Knowledge Graph for Machine Learning

Timon Zimmermann, Rhicheek Petra
Oracle Labs
Mar, 2019

ORACLE

Overview

1) Background & Problem Statement

2) Healthcare Dataset & Application

3) Baseline: Architecture & Application

4) KG-RNN: Architecture & Application

5) Results

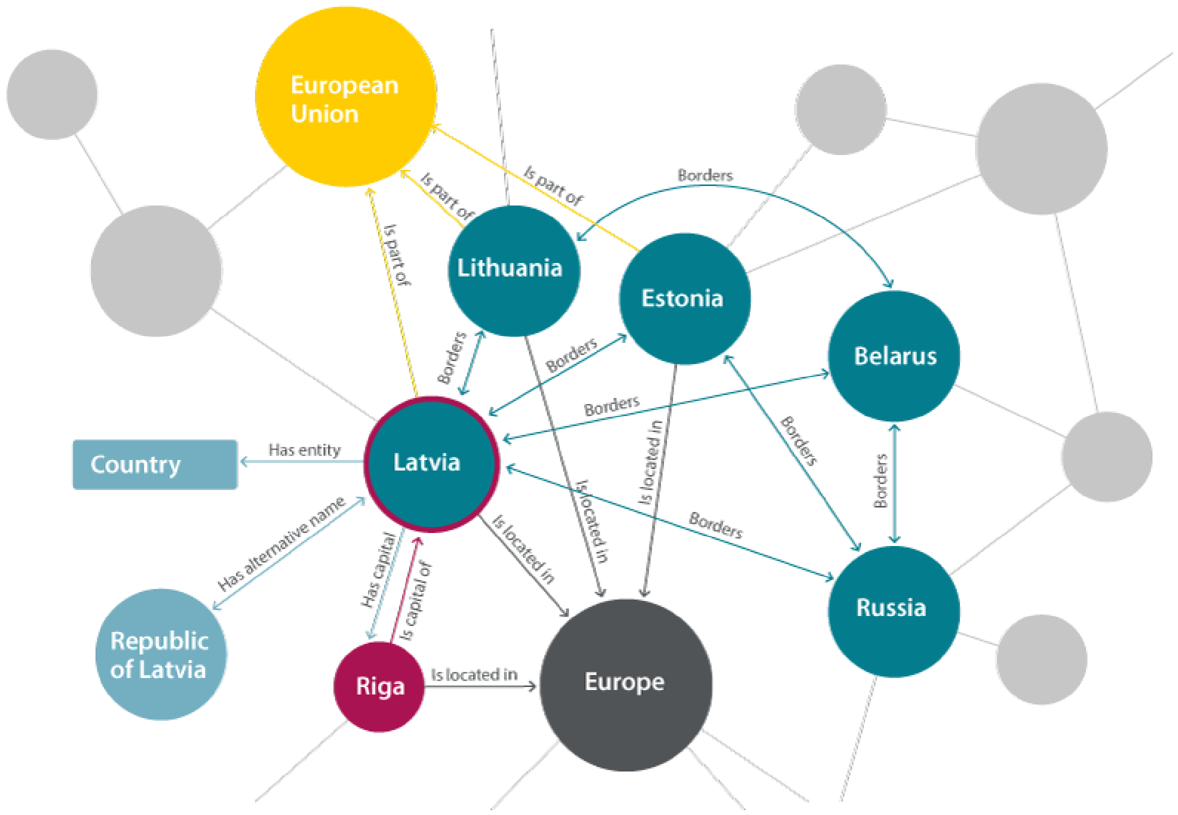
6) Concluding Remarks

ORACLE

Background & Problem Statement

Knowledge Graph

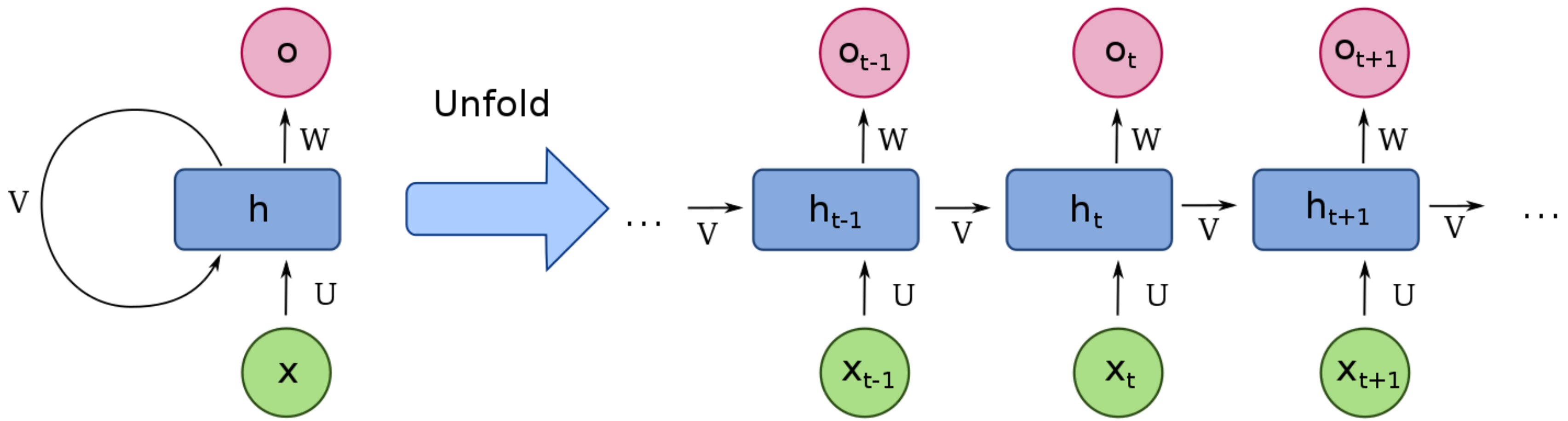
- Represent information as entities, properties and relations between entities
- Allows to **naturally** combine information from different sources
- Lot of activity and research around leveraging knowledge graphs in Machine Learning



ORACLE

Recurrent Neural Network

Allows to exploit the temporal dynamic behavior of a sequence



Use-Case Examples

- Stock Market Prediction
- Machine Translation

Machine Learning on Evolving Knowledge Graph

- How to learn from knowledge graph with a dynamic component?
  → Changing Knowledge Graph structure over time
  → Dynamic properties in entities

- **Idea**: Feed evolving entity to a Machine Learning model as a **sequence of entity states**

- **Solution**: Fusion of Graph Machine Learning and Recurrent Neural Network

ORACLE

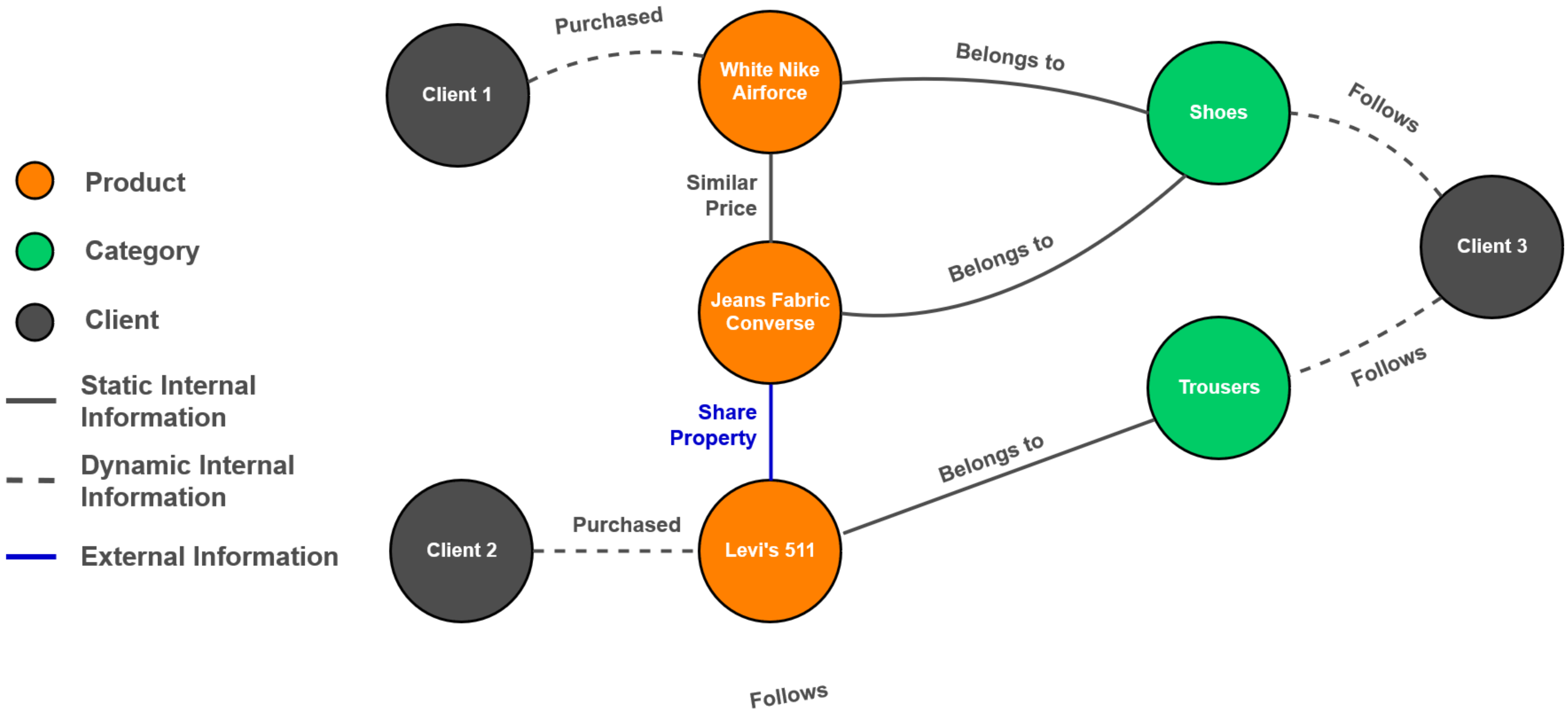
Evolving Knowledge Graph – Example Online shop

