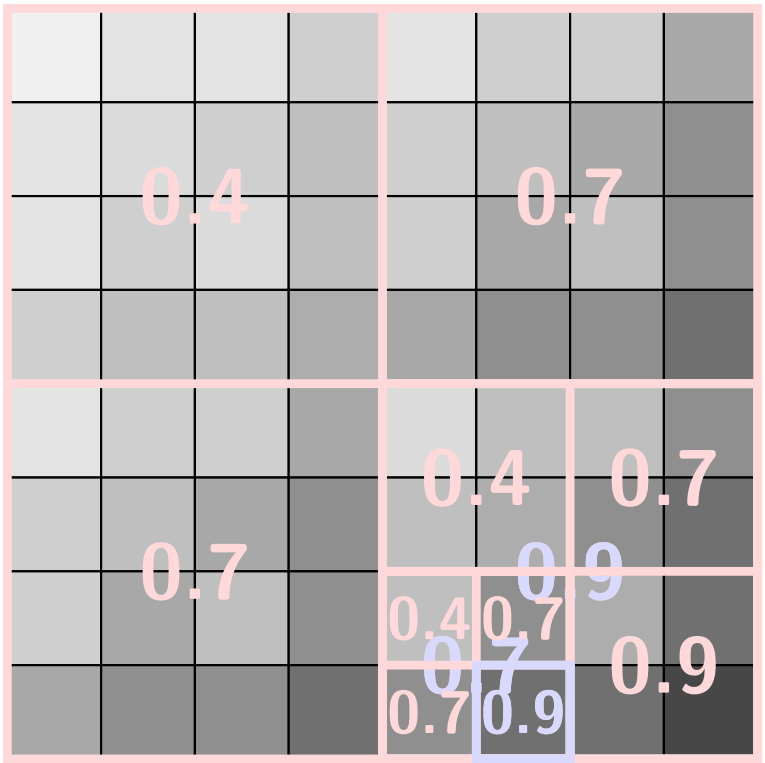


# Sampling Algorithm

- Naïve sampling: $\Omega(n^2)$ time
- A Ball Dropping Process (BDP)
  - exploits fractal structure: $O\left(\log_2\left(n\right)|E|\right)$

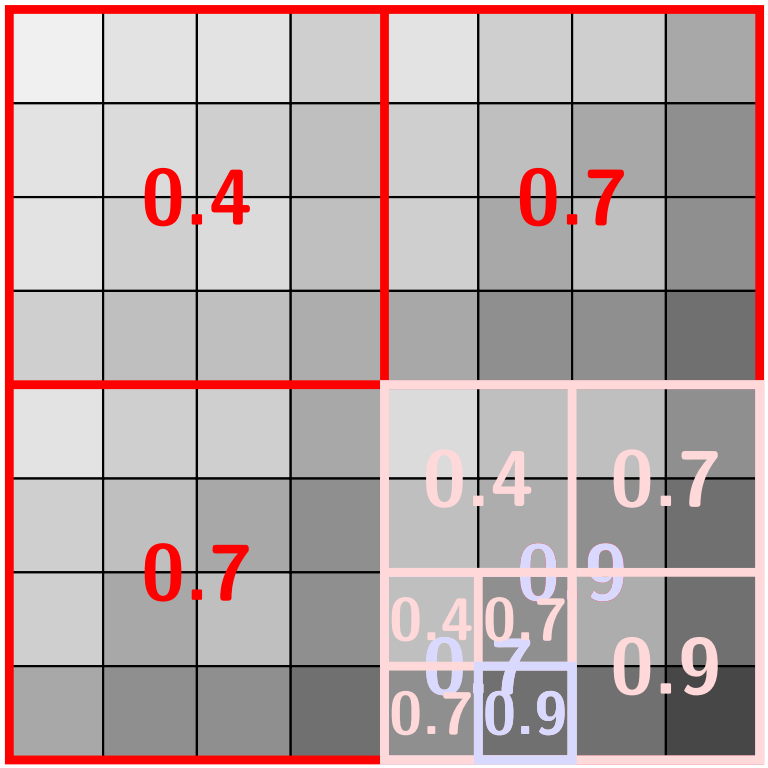
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



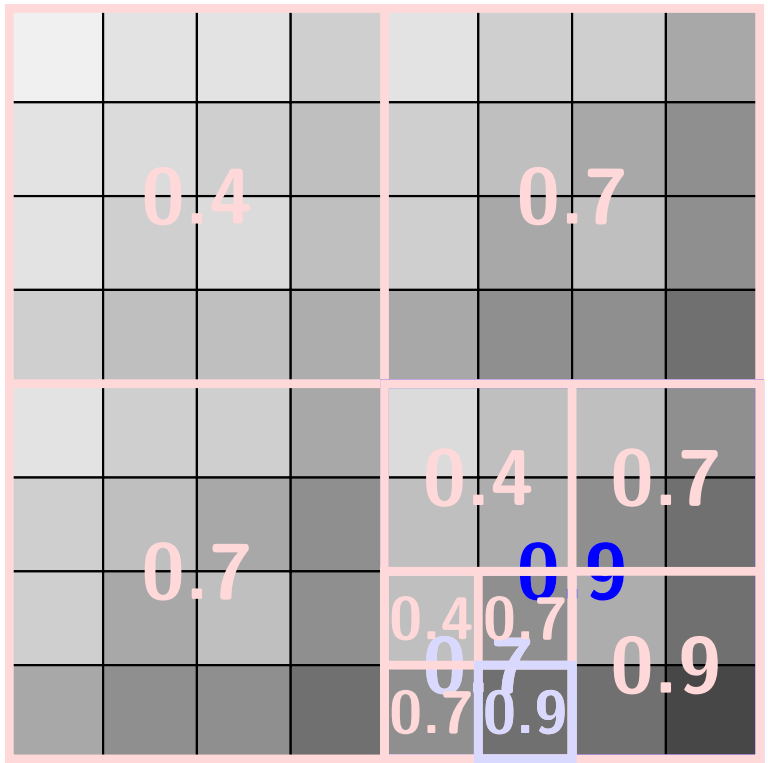
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



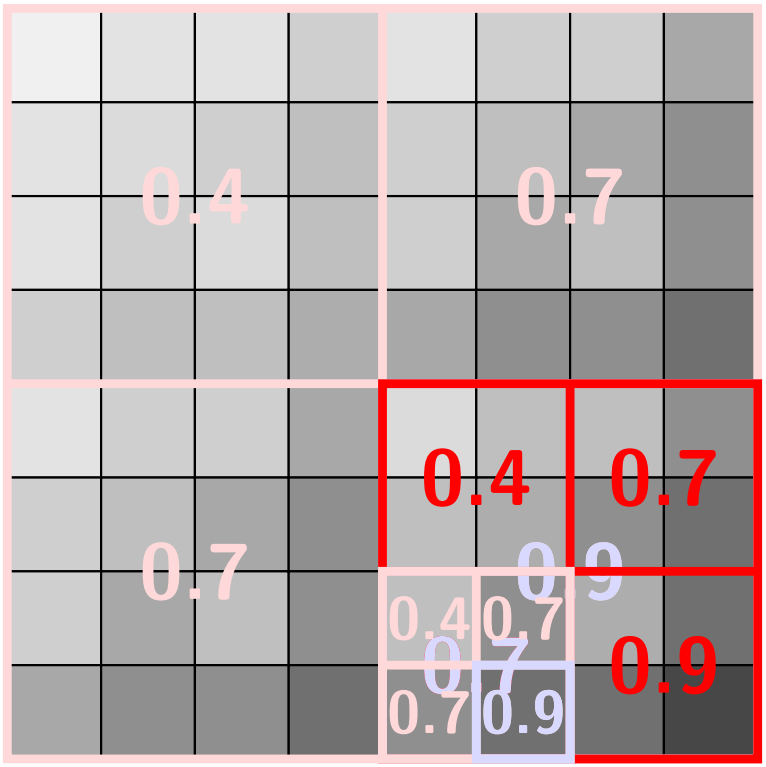
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



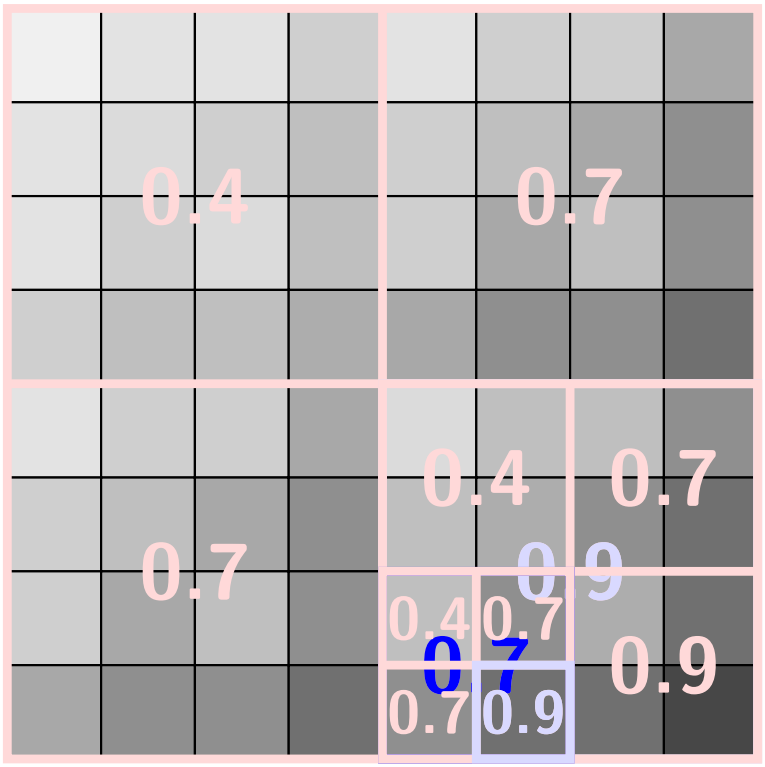
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



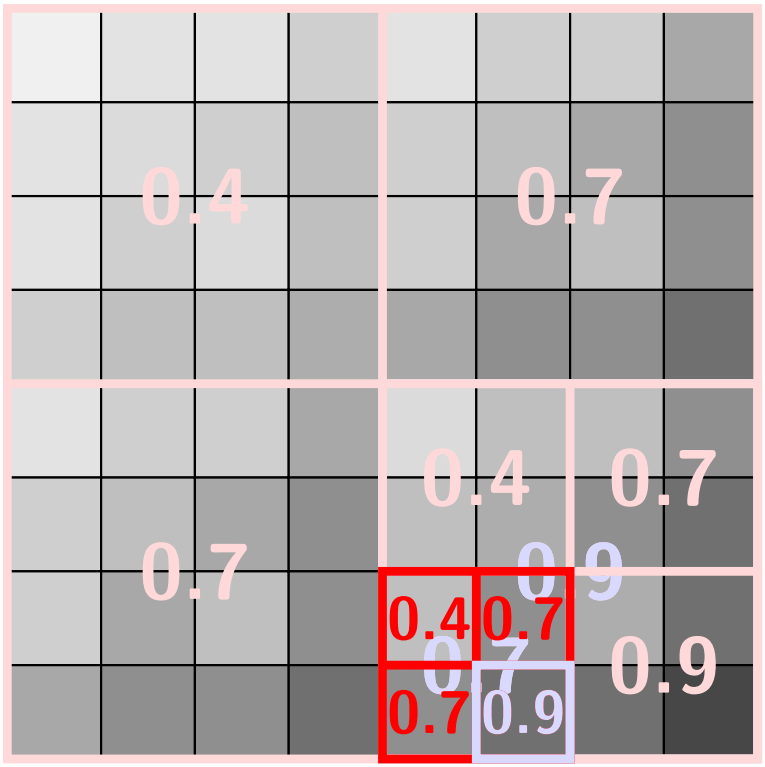
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



