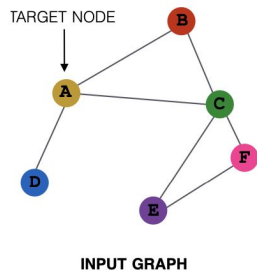


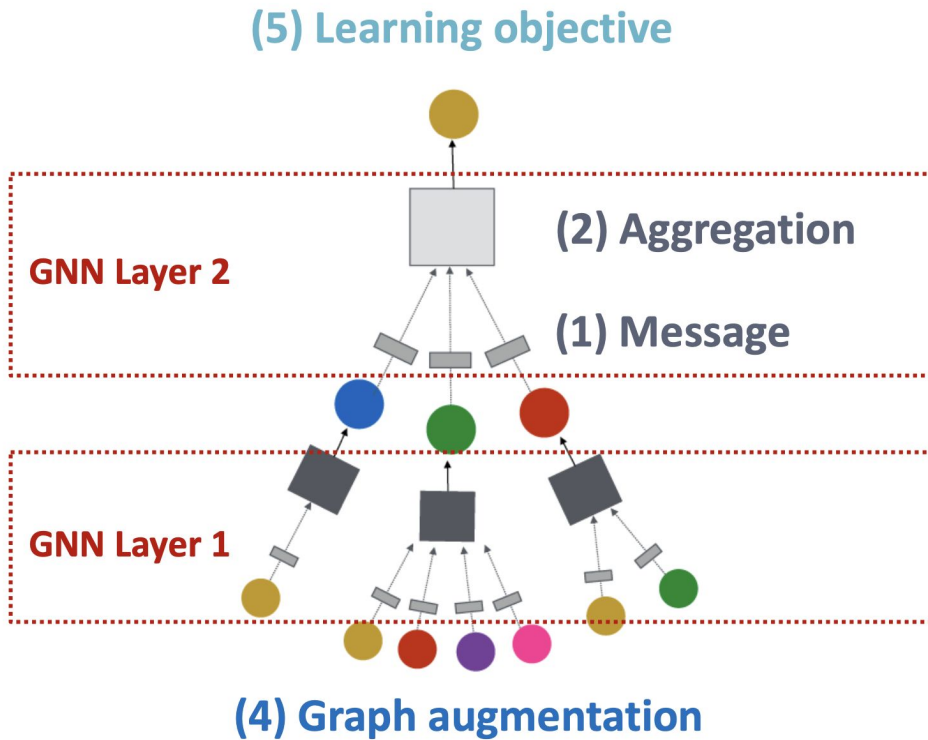
Анализ графовых данных и глубокое обучение