- Entities and Properties:
  - Client (First name, Last name)
  - Category (Name)
  - Product (Name, Price, Properties)

- Relationships:
  - Client – *purchased* – Product
  - Client – *follows* – Category
  - Product – *belongs to* – Category
  - Product – *shares property* – Product
  - Product – *has similar price* – Product

ORACLE

Evolving Knowledge Graph – Example Online shop



ORACLE

Healthcare Application

Dataset

ORACLE

Healthcare Dataset – Admission Point-of-View

- 4 kinds of event over time:
  - Fluids into patient
  - Fluids out of the patient
  - Lab test results
  - Drugs prescribed

- Medical codes (**ICD-9**) describing the diagnoses and procedures at discharge

Examples:

- Insulin, 5 ml/h
- Urine, 3ml at 10:30
- pH, 7.4 at 15:35
- Aspirin, 500mg/day

ICD-9 Codes:

- 401.0: Malignant essential hypertension
- 403.9: Unspecified hypertensive renal disease
- …

ORACLE

Machine Learning in Healthcare

Problem:

Analyzing thousands of events over time is very difficult for physicians:

- Humans can only process and correlate 4 variables at the same time
- Historical information from previous admissions of other patients 12 years ago, potentially at other locations

Goal:

Predict the medical codes that will be diagnosed at discharge of the patient

ORACLE

MIMIC-III

Dataset: MIMIC-III, v1.4

- MIMIC is a relational database containing data of patients who stayed within the intensive care units at Beth Israel Deaconess Medical Center
- Commonly found in the scientific research & literature

- 46’520 patients
- 58’976 admissions
- 21’000’000 input events
- 4’500’000 output events
- 28’000’000 lab tests
- 4’000’000 prescriptions

Input

We consider **admissions** as our main entity composed of many **events**

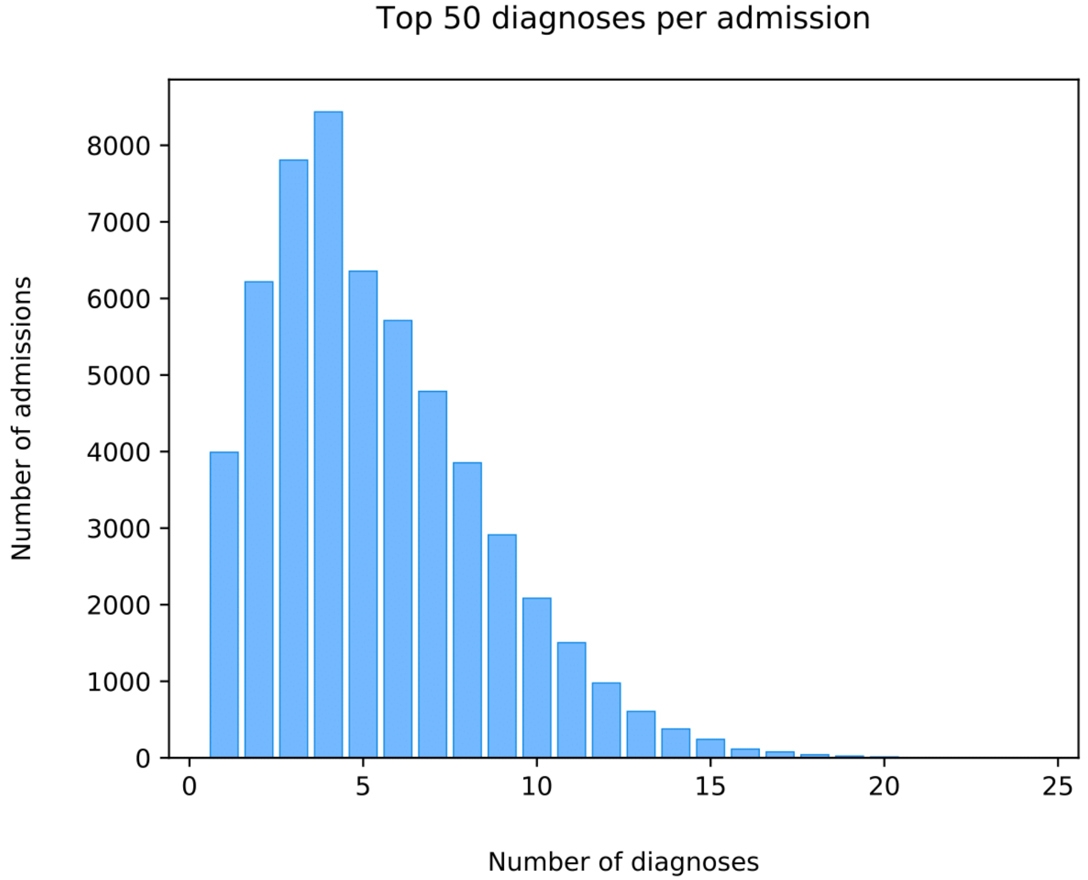
Output

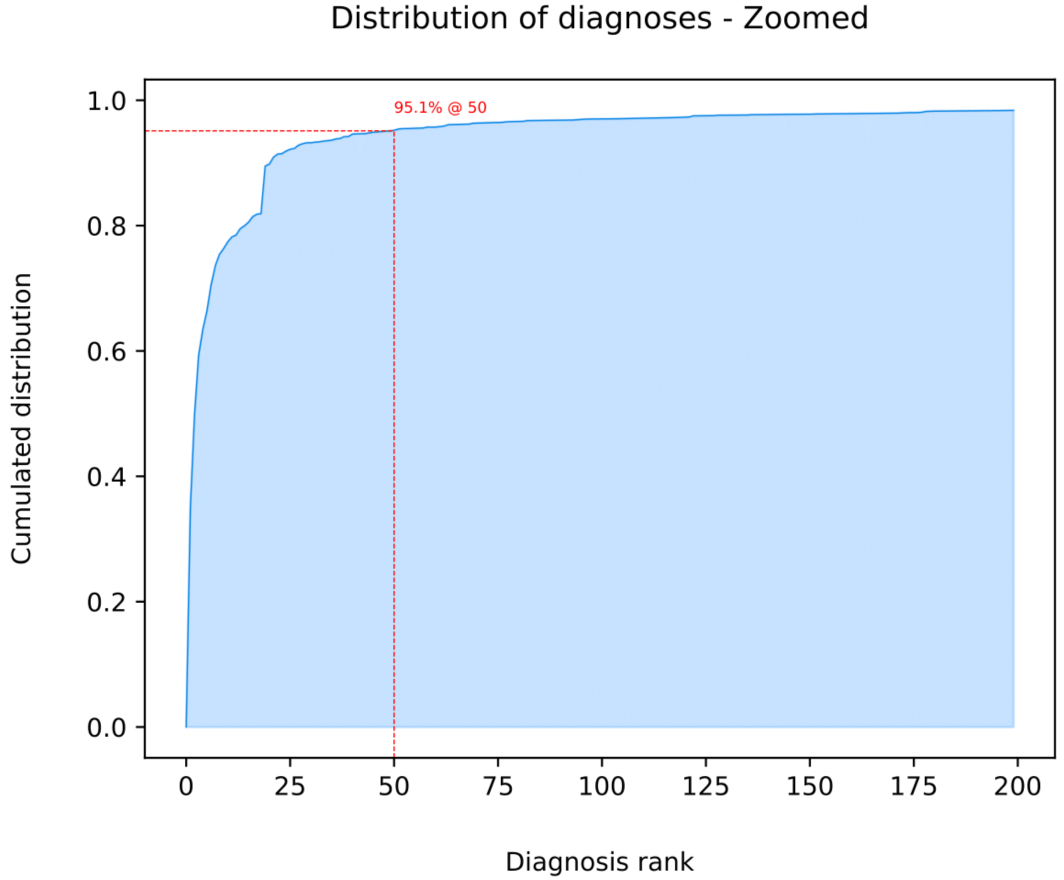
Multi-label multi-class classification problem:

