Lecture 1: Relational Data & Node Embeddings

Relational Learning

İsmail İlkan Ceylan
Advanced Topics in Machine Learning, University of Oxford

17.01.2022

Course Organization

Course Organization

Advanced Topics in Machine Learning: Research-oriented & less conventional course

- **Relational Learning**: 9 lectures by *İsmail İlkan Ceylan*
- **Bayesian Machine Learning**: 9 lectures by *Jiarui Gan and Yarin Gal*

Course Organization

Advanced Topics in Machine Learning: Research-oriented & less conventional course

- **Relational Learning**: 9 lectures by *İsmail İlkan Ceylan*
- **Bayesian Machine Learning**: 9 lectures by *Jiarui Gan and Yarin Gal*

Location and Time: Lecture Theatre A

- Week 1- 8: Monday's 14:00 - 16:00
- Week 1 & 2: Wednesday's 14:00 - 15:00

Course webpage: https://www.cs.ox.ac.uk/teaching/courses/2021-2022/advml/

Administrative inquiries: academic.administrator@cs.ox.ac.uk

Content inquiries: @ the respective lecturer

Course Organization

Advanced Topics in Machine Learning: Research-oriented & less conventional course

- **Relational Learning**: 9 lectures by *İsmail İlkan Ceylan*
- **Bayesian Machine Learning**: 9 lectures by *Jiarui Gan and Yarin Gal*

Assessment: Through a paper reproducibility challenge, as detailed in the assessment form:

- Students form groups of 3 - 4
- Each group bids on at least two assessment papers
- Each group delivers a report and a poster
- **Marking**: group report (25%), group poster (25%), individual viva (50%)
- Viva's at the beginning of Trinity and approximately 15 min's for each student

Course Organization

Advanced Topics in Machine Learning: Research-oriented & less conventional course

- **Relational Learning**: 9 lectures by *İsmail İlkan Ceylan*
- **Bayesian Machine Learning**: 9 lectures by *Jiarui Gan and Yarin Gal*

Practicals:

Practical 1: Building a graph neural network

Practical 2: Developing a Bayesian model

Practical 3 & 4: Discussing the assessment papers and group-formation

Practical 5 & 6: Kick-off projects

Demonstrators: Ralph Abboud (RL), Vit Ruzicka (RL), Ben Moseley (BML), Matthew Wicker (BML)

Course Structure: Relational Learning

Course Structure: Relational Learning

Relational learning: Very broad area covering machine learning over relational data!

Course Structure: Relational Learning

Relational learning: Very broad area covering machine learning over relational data!

Relational data and node embedding models (2 lectures)

- Relational data, graphs, shallow node embeddings

Course Structure: Relational Learning

Relational learning: Very broad area covering machine learning over relational data!

Relational data and node embedding models (2 lectures)

- Relational data, graphs, shallow node embeddings

Graph neural networks (7 lectures)

- Fundamentals (graph neural networks, relational inductive bias, node-level tasks, graph-level tasks, edge-level tasks, message passing neural network architectures)
- Foundations (expressive power of message passing neural networks, higher-order models, unique features, random features)
- Applications (drug discovery, recommender systems, combinatorial optimization, ...)

Course Structure: Relational Learning

Relational learning: Very broad area covering machine learning over relational data!

Relational data and node embedding models (2 lectures)

- Relational data, graphs, shallow node embeddings

Graph neural networks (7 lectures)

- Fundamentals (graph neural networks, relational inductive bias, node-level tasks, graph-level tasks, edge-level tasks, message passing neural network architectures)
- Foundations (expressive power of message passing neural networks, higher-order models, unique features, random features)
- Applications (drug discovery, recommender systems, combinatorial optimization, ...)

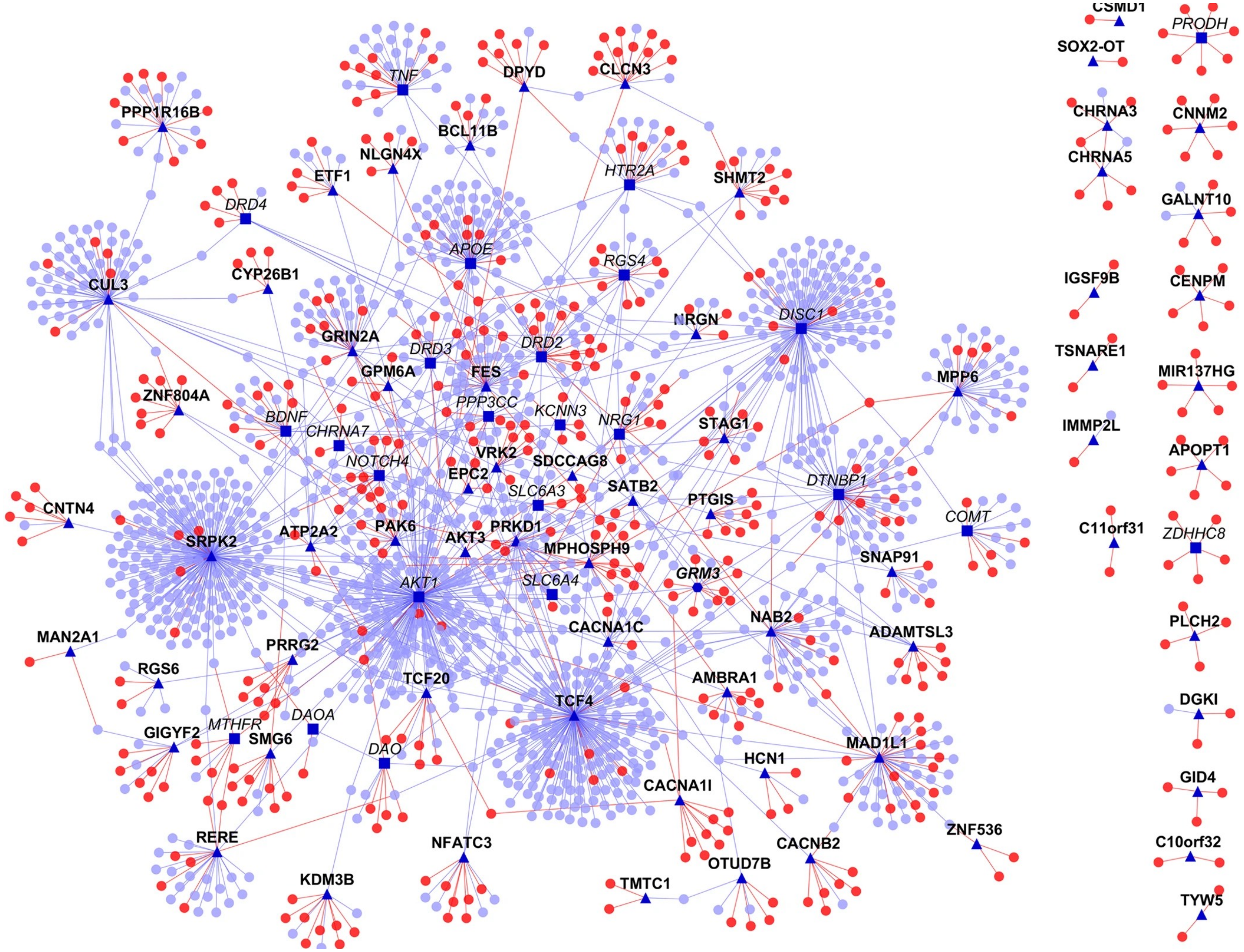
Reference book: William L. Hamilton. (2020). Graph Representation Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers.

Overview of the Lecture

- Relational data
- Graph representation learning
- Machine learning with knowledge graphs
- Knowledge graph embedding models
    - Model expressiveness
    - Model inductive capacity and inference patterns
    - Empirical evaluation: Datasets and metrics
- Summary

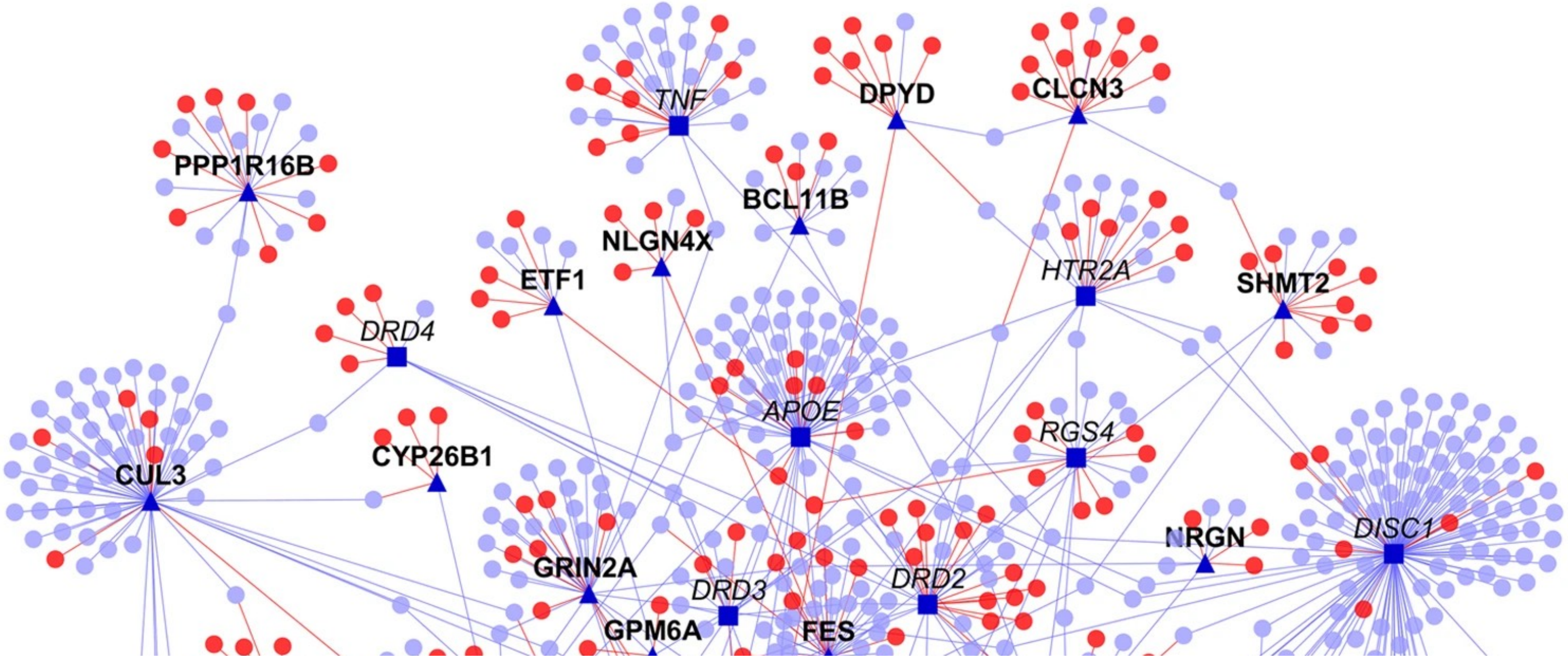
Relational Data

Relational Data



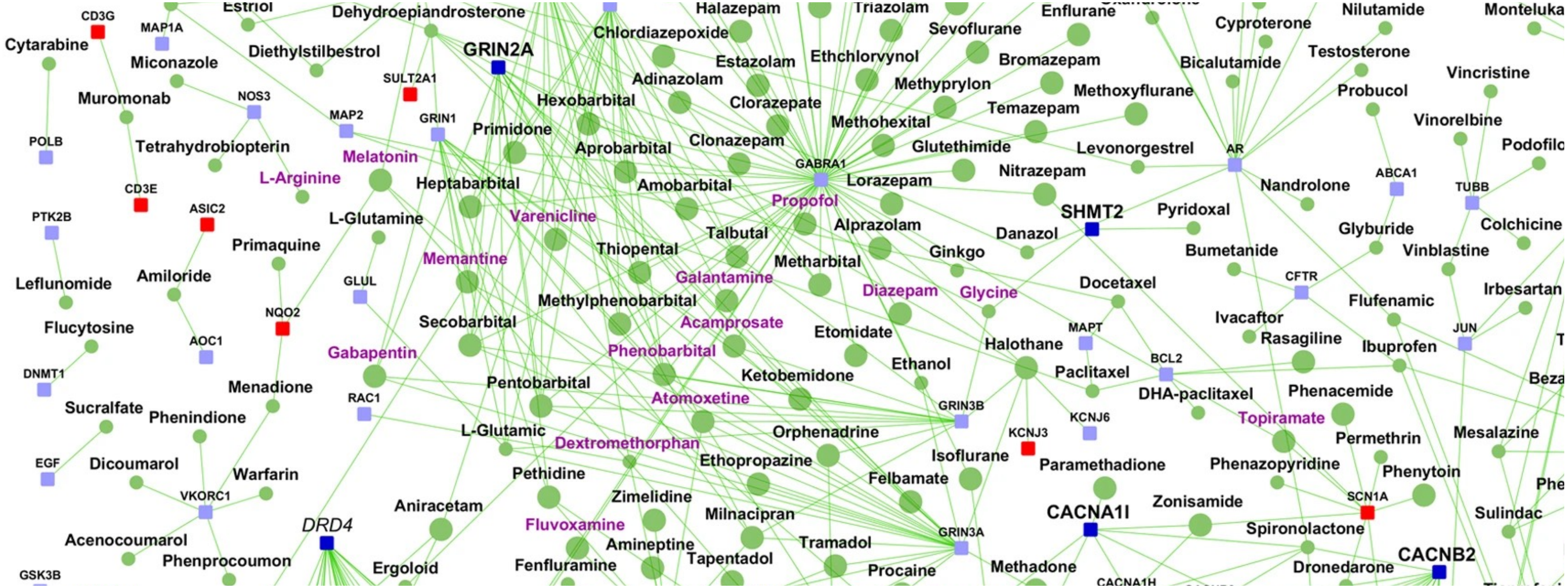
Protein networks: Figure illustrates schizophrenia interactome from (Ganapathiraju et al, 2016).

Relational Data



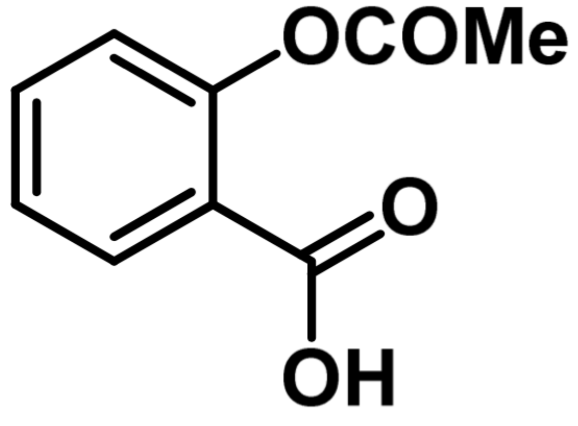
Excerpt from Schizophrenia interactome (Ganapathiraju et al, 2016): Genes are shown as nodes and PPIs as edges connecting the nodes. Schizophrenia-associated genes are shown as dark blue nodes, novel interactors as red color nodes and known interactors as blue color nodes. Red edges are the novel interactions, whereas blue edges are known interactions.

Relational Data

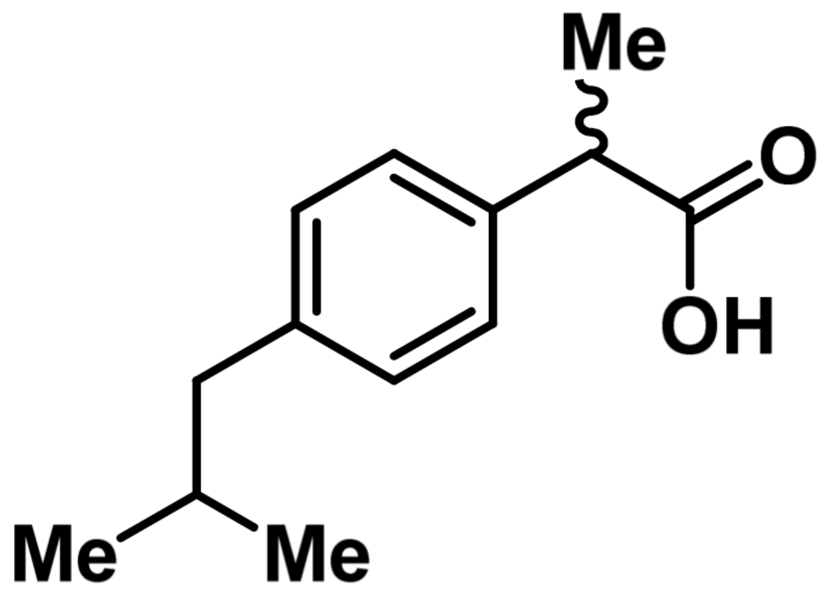


Excerpt from gene–drug interactome (Ganapathiraju et al, 2016): The network shows the drugs that target genes from the schizophrenia interactome. Drugs are shown as **round nodes** colored in green and genes as **square nodes** colored in dark blue (schizophrenia genes), light blue (known interactors), and red (novel interactors).

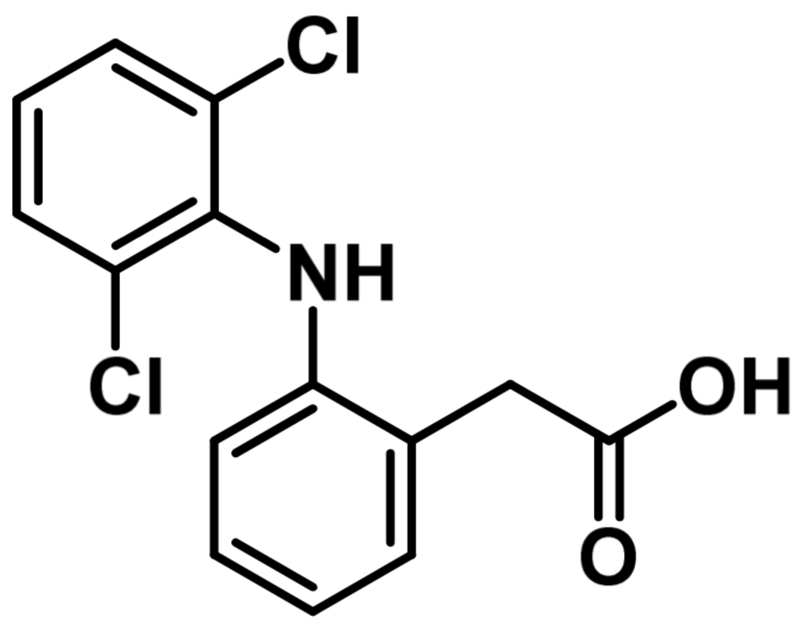
Relational Data



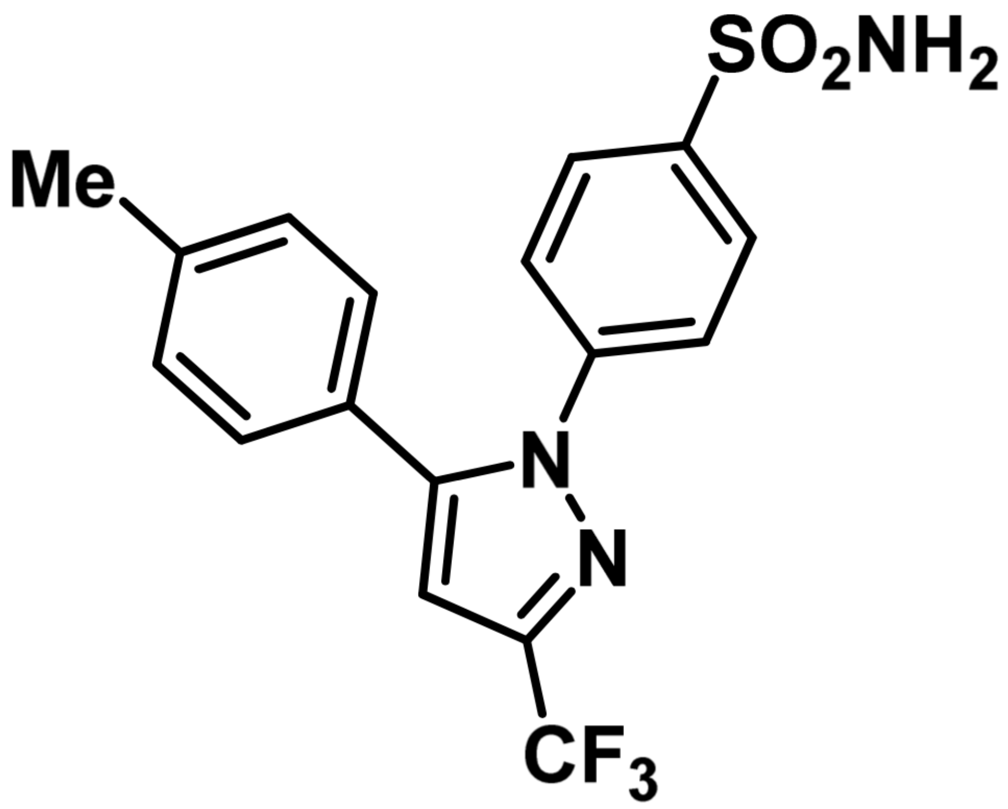
Aspirin (**1**)