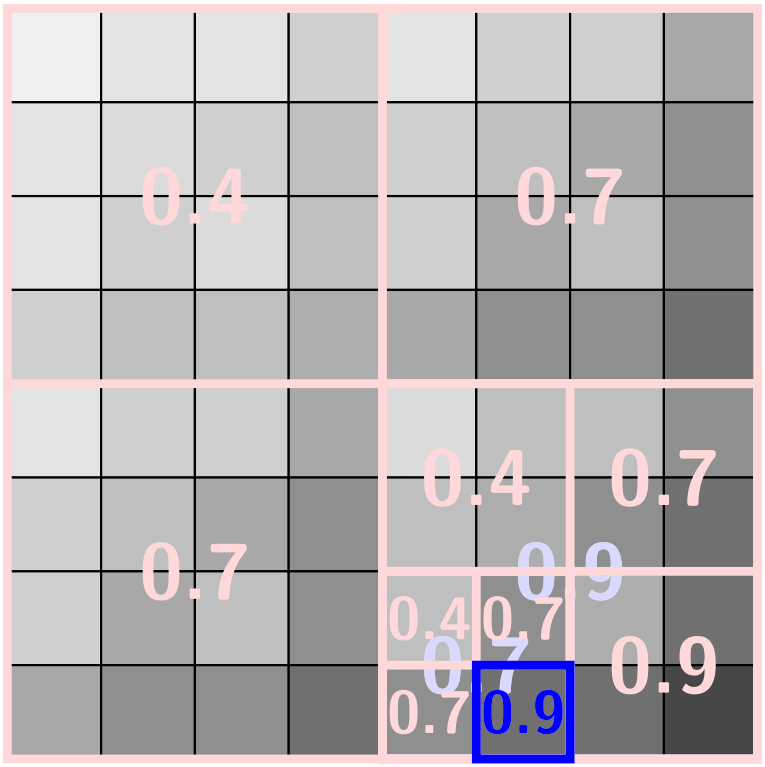
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



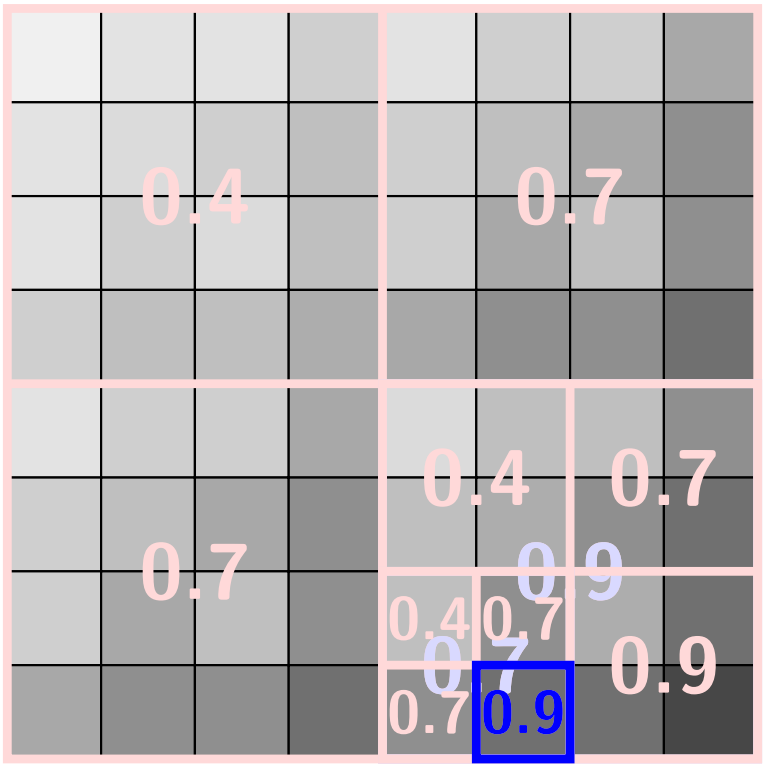
# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



# Ball Dropping Process

- Generate the number of edges
- Divide the matrix into four quadrants
- Choose one with proportional probability
- Repeat this edge number times



# Samples Produced by the BDP

**Theorem (Yun and Vishwanathan, 2012)**

*If a multi-graph $G$ is sampled from a BDP with parameters $\Theta$, then $A_{ij}$ follows an independent Poisson distribution with rate parameter $\Gamma_{ij} = \left[\otimes^d \Theta\right]_{ij}$.*

# Why don't we all use the KPGM?

- Some serious limitations
  - Empirically (Moreno and Neville, 2009)
  - Theoretically (Kolda et al., 2012)
- Parameter estimation requires finding a latent permutation (hard)

# Outline

# Idea

- Suppose there are $d$ attributes which describe each node.
- Each node either possesses or lacks each attribute.
- For example, each attribute can be understood to an answer to each question, such as:
  - Are you a statistician?
  - Do you speak Korean?
  - Do you read Engadget?

# Idea

- Questions are independent of each other
- However the probability of an answer being 1 can be estimated

$$\mu_{Stats} = 0.2,\ \mu_{Kor} = 0.1,\ \mu_{Eng} = 0.4$$

- There also is a $2 \times 2$ parameter matrix associated with each attribute

$$\Theta_{Stats} = \begin{pmatrix} 0.5 & 0.3 \\ 0.3 & 0.9 \end{pmatrix}, \Theta_{Kor} = \begin{pmatrix} 0.6 & 0.3 \\ 0.3 & 0.99 \end{pmatrix}, \Theta_{Eng} = \begin{pmatrix} 0.2 & 0.3 \\ 0.3 & 0.6 \end{pmatrix}.$$

# Model

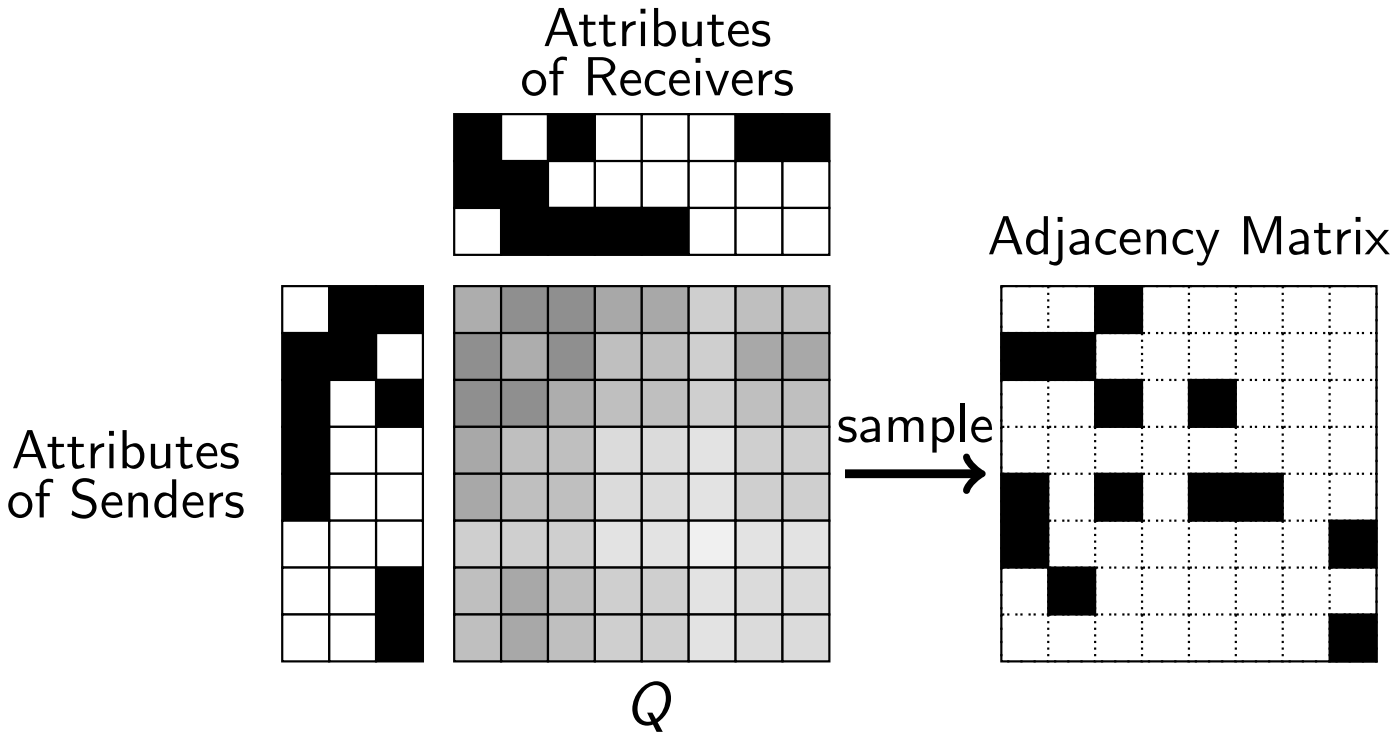
$$\Theta_{Stats} = \begin{pmatrix} 0.5 & 0.3 \\ 0.3 & 0.9 \end{pmatrix}, \Theta_{Kor} = \begin{pmatrix} 0.6 & 0.3 \\ 0.3 & 0.99 \end{pmatrix}, \Theta_{Eng} = \begin{pmatrix} 0.2 & 0.3 \\ 0.3 & 0.6 \end{pmatrix}.$$

- Consider two people:
  - Yun: (Stats 1, Korean 1, Engadget 0)
  - Vishy: (Stats 1, Korean 0, Engadget 1)
- The probability of an edge from Yun to Vishy is given by

$$P_{Yun,Vishy} = \underbrace{0.9}_{Stats} \cdot \underbrace{0.3}_{Kor} \cdot \underbrace{0.3}_{Eng} = 0.081$$

- The effect of each attribute is multiplicative (Kim and Leskovec 2010)

# Graphical Representation

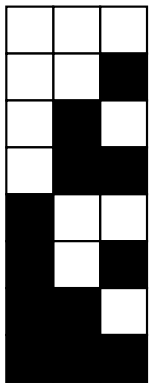



- $Q$ : edge probability matrix
- Each edge is independently sampled
- Naively $O\left(n^2 d\right)$. Can we do it faster?