- ~ 8’900 unique ICD-9 codes in total
- We focus on the 50 most frequent ones

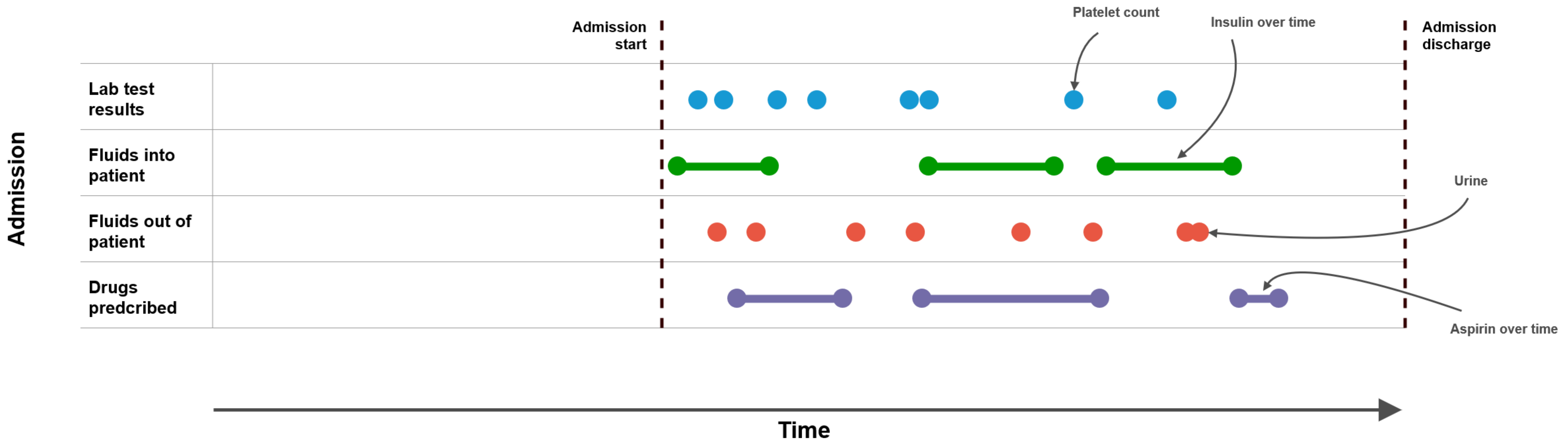
ORACLE

MIMIC-III – Statistics

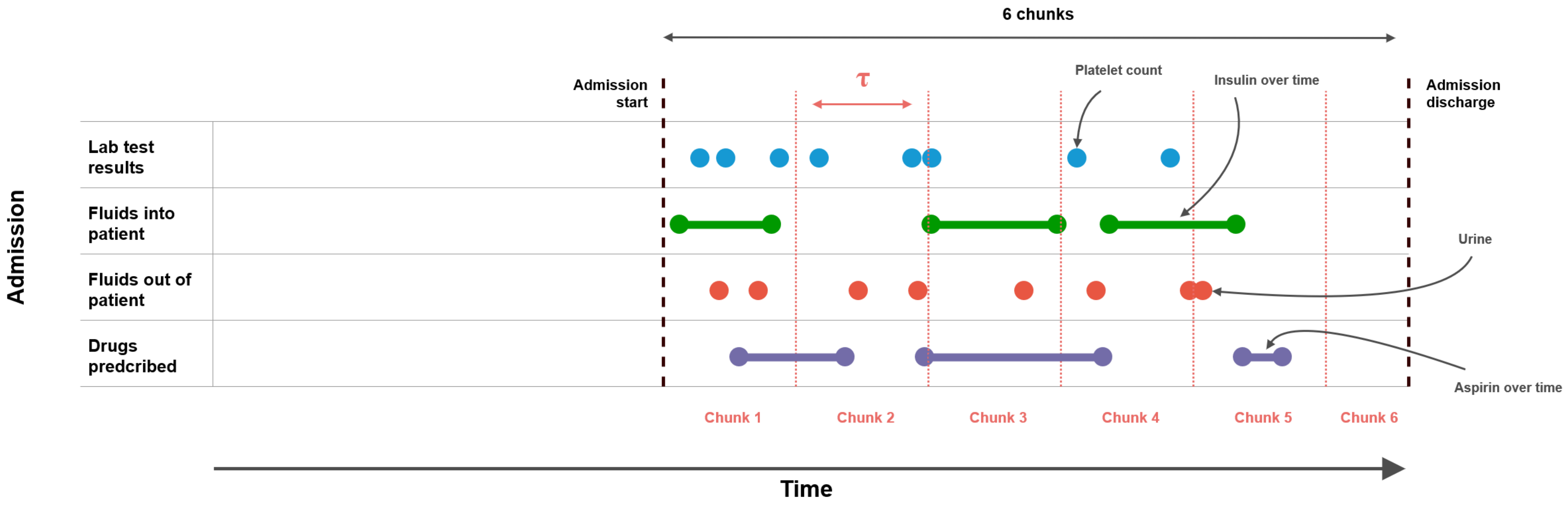




MIMIC-III – Data Formation – Original



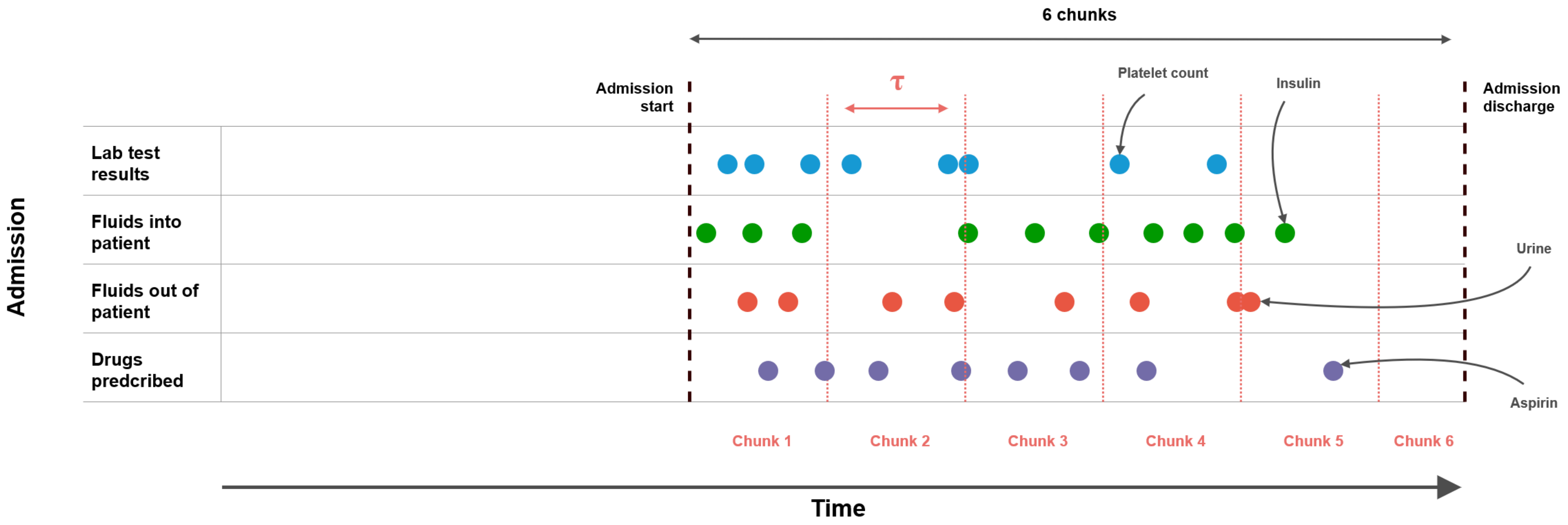
MIMIC-III – Data Formation – Step 1



Modifications

- Group events in same time interval (e.g. 3h)
- Consider each interval as a **graph state** or **chunk**