Азимов Рустам

В предыдущих сериях

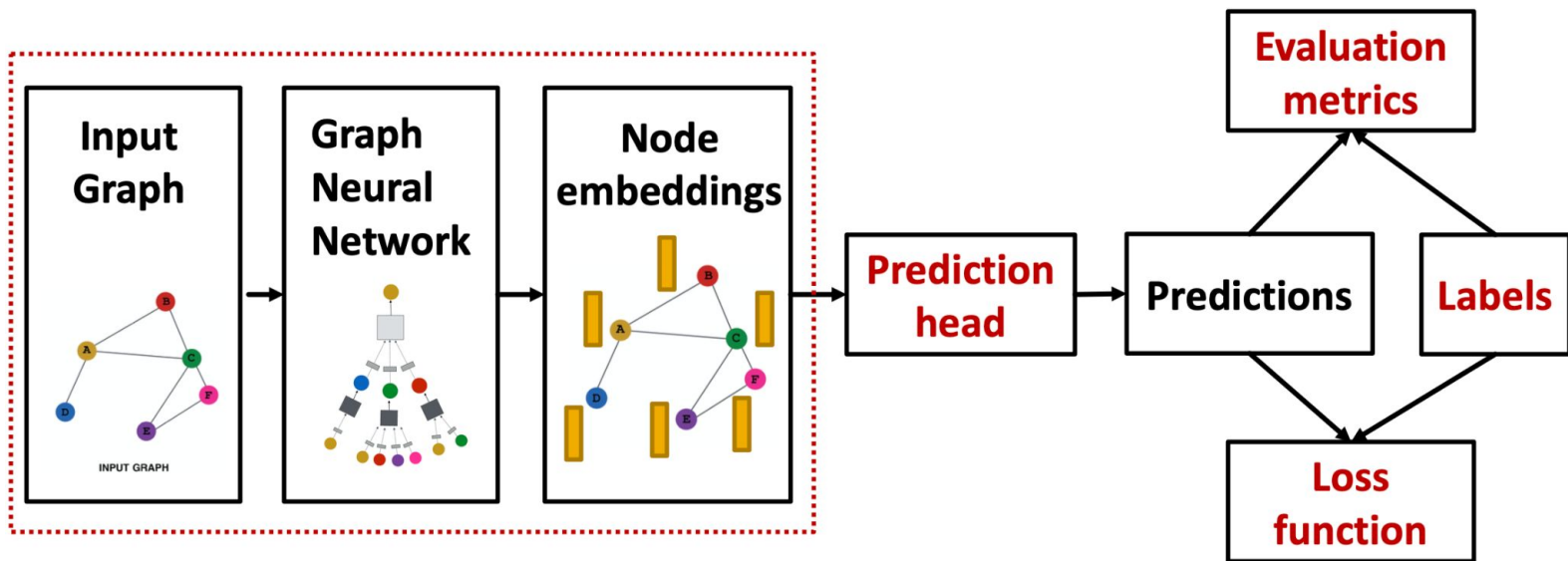


(3) Layer connectivity



GNN Training Pipeline

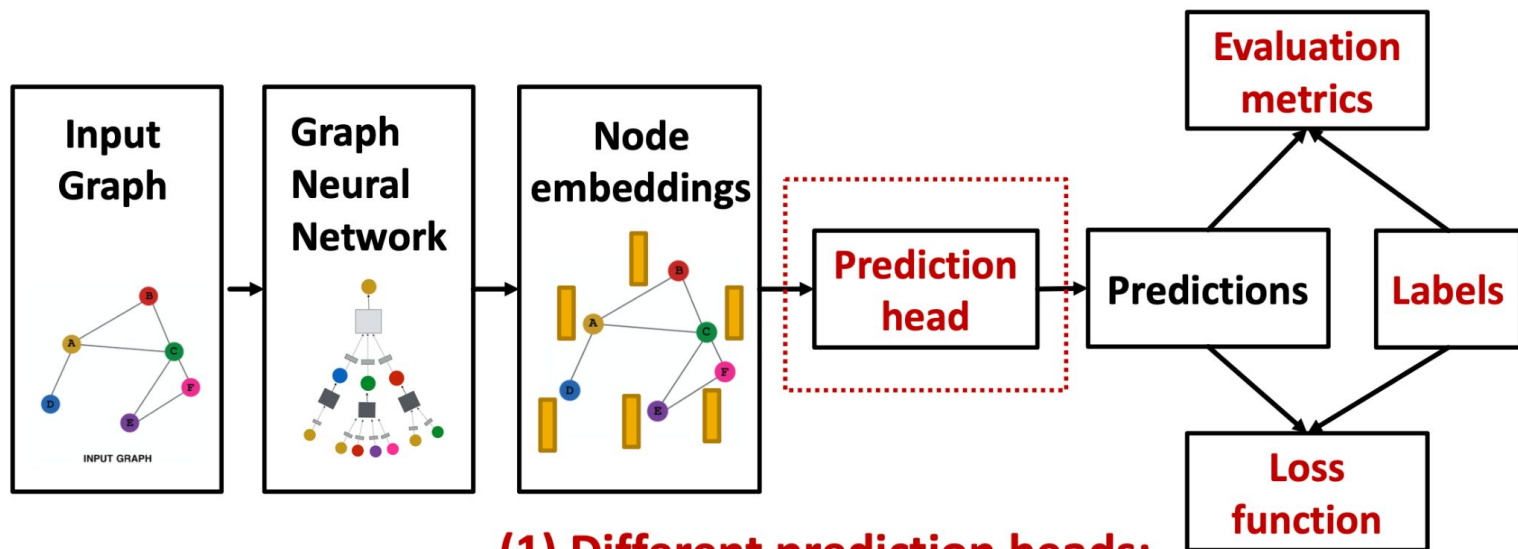
Training Pipeline



Output of a GNN: set of node embeddings

$$\{\mathbf{h}_v^{(L)}, \forall v \in G\}$$

Prediction head



(1) Different prediction heads:

- Node-level tasks
- Edge-level tasks
- Graph-level tasks

Node-level head

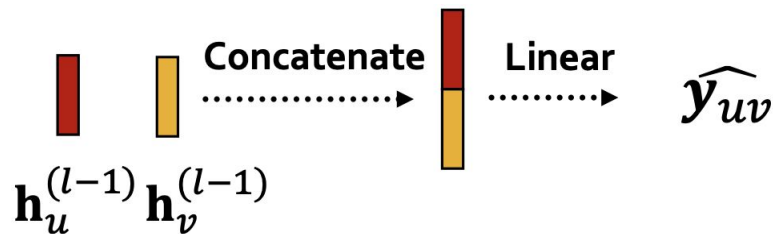
- Можно использовать линейный слой над обученными эмбедингами

$h_v^{(l)}$

$$\hat{\mathbf{y}}_v = \text{Head}_{\text{node}}(\mathbf{h}_v^{(L)}) = \mathbf{W}^{(H)} \mathbf{h}_v^{(L)}$$

- Размер линейного слоя зависит от количества целевых признаков u для задачи регрессии/классификации

Edge-level head (1)