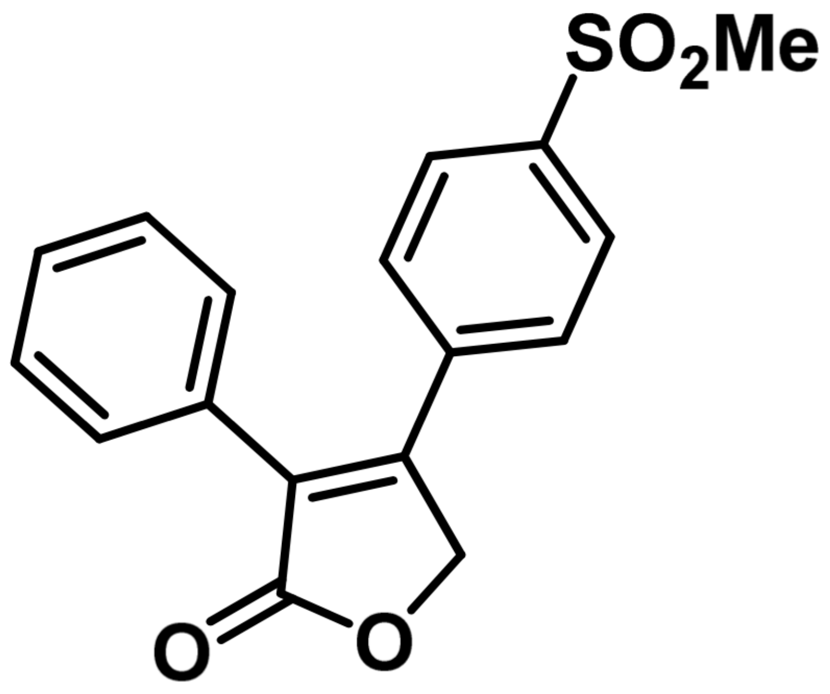
Ibuprofen (**2**)



Diclofenac (**3**)



Celecoxib (**4**)



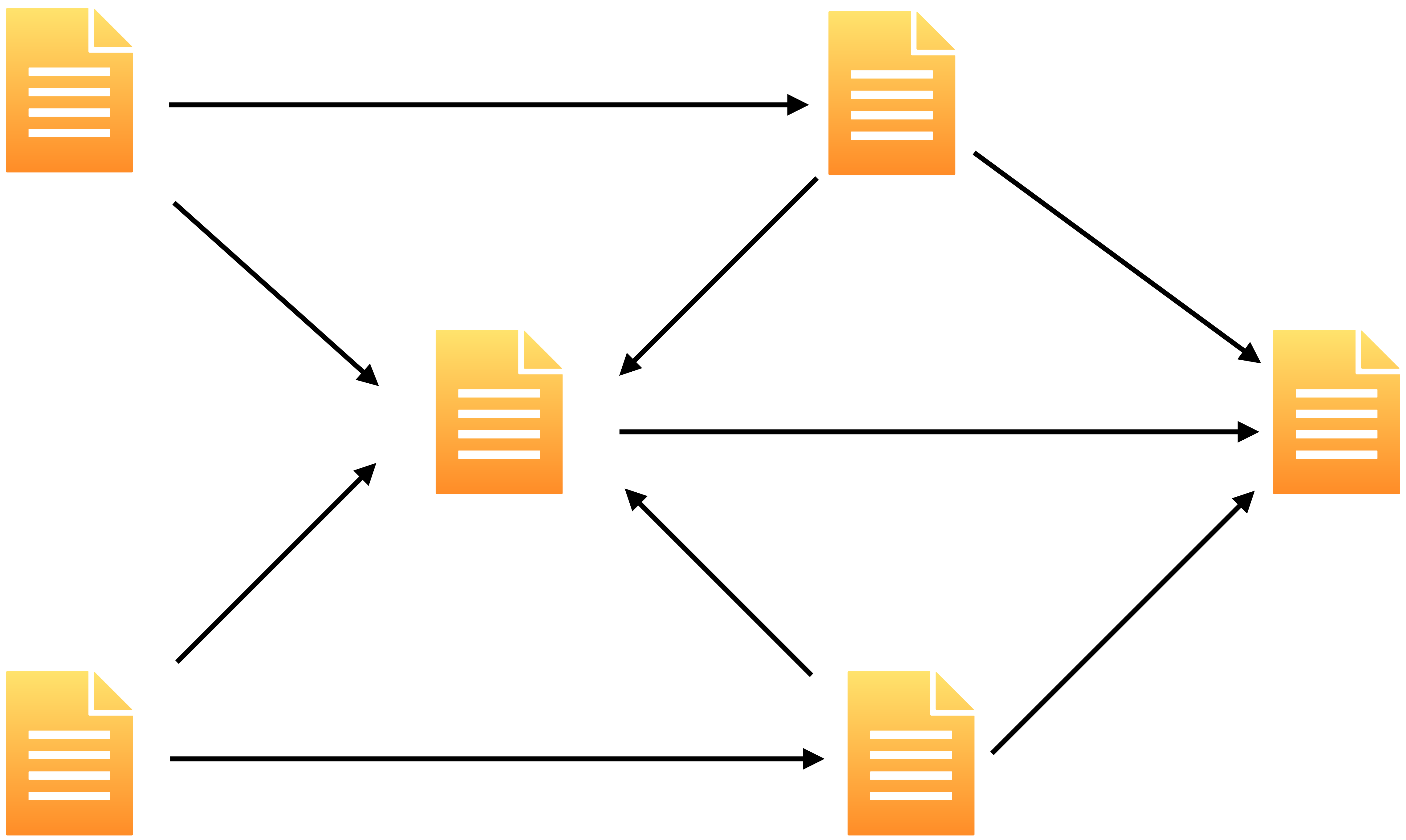
Rofecoxib (**5**)

Molecule Networks (Rao et al, 2013): Figure shows the molecule structure of NSAID drugs. “Me" is an abbreviation for "methyl" (CH3).

Relational Data

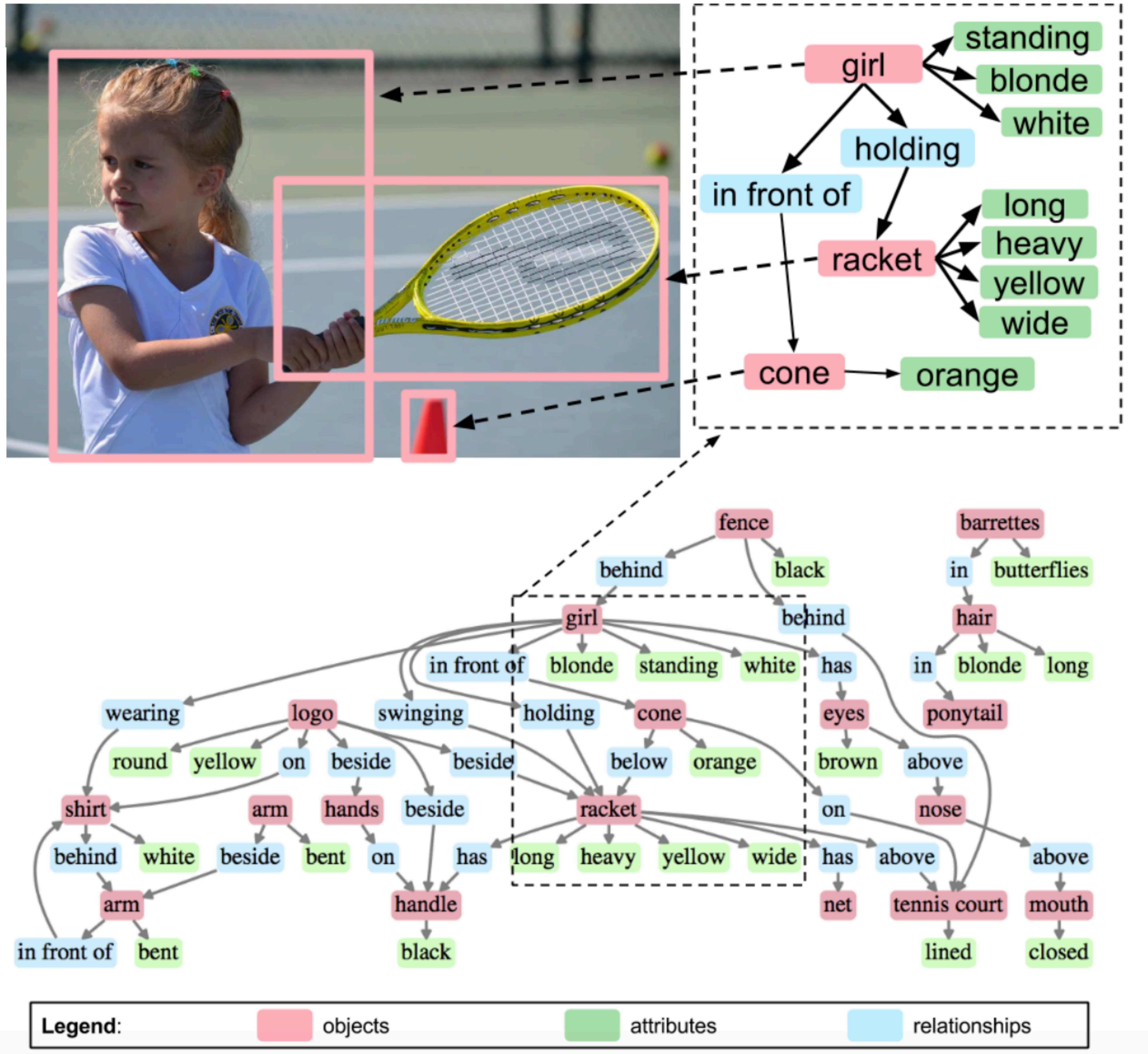
Social networks: Entities (e.g., individuals, groups, organizations) interacting with other entities on social platforms.

Relational Data



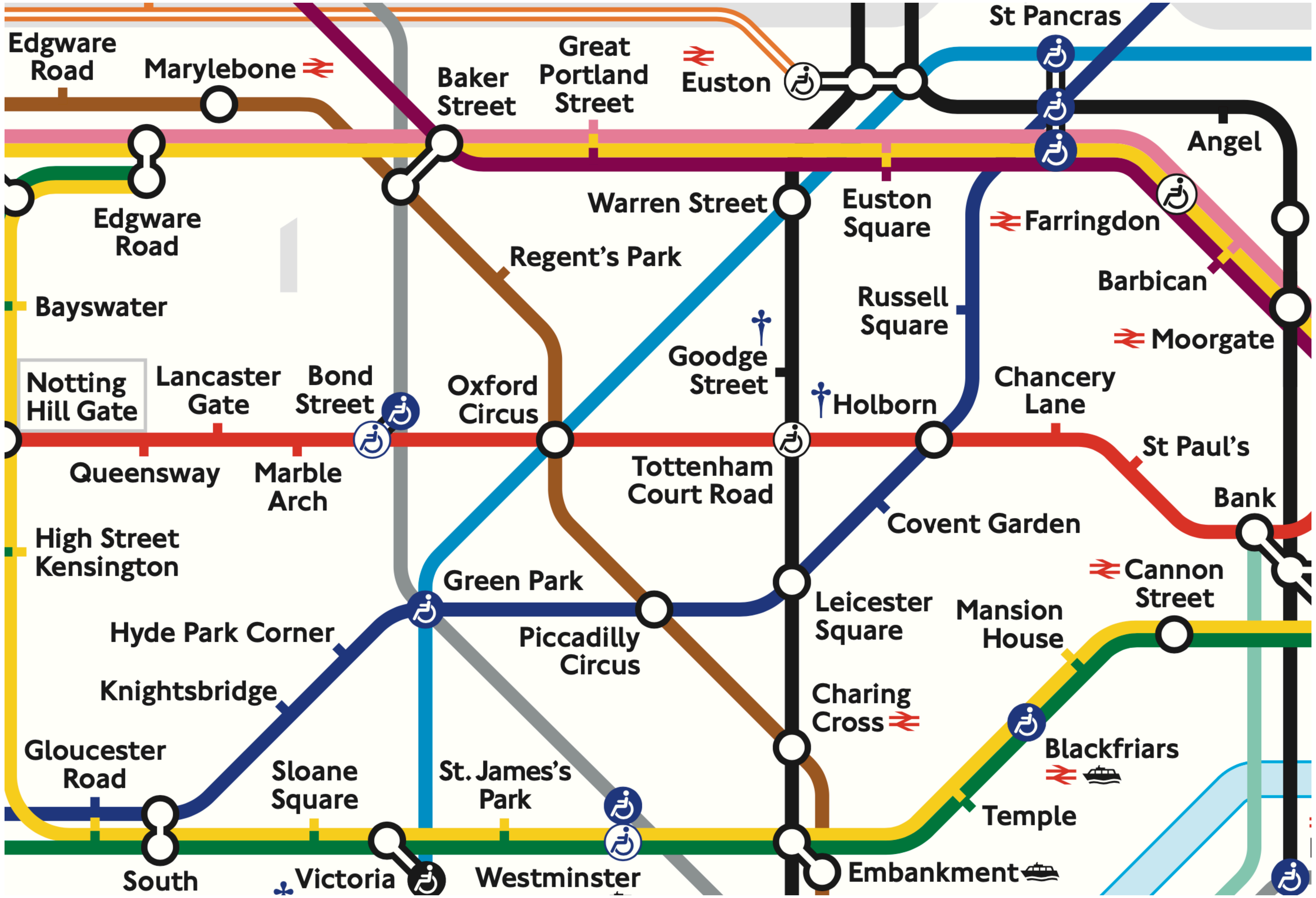
Citation networks: Each paper cites other papers, forming a citation graph across papers.

Relational Data



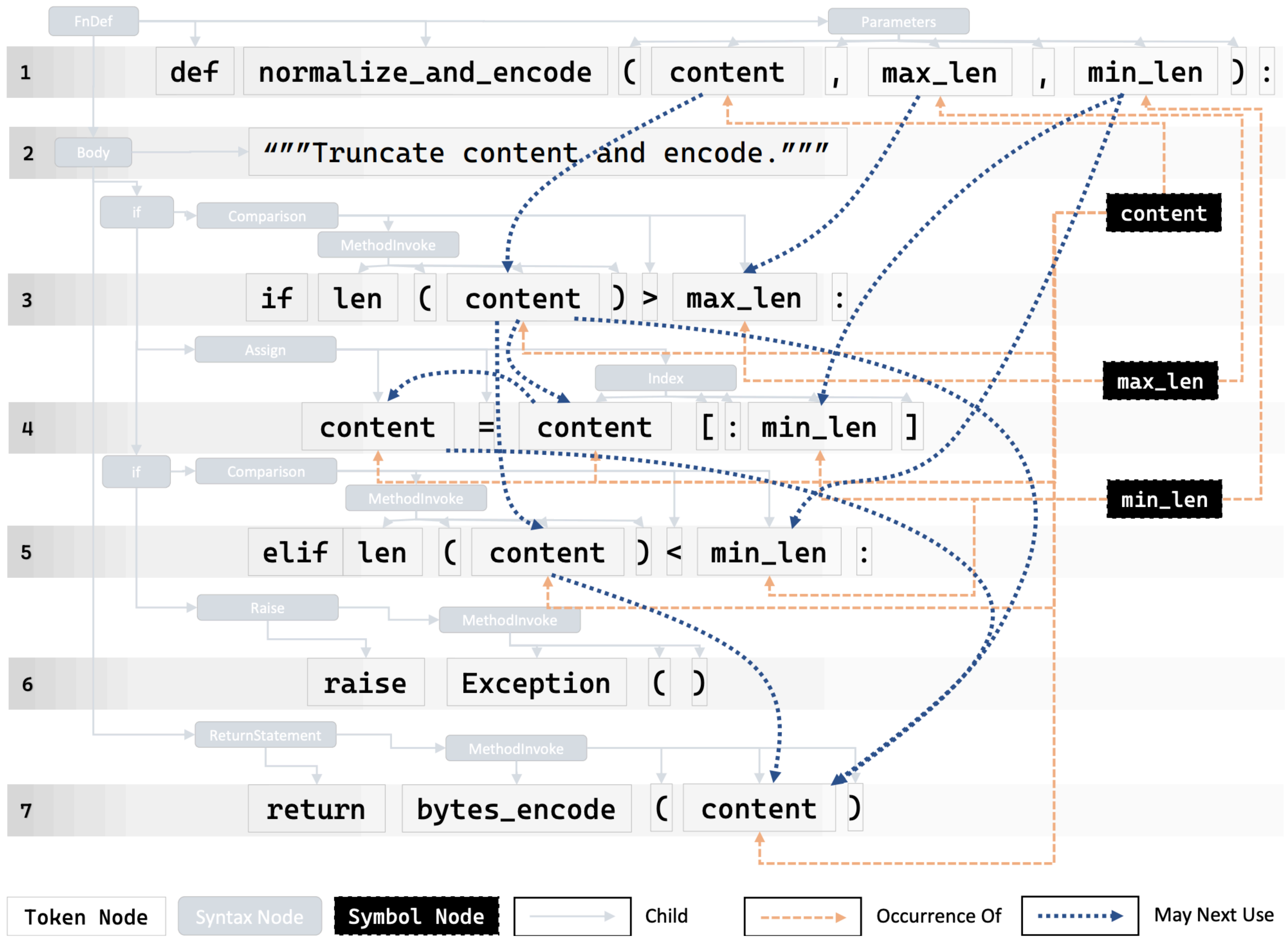
Scene graphs (Johnson et al., 2015): A scene as a graph.

Relational Data



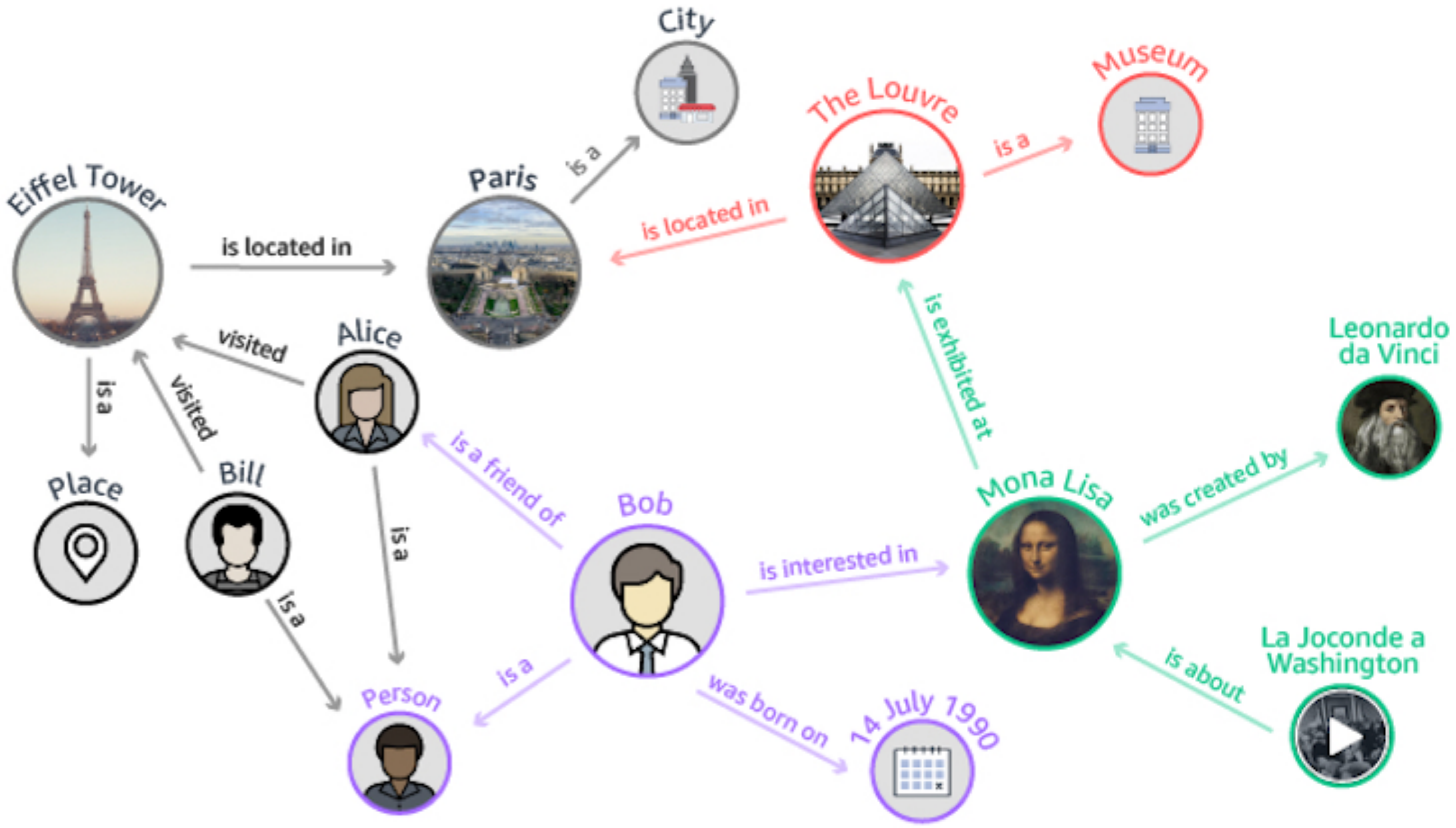
Traffic networks: An excerpt of the London Tube of Zone 1, showing different lines.

