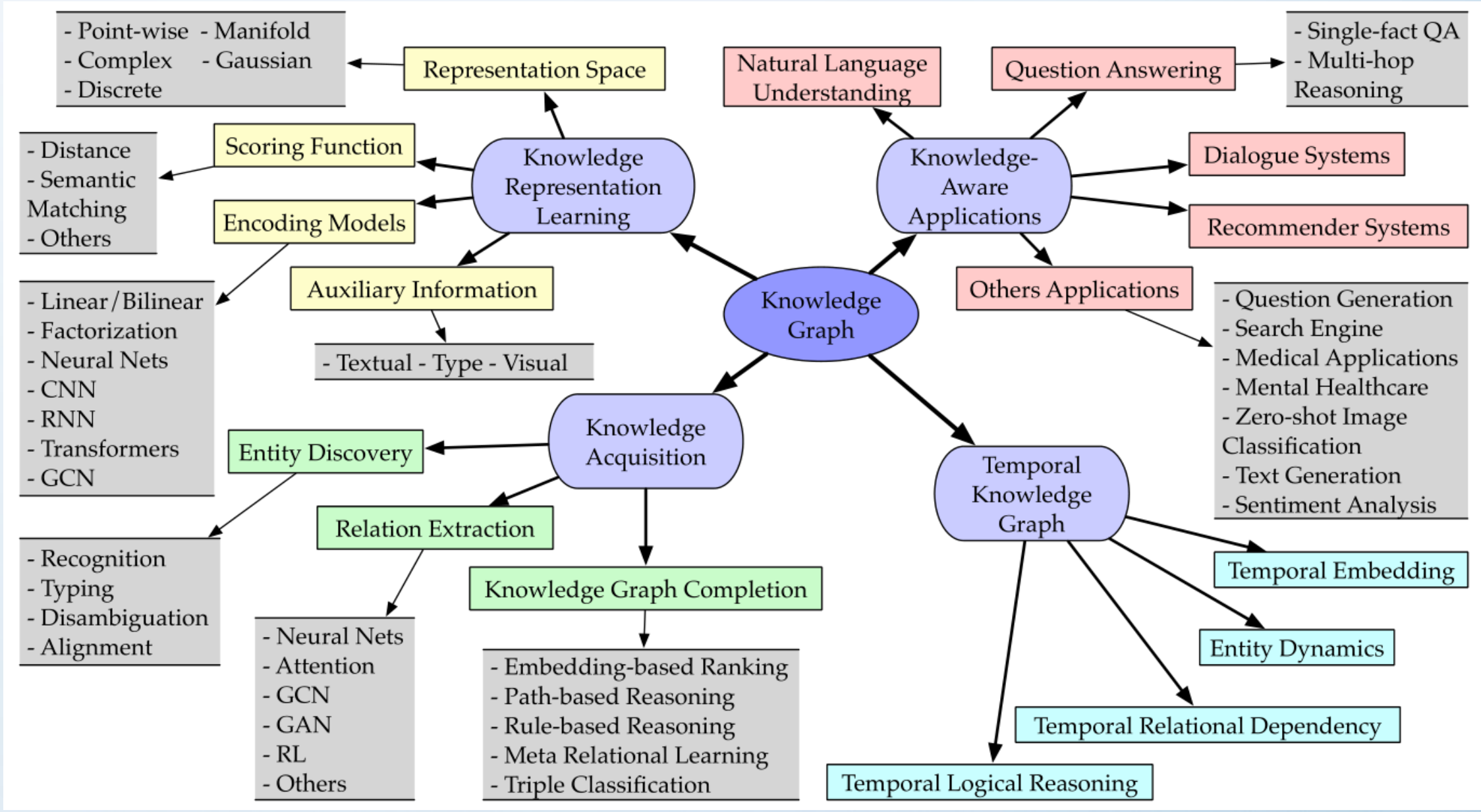
# Обработка графов знаний с использованием методов машинного обучения

# Графы знаний

- Графы знаний как средство улучшения искусственного интеллекта
- Граф знаний – это база знаний, использующая графовую (мульти, гипер-графовую) модель. Основная задача – обеспечение полноты знаний.
  - Тезаурус – специализированная база знаний, предназначенная для хранения (служебной) лингвистической информации.
  - Онтология – прежде всего схема данных, ориентированная на логический вывод. Основная задача – обеспечение непротиворечивости знаний.
- Наиболее известные проекты:
  - DBpedia
  - Yago
  - ConceptNet
    - Статья
  - Atomic 2020 (ориентирован на ситуации и причинно-следственные связи)
- Задачи, решаемые на графах знаний:
  - Предсказание вершин и связей.
  - Логический вывод на графе знаний (эта задача также решается на онтологиях)
  - Основой решения задач машинного обучения является embedding (векторное представление) графов знаний.
    - Статья «A Survey on Knowledge Graphs: Representation, Acquisition and Applications» https://arxiv.org/abs/2002.00388
    - Статья «Knowledge Graph Embedding: A Survey of Approaches and Applications» https://www.researchgate.net/publication/319947524_Knowledge_Graph_Embedding_A_Survey_of_Approaches_and_Applications