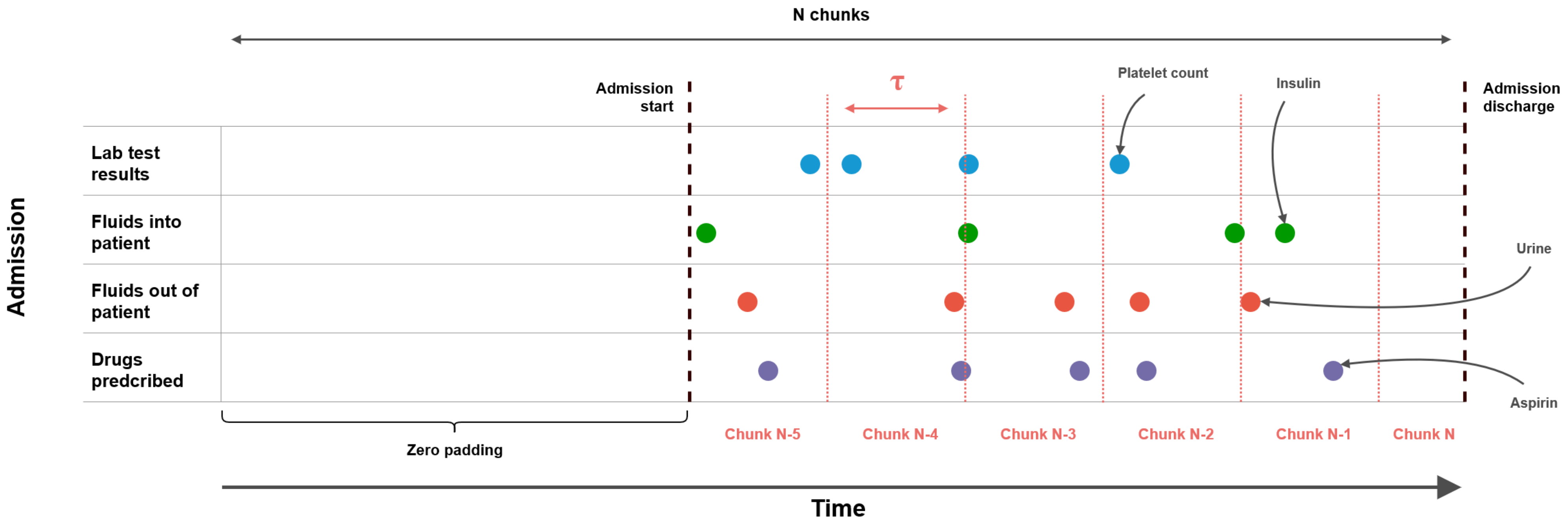
MIMIC-III – Data Formation – Step 2



Modifications

- Conversion of **time-range** events to **one-time** events
- For the same event, convert all units to same one

MIMIC-III – Data Formation – Step 3



Modifications

- Down-sample events per type and per chunk to some fixed number (**K**)
- Add zeros before admission's beginning to make it fixed length (**N**)

From MIMIC-III to an Evolving Graph

- We now have an evolving entity for each admission
- Solves many challenges inherent to healthcare:
  - Many missing values in events
  - Non-constant intervals among different events:
    - mmHg/h for **blood pressure**
    - mg/30min for **aspirin**
- More natural than existing approaches:
  - We are interested in the evolution of a patient's state
  - Not his exact vitals at every T, where T is the finest-grain event interval

ORACLE

Baseline

Single admission approach

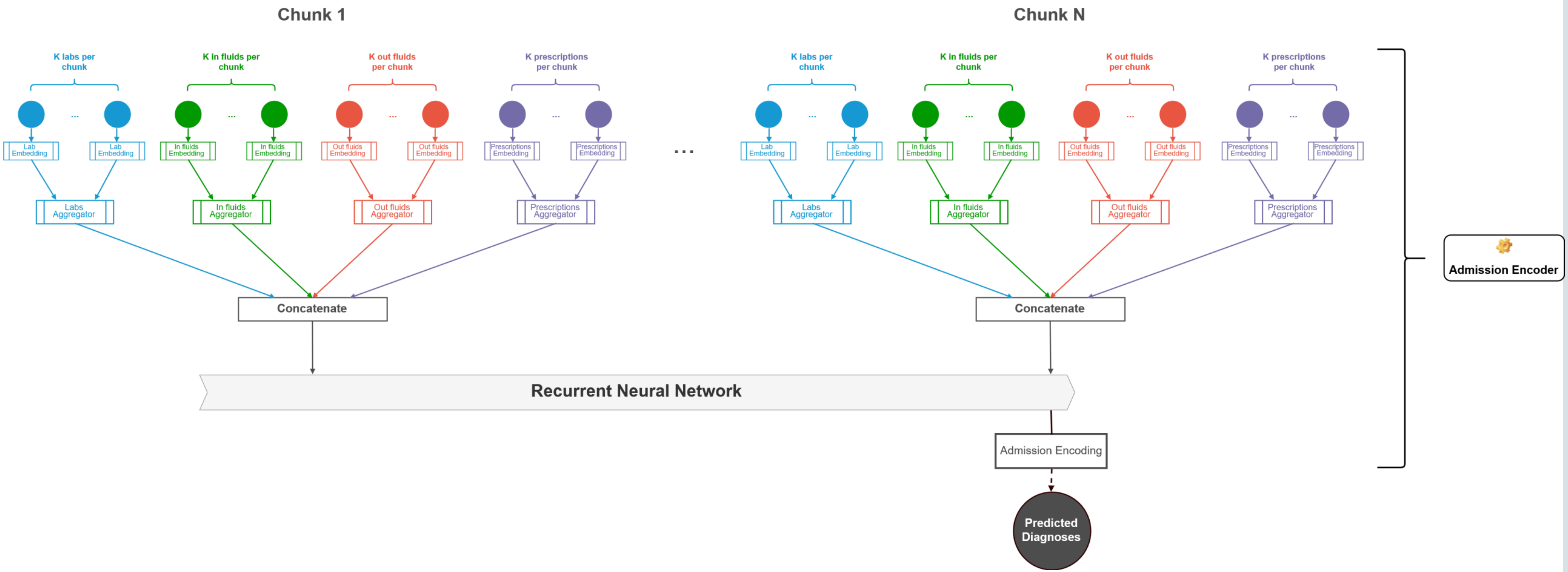
ORACLE

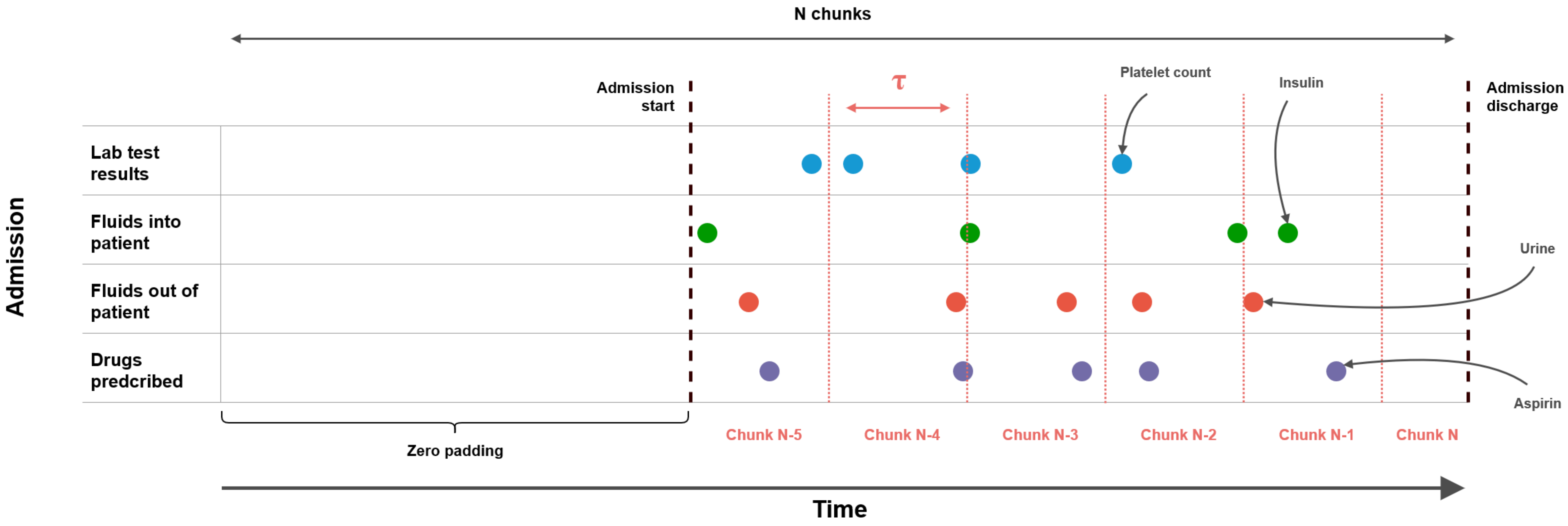
Baseline

Main idea

1. Create a vector representation of each event (called **embedding**)
2. **Aggregate** events of the same type within the same chunk together
3. **Concatenate** aggregated events for each type
4. **Feed** aggregated chunks into a Recurrent Neural Network
5. Predict the final diagnoses from the output of the RNN

ORACLE


Chunk 1
Chunk N
K labs per chunk
K in fluids per chunk
K out fluids per chunk
K prescriptions per chunk
Lab Embedding
In fluids Embedding
Out fluids Embedding
Prescriptions Embedding
Labs Aggregator
In fluids Aggregator
Out fluids Aggregator
Prescriptions Aggregator
Concatenate
Recurrent Neural Network
Admission Encoding
Predicted Diagnoses
Admission Encoder



N chunks
Admission start
τ
Platelet count
Insulin
Admission discharge
Lab test results
Fluids into patient
Fluids out of patient
Drugs predcribed
Urine
Aspirin
Admission
Zero padding
Chunk N-5
Chunk N-4
Chunk N-3
Chunk N-2
Chunk N-1
Chunk N
Time