# Outline

# KPGM and MAGM are closely related

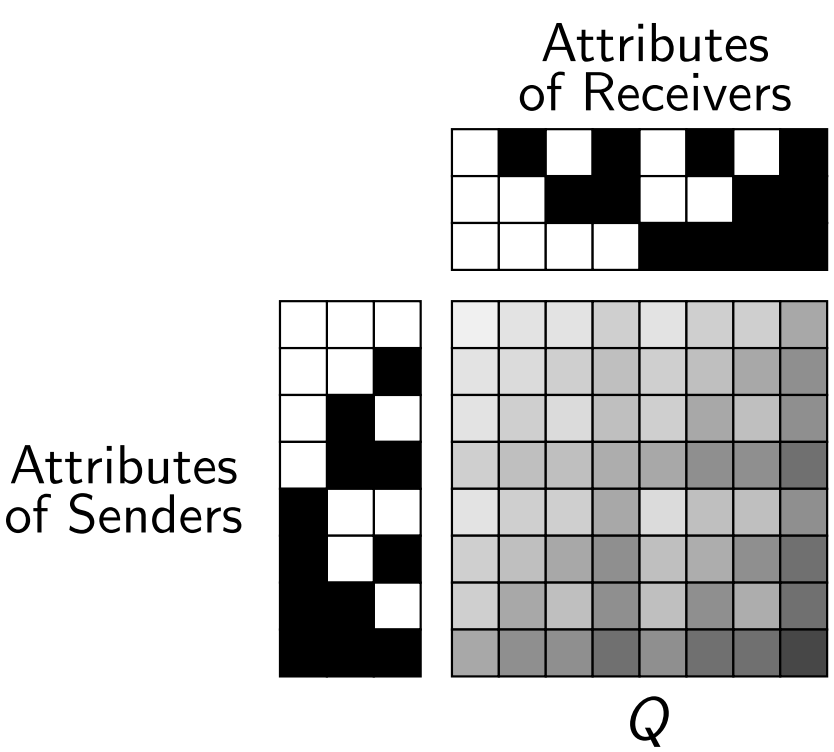
- Suppose we have a set of attributes which looks like the following:



- The parameter for each attribute is the same: $\Theta = \begin{pmatrix} 0.4 & 0.7 \\ 0.7 & 0.9 \end{pmatrix}$
- How does the edge probability matrix $Q$ look like?

# KPGM and MAGM are closely related

Looks familiar?



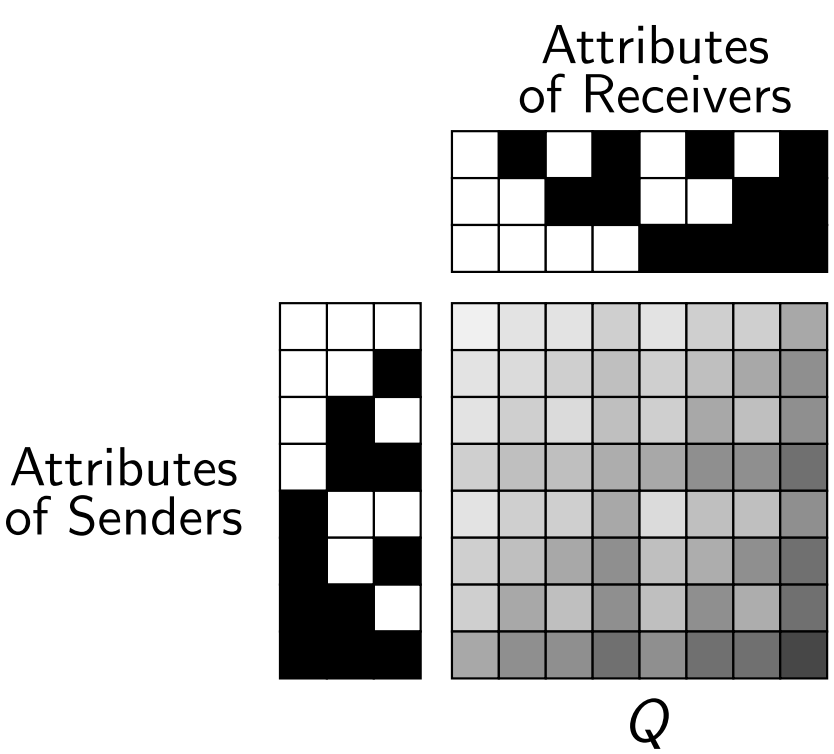


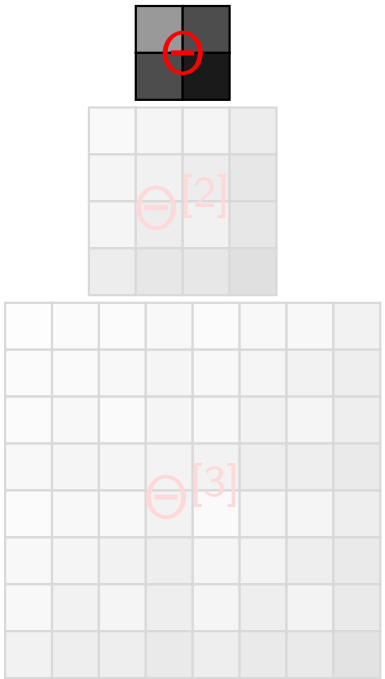


Kronecker Power of the Matrix!

# KPGM and MAGM are closely related

Looks familiar?

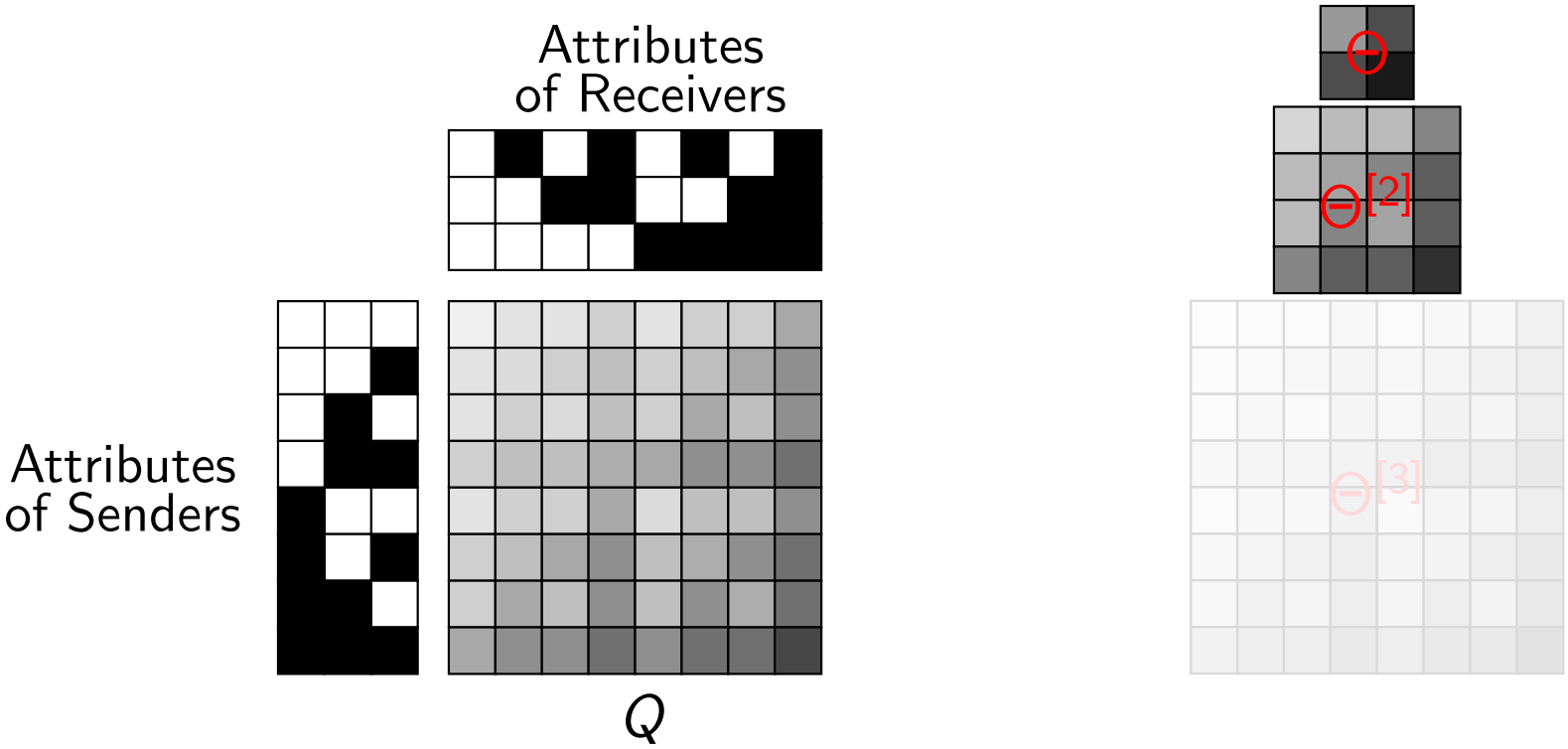



Kronecker Power of the Matrix!

# KPGM and MAGM are closely related

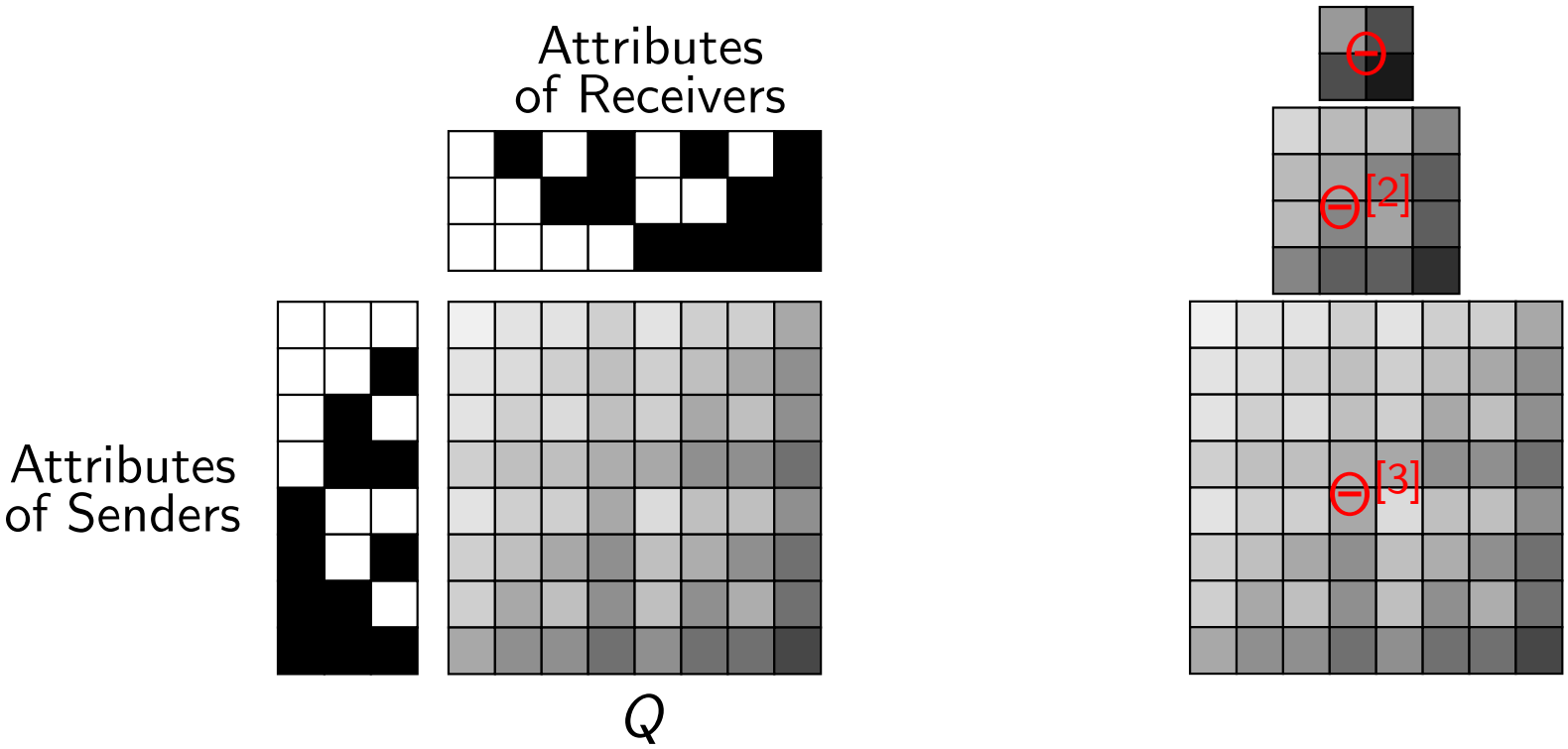
Looks familiar?