Relational Data



Program dependency graphs (Allamanis, 2021): Figure shows a Python program and its dependencies represented as a graph.

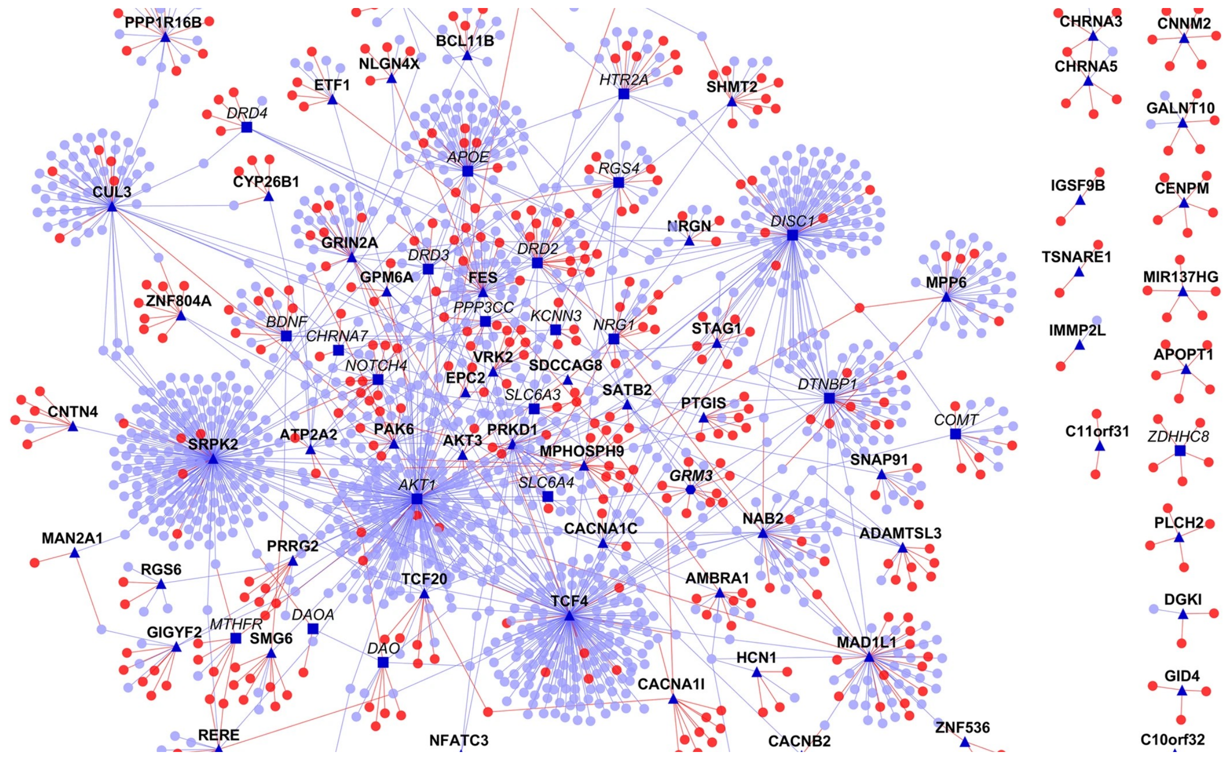
Relational Data

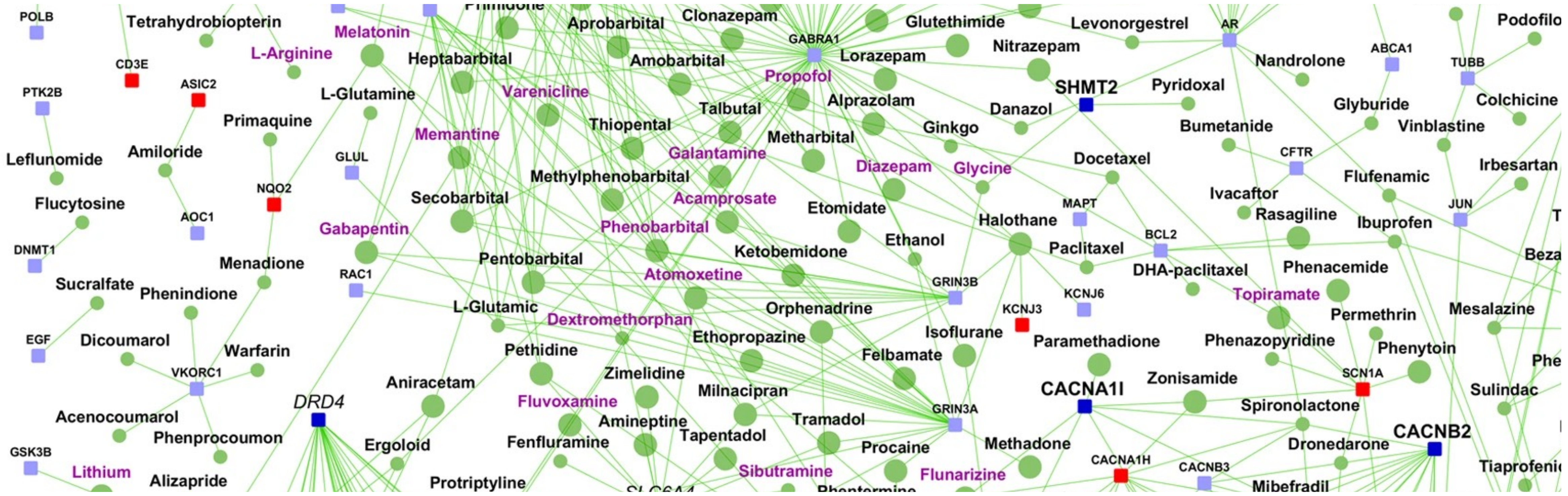


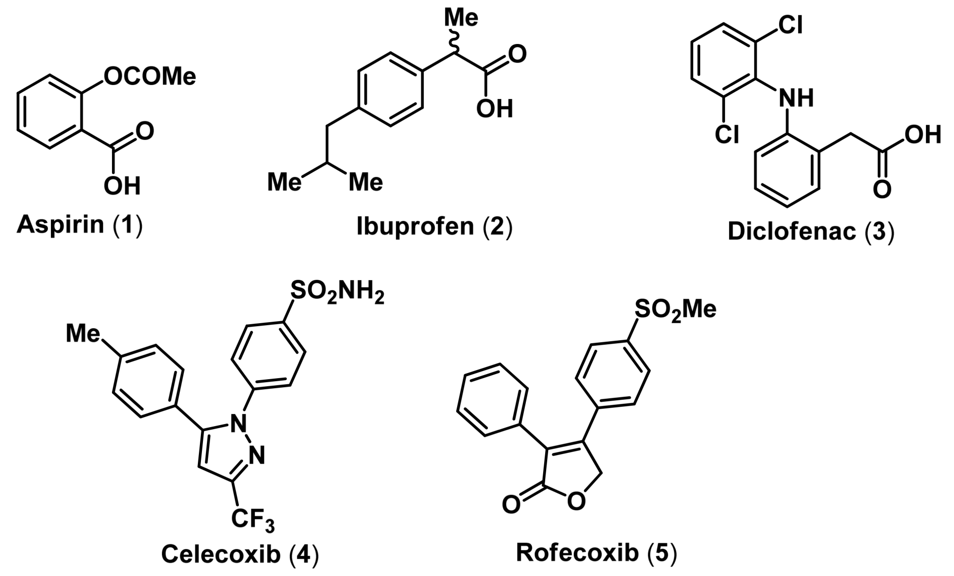
Knowledge graphs: Graph-structured data models, storing relations (e.g., isFriendOf) between entities (e.g., Alice, Bob) and thereby capture structured knowledge.