# Направления исследований



# Векторное представление графа знаний – 1

- Векторное представление (эмбеддинг) графа знаний – это математическое преобразование графа в вектор или в набор векторов в заданном векторном пространстве. Его можно проводить отдельно для вершин графа, для вершин и ребер графа, и даже для всего графа целиком. В первых двух случаях результатом будет набор векторов, в последнем – один вектор для целого графа. Главное условие состоит в том, что такое преобразование должно адекватно передавать семантику и топологию исходного графа. Эмбеддинг позволяет получить сразу несколько преимуществ при работе с графом знаний:
    1. Первое преимущество заключается в оптимизации использования памяти, ведь вектор – это сжатое представление информации из графа. Для действительно больших графов использование матрицы смежности напрямую становится сложно выполнимым, ведь для миллиона вершин размерность матрицы смежности составляет миллион в квадрате ячеек. Эмбеддинг в данном случае является более применимым решением, так как представляет узлы графа с помощью векторов гораздо меньшей размерности.
    2. Второе преимущество состоит в том, что мы получаем возможность использовать уже накопленный математический аппарат для методов машинного обучения, который позволяет работать с векторами.
    3. Третье преимущество состоит в производительности результирующих алгоритмов, использующих физическое представление графа знаний. В этом случае выполнение операций над векторами более производительно, чем выполнение операций над традиционной графовой моделью в виде множеств вершин и ребер. Кроме того, на сегодняшний день видеокарты позволяют еще больше ускорить работу с векторами.

# Векторное представление графа знаний – 2

- Процесс эмбеддинга графа знаний можно условно разделить на три этапа:

1. Выбор того, каким образом будут представлены в векторном пространстве сущности и отношения. На данном этапе происходит задание способа представления сущностей и отношений в непрерывном векторном пространстве. Сущности обычно представляются в виде векторов (точек в векторном пространстве), а отношения задаются в виде операторов в этом векторном пространстве. В зависимости от используемых методов, отношения могут быть заданы в виде векторов (например, вектор суммы или разности), матриц, тензоров, Гауссовских распределений и их смесей.

2. Задание функции правдоподобия преобразования (англ. scoring function). Для каждого RDF-триплета ***<h, r, t>*** задается функция правдоподобия ***f(h, t)***. Чем больше значение этой функции, тем более вероятно то, что факт, описываемый триплетом, является истинным. Таким образом, в соответствии с функцией правдоподобия, факты, содержащиеся в графе знаний, набирают больше баллов, чем факты, которых нет в графе знаний.

3. Обучение модели эмбеддинга подразумевает решение оптимизационной задачи, заключающейся в максимизации суммарной функции правдоподобия для всех триплетов, содержащихся в графе знаний.

# Векторное представление графа знаний – 3

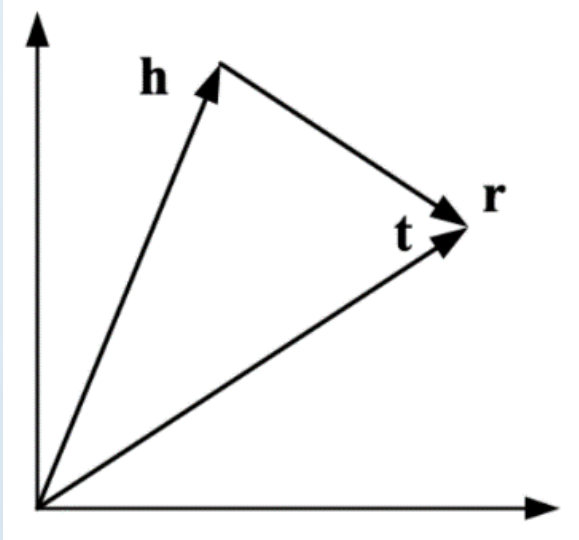
- На сегодняшний день существуют десятки различных техник эмбеддинга. Они различаются способами представления сущностей и отношений в векторном пространстве и способами задания функций правдоподобия. Основные модели эмбеддинга можно разделить на две группы: модели параллельного переноса и модели семантического соответствия:

1. Модели параллельного переноса (англ. «translational distance models», TDM) используют функции правдоподобия, основанные на измерении расстояния между двумя вершинами графа. Причем оператор перемещения из одной вершины в другую задается на основе отношения между этими узлами в исходном графе.

2. Модели семантического соответствия (англ. «semantic matching models», SMM) используют функции правдоподобия, основанные на семантических характеристиках вершин и отношений исходного графа.

# Модель TransE (TDM)

- Эта модель представляет и узлы, и связи как векторы в одном и том же векторном пространстве. Причем оператор отношения рассматривается как вектор перемещения между субъектом и объектом. Это позволяет на интуитивном уровне уловить семантический смысл операции сложения над векторами: если к субъекту прибавить отношение, то результат будет примерно равен объекту.