Kronecker Power of the Matrix!

# KPGM and MAGM are closely related

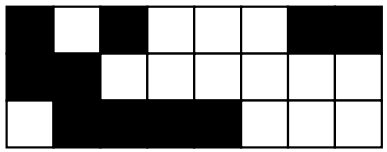
Looks familiar?



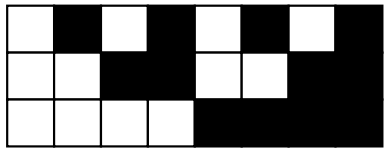

Kronecker Power of the Matrix!

# Connection

- If every node has unique color, it suffices to sample from a KPGM.
- Problem is, there can be duplicates . . .



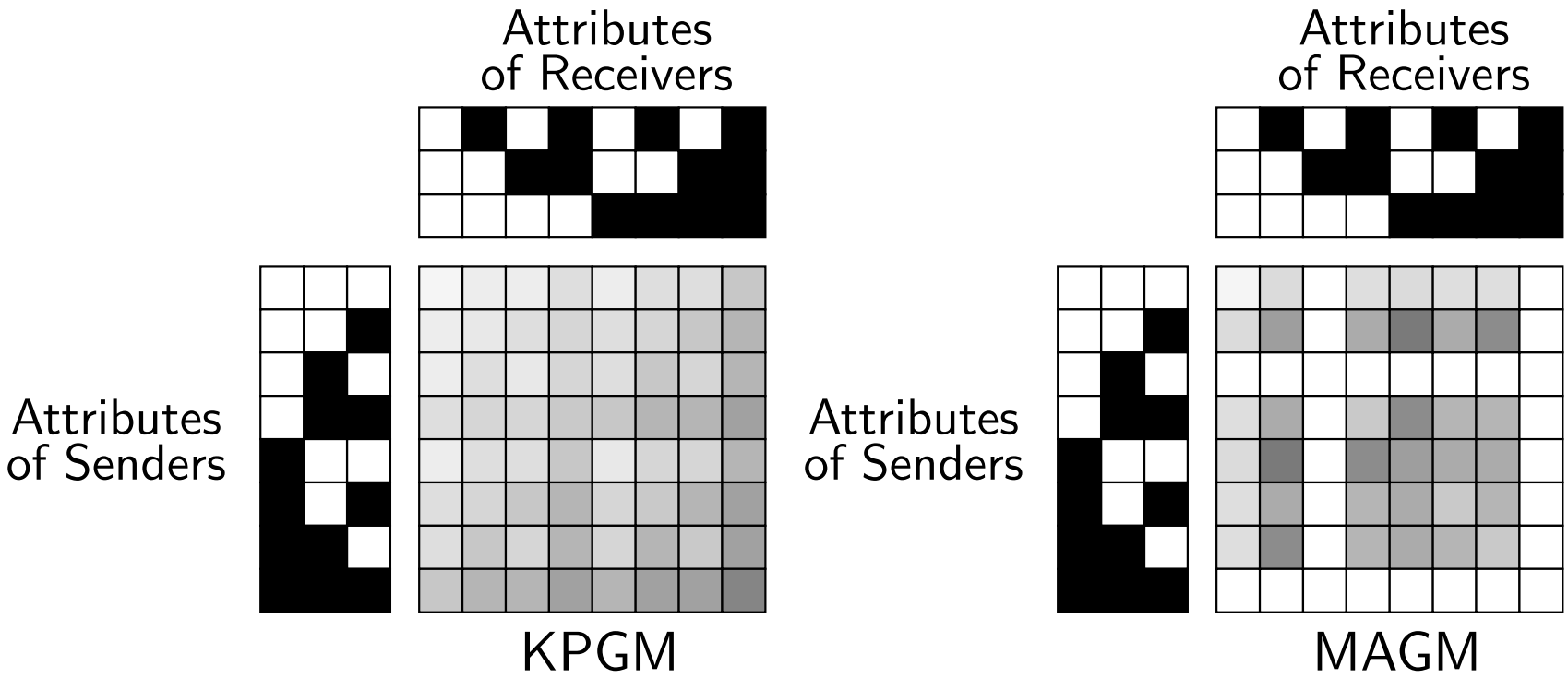
instead of



- Let $m$ be the maximum number one color is repeated.
  (for example, $m = 2$ in the example above)
- If $m$ is a small number, is there a good way of sampling the MAGM?

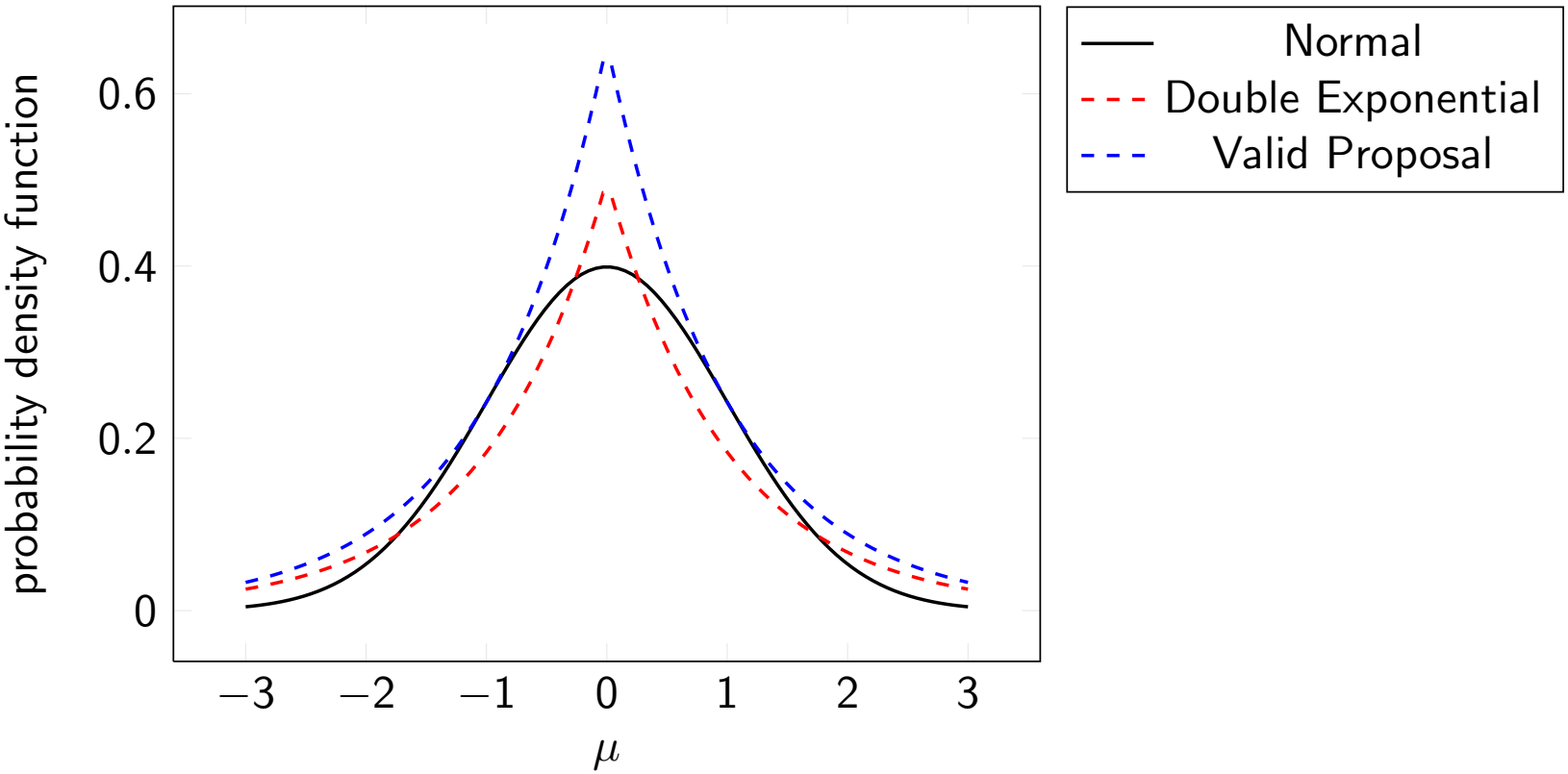
# Simple Illustrative Proposal

- Expected number of edges between colors:



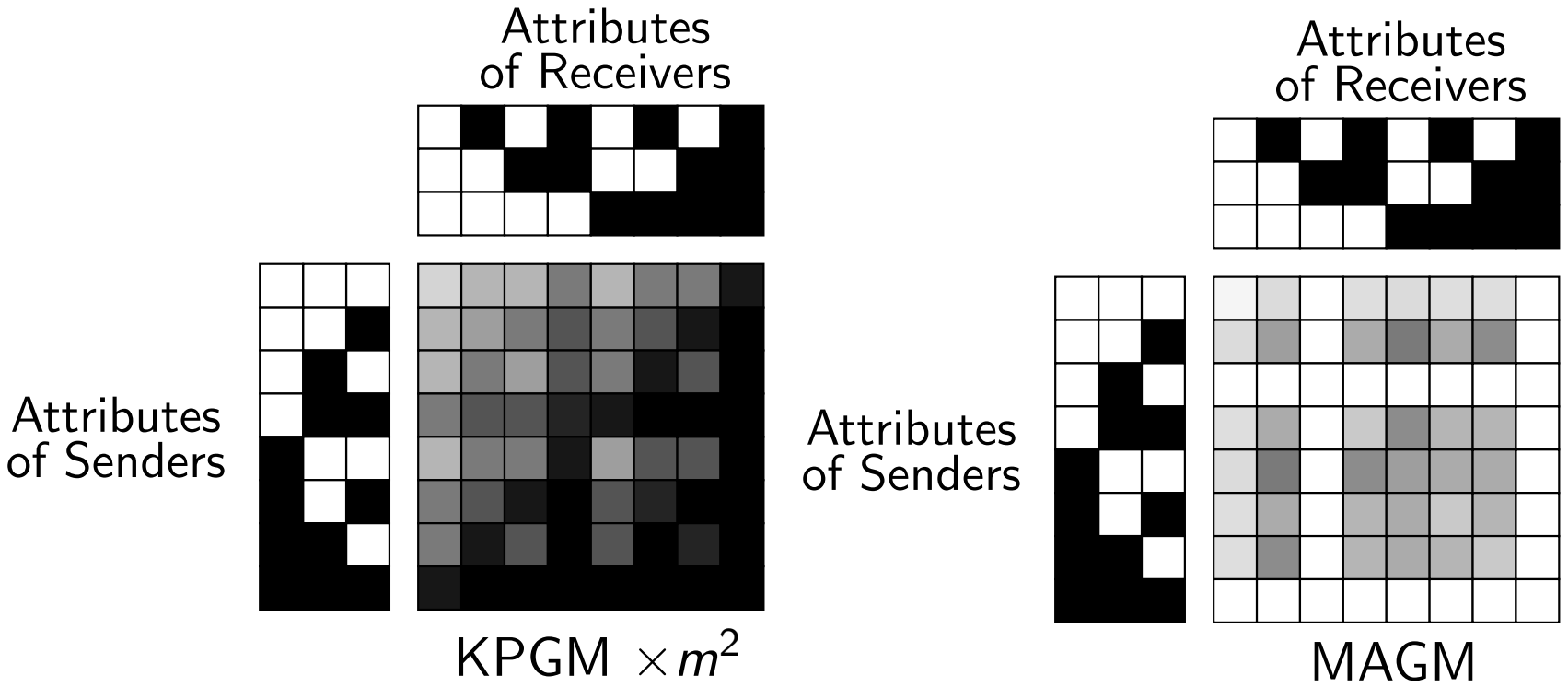
# Simple Illustrative Proposal

- KPGM itself is not a valid proposal for MAGM.
- Remember the basic principle of accept-reject sampling!