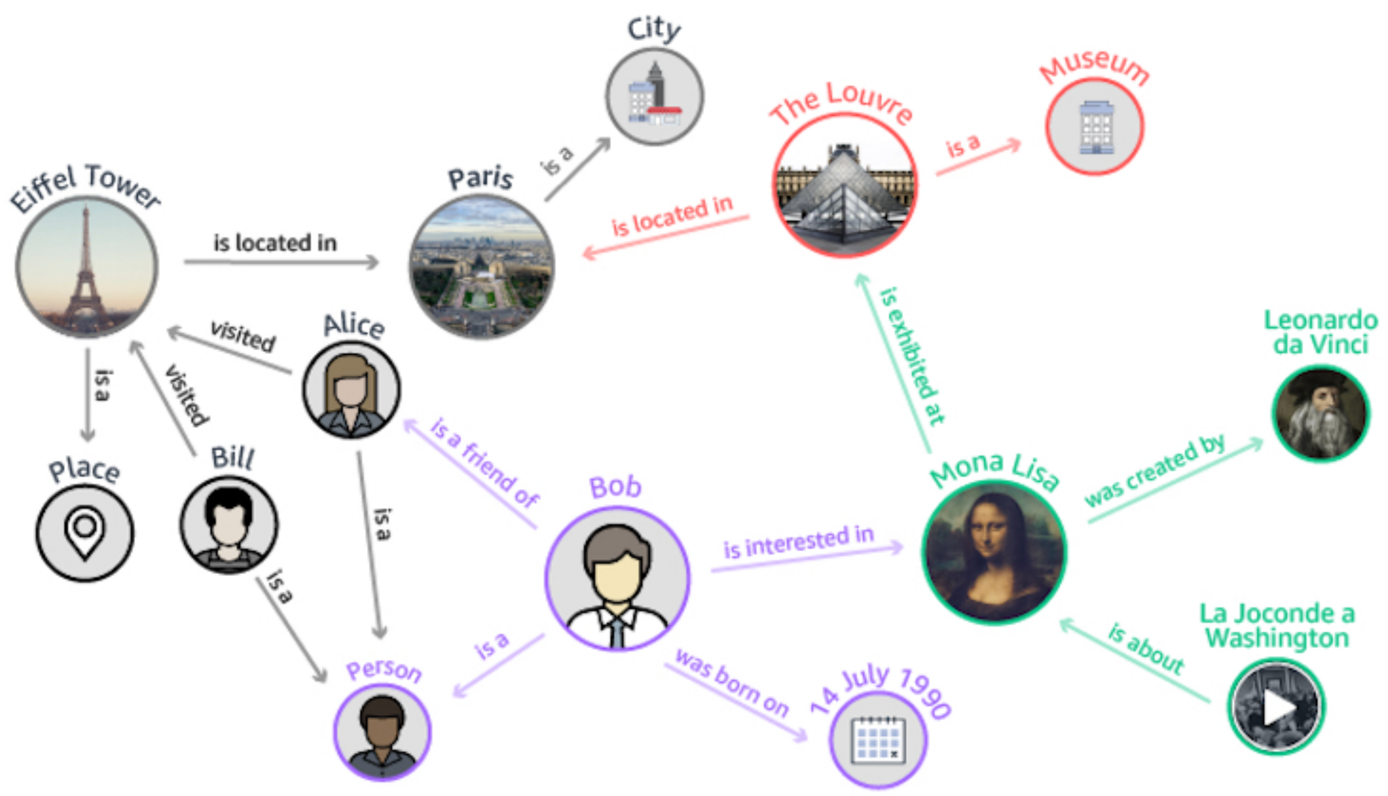
Graph Representation Learning

Beyond Euclidian Spaces

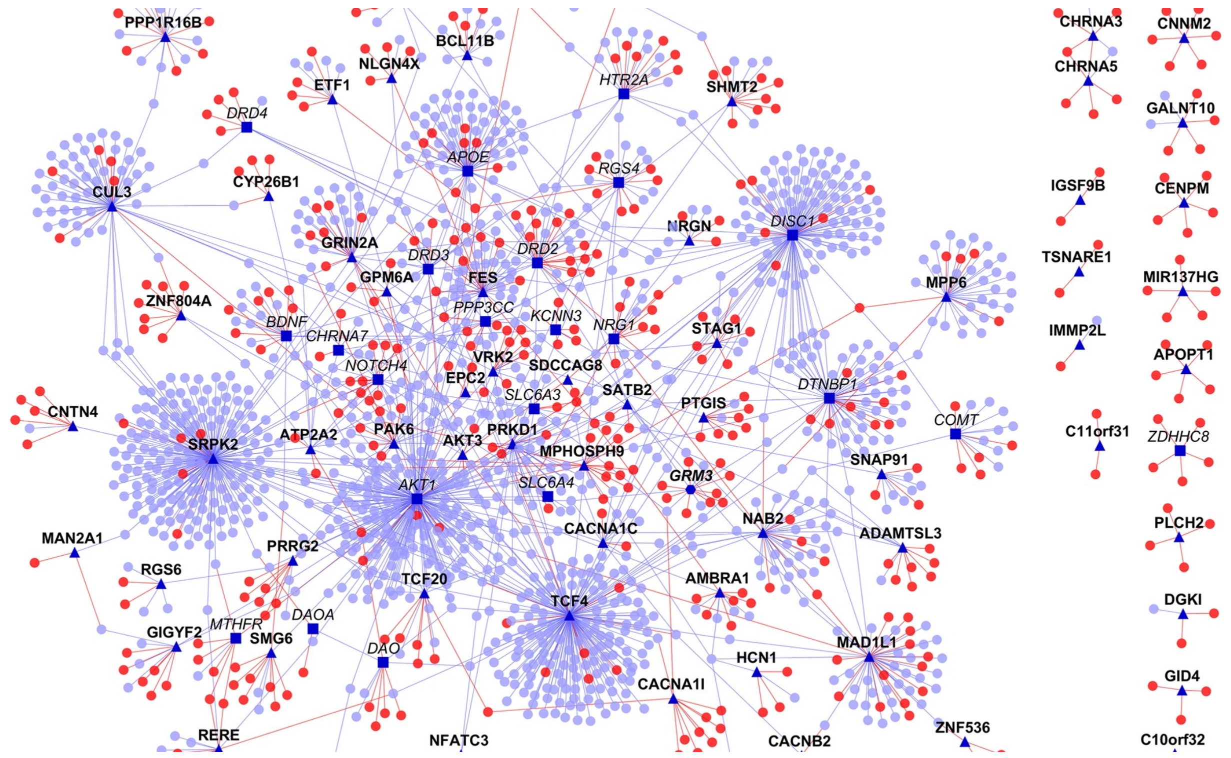


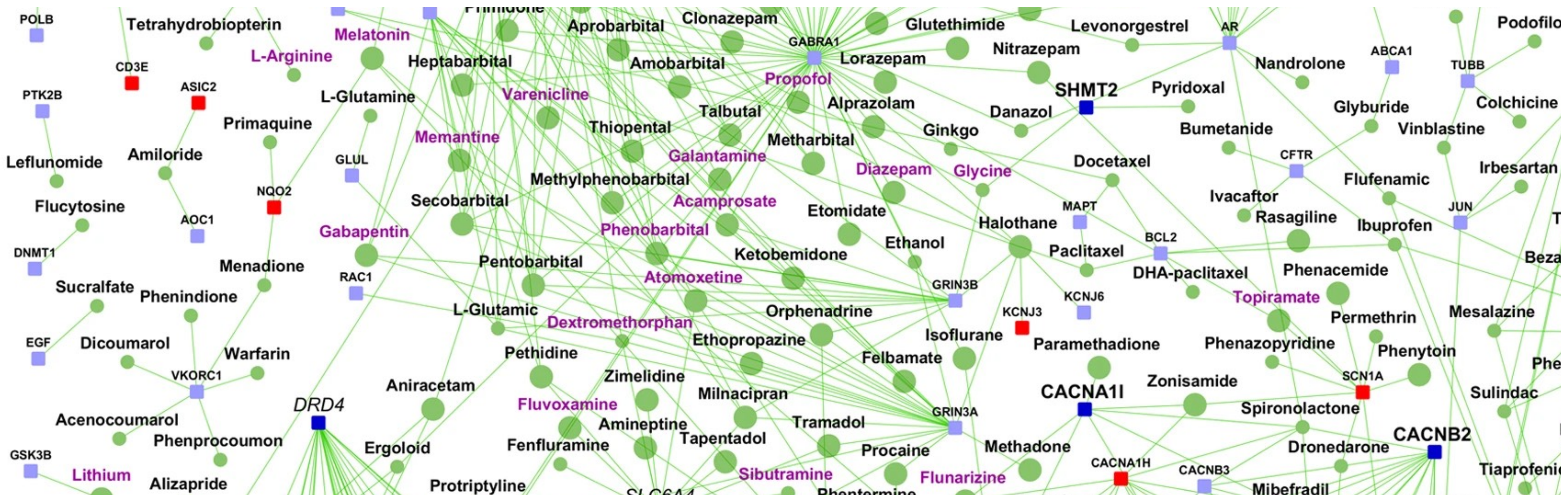


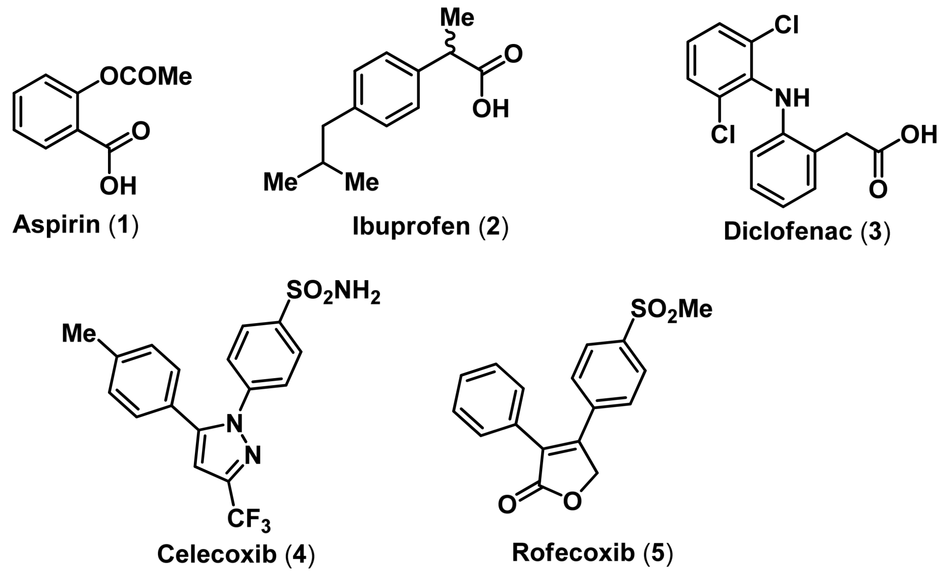


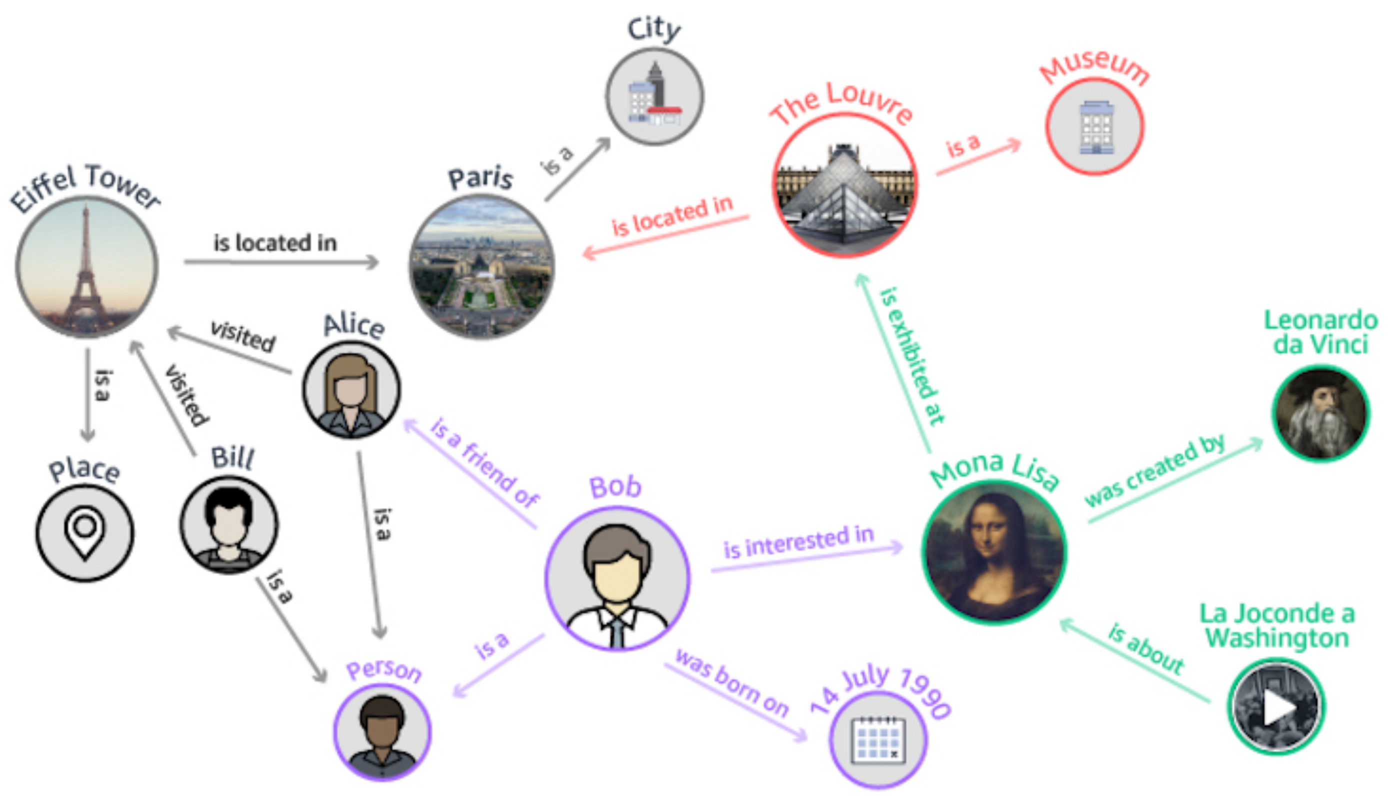


Beyond Euclidian Spaces





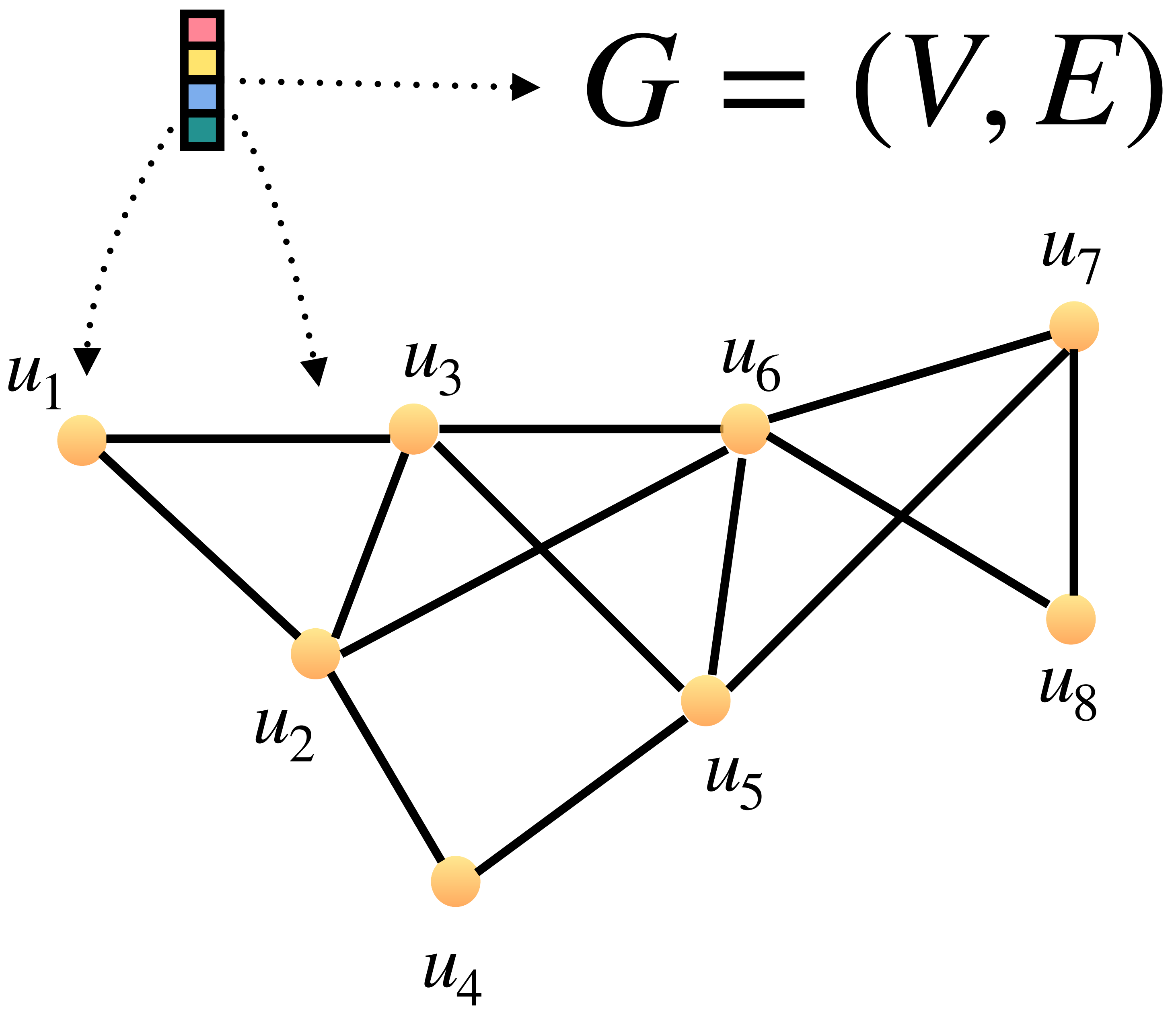




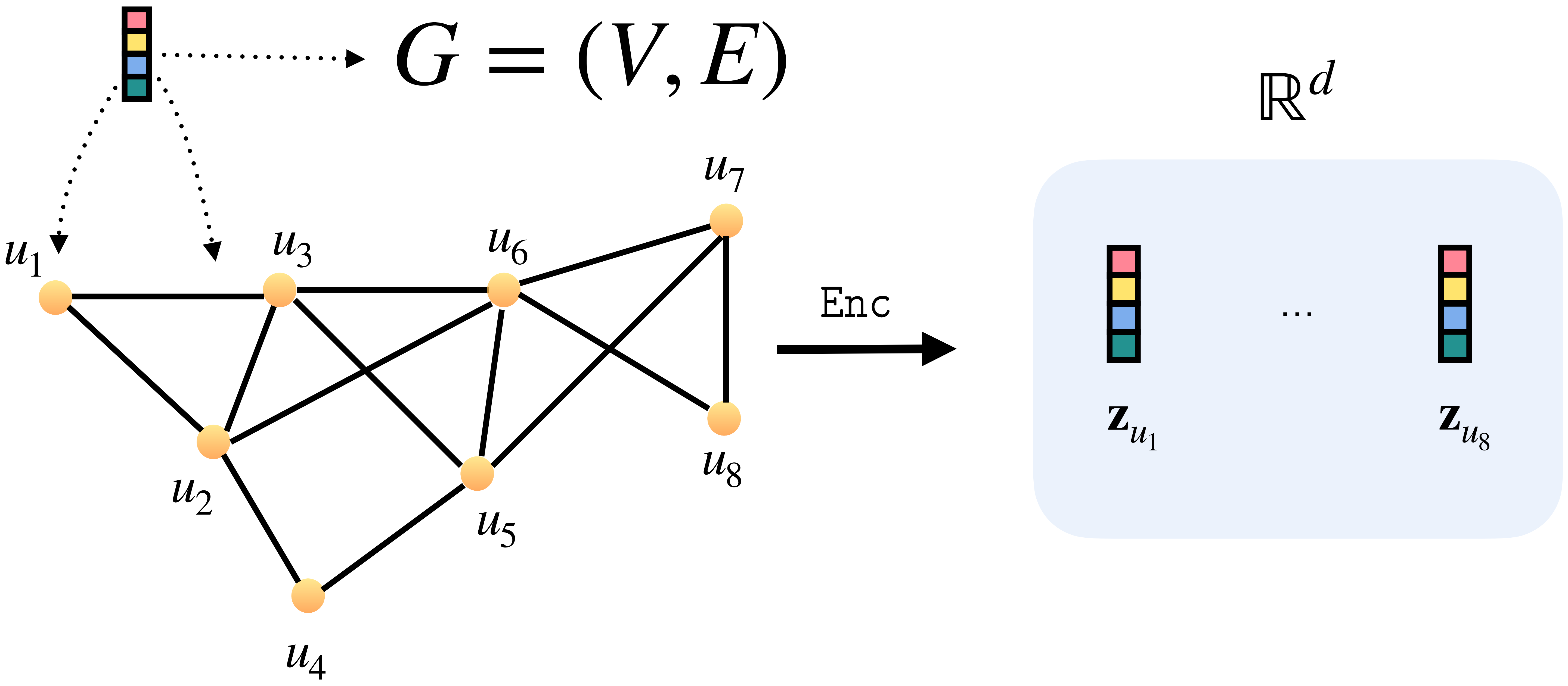
Graph representation learning is an important branch of geometric deep learning which is an umbrella term for deep learning over (non)-Euclidian spaces (Bronstein et al.).

An Encoder-Decoder Perspective

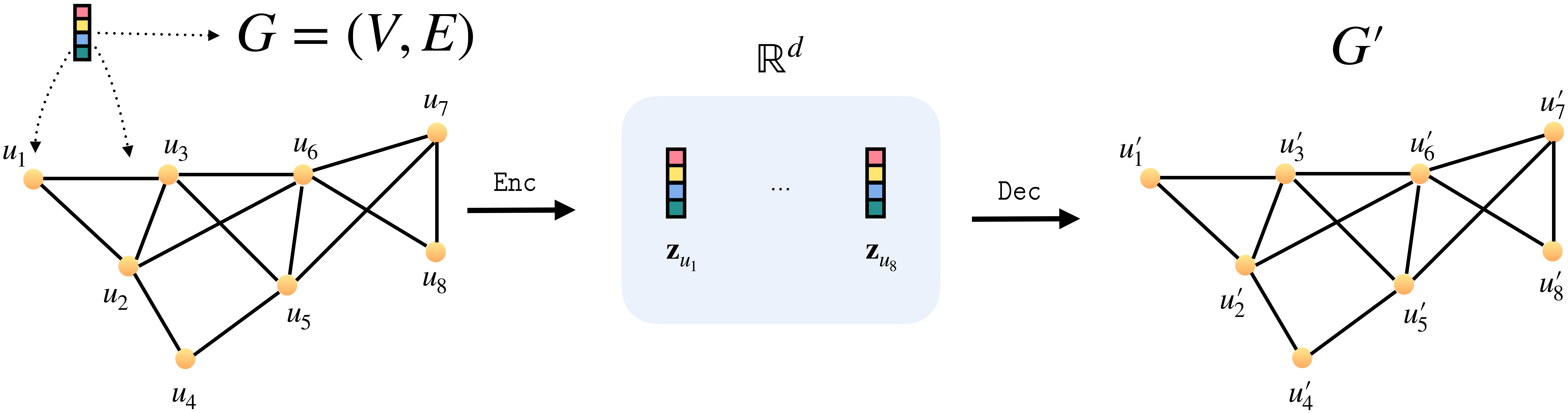
An Encoder-Decoder Perspective



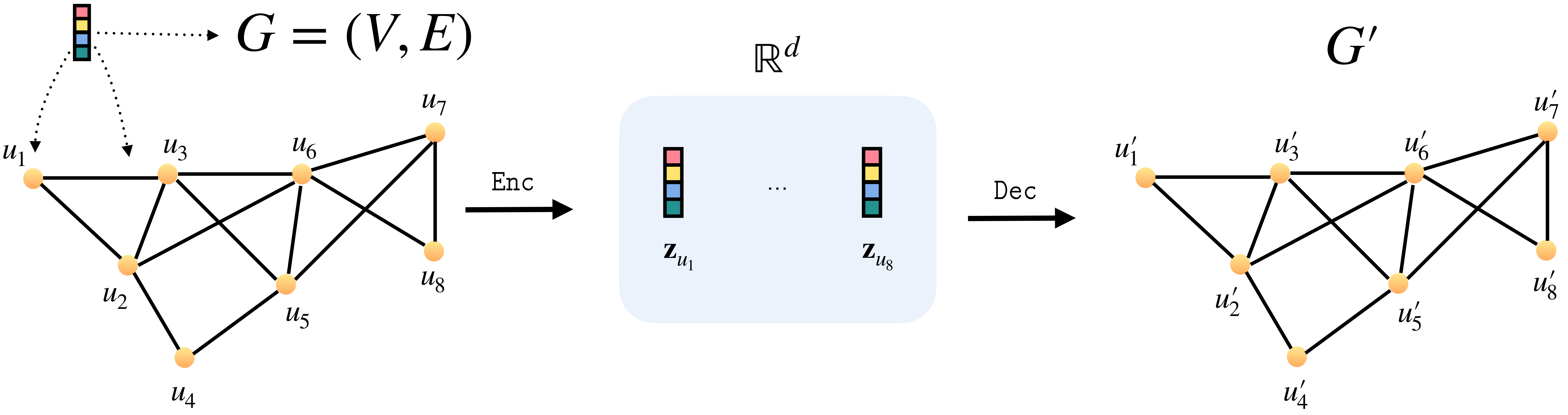
An Encoder-Decoder Perspective



An Encoder-Decoder Perspective



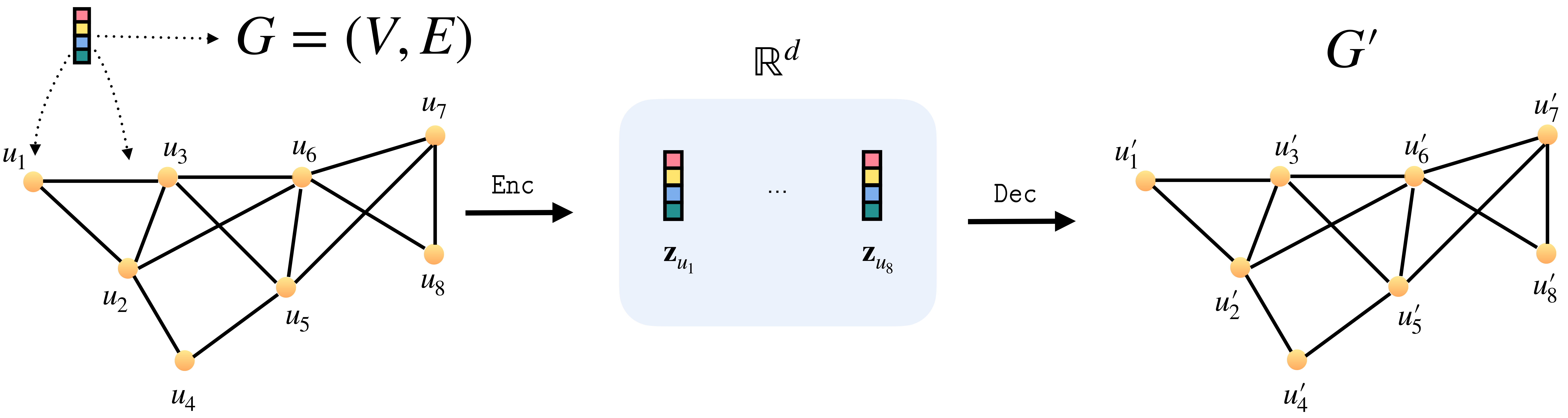
An Encoder-Decoder Perspective



Goal: Embedding nodes, edges, graphs, along with their features, and use these embeddings for predicting node-level, edge-level, or graph-level properties.

Intuition: Nodes/edges/graphs with "similar properties" should have representations closer to each other than nodes/edges/graphs with "dissimilar properties".

An Encoder-Decoder Perspective

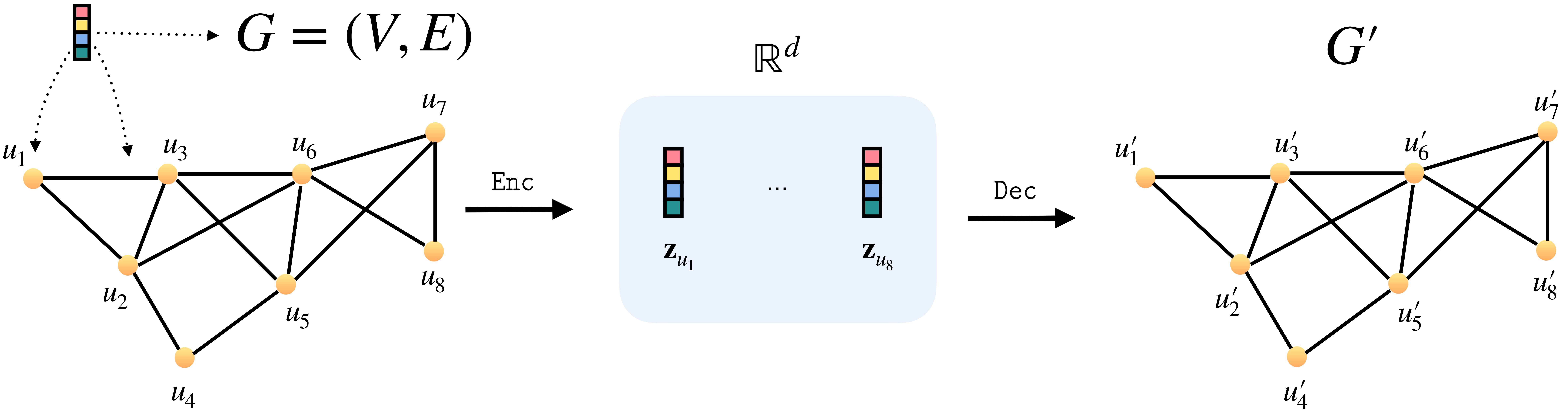


Training: Let $\mathbf{S}[u, v]$ be a similarity measure between the nodes u, v and suppose:

$$\texttt{Enc} : V \to \mathbb{R}^d \qquad\qquad \texttt{Dec} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$$

Optimization: $\forall u, v \in V : \texttt{Dec}(\texttt{Enc}(u), \texttt{Enc}(v)) = \texttt{Dec}(\mathbf{z_u}, \mathbf{z_v}) \sim \mathbf{S}[u, v]$, i.e., minimize the reconstruction loss.

An Encoder-Decoder Perspective



Graph representation learning tasks: Various node/edge/graph level tasks are of interest.