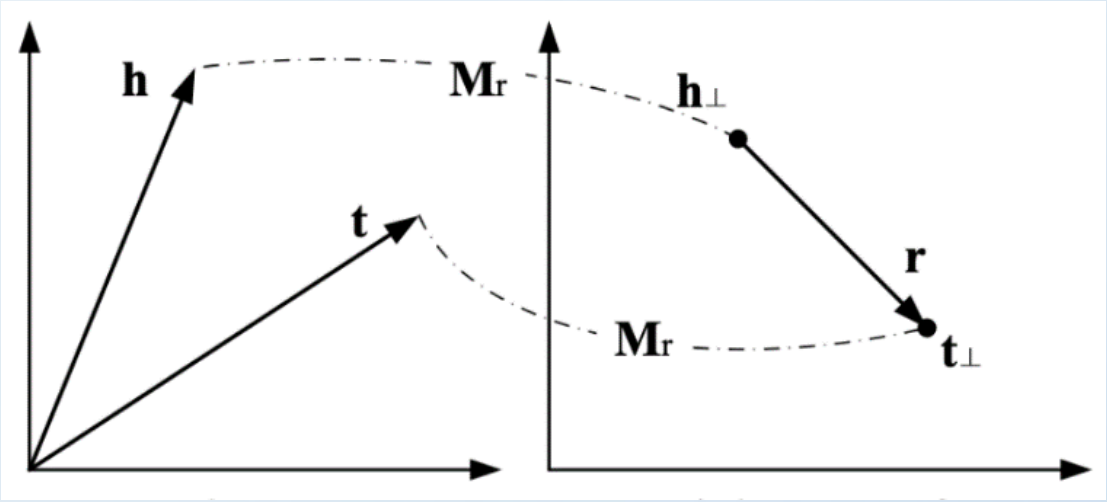



- Основным недостатком модели TransE является описание отношений один-ко-многим и много-ко-многим. Например, если у триплетов совпадают значения субъекта или объекта, то в этом случае будут сгенерированы почти совпадающие эмбеддинги.

# Модель TransR (TDM)

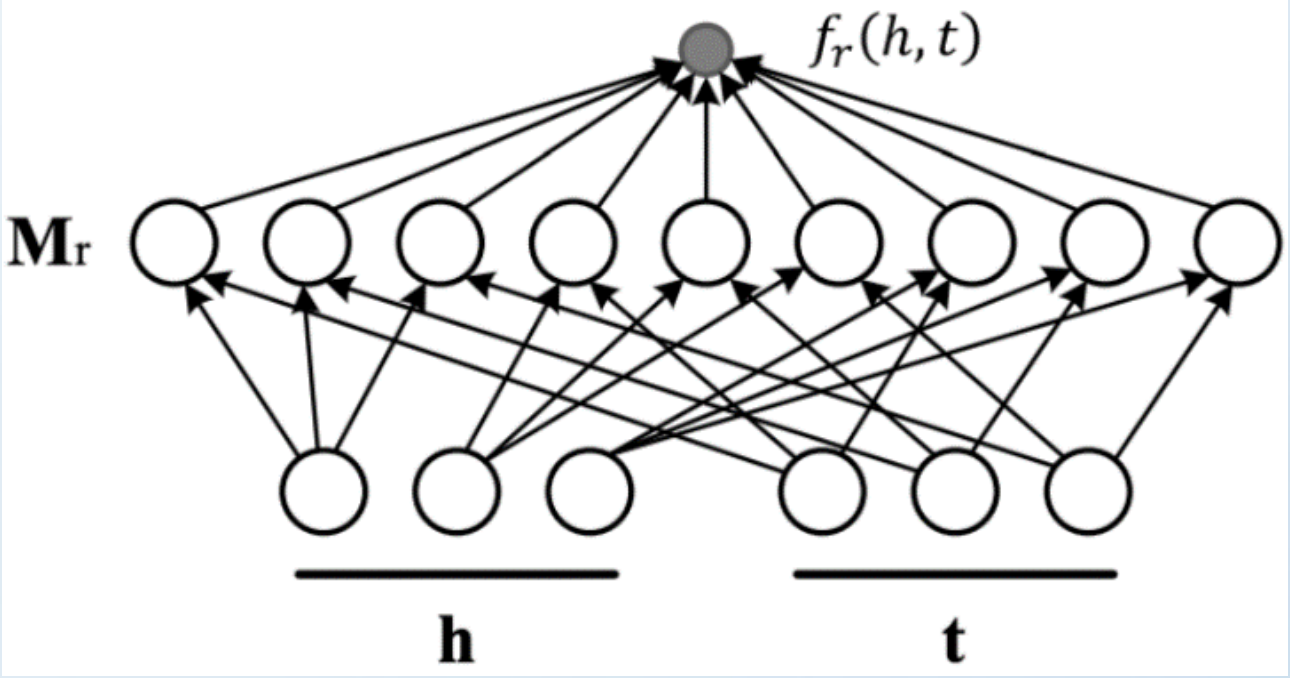
- Модель очень похожа на TransE, но позволяет преодолеть недостатки описания отношений один-ко-многим и много-ко-многим. Для этого модель использует два пространства, одно для встраивания сущностей, а второе для встраивания отношений. При этом, у сущности будут различные проекции на пространство отношений в зависимости от того, в каких отношениях используется данная сущность.