KG-RNN: Architecture

Multi admission approach

Weighted Knowledge Graph **Construction**

From tables and external information

Files

Databases

Entities at t = 0

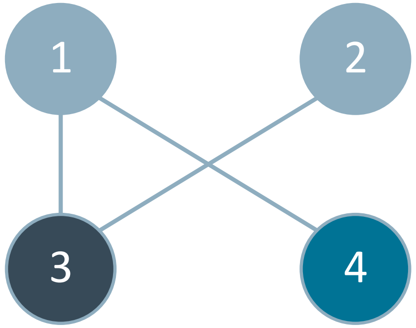
ID	Type	Property A
1	Entity A	1
2	Entity A	5
3	Entity B	0
4	Entity C	2

Entities at t = T

ID	Type	Property A
1	Entity A	29
2	Entity A	3
3	Entity B	7
4	Entity C	12

Relations

Entity 1	Entity 2	Type	Weight
1	3	X	0.95
3	2	Y	0.5
4	1	X	1



ORACLE

Weighted Knowledge Graph **Extraction**

- From this Knowledge Graph:
  → Extract the **M** nearest neighbors for each input entity

- Nearest: Can be any distance metric between entities

- Any graph sampling technique can be applied (weighted or not):
  – Random sampling
  – Snow-Ball sampling
  – Forest Fire sampling
  – …

ORACLE

KG-RNN

Main idea

1. Feed graph states sequence, focused on an entity, through the **Encoder** module (c.f. *Baseline*):
   - Input entity
   - Neighboring entities

2. Gather the encoded entities with the **Main** (c.f. *next slides*) module

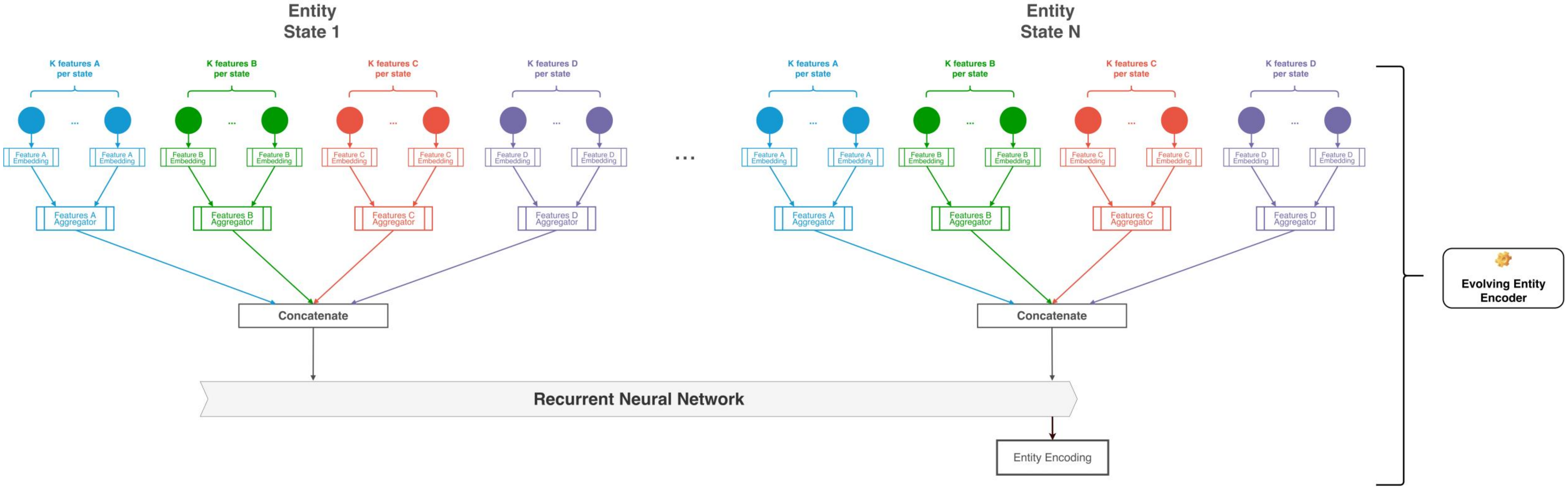
3. Attach static information of entities to their encoding vector