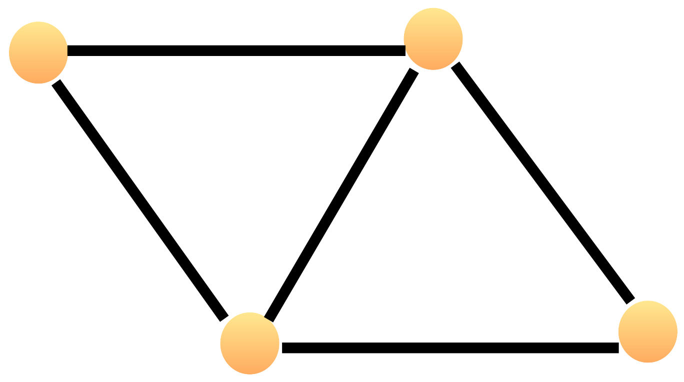
Node-level: Node classification/clustering/regression

Edge-level: Link prediction, knowledge graph completion

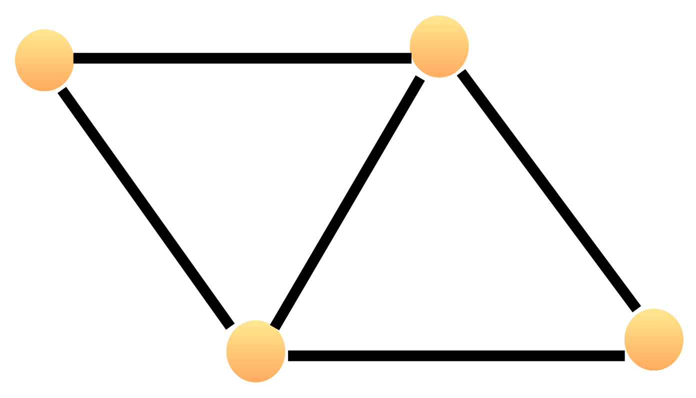
Graph-level: Graph classification/clustering/regression/generation

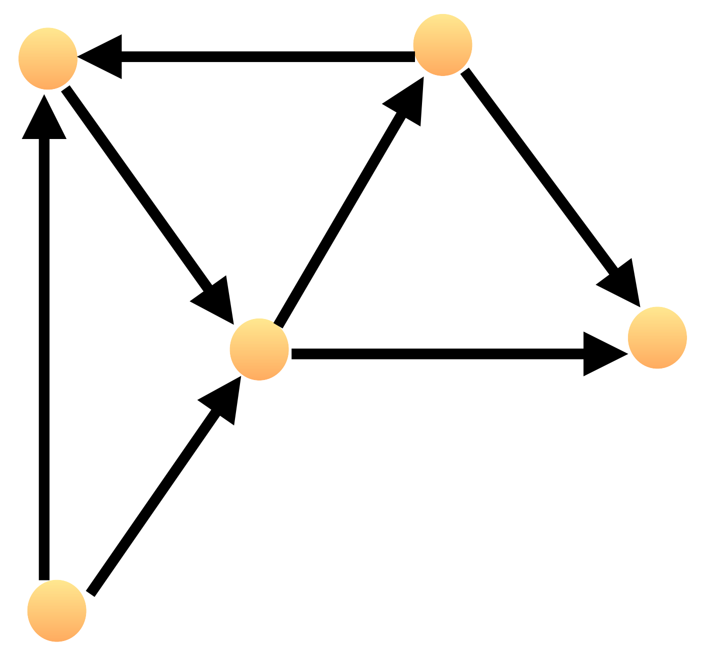
What Kind of Graphs?

What Kind of Graphs?

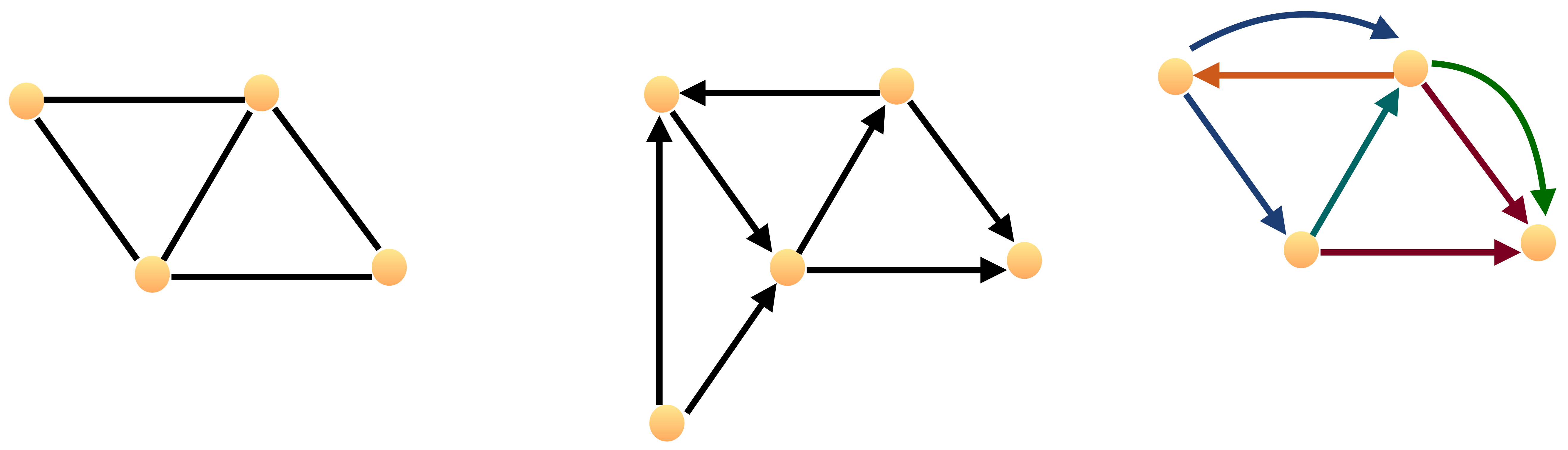


What Kind of Graphs?

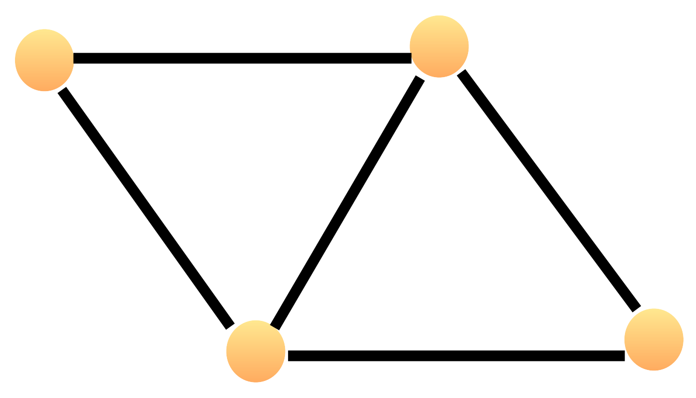


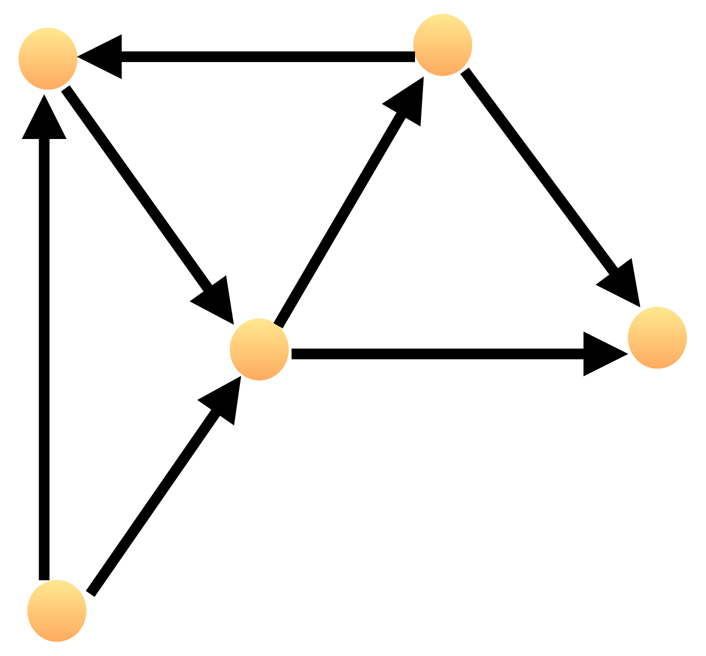


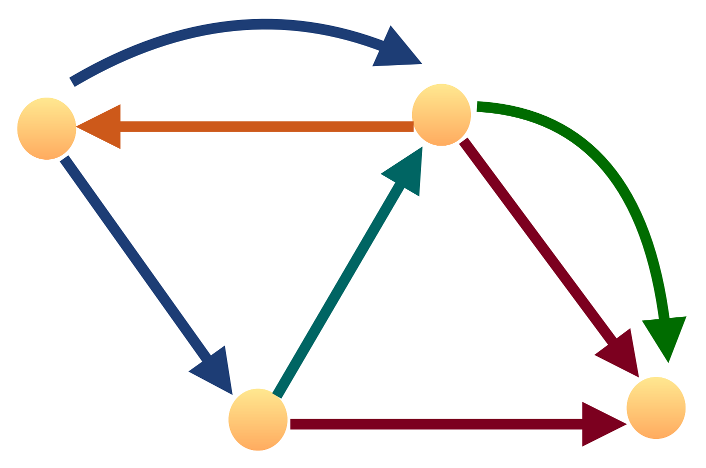
What Kind of Graphs?

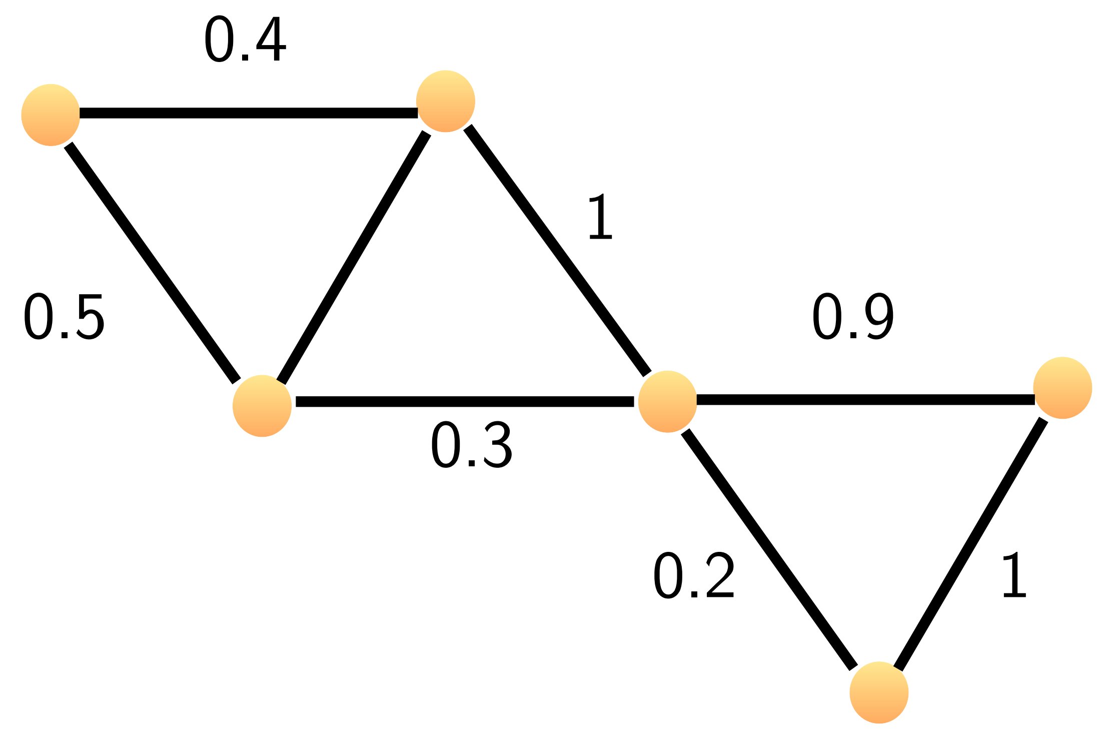


What Kind of Graphs?

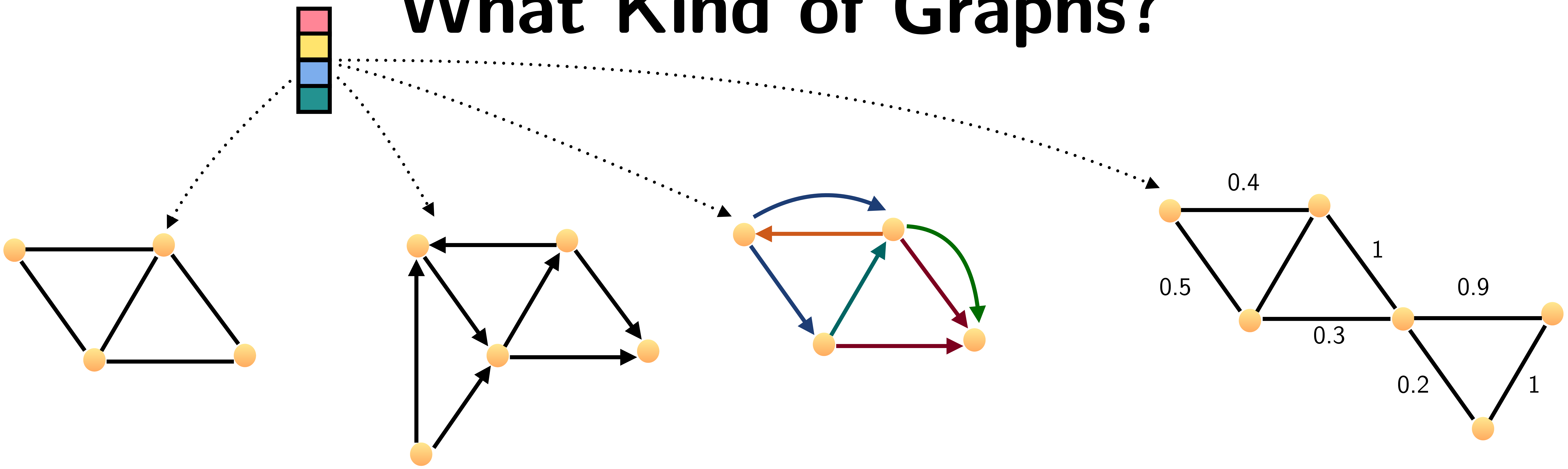




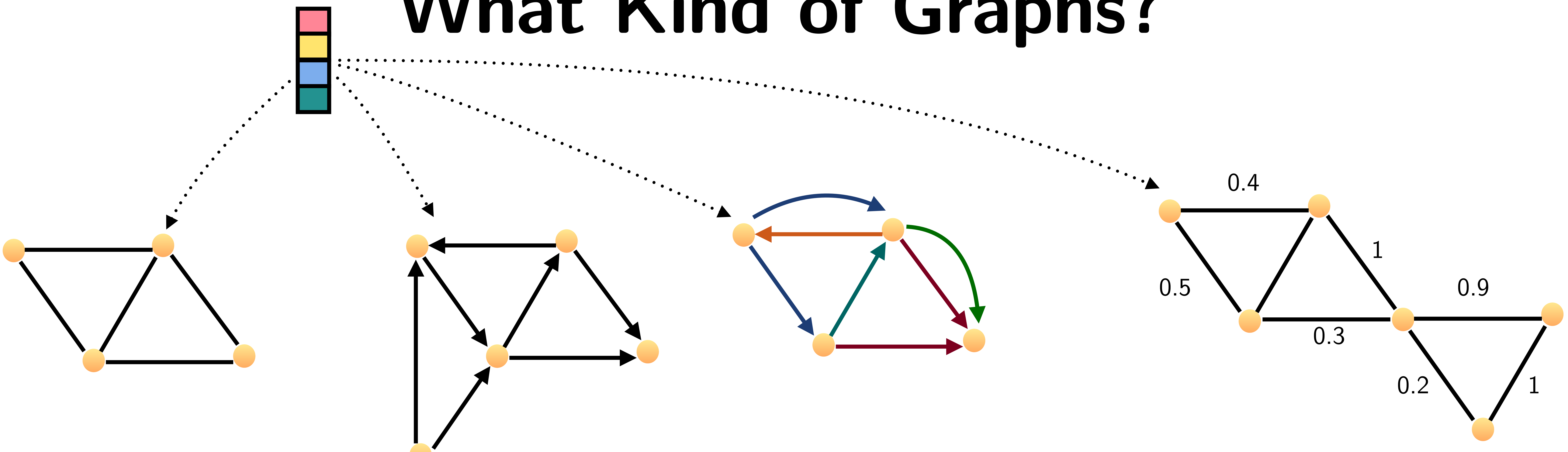


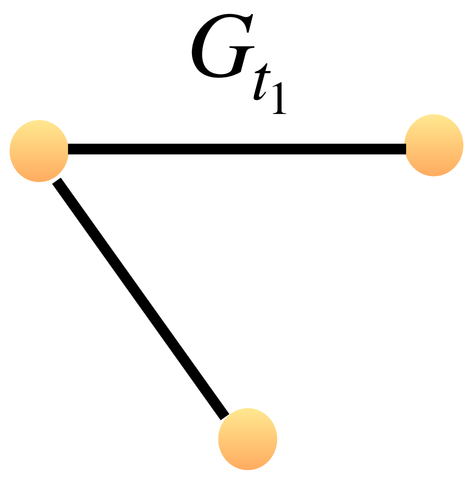


What Kind of Graphs?

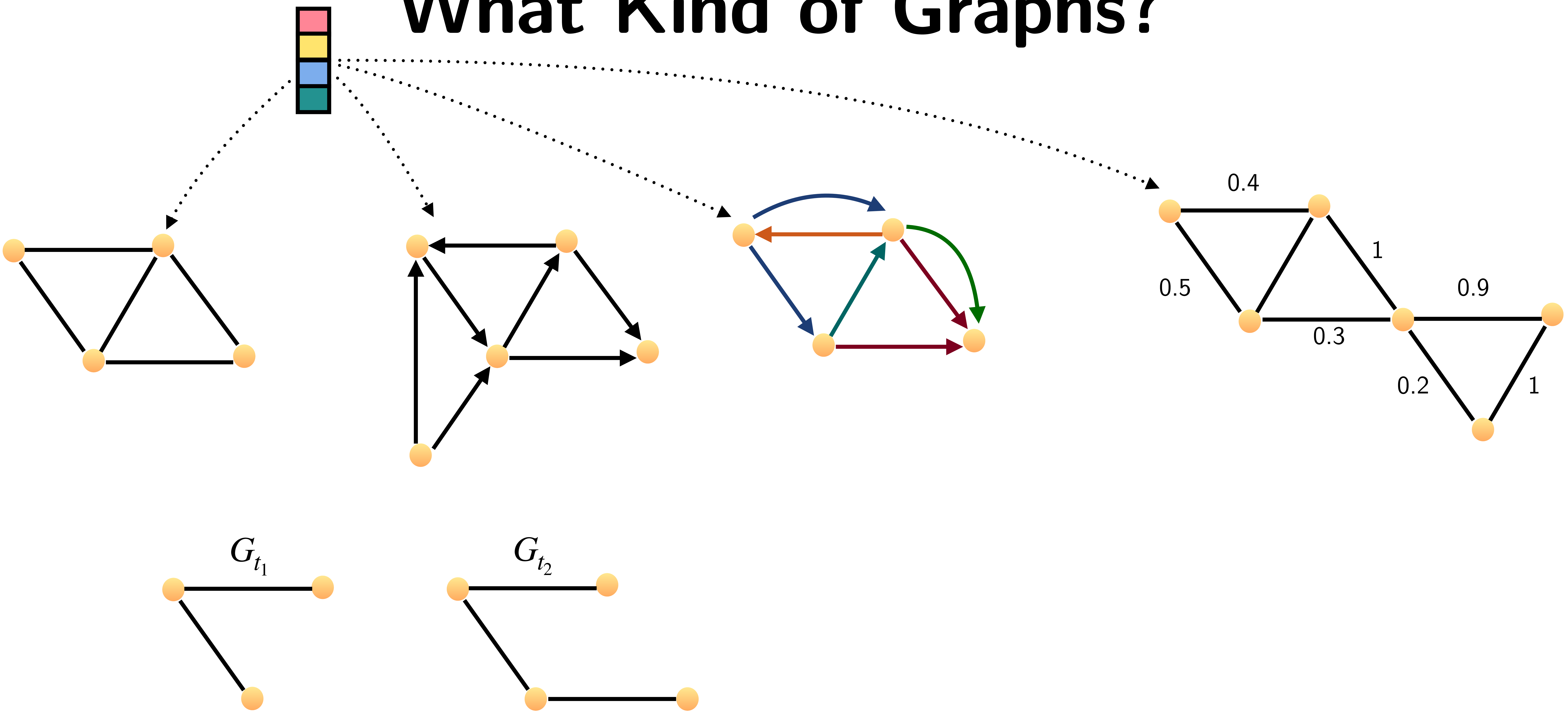


What Kind of Graphs?