- Основной недостаток модели TransR состоит в том, что из-за необходимости вычисления проекционной матрицы для каждого отношения, модель теряет простоту и эффективность TransE. Обучение модели TransR займет больше времени и ресурсов чем обучение модели TransE.



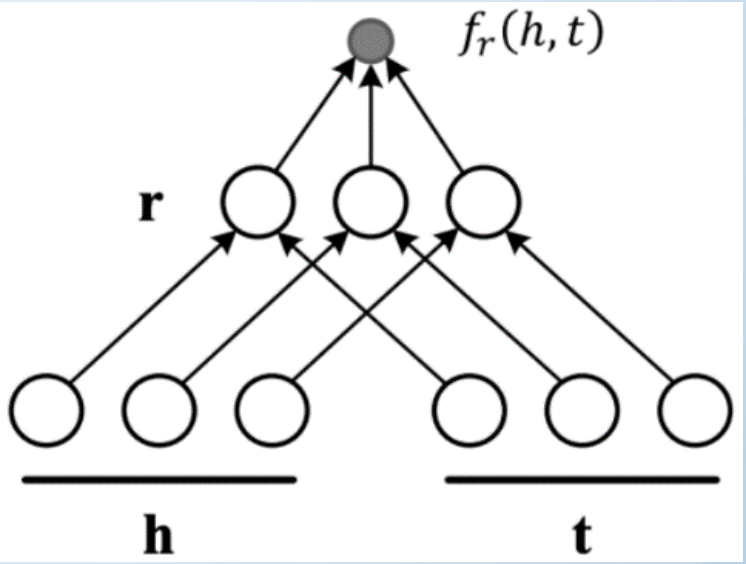
# Модель RESCAL (SMM)

- Модель ассоциирует каждую сущность с вектором и старается передать скрытые смыслы (факторы) этой сущности. Каждое отношение представлено в виде матрицы, которая моделирует попарные взаимодействия скрытых факторов.
- К недостаткам модели можно отнести ее вычислительную сложность, связанную с большой размерностью матрицы отношений ***Mr***.



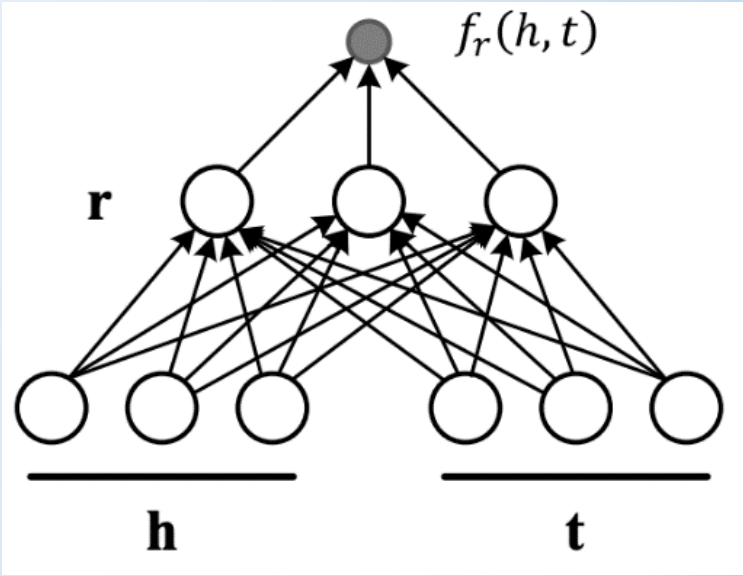
# Модель DistMult (SMM)

- Упрощает модель RESCAL, накладывая ограничение на матрицу отношений. В DistMult матрица отношений обязательно должна быть диагональной для каждого отношения ***r***.
- Функция правдоподобия отражает попарные взаимодействия только между компонентами сущностей, лежащими в одинаковых измерениях. За счет этого сложность модели становится линейной, а не квадратичной. Однако эта модель слишком сильно упрощена и может работать только с симметричными отношениями. Этого явно недостаточно для абстрактного графа знаний, в котором отношения могут быть произвольными.



# Модель HolE (SMM)

- Модель голографического эмбеддинга. Она совмещает в себе мощность RESCAL и простоту DistMult. В HolE к представлениям сущностей сначала применяется оператор взаимной корреляции. После этого, для определения значения функции правдоподобия, полученный вектор совмещается с вектором представления связи.
- Взаимная корреляция сжимает попарные взаимодействия компонентов сущностей. Поэтому сложность у модели линейная, что эффективнее, чем у RESCAL. Но при этом, модель HolE может работать с асимметричными отношениями, так как взаимная корреляция не коммутативна. Эта особенность недоступна в модели DistMult.