$$\hat{\mathbf{y}}_{uv} = \text{Linear}(\text{Concat}(\mathbf{h}_u^{(L)}, \mathbf{h}_v^{(L)}))$$

Edge-level head (2)

- 1-way prediction

$$\hat{\mathbf{y}}_{uv} = (\mathbf{h}_u^{(L)})^T \mathbf{h}_v^{(L)}$$

- k -way prediction (идея, схожая с multi-head attention в GAT)

$$\hat{\mathbf{y}}_{uv}^{(1)} = (\mathbf{h}_u^{(L)})^T \mathbf{W}^{(1)} \mathbf{h}_v^{(L)}$$

...

$$\hat{\mathbf{y}}_{uv}^{(k)} = (\mathbf{h}_u^{(L)})^T \mathbf{W}^{(k)} \mathbf{h}_v^{(L)}$$

$$\hat{\mathbf{y}}_{uv} = \text{Concat}(\hat{\mathbf{y}}_{uv}^{(1)}, \dots, \hat{\mathbf{y}}_{uv}^{(k)}) \in \mathbb{R}^k$$

Graph-level head

(1) Global mean pooling

$$\hat{\mathbf{y}}_G = \text{Mean}(\{\mathbf{h}_v^{(L)} \in \mathbb{R}^d, \forall v \in G\})$$

(2) Global max pooling

$$\hat{\mathbf{y}}_G = \text{Max}(\{\mathbf{h}_v^{(L)} \in \mathbb{R}^d, \forall v \in G\})$$