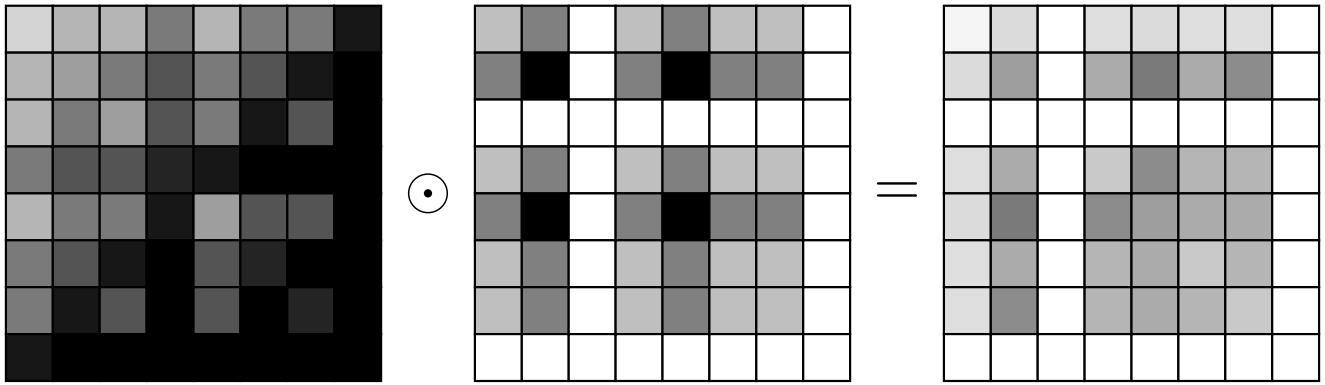
# Simple Illustrative Proposal

- If the left matrix is multiplied by $m^2$, its every entry is strictly higher that the corresponding entry in the right.



# Simple Illustrative Proposal

- First generate $m^2$ times more edges from BDP.
- Then, reject some of them according to the acceptance ratio.
- Resulting graph will follow MAGM




- Recall when $X \sim Poi(\lambda)$, $Y \mid X \sim Bin(X, p)$, then $Y \sim Poi(\lambda \cdot p)$.

# Analysis of Simple Proposal

- Let $e_K$ be the expected number of edges in the KPGM.
- We are sampling $m^2 \cdot e_K$ edges from the proposal distribution.
- Sampling each edge will take $O(\log_2(n))$ time.
- Therefore, the overall time complexity is: $O\left(m^2 \cdot e_K \cdot \log_2(n)\right)$.
- It is important to have small $m$!

# Bounding $m$

$$m := \max_{0 \leq c \leq n-1} |\mathcal{V}_c| ,$$

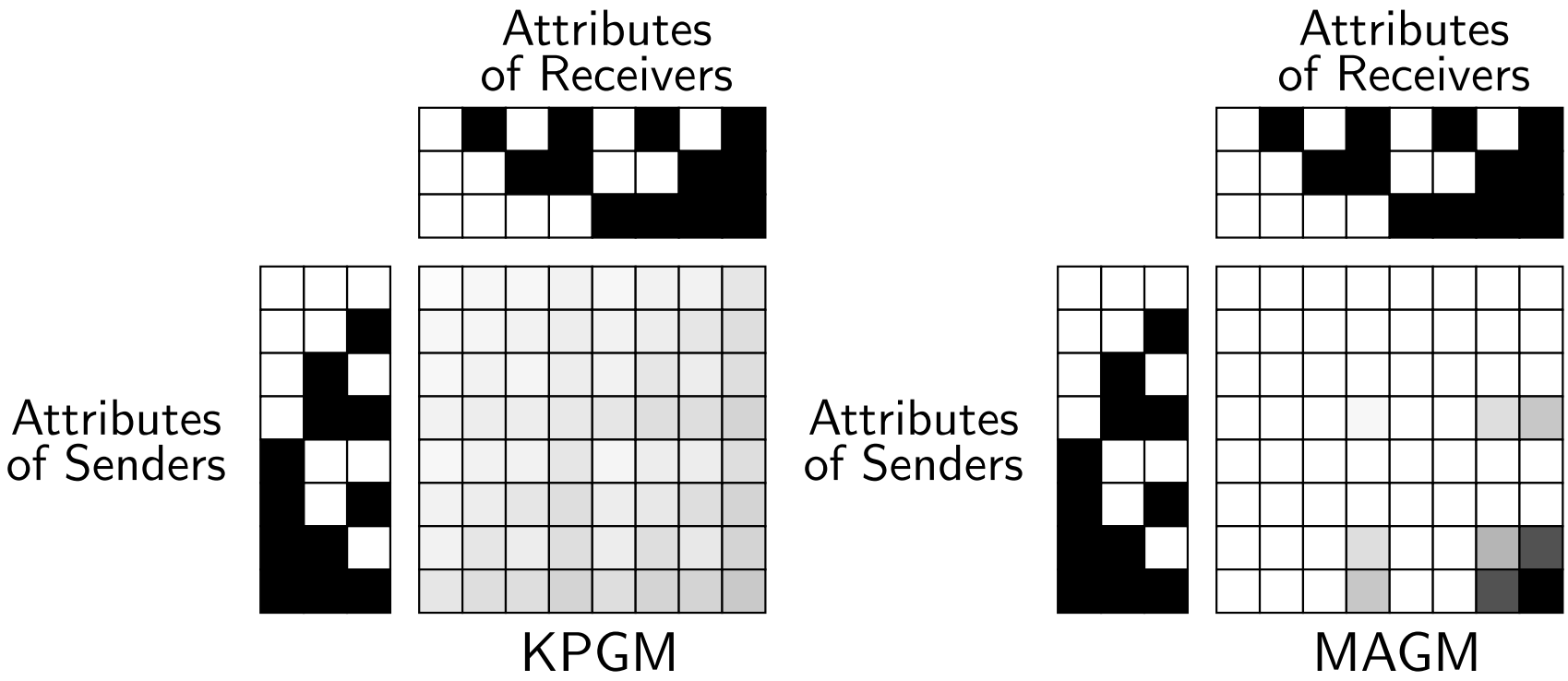
- $c$ denotes the color
- $|\mathcal{V}_c|$ denotes the number of nodes with color $c$

**Theorem (Yun and V., 2012)**

*When* $\mu^{(1)} = \mu^{(2)} = \cdots = \mu^{(d)} = 0.5$, *and* $n = 2^d$, *with high probability* $m \leq \log_2(n)$.

# When $\mu$ is Large or Small . . .

# Directly Bounding by $m^2$ is Inefficient



# Partitioning Colors

- Partition colors into set of frequent colors $\mathcal{F}$ and infrequent colors $\mathcal{I}$:

$$\mathcal{F} := \{c : \mathbb{E}[|\mathcal{V}_c|] \geq 1\}$$
$$\mathcal{I} := \{c : \mathbb{E}[|\mathcal{V}_c|] < 1\} = \mathcal{F}^c$$

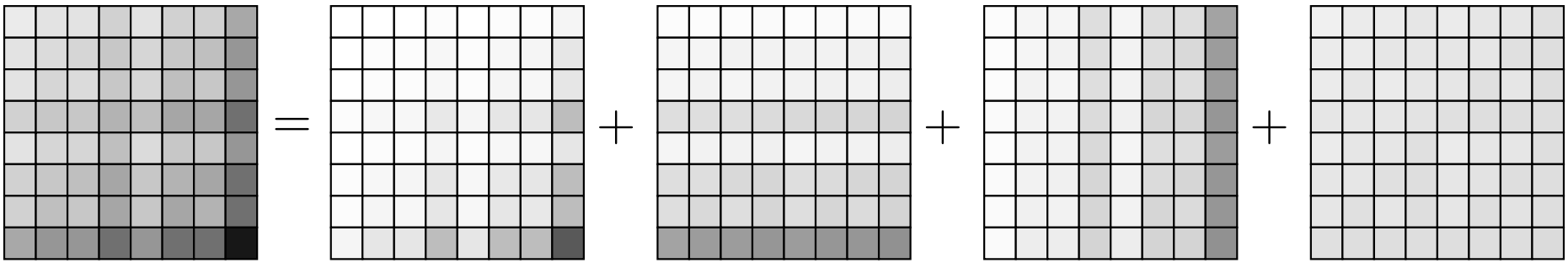
- Every node chooses its attribute independently
  - $|\mathcal{V}_c|$ are binomial random variables
- Using standard Chernoff-Hoeffding bounds, with high probability

$$m_{\mathcal{F}} := \max_{c \in \mathcal{F}} \frac{|\mathcal{V}_c|}{\mathbb{E}[|\mathcal{V}_c|]} \leq \log_2(n),$$
$$m_{\mathcal{I}} := \max_{c \in \mathcal{I}} |\mathcal{V}_c| \leq \log_2(n),$$

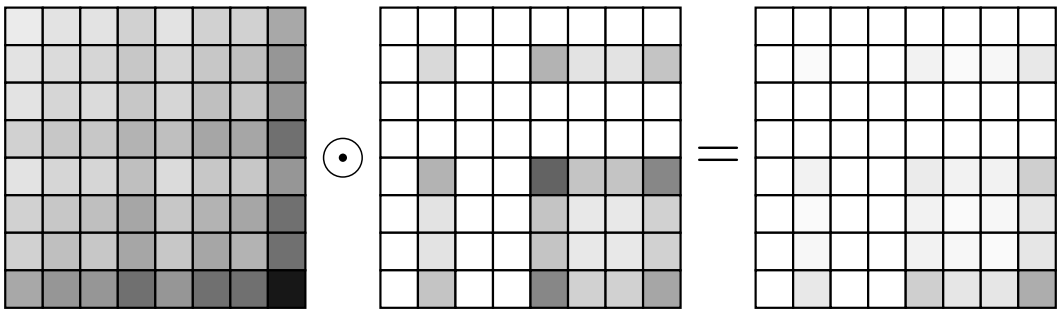
- Compare with using single $m := \max_{0 \leq c \leq n-1} |\mathcal{V}_c|$,

# Compact Proposal

- One can design a more compact proposal by partitioning colors.