# Векторное представление гиперграфа – 1

- Статья Hyperedge2vec: Distributed Representations for Hyperedges

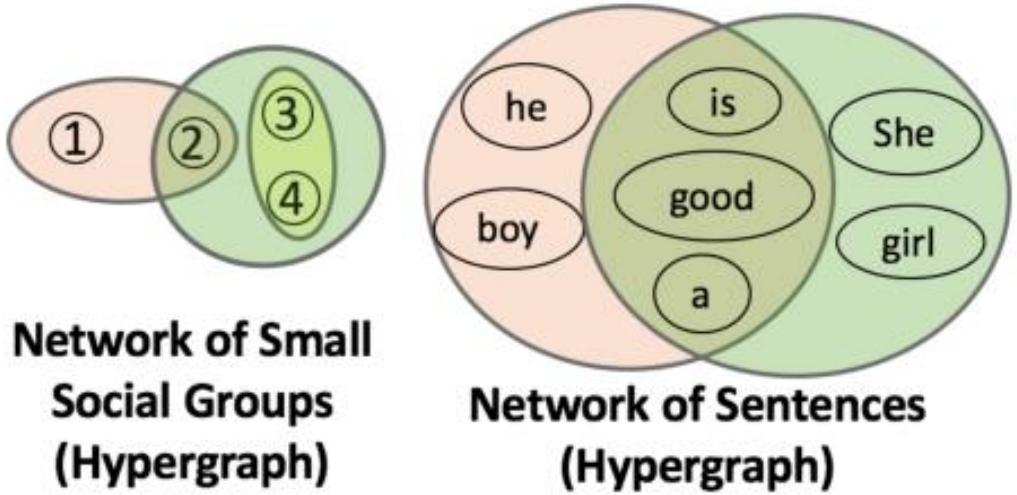
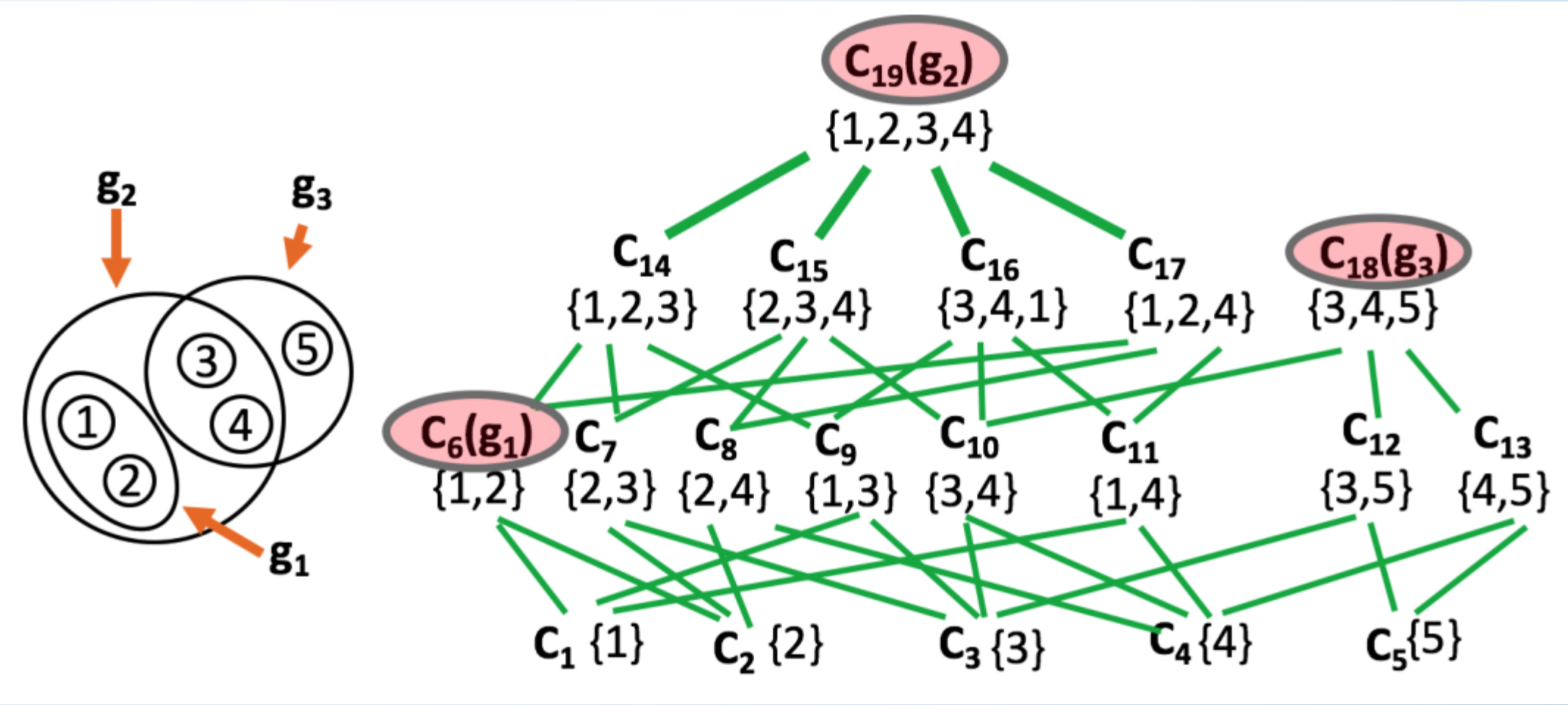



**Figure 1: Example illustrating "set-like" hypergraph structure from two domains. Left is a collaboration network between four individuals and on right is a network resulting from two sentences: "He is a good boy" & "She is a good girl"; and eight word nodes.**

# Векторное представление гиперграфа – 2

- Структура гиперребер может быть представлена в виде диаграммы Хассе.