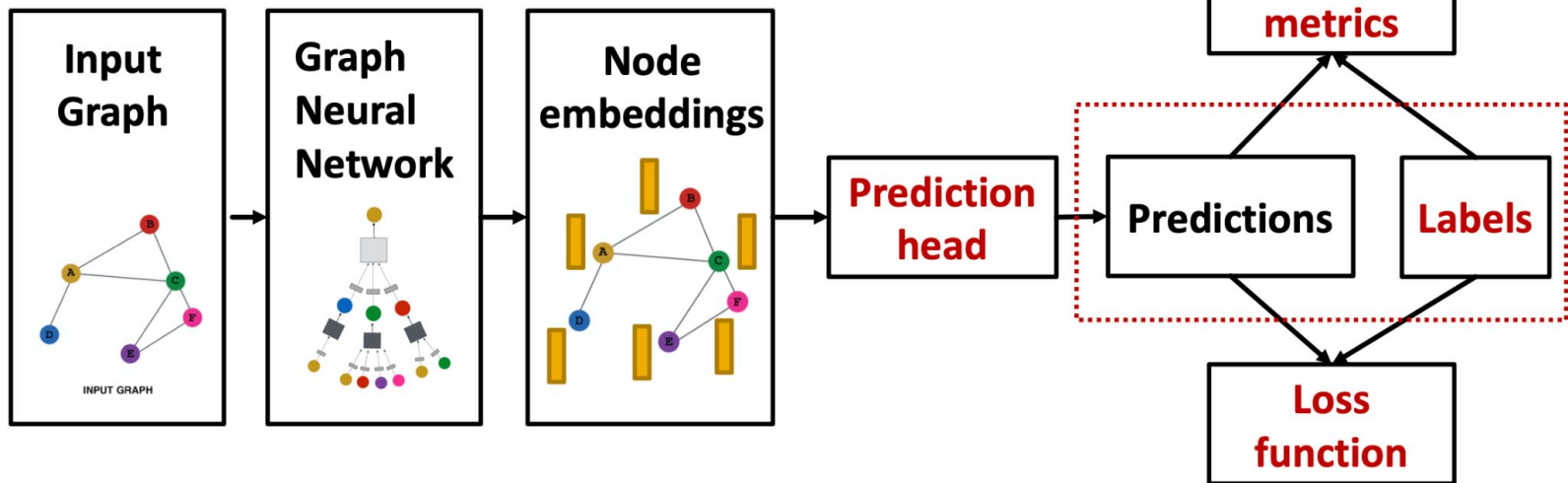
(3) Global sum pooling

$$\hat{\mathbf{y}}_G = \text{Sum}(\{\mathbf{h}_v^{(L)} \in \mathbb{R}^d, \forall v \in G\})$$

Training Pipeline

(2) Where does ground-truth come from?