4. Get the final prediction from these static information + encoding vectors

ORACLE

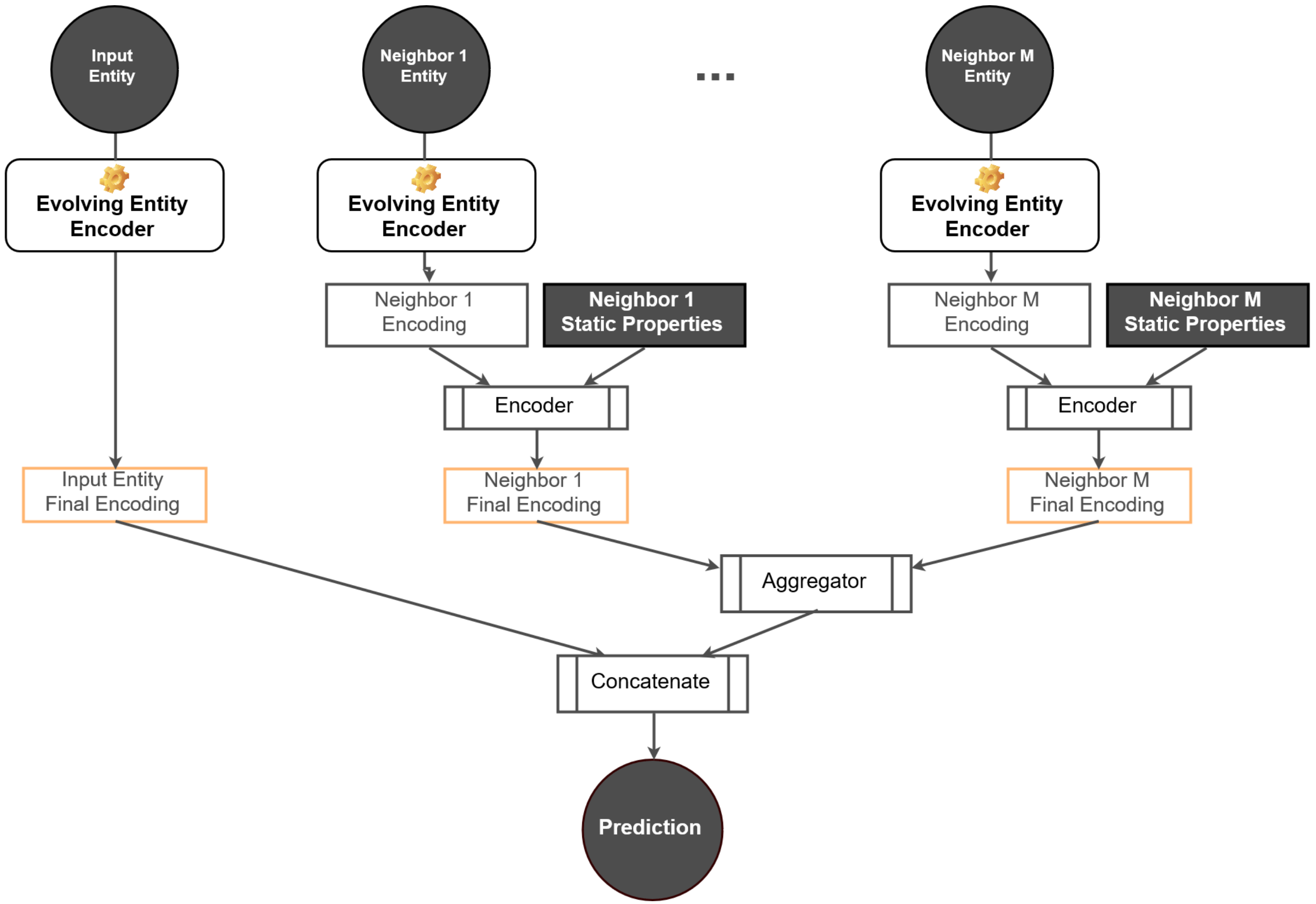
KG-RNN: Entity Encoder

Predicting from graph states



KG-RNN: Main

Predicting from graph states



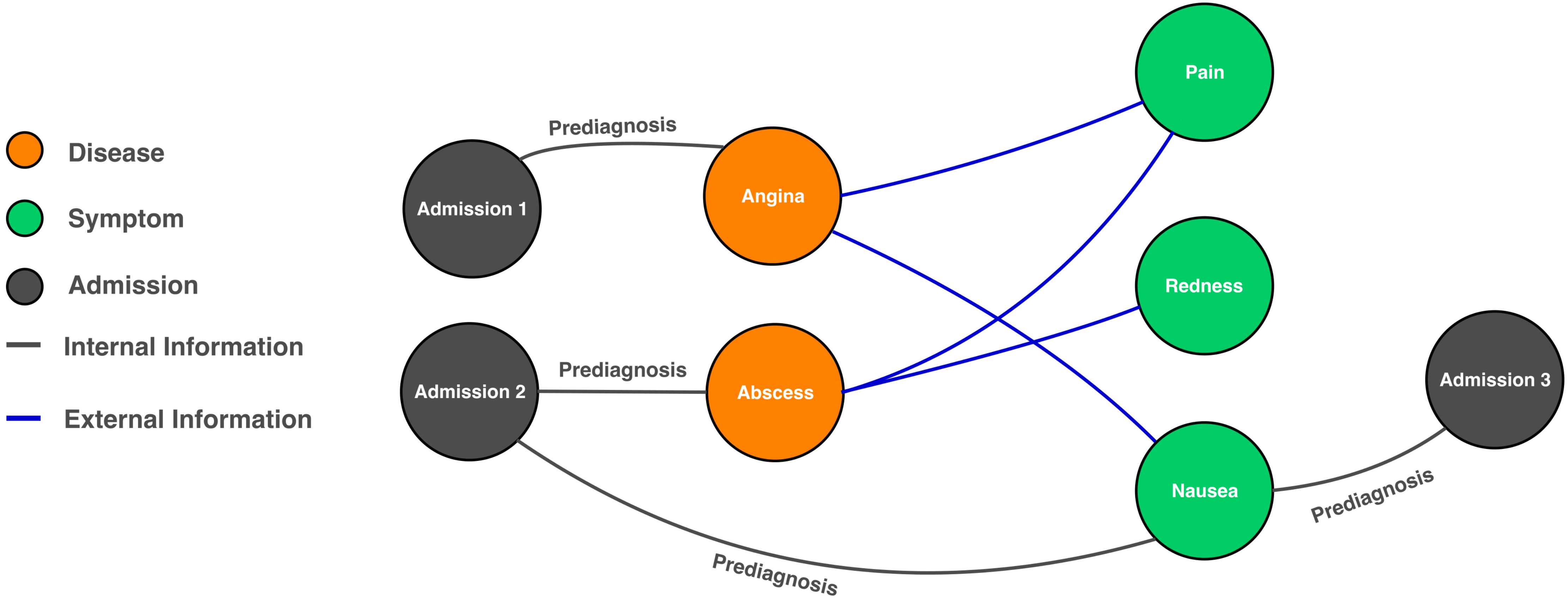
ORACLE

KG-RNN: Application

Healthcare Dataset

Weighted Knowledge Graph Construction

Healthcare example



ORACLE

Weighted Knowledge Graph Construction

Healthcare example

- External information from derived Knowledge Graph:
  - Learning a Health Knowledge Graph from Electronic Medical Records (Rotmensch et al., Nature 2017)
  - They link diseases and symptoms:
    - *Migraine – Headache (0.384), nausea (0.316), sensitivity to light (0.223), …*
  - Each link comes with a relevance score → **Edge weight**
- String matching between *Internal KG* and *External KG*:
  - NGrams + TF-IDF → Cosine similarity
  - Admission Prediagnosis – Disease
  - Admission Prediagnosis – Symptom
  - Cosine similarity score between 0 and 1 → **Edge weight**

ORACLE

Weighted Knowledge Graph Extraction

Healthcare example

- Input entity: **Admission**
  - We extract **10** nearest neighbors per admission

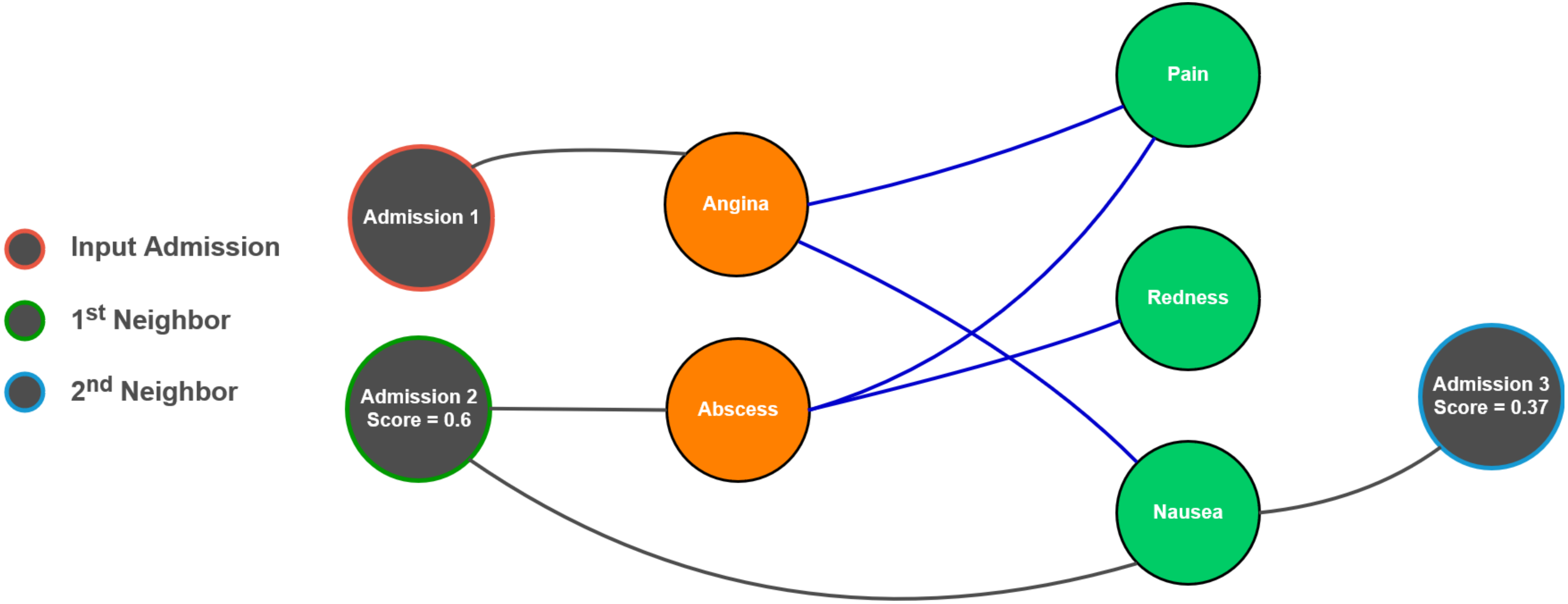
- Distance metric:
  - Disease – Symptom: Weight already defined in external KG
  - Admission – Symptom/Disease: Cosine similarity score between texts

- Sampling: Importance Sampling
  - Compute *Weighted Personalized PageRank* score focused on input admission
  - Take **top-10** admissions with respect to WPPR score