# Векторное представление гиперграфа – 3

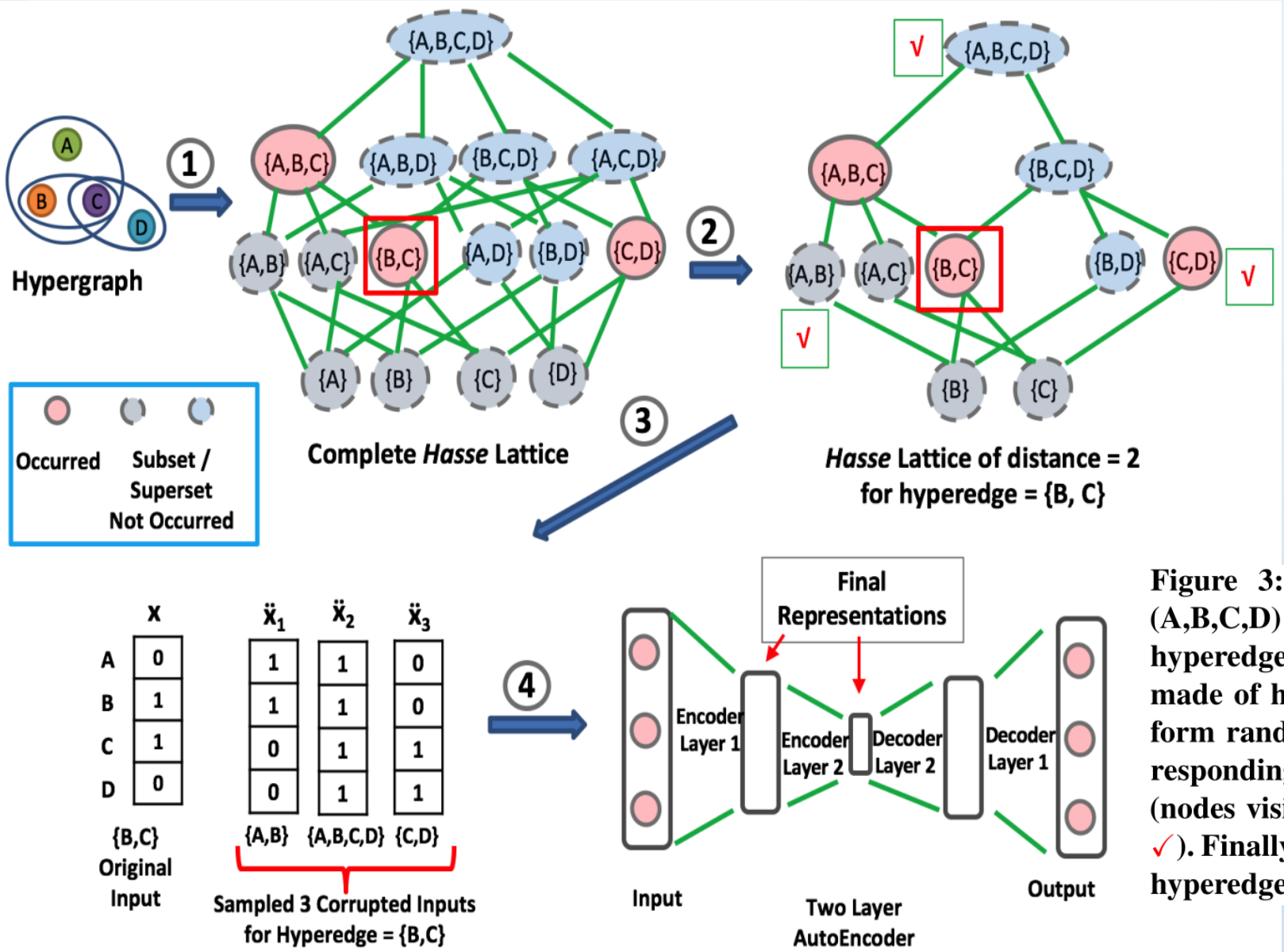



**Figure 3:** For the given hypergraph between four nodes (A,B,C,D) we consider the complete *hasse* lattice. For a given hyperedge {B,C} (square box) we then construct the sub-lattice made of hyperedges with distance $h = 2$ from {B,C}. We perform random walk (with $\tau = 0.2$) starting from the node corresponding to hyperedge {B,C} and sample $p = 3$ hyperedges (nodes visited by the random walk; shown with a check-mark ✓). Finally, we train the autoencoder to reconstruct the original hyperedge from these $p$ noisy hyperedges.