- **Supervised labels**
- **Unsupervised signals**



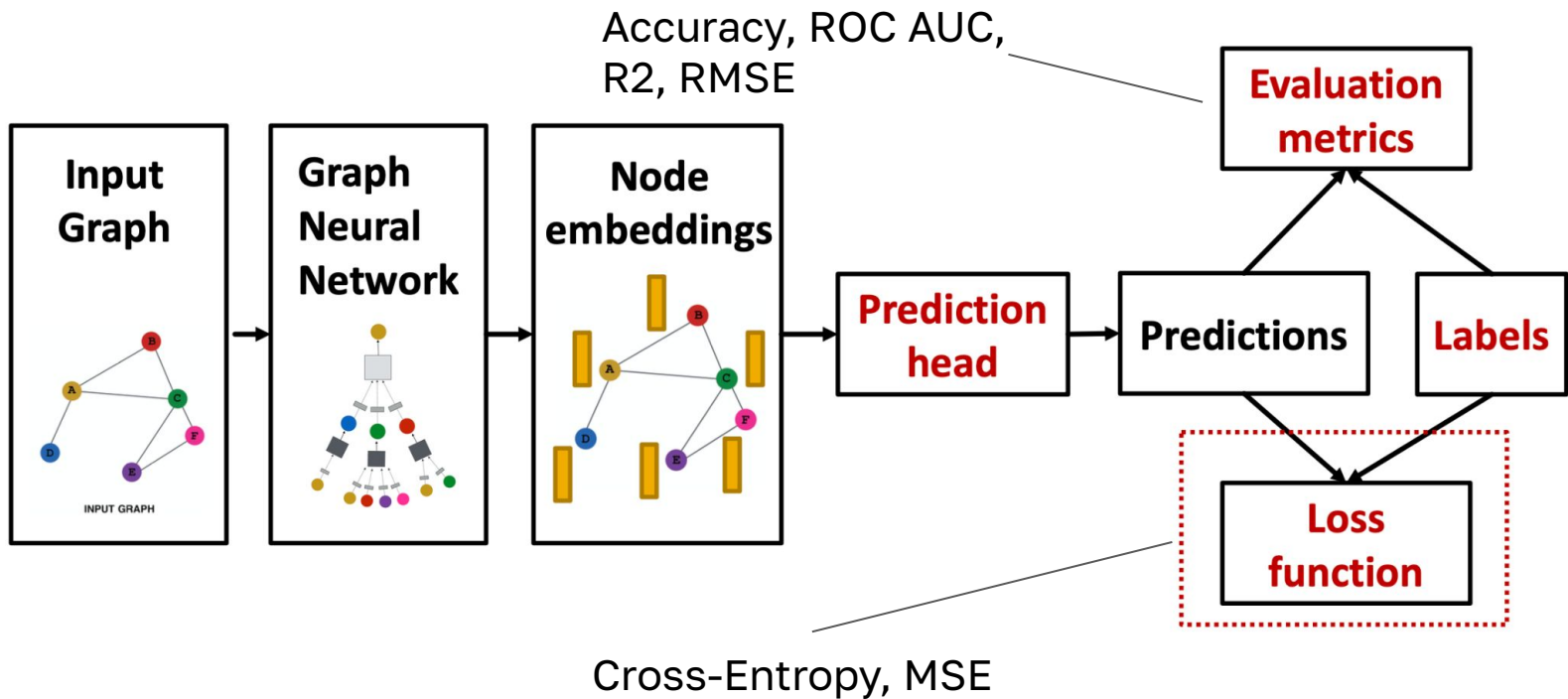
Supervised Labels

- Метки вершин, рёбер или графов уже даны
- Например, каждый граф — лекарство и известны некоторые токсичные и нетоксичные лекарства

Unsupervised Signals

- Дан только граф/графы без меток
- Можно самостоятельно выделить метки для предсказания
- Для вершин, например, предсказывать clustering coefficient, PageRank, ...
- Или к какому кластеру принадлежит вершина
- Для рёбер — убрать некоторые из них и научиться предсказывать их наличие
- Для графов, например, предсказывать изоморфны ли два графа

Training Pipeline



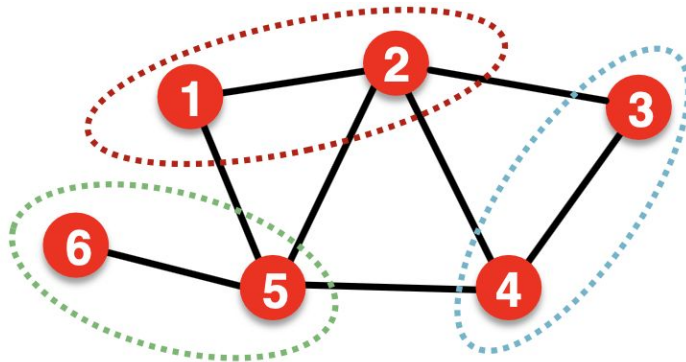
Dataset Splitting

- Как разделить набор графовых данных на train/validation/test?
- Легко, если набором является множество графов
- В чём сложность, когда имеется только один граф?