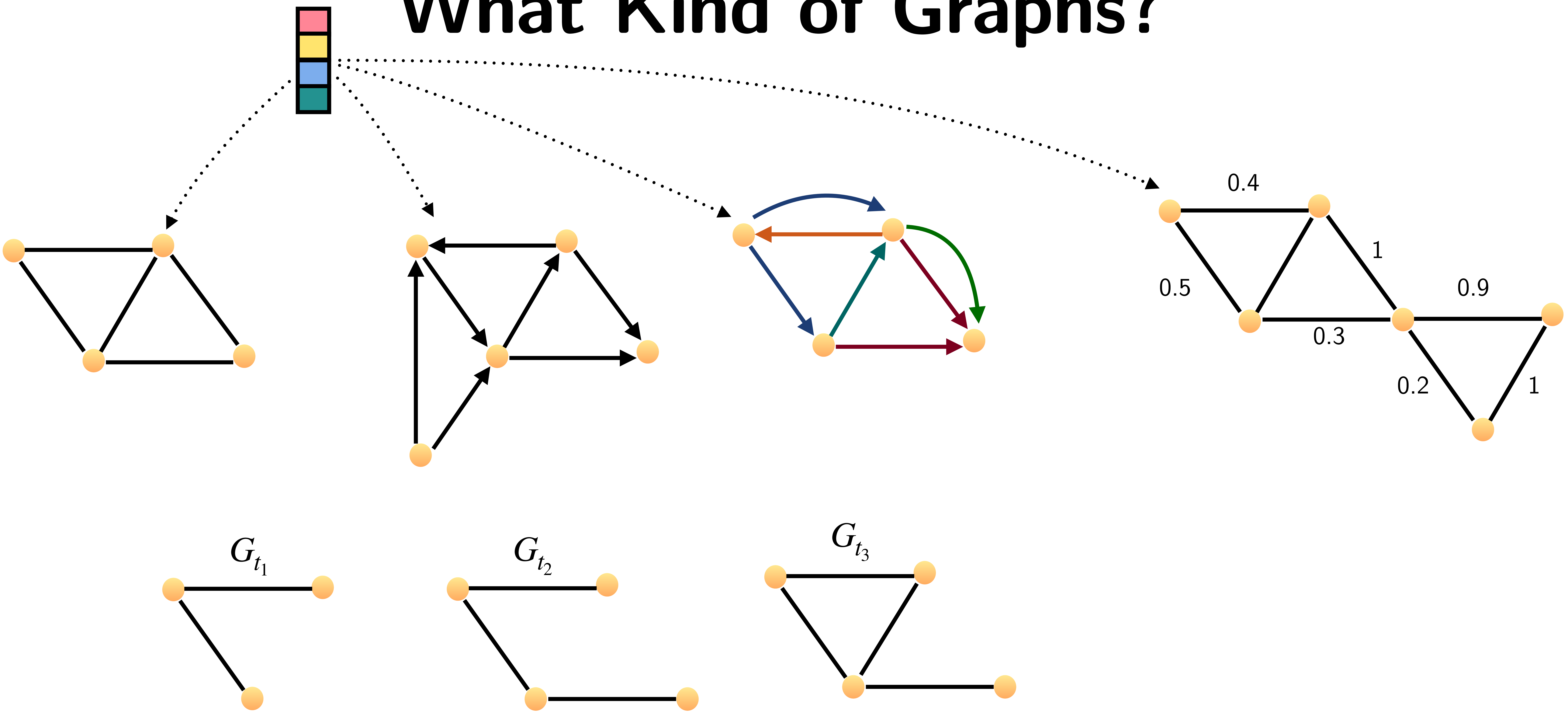




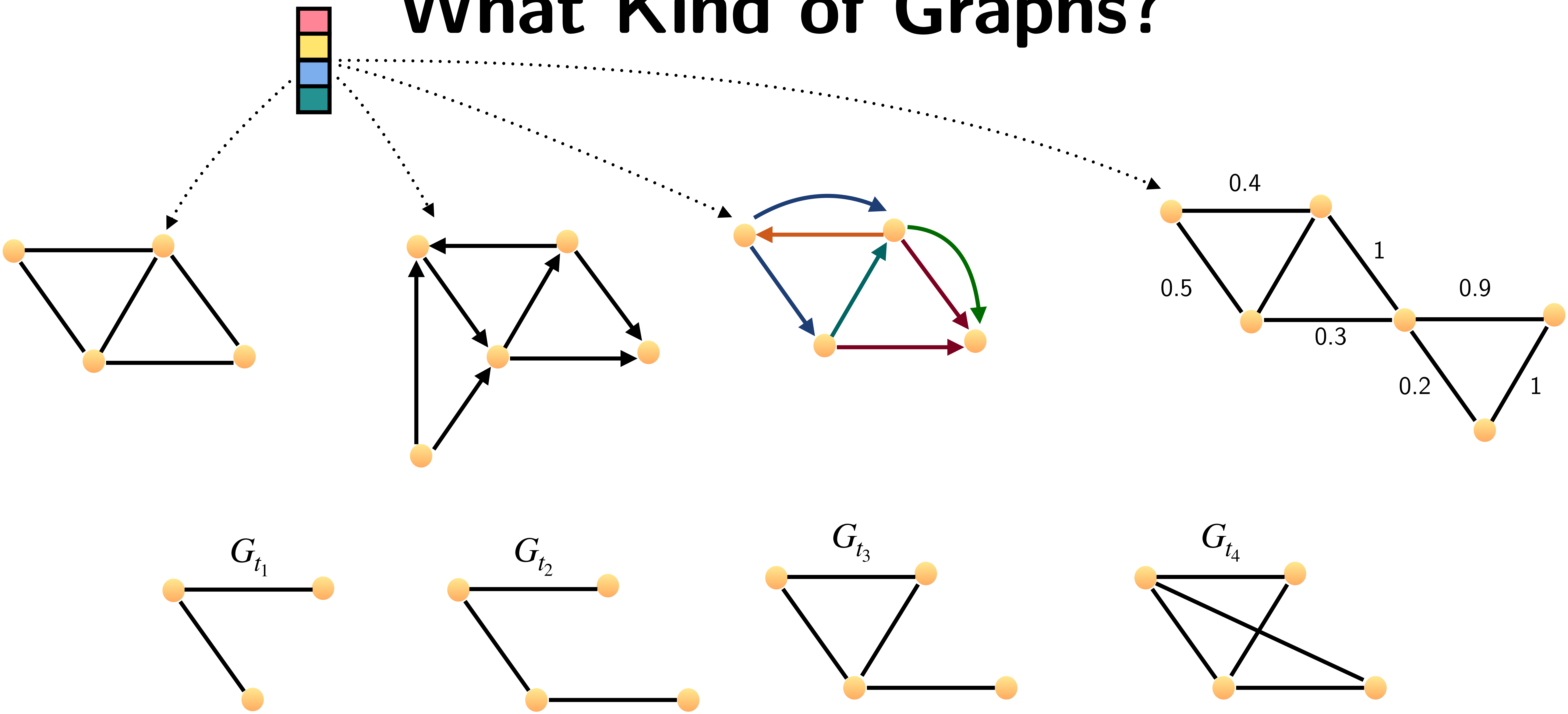
What Kind of Graphs?



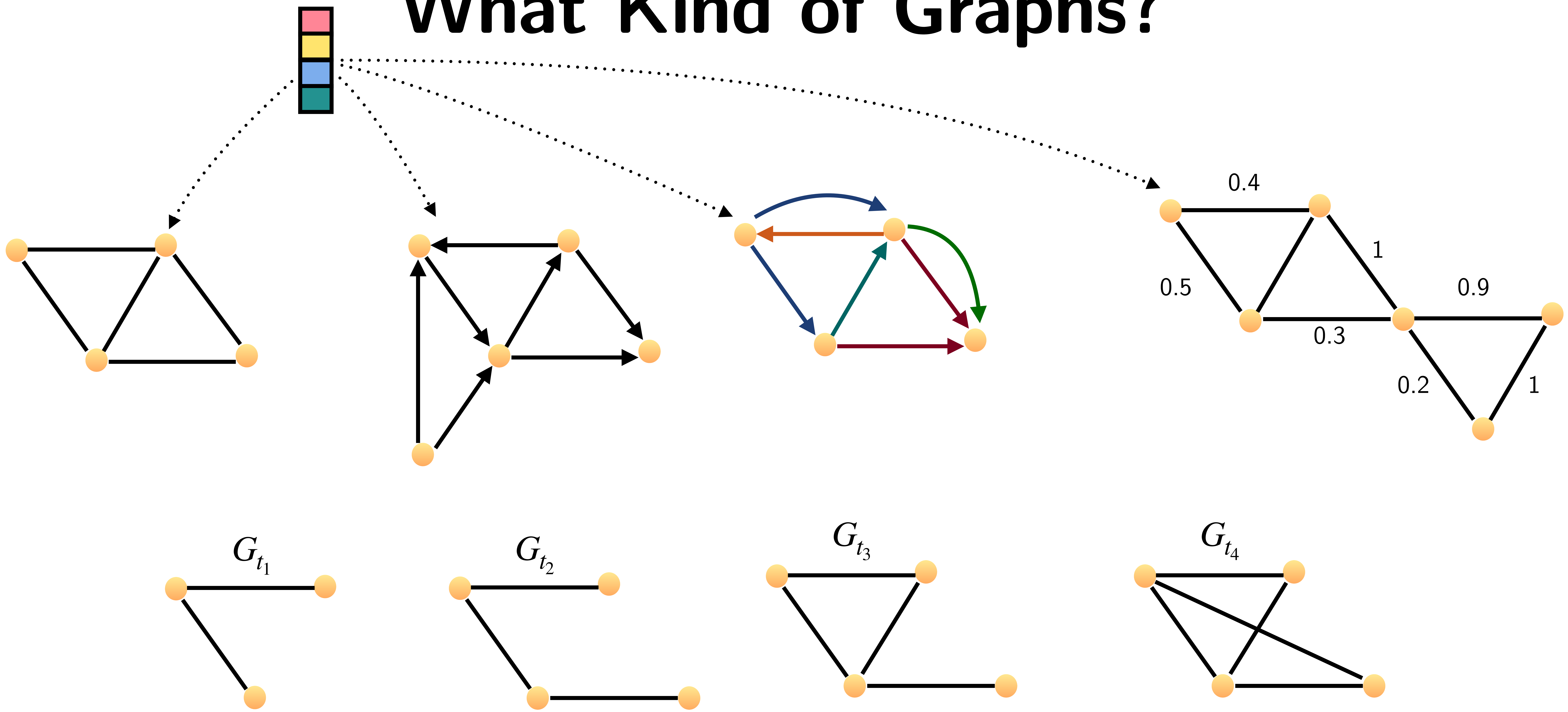
What Kind of Graphs?



What Kind of Graphs?



What Kind of Graphs?



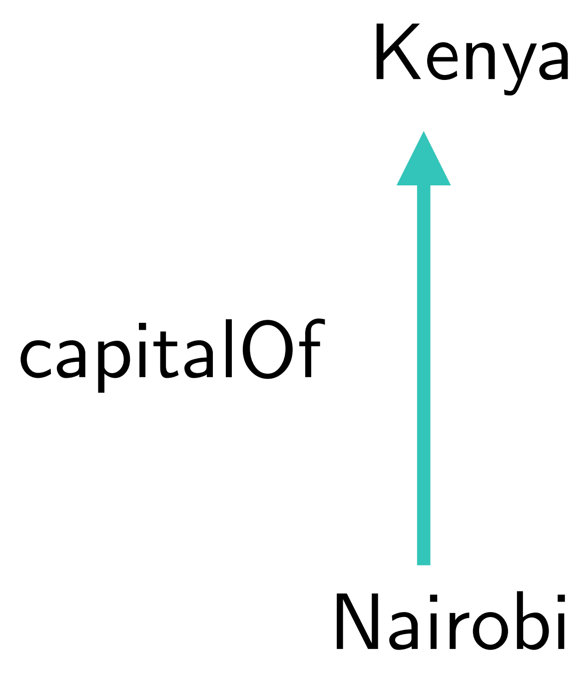
Lecture 1 - 2: Learning with **knowledge graphs** (no features) using shallow embedding models.

Lecture 3 - 9: Learning with (mostly) **undirected graphs** + features using graph neural networks.