ORACLE

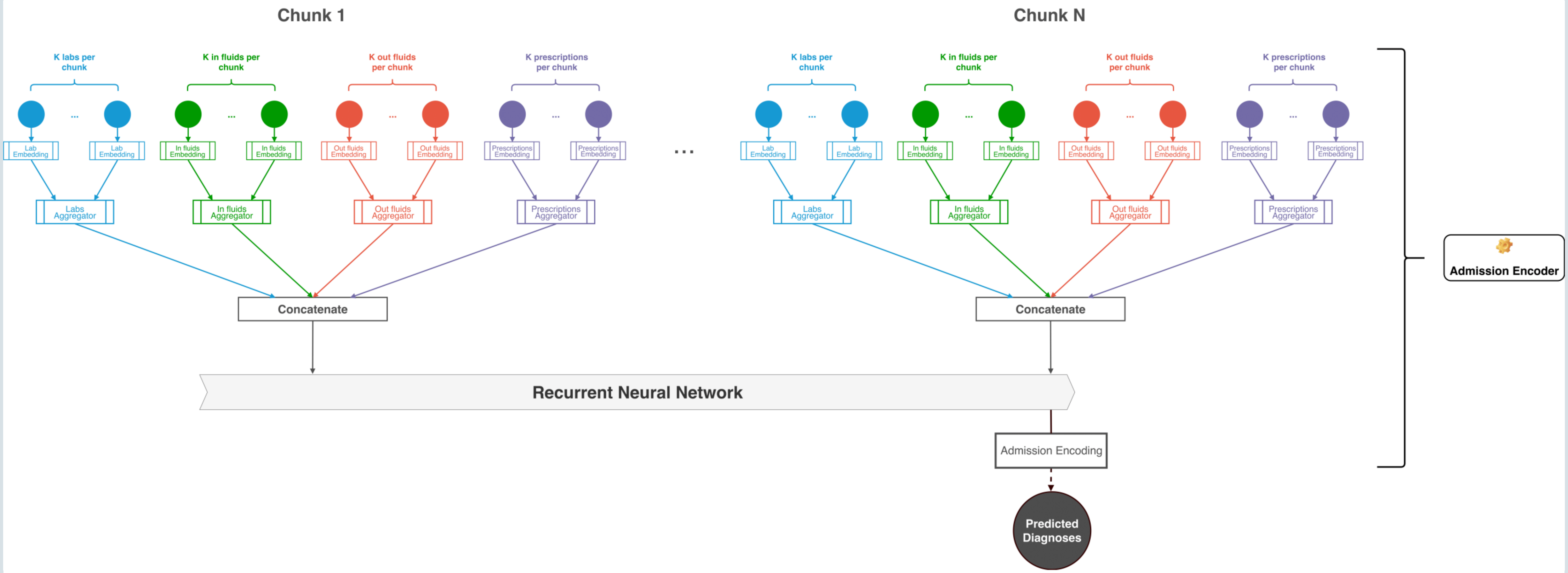
Weighted Knowledge Graph Construction

Healthcare example



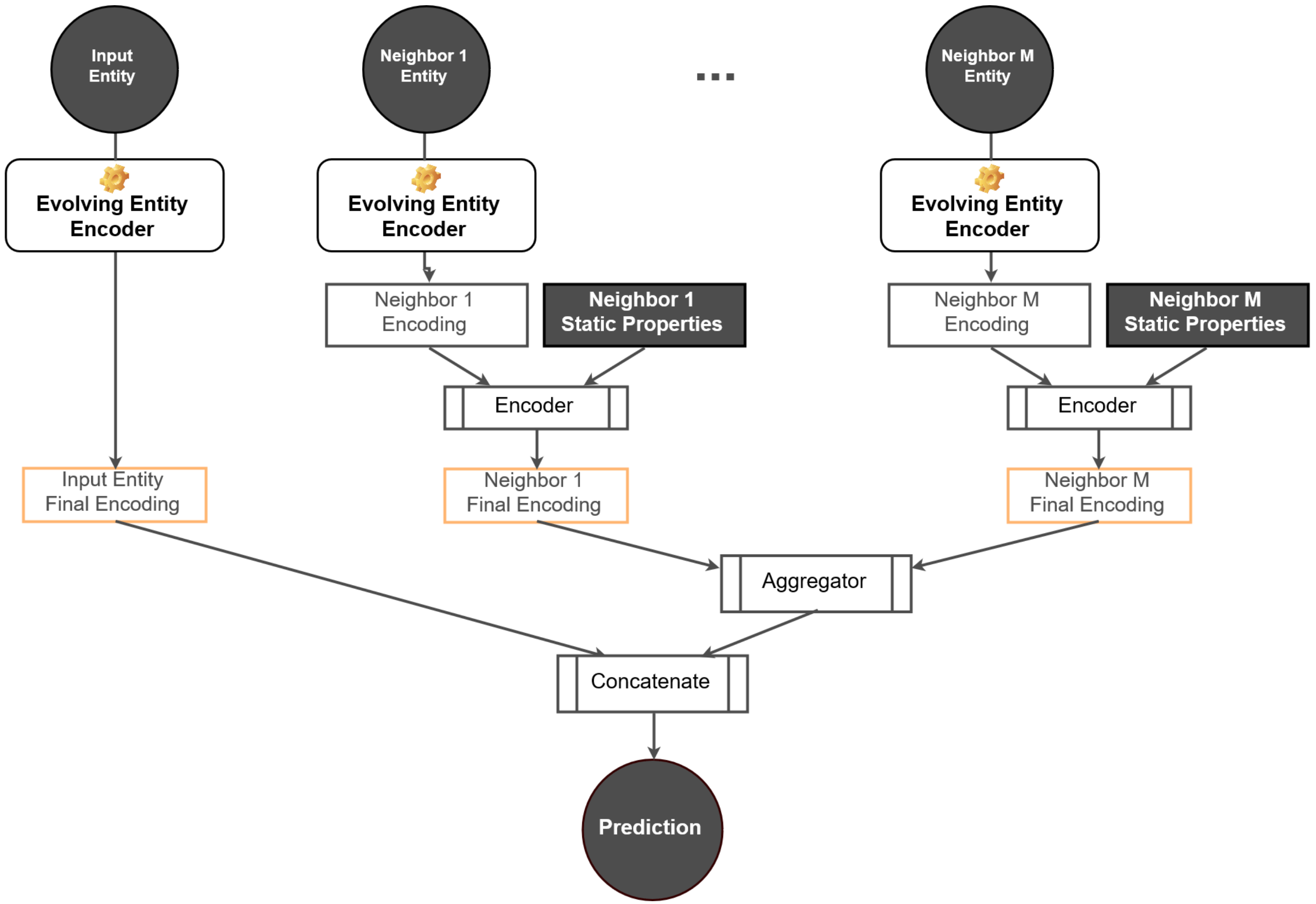
KG-RNN: Entity Encoder, reminder

Predicting from graph states



KG-RNN: General Model

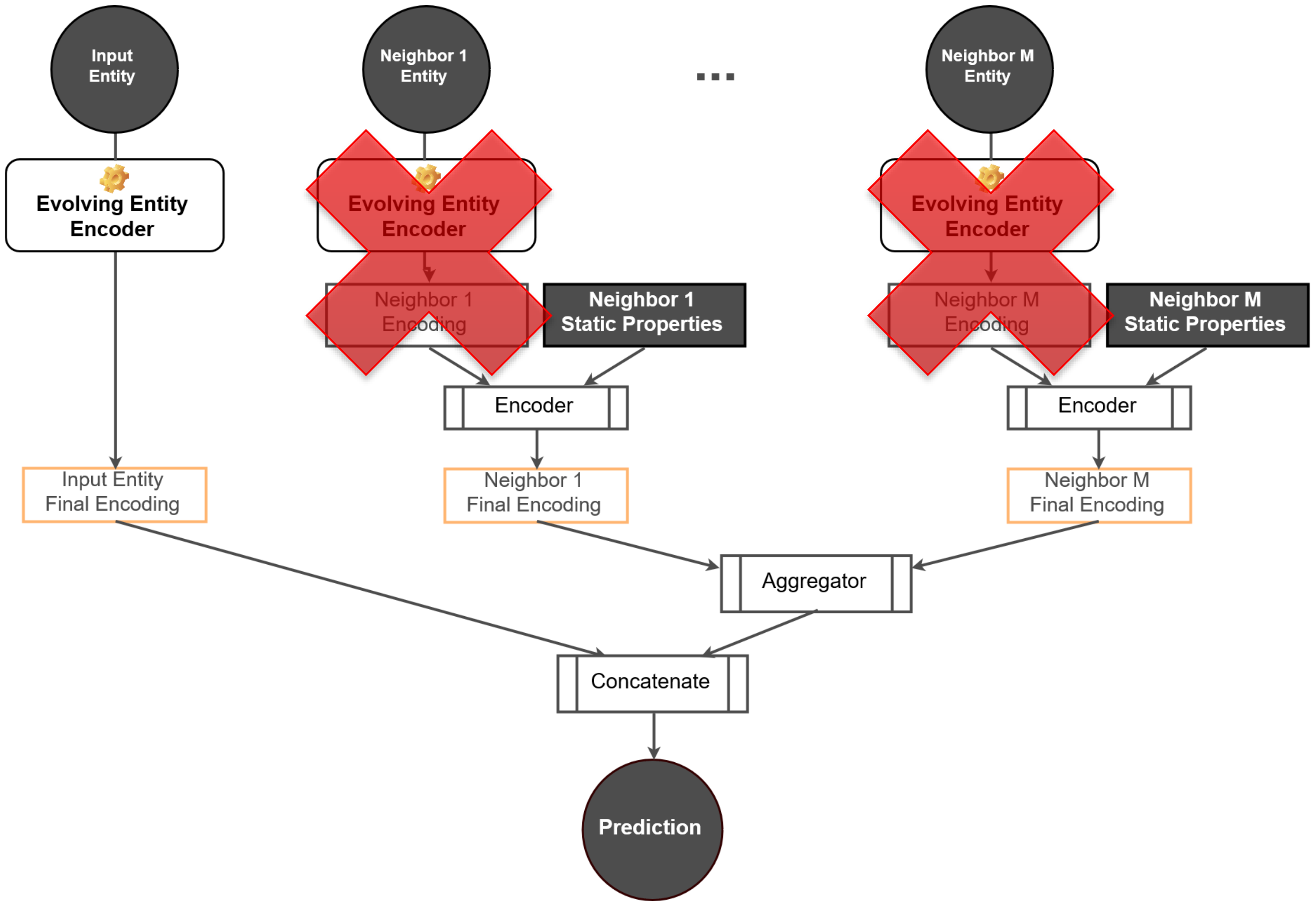
Predicting from graph states



ORACLE

KG-RNN: Adapted Model

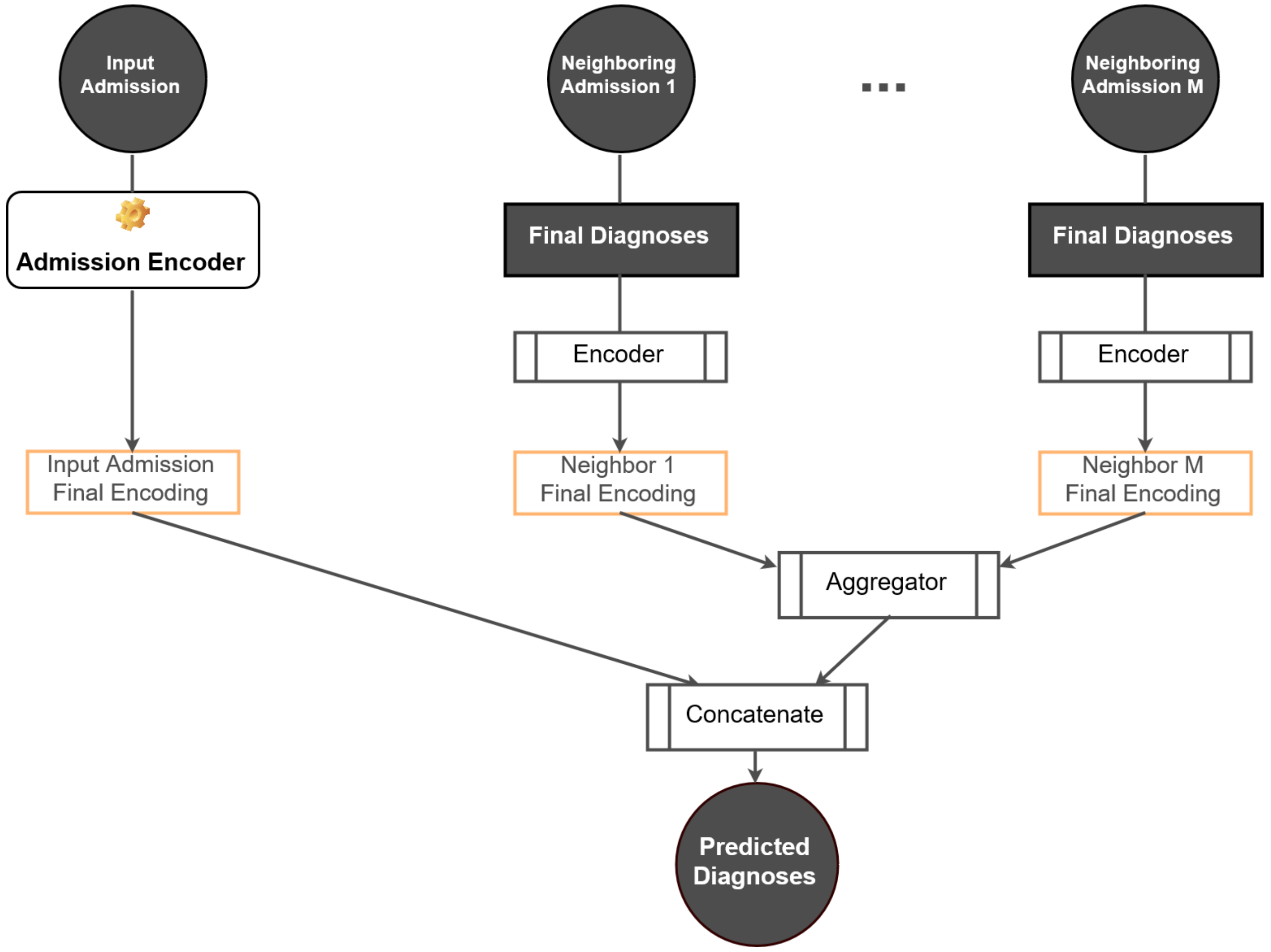
Predicting from graph states



ORACLE

KG-RNN: Adapted Model

Predicting from graph states



ORACLE

Demo

Live Prediction Example

Results

Baseline vs. KG-RNN

Quantitative Results

Metric	Average	Model	Score
F1	Macro	Baseline	**36.52%**
		KG-RNN	**37.90%** (+1.38%)
	Micro	Baseline	**51.55%**
		KG-RNN	**53.47%** (+1.92%)
AUROC	Macro	Baseline	**85.24%**
		KG-RNN	**86.29%** (+1.05%)
	Micro	Baseline	**90.55%**
		KG-RNN	**91.03%** (+0.48%)
Accuracy	-	Baseline	**92.22%**
		KG-RNN	**92.36%** (+0.14%)

Properties

3 hours per chunk
200 chunks per admission
25 events per type and per chunk
10 neighbors sampled per input admission

Qualitative Results

- After training, check the closest events to a particular event in terms of embedding vectors:

Neighbors for **White Blood Cells**