Training

Validation

Test



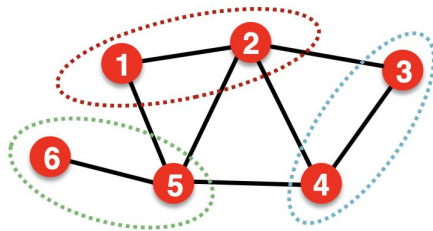
Transductive Setting

- Позволяем быть влиянию частей набора данных друг на друга
- На каждом этапе доступен весь граф
- При обучении вычисляются эмбединги с использованием всего графа, но loss оптимизируется только для тренировочных данных
- Аналогично при валидации метрику подсчитываем на валидационных данных

Training

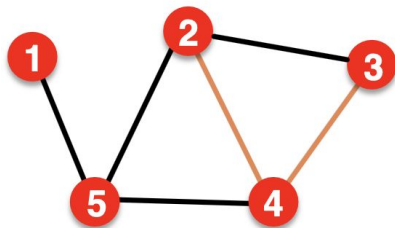
Validation

Test

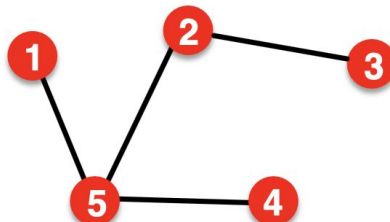


Пример: Link Prediction

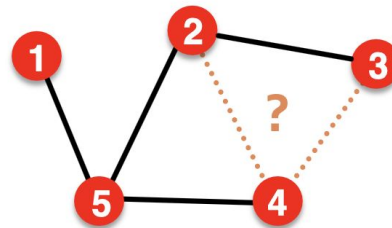
- Отдельного внимания заслуживает задача предсказания связей в графе
- Прячем некоторые рёбра, чтобы на них оптимизировать предсказания



Original graph



Input graph to GNN

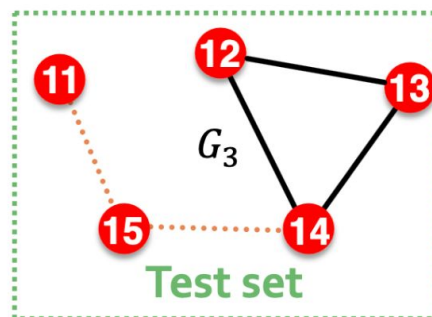
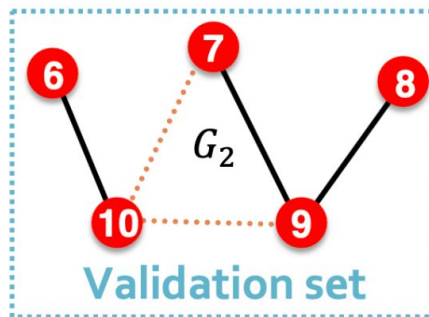
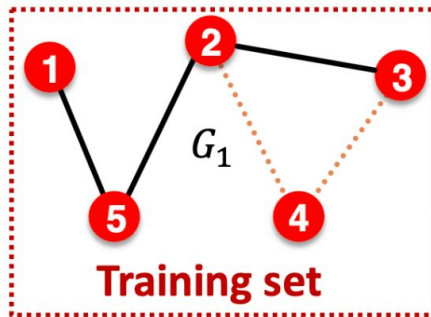