- The target distribution is recovered by the accept-reject scheme



# The Proposal

$$\Theta'^{(\mathcal{F}\mathcal{F})(k)} := (n\, m_{\mathcal{F}})^{\frac{2}{d}} \begin{bmatrix} \left(1-\mu^{(k)}\right)^2 \theta_{00}^{(k)} & \left(1-\mu^{(k)}\right)\mu^{(k)}\theta_{01}^{(k)} \\ \mu^{(k)}\left(1-\mu^{(k)}\right)\theta_{10}^{(k)} & \left(\mu^{(k)}\right)^2 \theta_{11}^{(k)} \end{bmatrix}$$

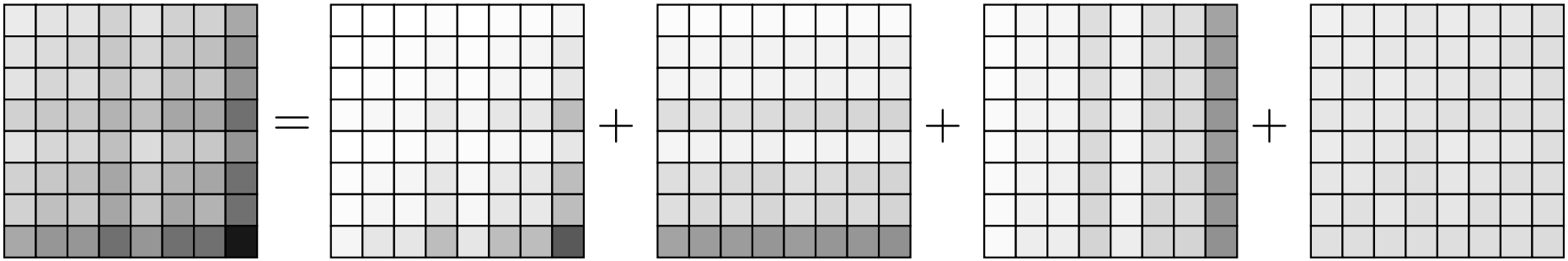
$$\Theta'^{(\mathcal{F}\mathcal{I})(k)} := (n\, m_{\mathcal{F}}\, m_{\mathcal{I}})^{\frac{1}{d}} \begin{bmatrix} \left(1-\mu^{(k)}\right)\theta_{00}^{(k)} & \left(1-\mu^{(k)}\right)\theta_{01}^{(k)} \\ \mu^{(k)}\theta_{10}^{(k)} & \left(\mu^{(k)}\right)\theta_{11}^{(k)} \end{bmatrix}$$

$$\Theta'^{(\mathcal{I}\mathcal{F})(k)} := (n\, m_{\mathcal{I}}\, m_{\mathcal{F}})^{\frac{1}{d}} \begin{bmatrix} \left(1-\mu^{(k)}\right)\theta_{00}^{(k)} & \mu^{(k)}\theta_{01}^{(k)} \\ \left(1-\mu^{(k)}\right)\theta_{10}^{(k)} & \mu^{(k)}\theta_{11}^{(k)} \end{bmatrix}$$

$$\Theta'^{(\mathcal{I}\mathcal{I})(k)} := (m_{\mathcal{I}})^{\frac{2}{d}} \begin{bmatrix} \theta_{00}^{(k)} & \theta_{01}^{(k)} \\ \theta_{10}^{(k)} & \theta_{11}^{(k)} \end{bmatrix}$$

# Time Complexity

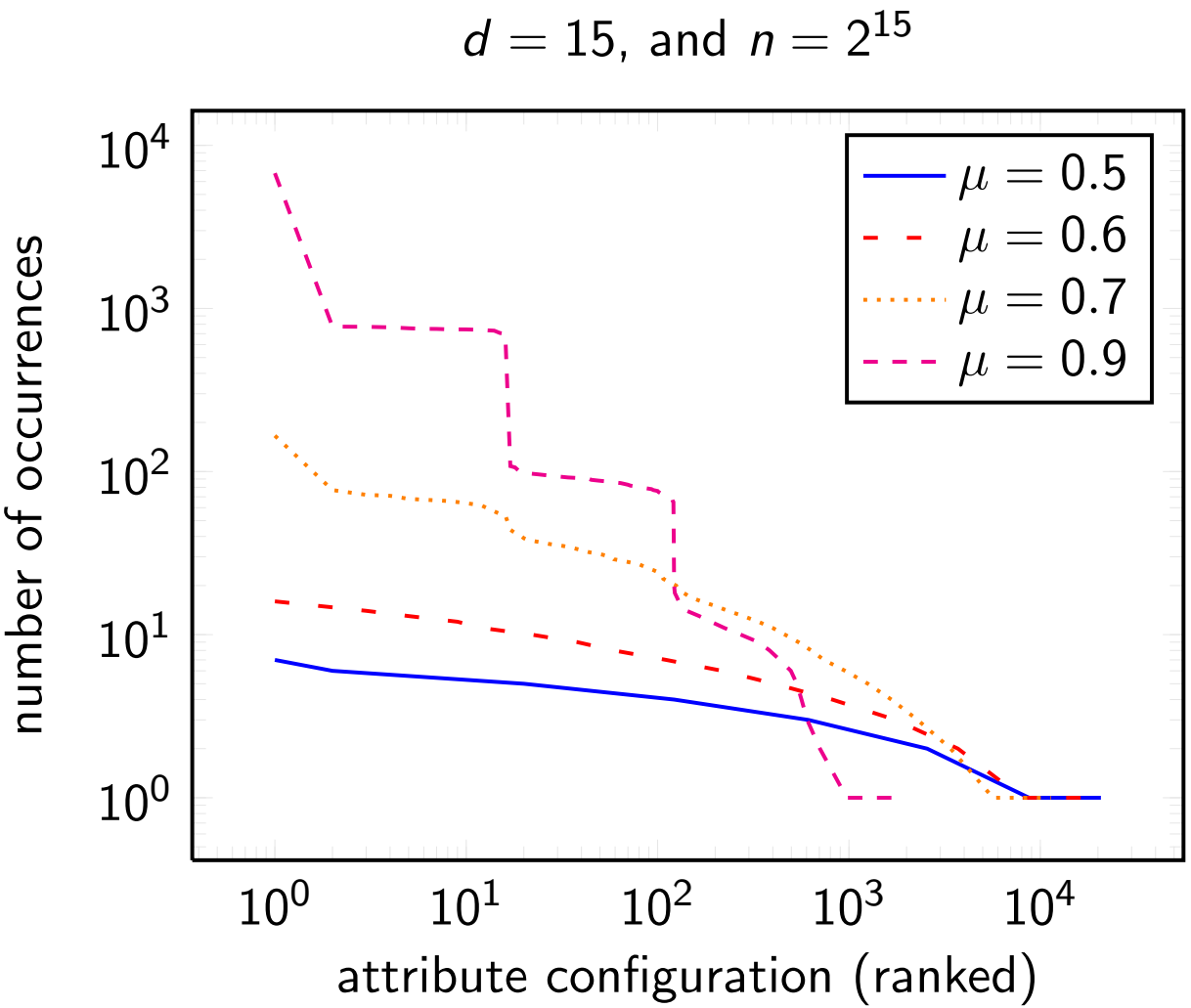
- The number of expected edges in each component is:




- $e_M$ = number of edges in MAGM, $e_K$ = number of edges in KPGM
- Sampling each edge takes $O(\log_2(n))$ time
- With high probability $m_{\mathcal{F}}, m_{\mathcal{I}} \leq \log_2(n)$
- Empirically we observe that $e_{MK}, e_{KM} \leq \max(e_M, e_K)$
- Overall time complexity: $O\left((\log_2(n))^3 (e_M + e_K)\right)$

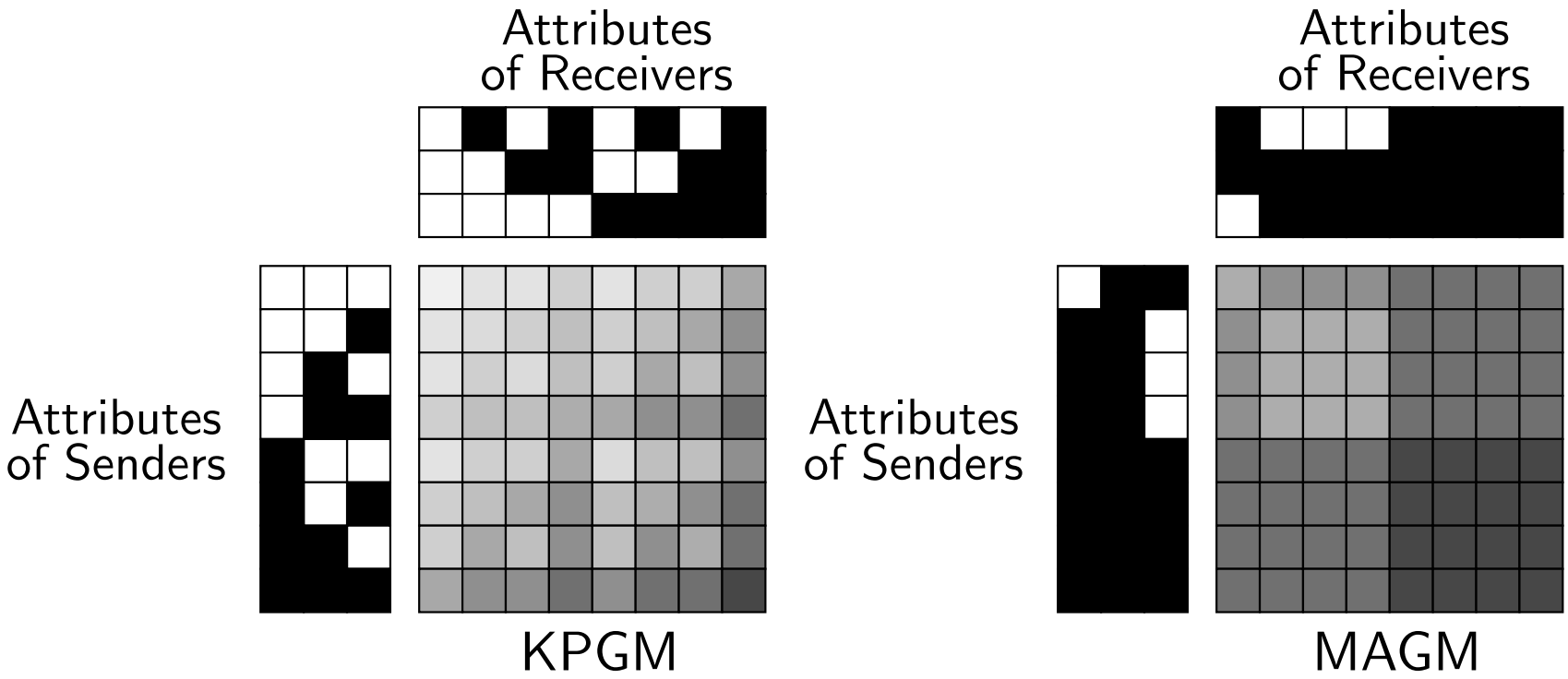
# Loss of Variety

- Ranked colors based on their frequency of occurrence.



# Block Structure

- When $\mu$ is high, small number of colors dominate and the MAGM exhibits block structure (stochastic blockmodel).