Rank	Laboratory measure
1	WBC *Alternative name*
3	White Cells *Alternative name*
6	Immunoglobulin A *Antibodies produced by white blood cells*

Neighbors for **Sodium**

Rank	Prescription
1	Sodium Chloride Nasal *Related prescription*
2	Sodium Chloride 0.9% Flush *Related prescription*
5	Famotidine Often given in “Sodium Chloride 0.9%“

ORACLE

Qualitative Results

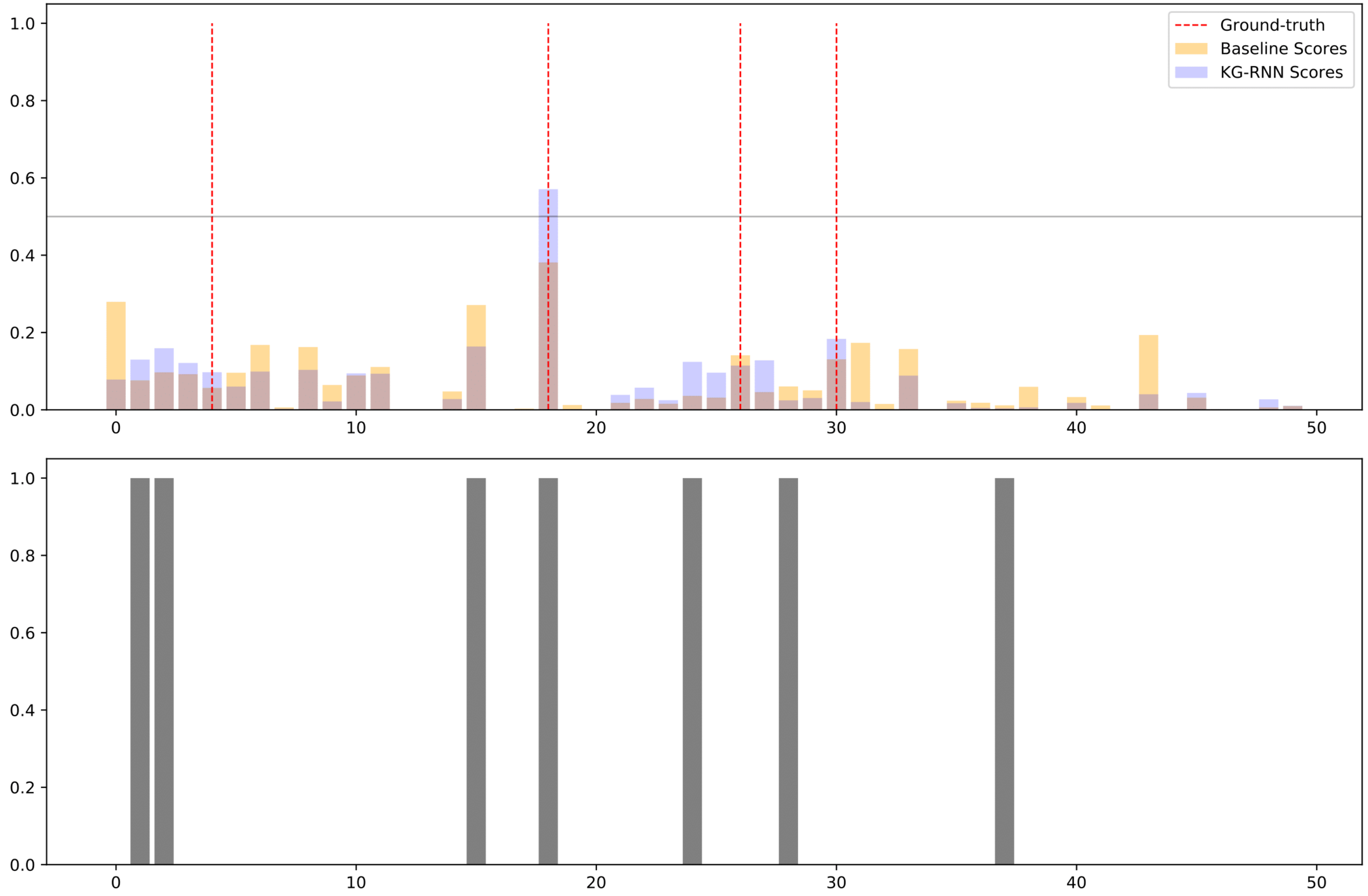
Take-Home Message

→ **KG-RNN not only learned similar events**
but also intrinsically linked and correlated events

ORACLE

Qualitative Results

- Baseline does not detect any diagnosis
- Neighbor helped pushing the confidence of KG-RNN upward for 2nd diagnosis
- Reveals how much KG-RNN relies on its neighbors



ORACLE

Concluding Remarks

ORACLE

Conclusion

- Proposed a novel architecture and pipeline for processing evolving knowledge graphs
- Our enriched knowledge graph has been designed to naturally cope with many of the challenges inherent to our application
- Our solution shows significant improvement over a simple single-admission setting
- Qualitative analysis revealing interesting findings in learning behavior

ORACLE

ORACLE®