# Графовые нейронные сети (GNN)

- Обзорная статья профессора А.Г. Дьяконова
- Общие принципы работы GNN
- Статья с примерами кода «What Are Graph Neural Networks? How GNNs Work, Explained with Examples»
- Серия статей «Graph Representation Learning»
- Graph Convolutional Network (GCN)
  - Краткая статья «Graph Convolutional Networks – Deep Learning on Graphs»
  - Детальная статья «Graph convolutional networks: a comprehensive review»
- Graph Attention Network (GAT)
  - Статья Understanding Graph Attention Networks (GAT)
- Графовые трансформеры
  - Статья с пояснением
  - Оригинальная статья

# Гиперграфовые нейронные сети – 1

- Статья «Hypergraph Neural Networks»

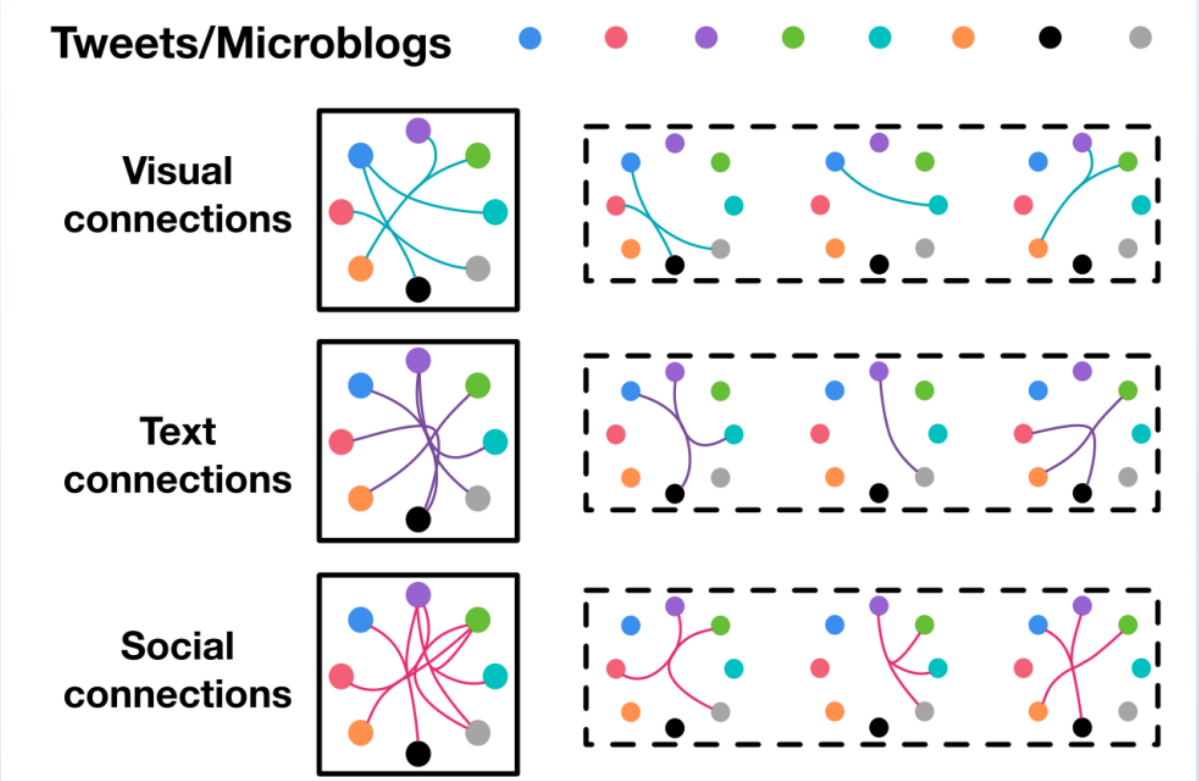
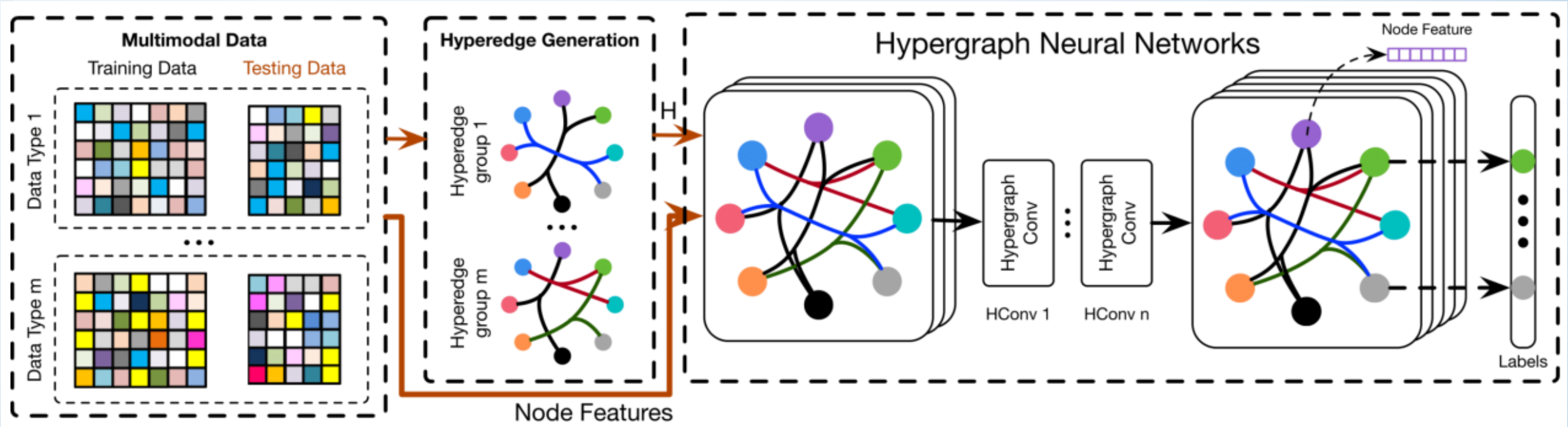