Predictions made by GNN

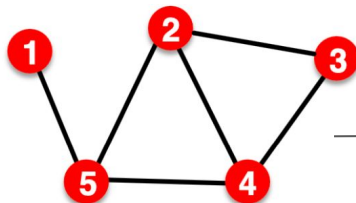
Inductive Link Prediction

Message
edge —

Supervision
edge ·····

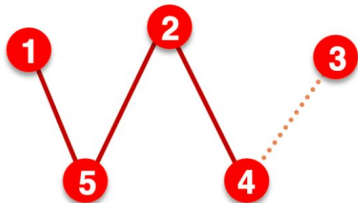


Transductive Link Prediction

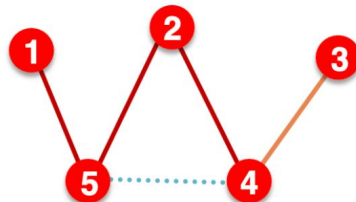


The original graph

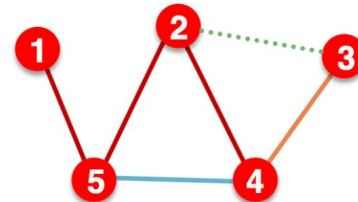
Training message edges
Training supervision edges
Validation edges
Test edges



(1) At training time:
Use **training message edges** to predict **training supervision edges**

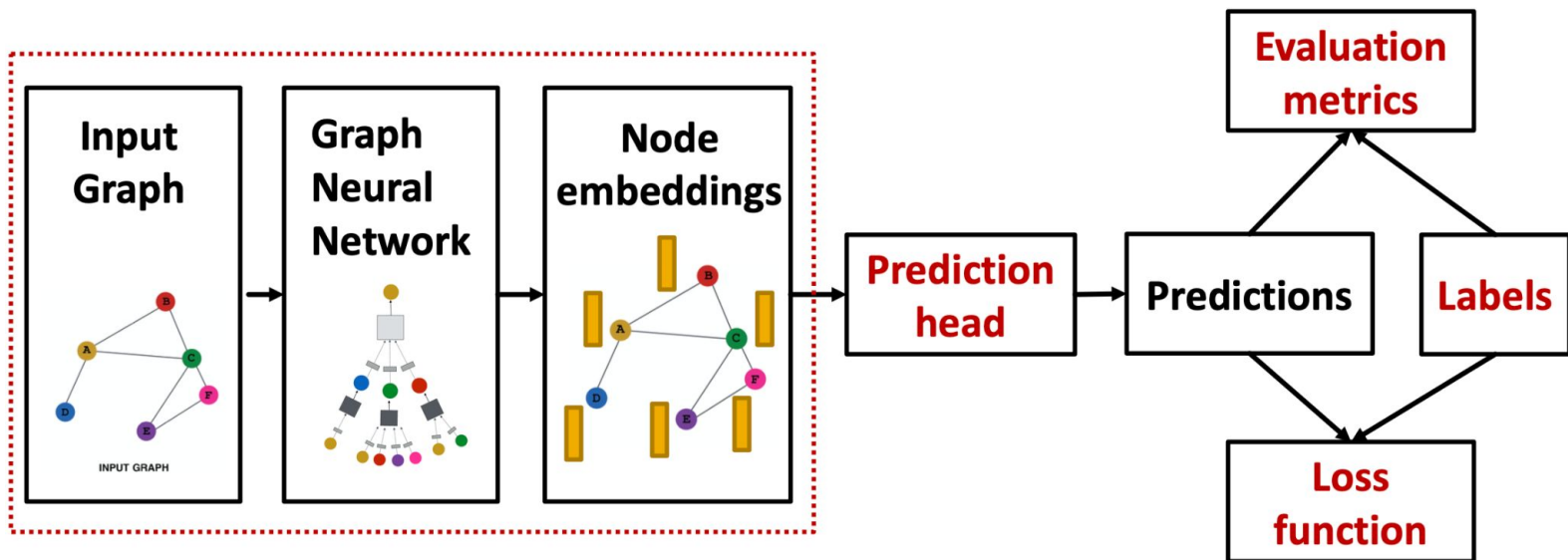


(2) At validation time:
Use **training message edges** & **training supervision edges** to predict **validation edges**



(3) At test time:
Use **training message edges** & **training supervision edges** & **validation edges** to predict **test edges**

Заключение



Output of a GNN: set of node embeddings

$$\{\mathbf{h}_v^{(L)}, \forall v \in G\}$$