# Outline

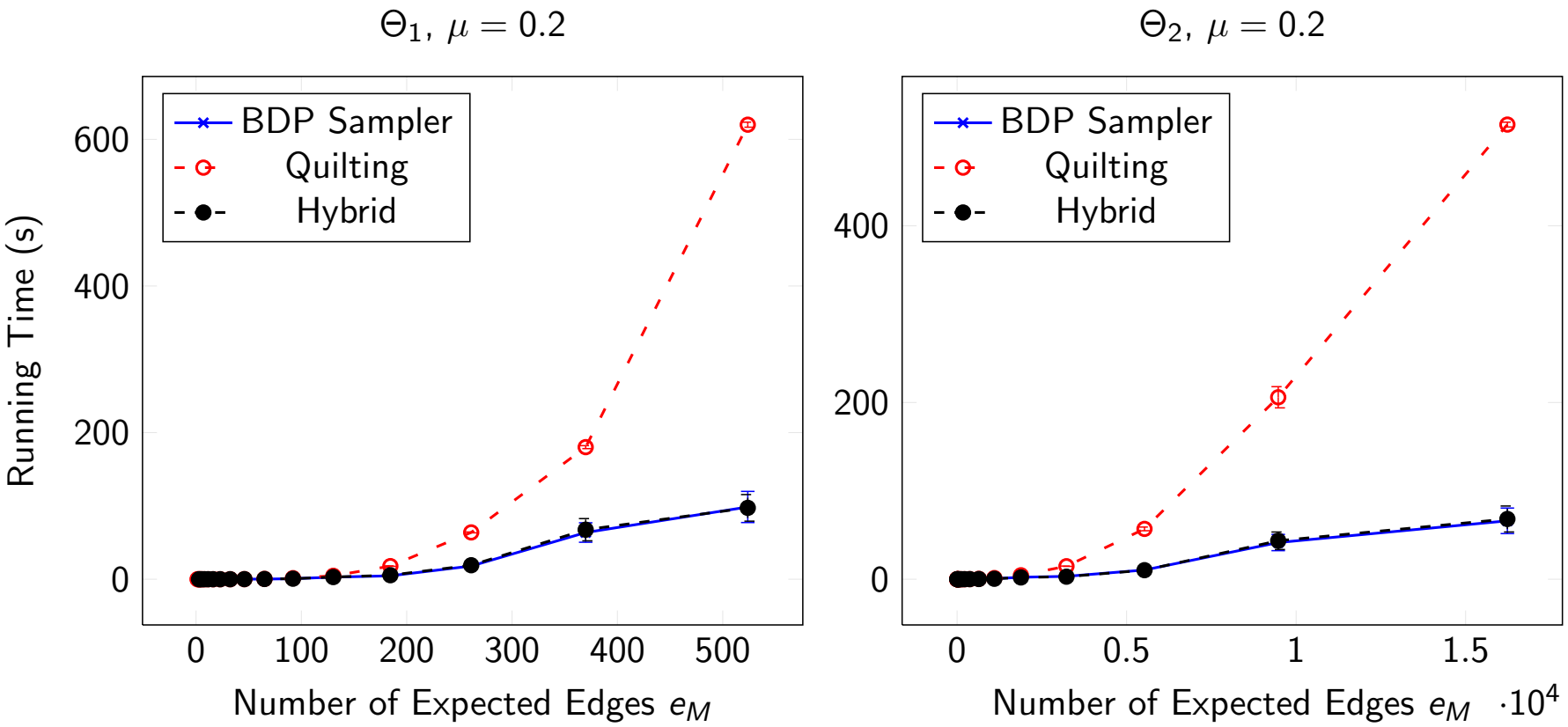
# Setup

- We chose two parameter matrices from the literature (Kim and Leskovec 2010, Moreno and Neville 2009)

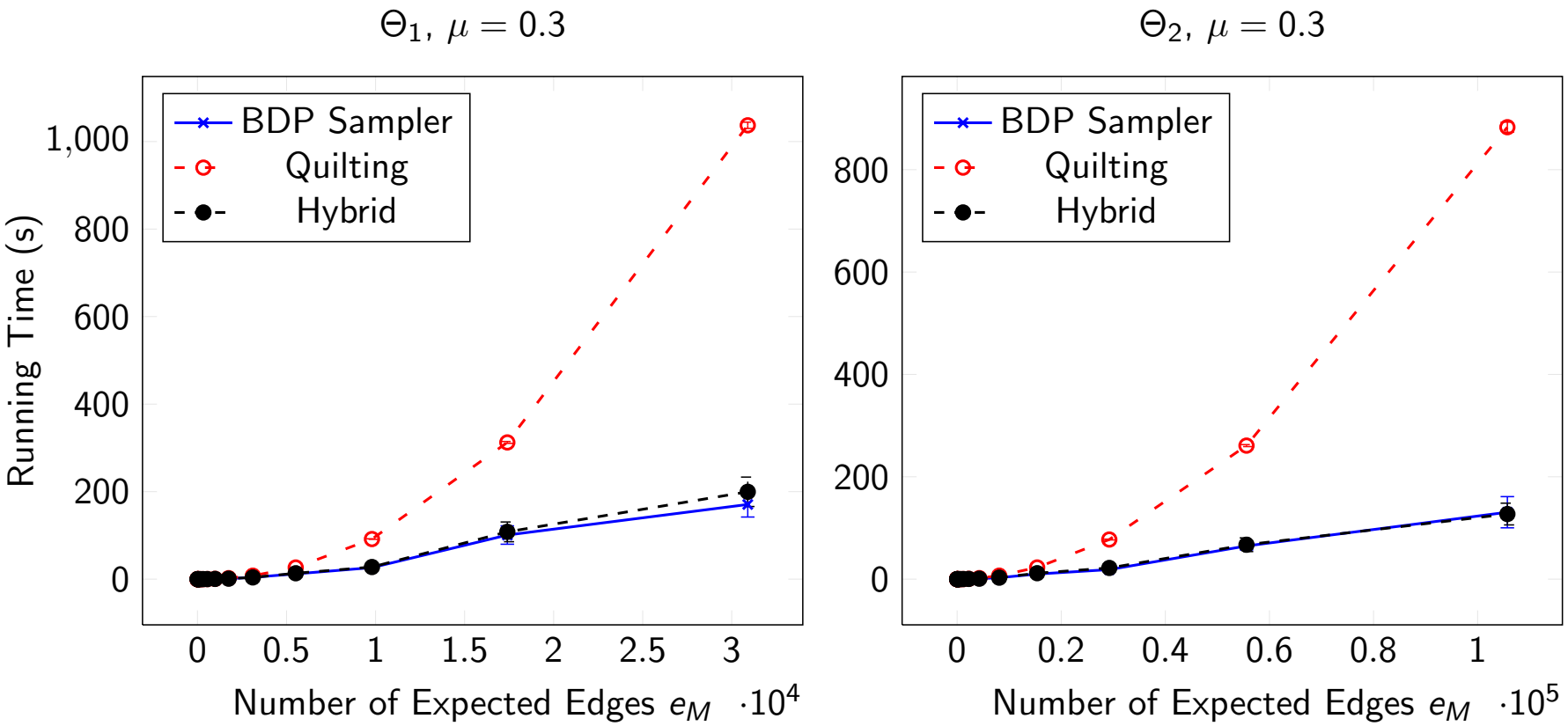
$$\Theta_1 = \begin{bmatrix} 0.15 & 0.7 \\ 0.7 & 0.85 \end{bmatrix} \text{ and } \Theta_2 = \begin{bmatrix} 0.35 & 0.52 \\ 0.52 & 0.95 \end{bmatrix}$$

- Number of attributes $d = \log_2 n$

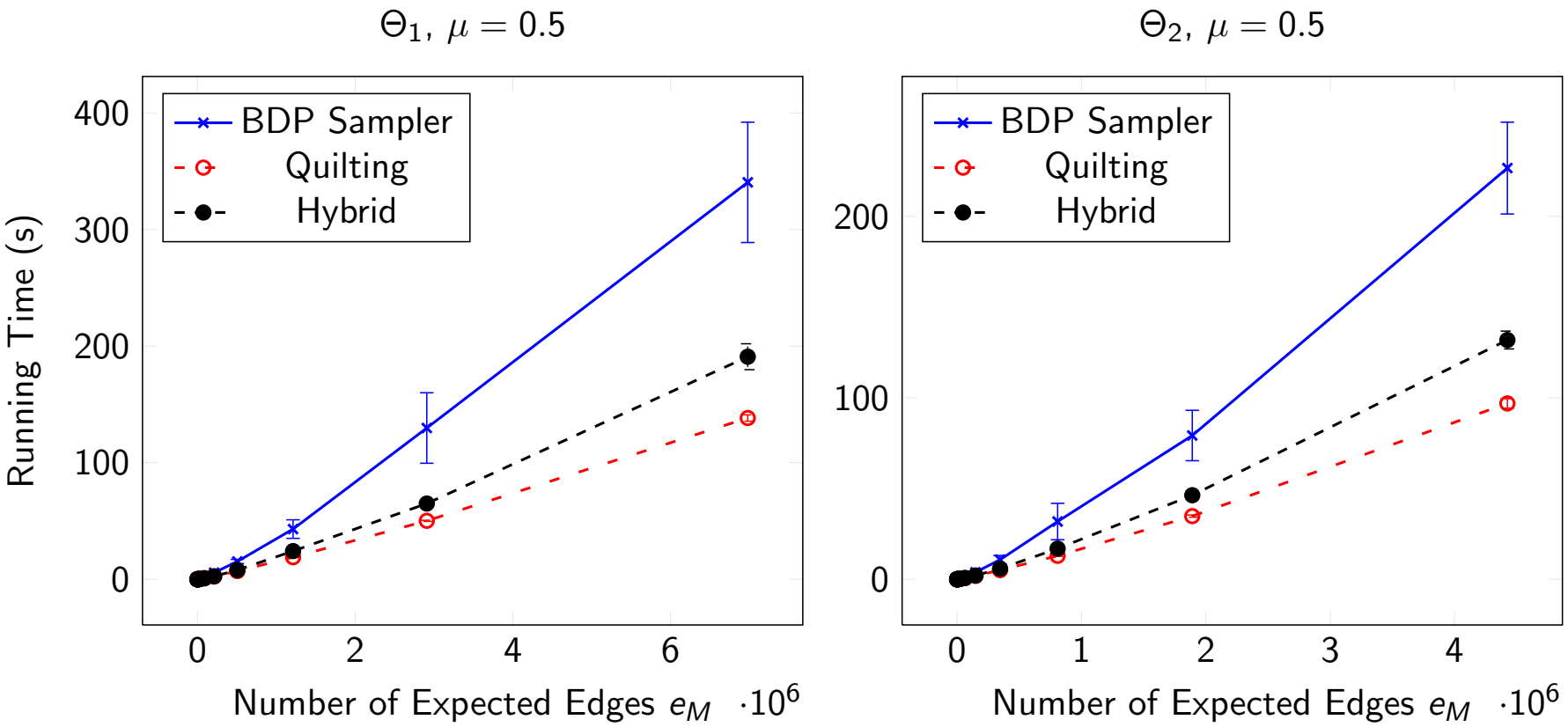
# Running Time as Function of Number of Expected Edges