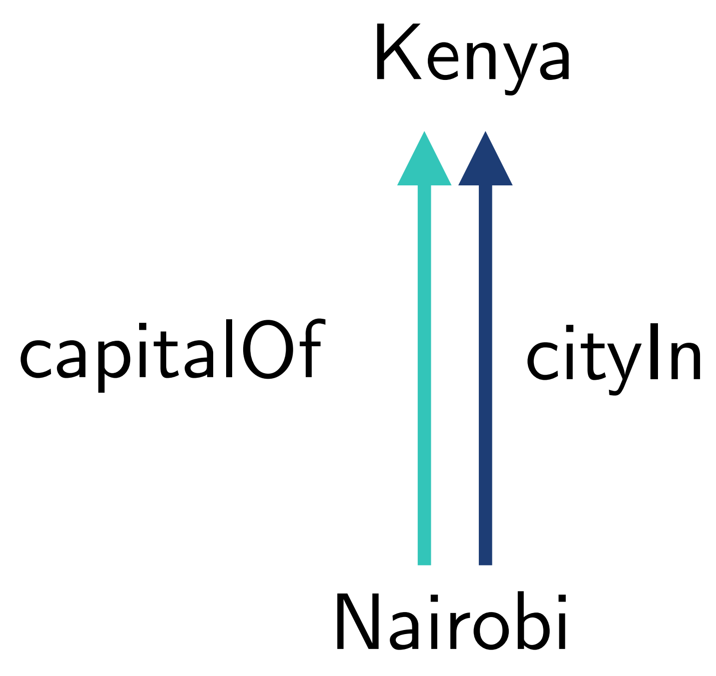
Knowledge Graphs

Knowledge Graphs

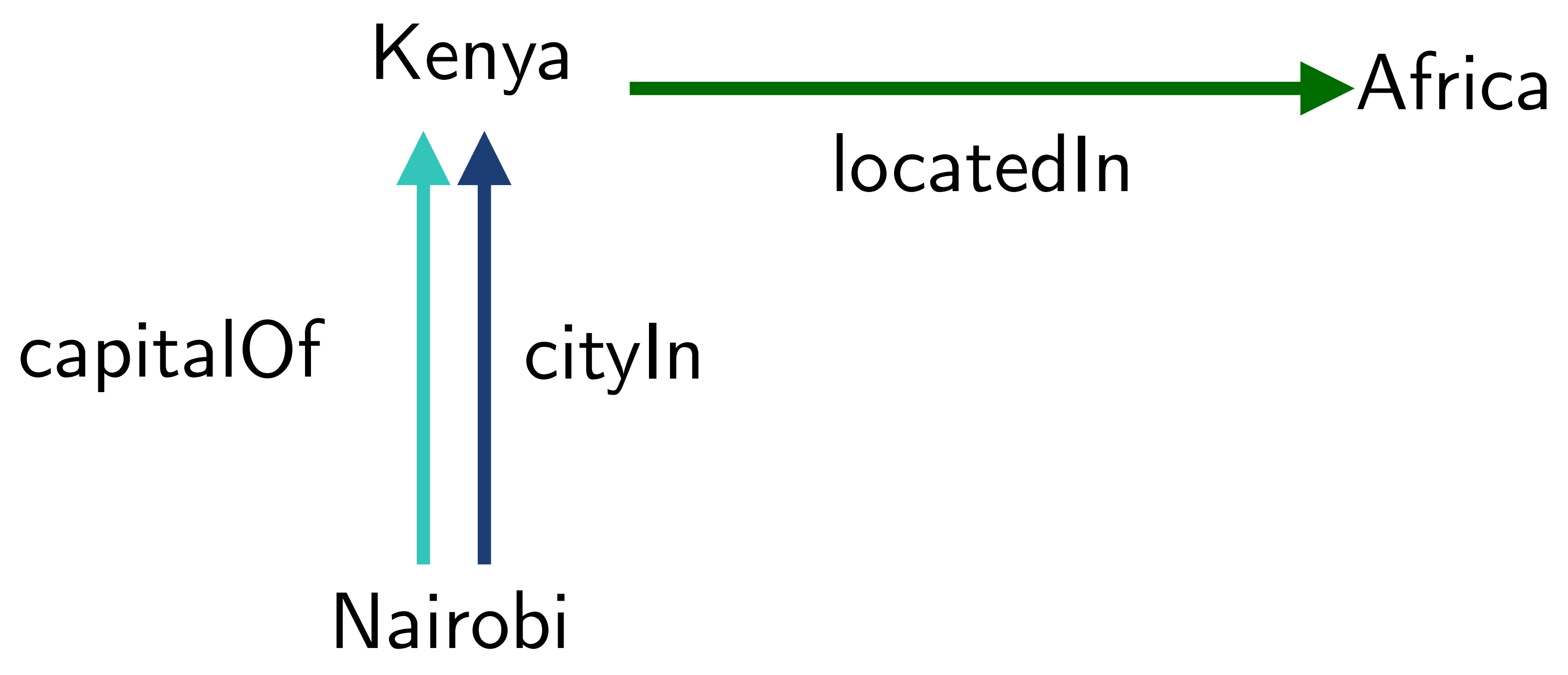
Knowledge Graphs



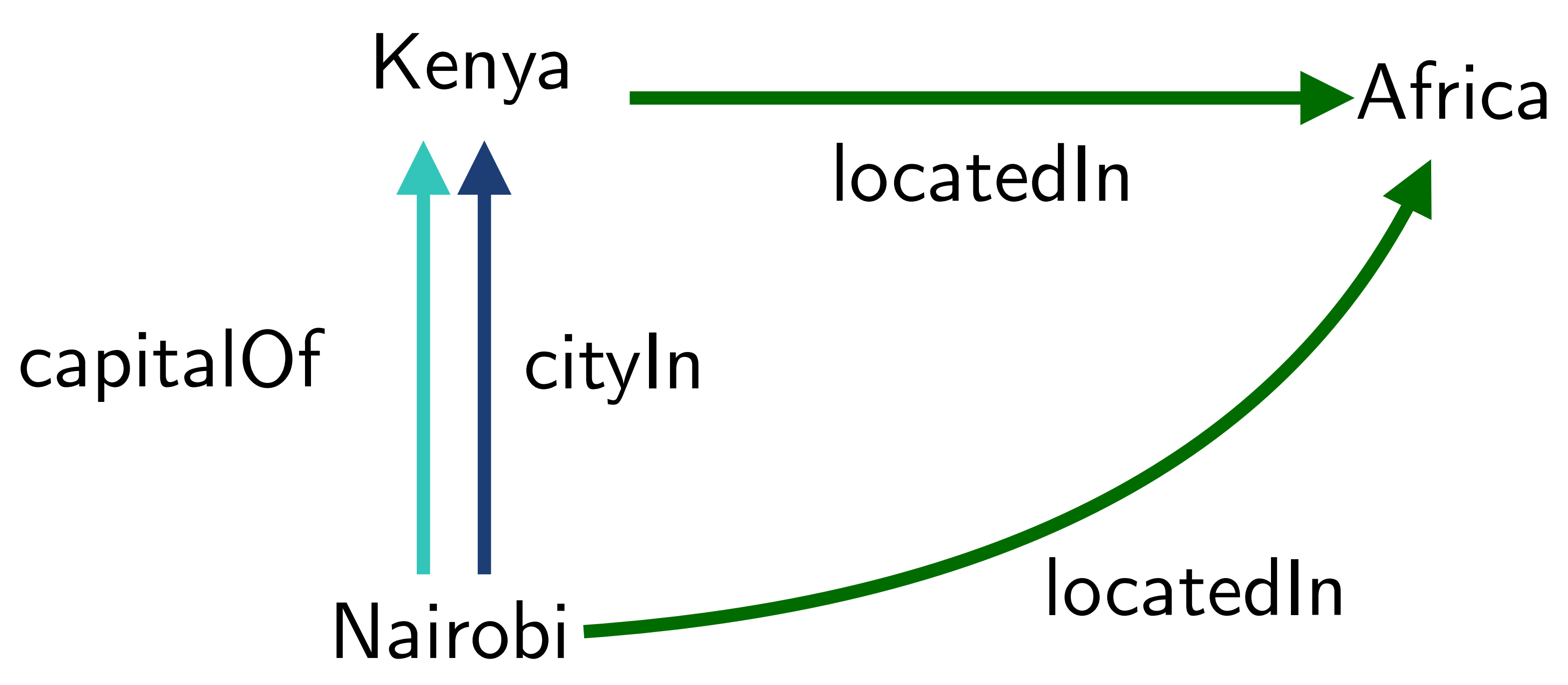
Knowledge Graphs



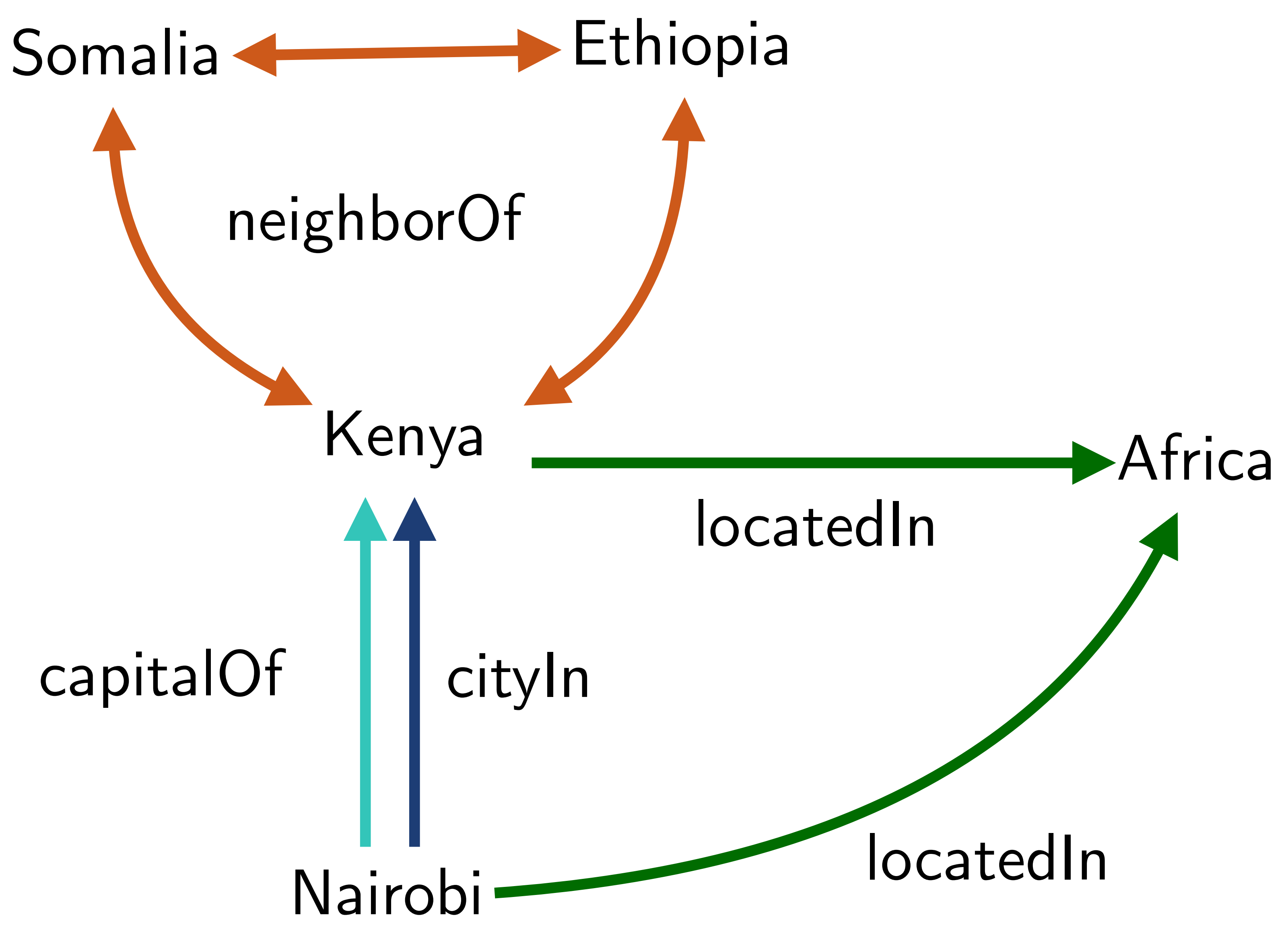
Knowledge Graphs



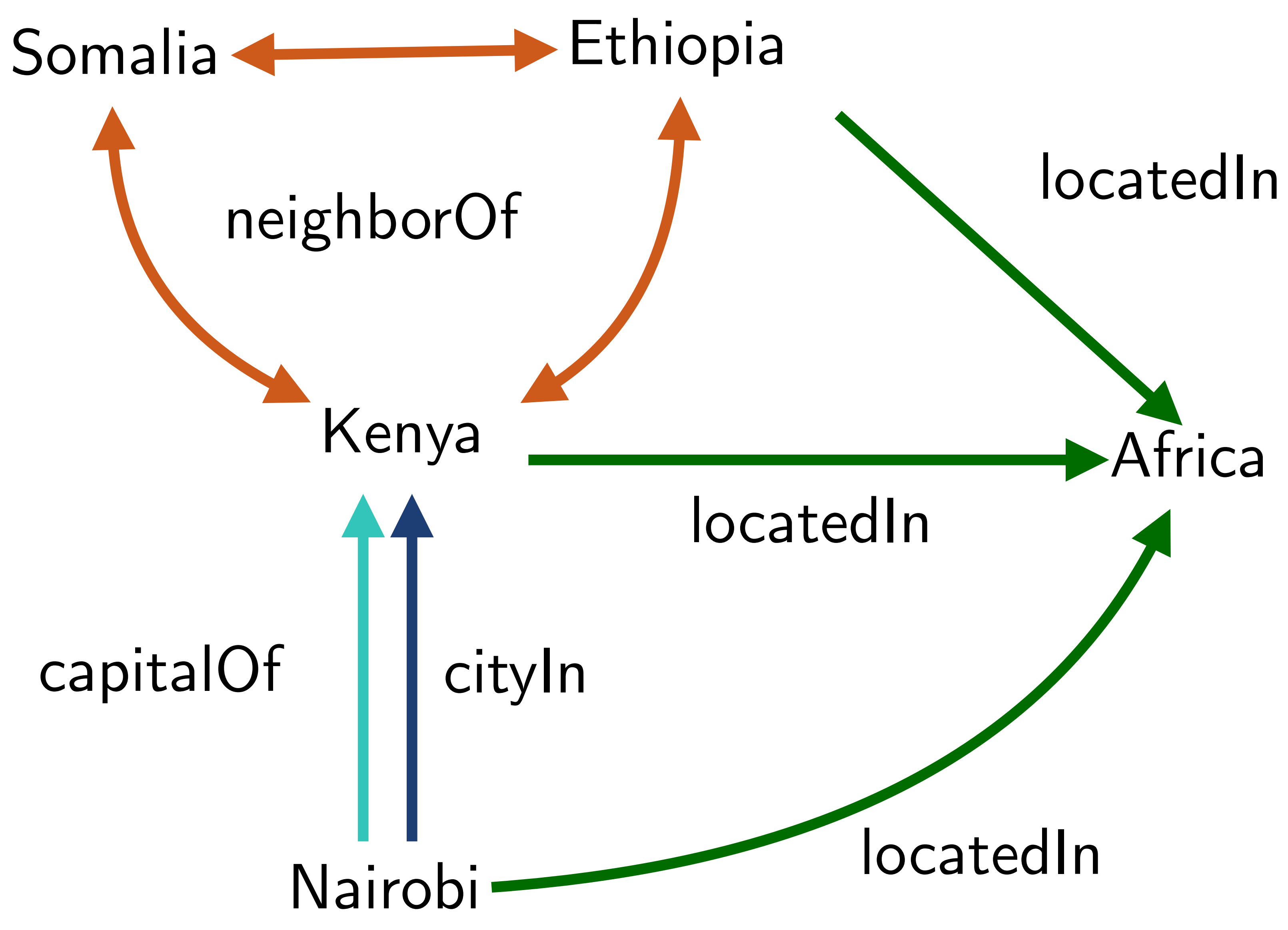
Knowledge Graphs



Knowledge Graphs

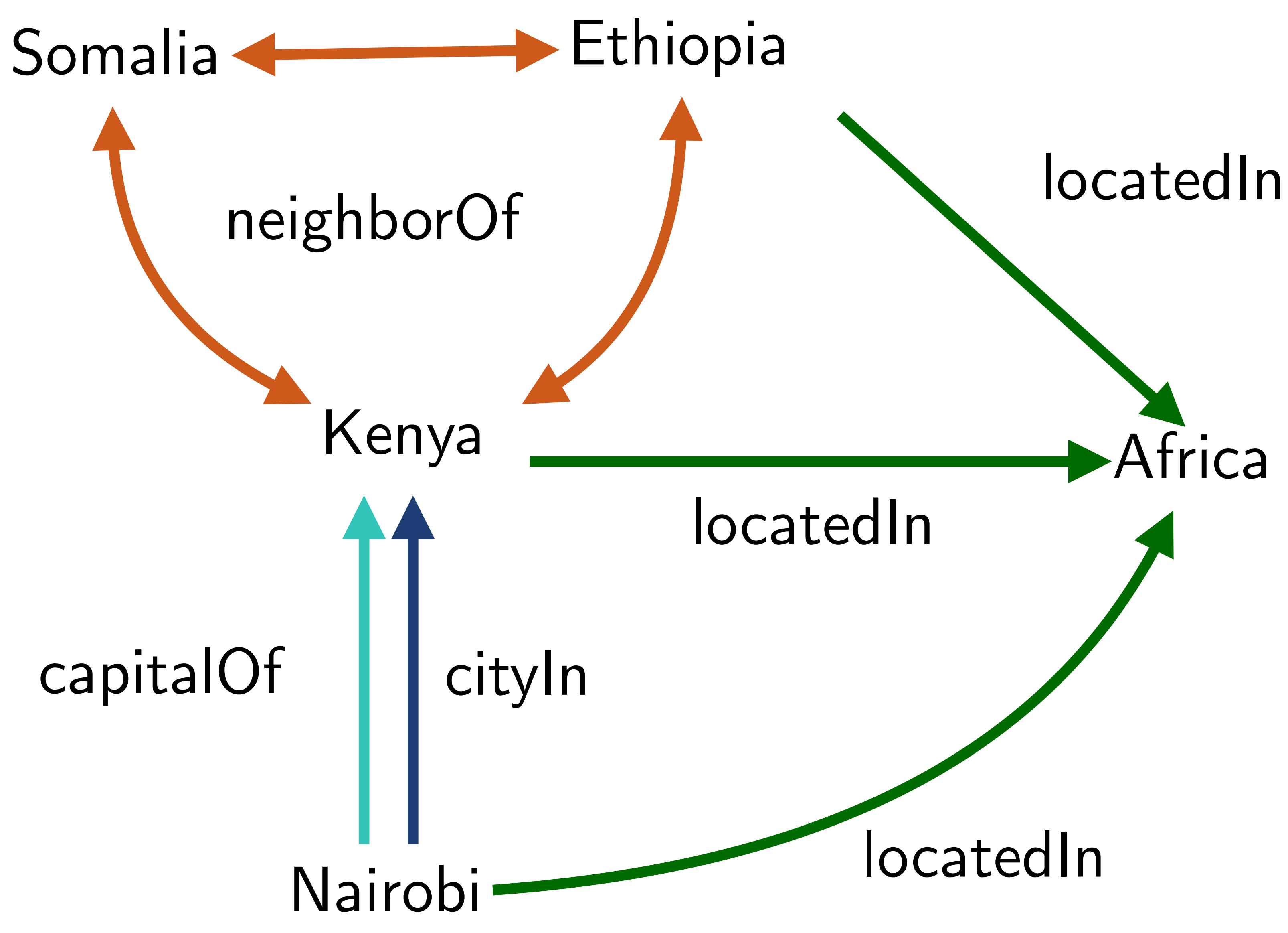


Knowledge Graphs



Knowledge Graphs

- We consider a **relational vocabulary** that consists of a finite set E of entities, and a finite set R of relations.
- A fact is of the form $r(h, t)$, where $r \in R$, and $h, t \in E$.
- We refer to h as the head and t as the tail entity in a fact $r(h, t)$. Such facts are sometimes denoted as triples of the form (h, r, t), i.e., as "subject, predicate, object" triples.
- A knowledge graph (KG) G is a set of facts over E and R; equivalently, a directed, labelled multigraph $G = (E, R)$.
- U is the set of all possible facts over E and R.



Why Knowledge Graphs?

Why Knowledge Graphs?

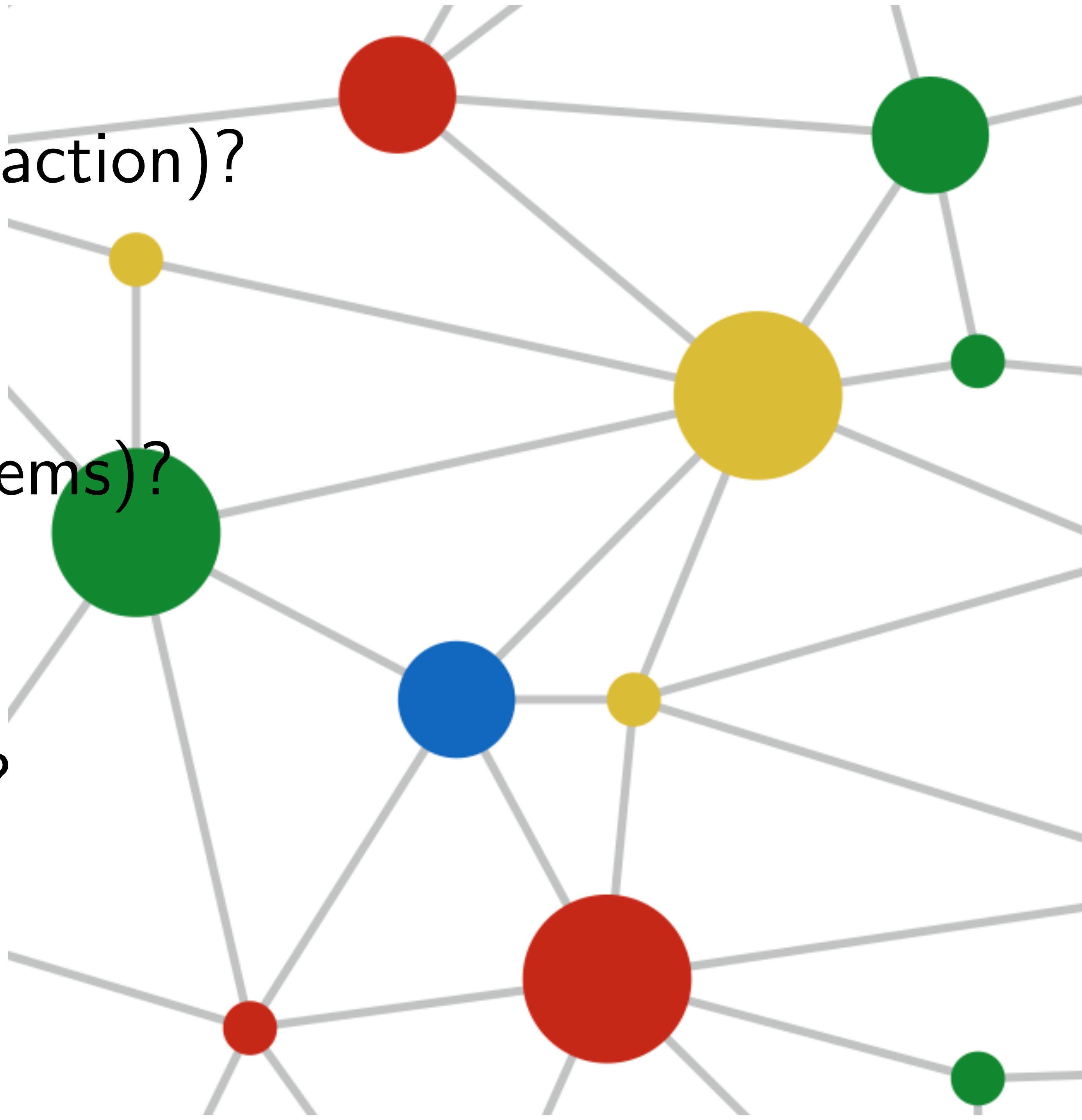
- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.

Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"

Why Knowledge Graphs?

- KGs provide means for storing, processing, and managing structured data, and are part of modern information technologies.
- KGs can be used for reasoning (in conjunction with ontologies), and for query answering, i.e., "Who has co-authored a paper with Marie Curie and Pierre Curie?"
- KGs pose (or, relate to) various challenges in AI & machine learning:
  - How to automatically construct KGs (e.g., relation extraction, open information extraction)?
  - How to populate an existing KG with new facts (e.g., KG completion)?
  - How to improve/personalize information systems using KGs (e.g., recommender systems)?
  - How to learn on top of KGs, while complying with the existing knowledge?
  - Can KGs be mediators for developing more reliable and interpretable models for ML?

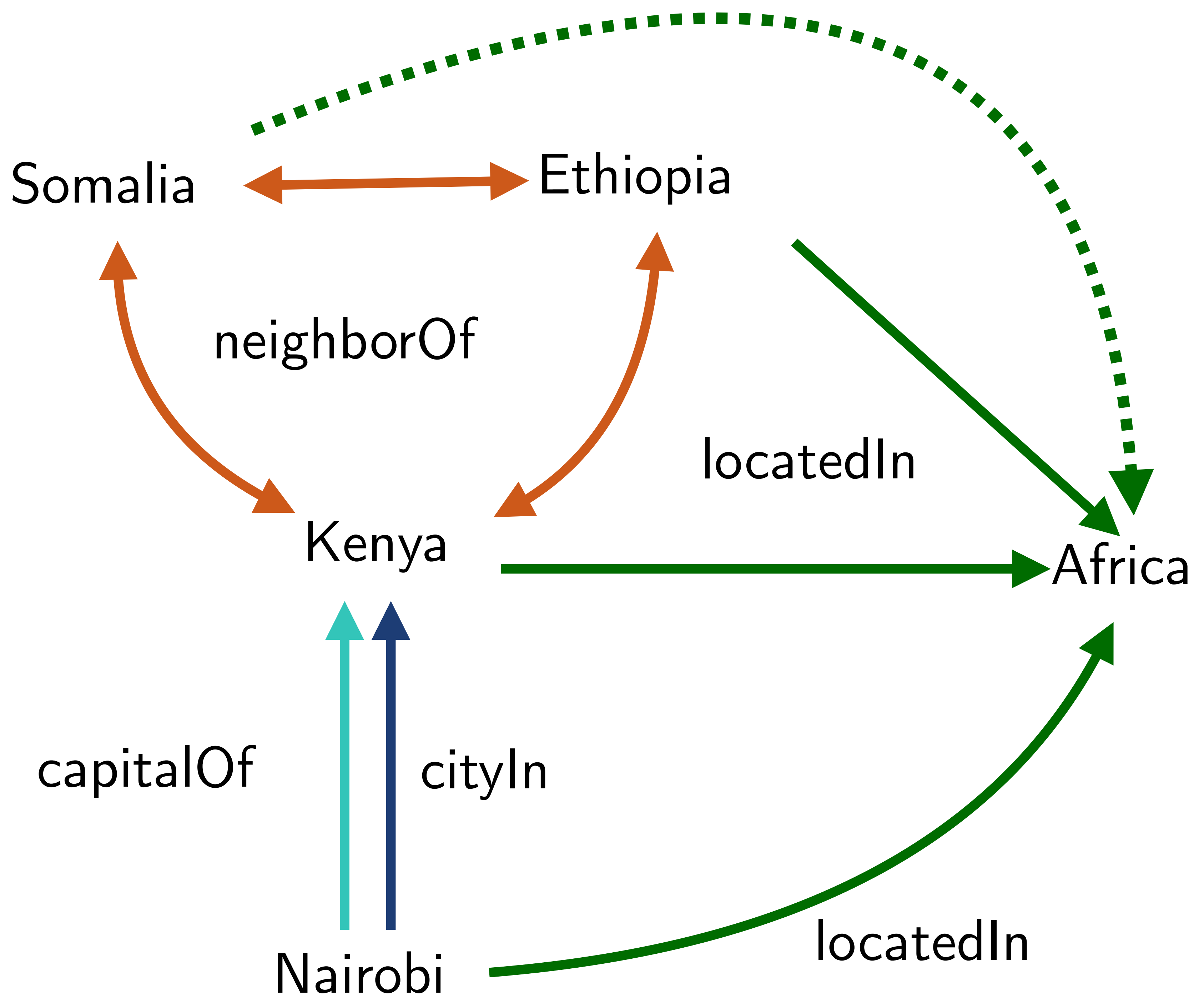


Knowledge Graph Completion

Problem: KGs are typically highly incomplete, which makes their downstream use more challenging. For example, 71% of individuals in Freebase lack a connection to a place of birth.