Figure 1: Examples of complex connections on social media data. Each color point represents a tweet or microblog, and there could be visual connections, text connections and social connections among them.

# Гиперграфовые нейронные сети – 2




1. Мультимодальный набор данных делится на обучающие и тестовые выборки, каждая выборка содержит набор вершин с их признаками.
2. Далее на основе выборок (с учетом их корреляции) генерируются группы гиперребер.
3. Группы гиперребер объединяются, чтобы сформировать общую матрицу смежности гиперграфа.
4. Матрица смежности и признаки вершин подаются на вход HGNN-сети, на выходе сети формируются метки выходных вершин.

# Гиперграфовые нейронные сети – 3

HGNN-сеть использует сверточные слои:

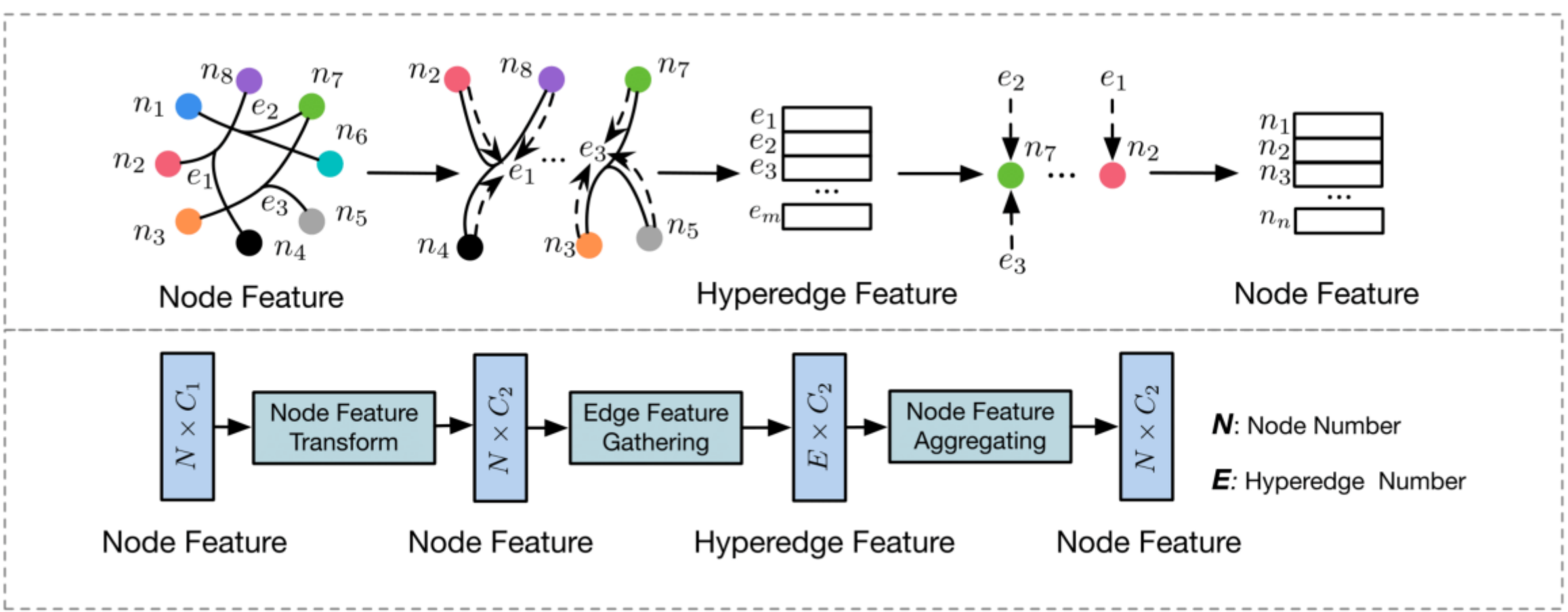



Figure 4: The illustration of the hyperedge convolution layer.