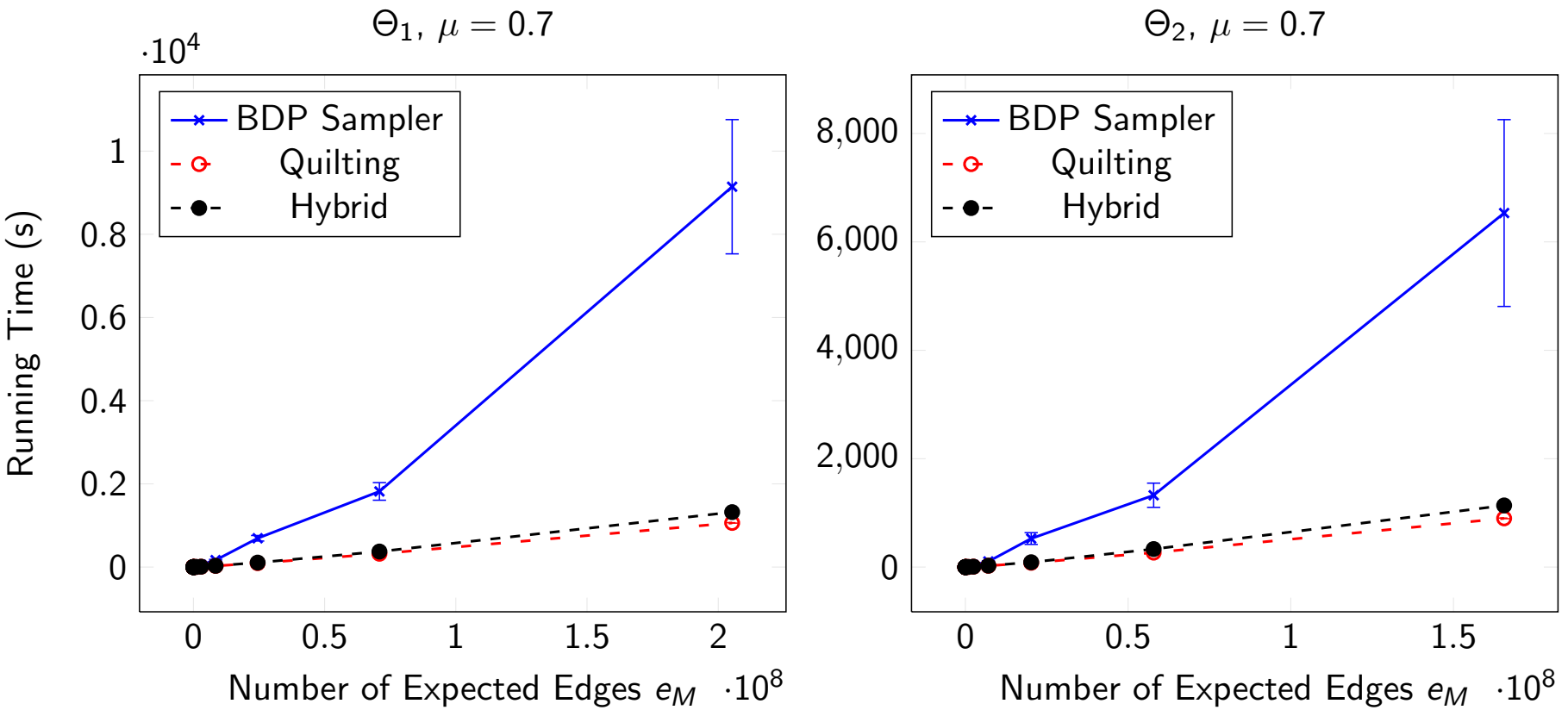
# Running Time as Function of Number of Expected Edges



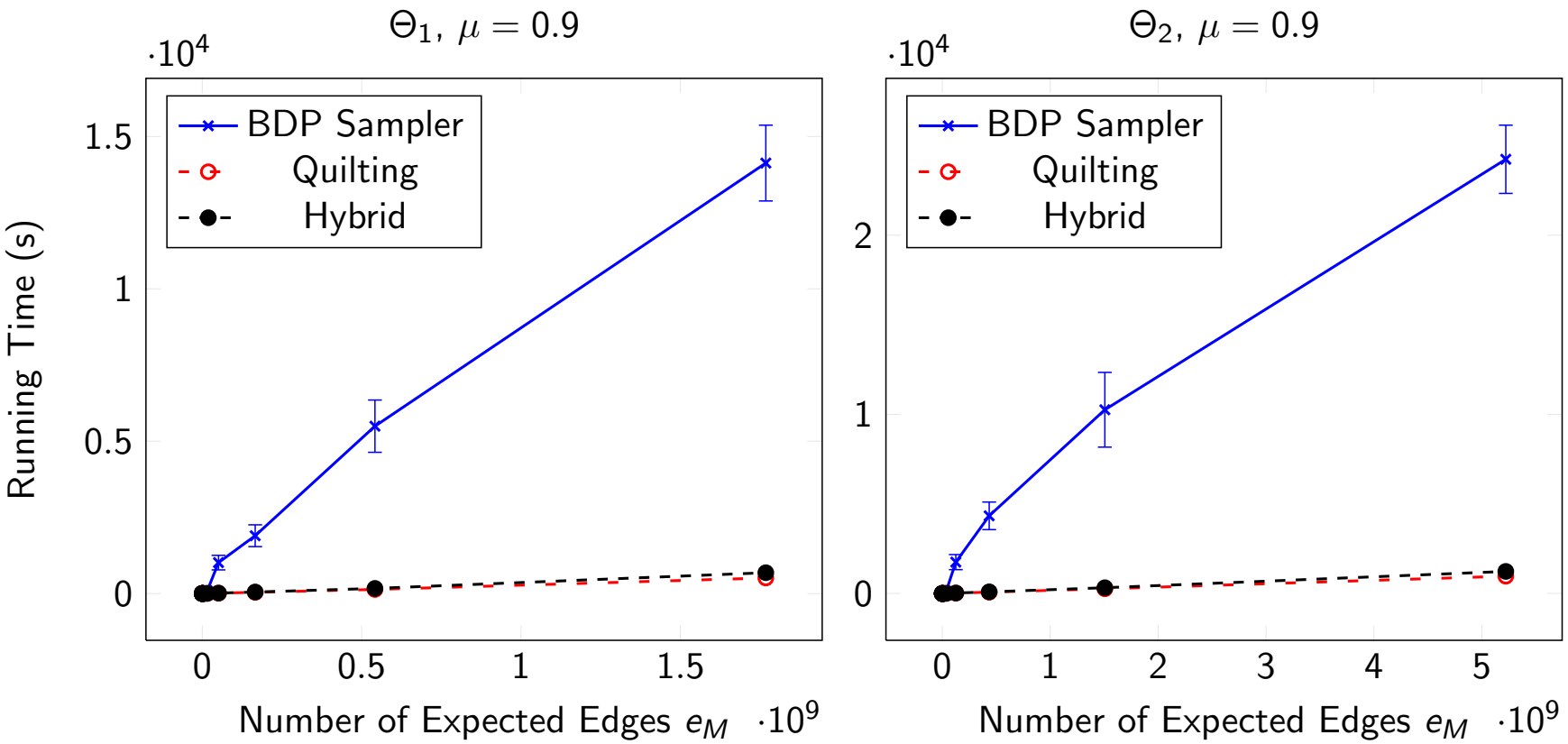
# Running Time as Function of Number of Expected Edges



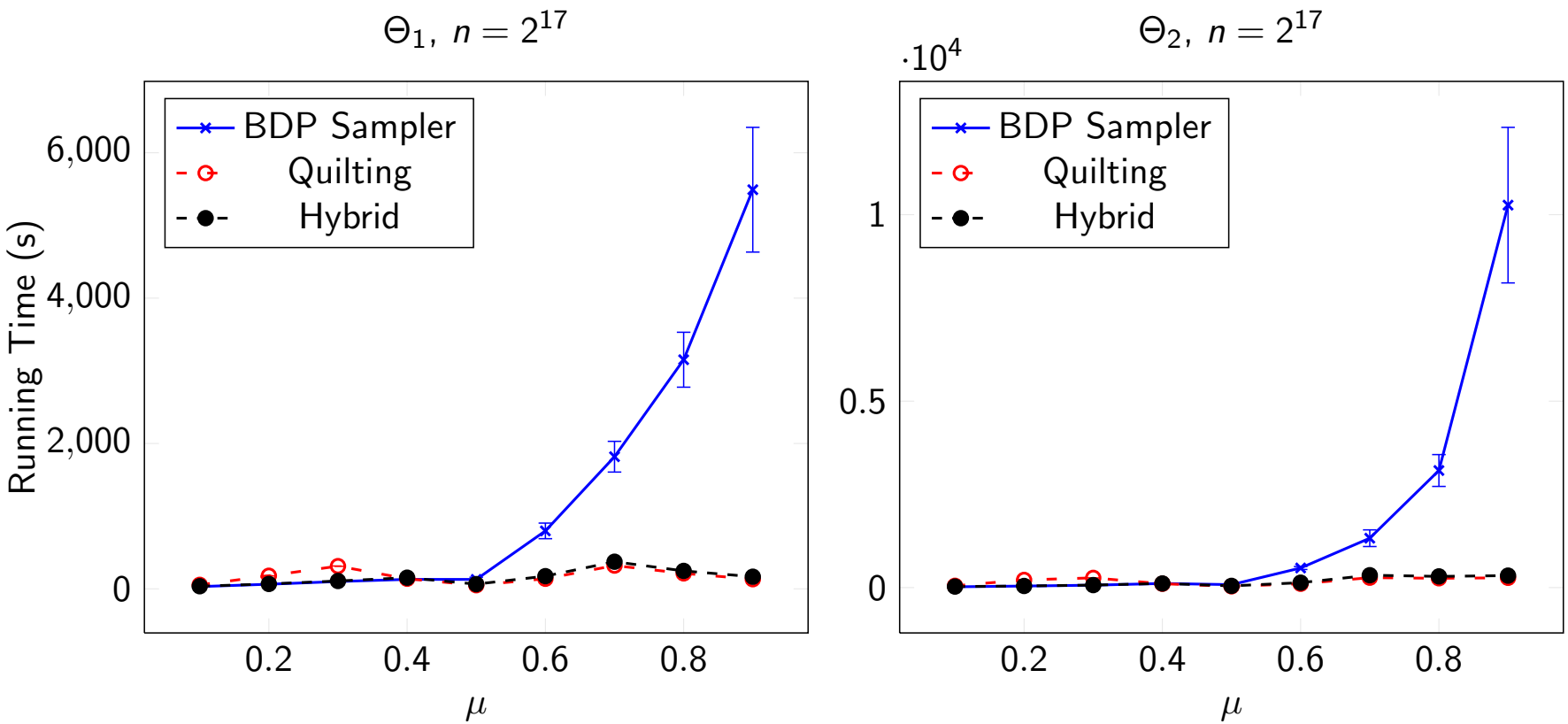
# Running Time as Function of Number of Expected Edges



# Running Time as Function of Number of Expected Edges



# Running Time as Function of $\mu$



# Outline

# Summary

**Question**

How to *efficiently* sample graphs from the Multiplicative Attribute Graph Model?

- We introduce the first sub-quadratic sampling algorithm for sampling Multiplicative Attribute Graphs
- **Time complexity:** $O\left(\left(\log_2(n)\right)^3 |E|\right)$ under some mild conditions
  - $n$ : the number of *nodes* in the graph
  - $|E|$ : number of *edges*
- Exploit the close connection between Kronecker Product Graph Model (KPGM) and Multiplicative Attribute Graph Model (MAGM)
- Can sample a graph with 8 million nodes and 20 billion edges in under 6 hours (naïve algorithm will take 93 days)

# Open Question

- Can we remove the runtime dependence on $e_K$?
- Dependence on $e_K$ can be pathological when $d \gg \log_2 n$.

# Acknowledgments