Question: Can we automatically find new facts for our KG, solely based on the existing information in the KG?

Task: Given a KG G, the task of knowledge graph completion is to predict facts that are missing from G.



Inspiration from Word Vector Representations

"The word representations computed using NNs are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns.

Somewhat surprisingly, many of these patterns can be represented as linear translations...

vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vector."

(Mikolov et. al, 2013)

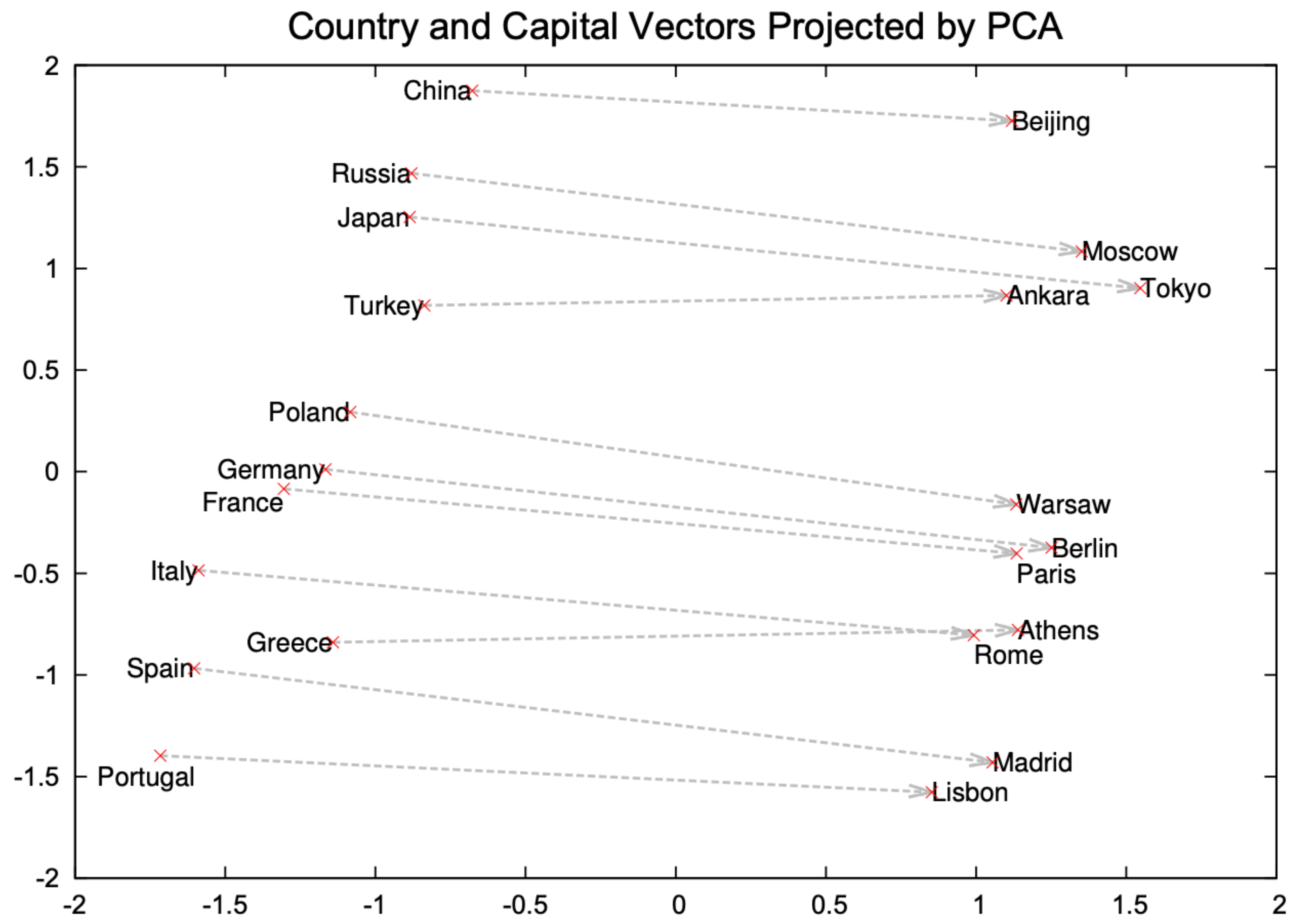


Figure 2 (Mikolov et. al, 2013): 2-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training no supervised information about what a capital city means is given.

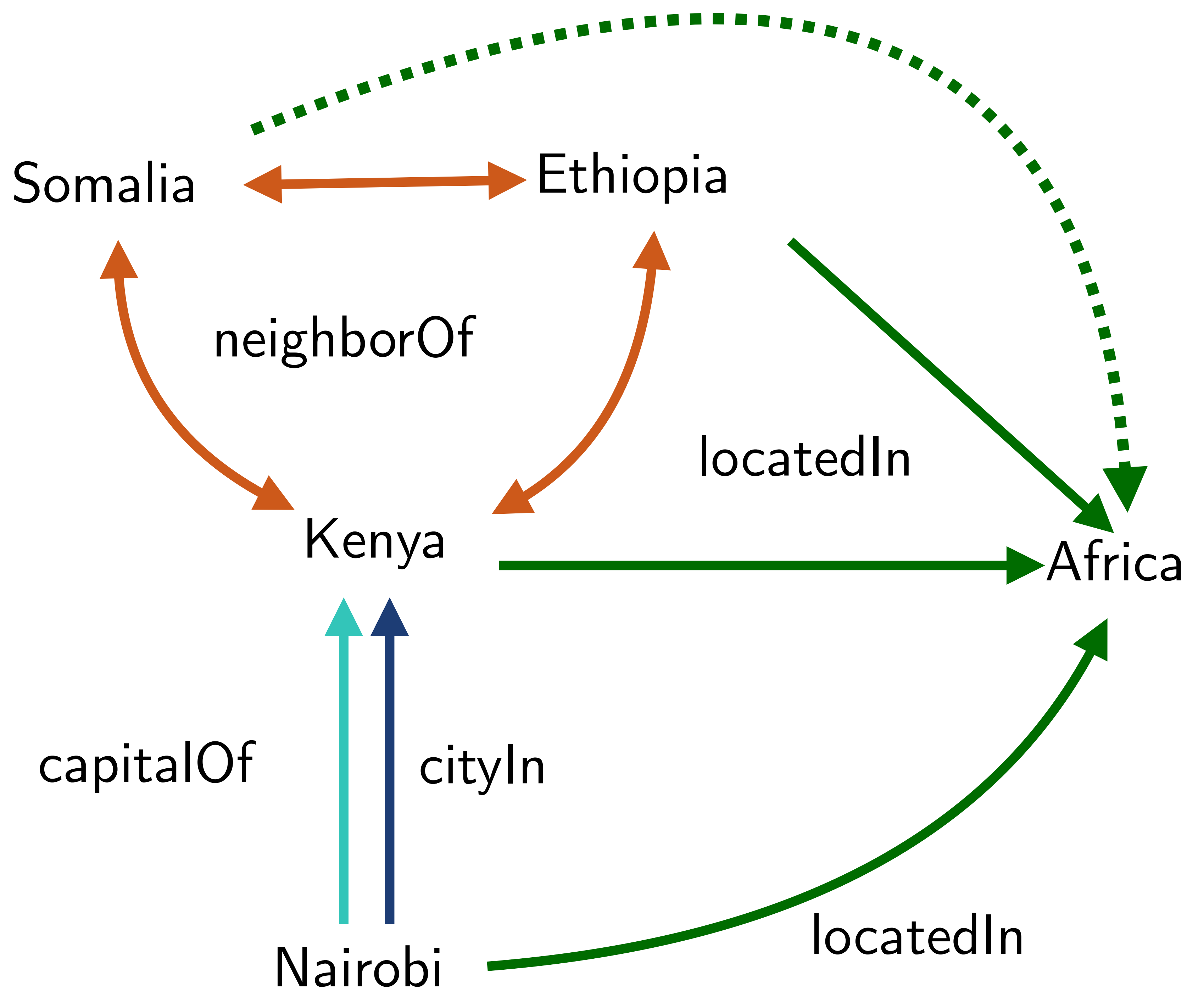
Knowledge Graph Completion

Task: Predict new facts for our KG, solely based on the existing information in the KG?

Intuition: Real-world data lies in low dimensional manifolds, so if existing facts exhibit patterns then one can embed them into low-dimensional spaces and use to predict new facts.

Encoder: Represent entities and relations as embeddings, while capturing latent properties of the KG: similar entities and relationships represented with similar embeddings.

Decoder: Score the facts using the learned similarities and rank the predictions.



Knowledge Graph Embedding Models

KG Embedding Models

KG Embedding Models

Most of the existing approaches can be described in term of the following criteria:

(i) **Model representation (Encoder)**: How are the entities and relations represented?

(ii) **Scoring function (Decoder)**: How is the likelihood of a fact to be true defined?

...and an appropriate loss function to minimize the objective function.

KG Embedding Models

Most of the existing approaches can be described in term of the following criteria:

(i) **Model representation (Encoder)**: How are the entities and relations represented?

(ii) **Scoring function (Decoder)**: How is the likelihood of a fact to be true defined?

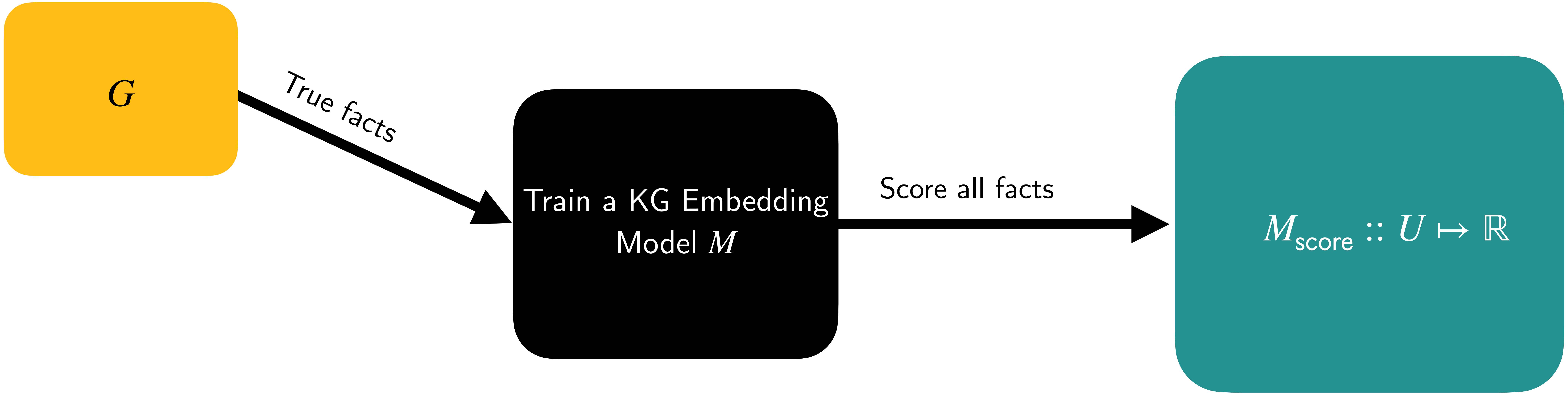
...and an appropriate loss function to minimize the objective function.

Well-known families of models classified in terms of model representation:

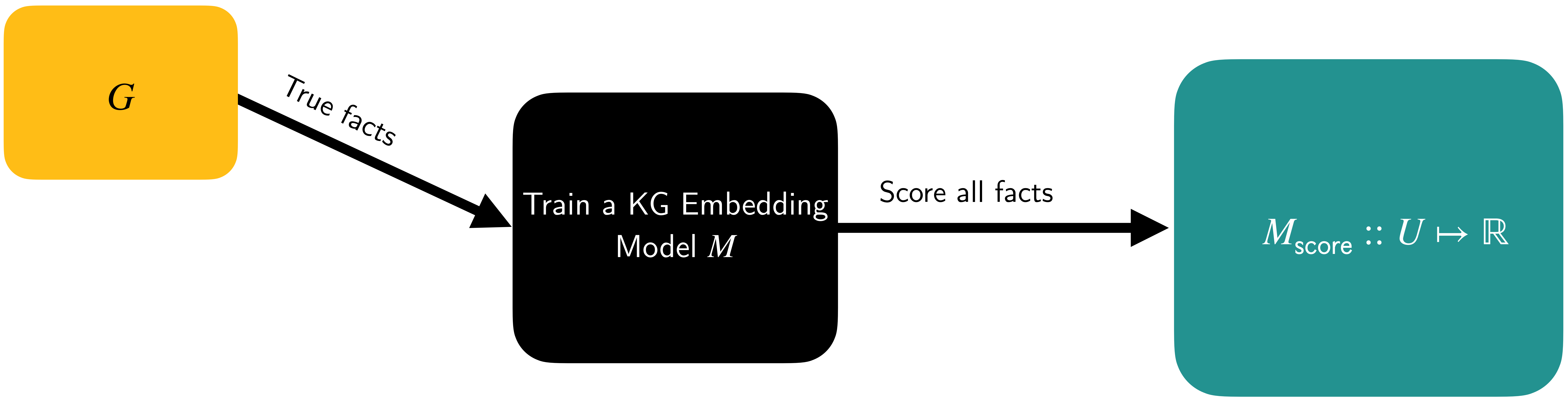
- **Translational**: Entities as points in the space, relations as translations operating on entity embeddings.
- **Bilinear**: Entities as points in the vector space, and relations as a bilinear map between entity embeddings.
- **Neural**: Entities and relations embedded using a neural network (e.g., convolutional neural network).

KG Embedding Models: Basic Idea

KG Embedding Models: Basic Idea

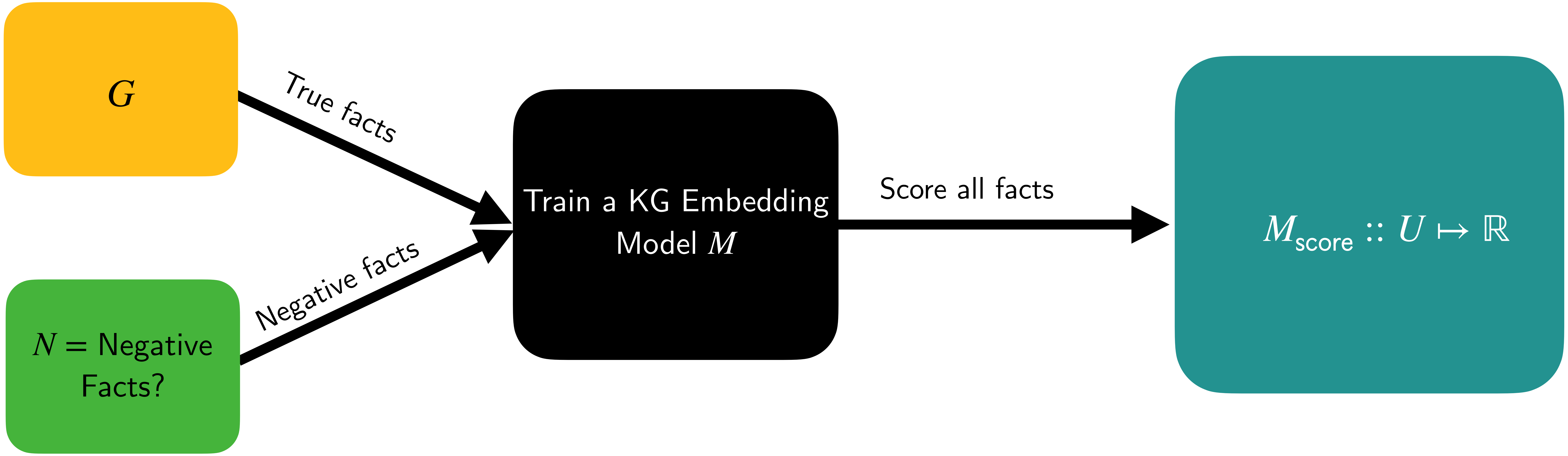


KG Embedding Models: Basic Idea



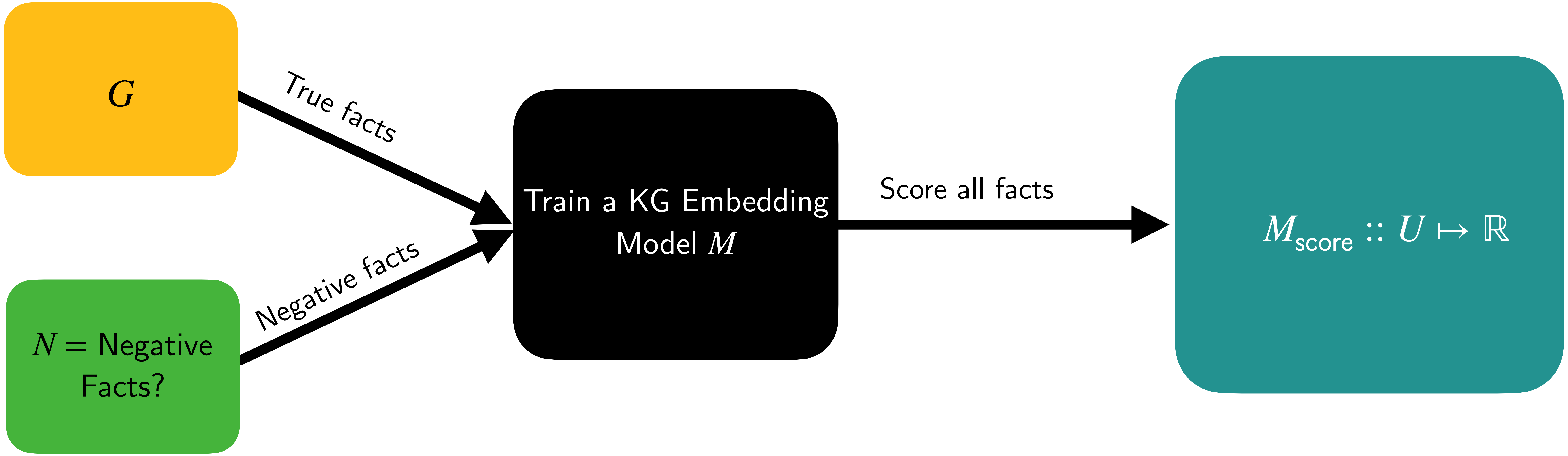
Optimization goal: Find a representation that scores/ranks "true facts" higher than "false facts" in accordance to a dissimilarity measure.

KG Embedding Models: Basic Idea



Optimization goal: Find a representation that scores/ranks "true facts" higher than "false facts" in accordance to a dissimilarity measure.

KG Embedding Models: Basic Idea



Optimization goal: Find a representation that scores/ranks "true facts" higher than "false facts" in accordance to a dissimilarity measure.

Problem: KGs typically store only positive information, and so encode only the facts that are true. There are no real negative examples to train with!

Negative Sampling

Negative Sampling

Idea: Corrupt true facts (i.e., facts from the KG) and use some of these as negative examples and a corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G.

Negative Sampling

Idea: Corrupt true facts (i.e., facts from the KG) and use some of these as negative examples and a corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G.

For a true fact $r(h, t) \in G$, we define the set of all corrupted facts as:

$$C^{r(h,t)} = \{r(e, t) \mid e \neq h \in \mathbf{E}, r(e, t) \notin G\} \cup \{r(h, e) \mid e \neq t \in \mathbf{E}, r(h, e) \notin G\} .$$

A negative fact for a given true fact $r(h, t)$, is a fact randomly sampled from $C^{r(h,t)}$.

The set of negative facts sampled for a given true fact $r(h, t)$ is $N^{r(h,t)}$.

Various negative sampling techniques are used, e.g., uniform sampling, adversarial sampling, etc.

Negative Sampling

Idea: Corrupt true facts (i.e., facts from the KG) and use some of these as negative examples and a corrupted fact is obtained by replacing only the head (resp., only the tail) entity in a true fact in G.

For a true fact $r(h,t) \in G$, we define the set of all corrupted facts as:

$$C^{r(h,t)} = \{r(e,t) \mid e \neq h \in \mathbf{E}, r(e,t) \notin G\} \cup \{r(h,e) \mid e \neq t \in \mathbf{E}, r(h,e) \notin G\} .$$

A negative fact for a given true fact $r(h,t)$, is a fact randomly sampled from $C^{r(h,t)}$.

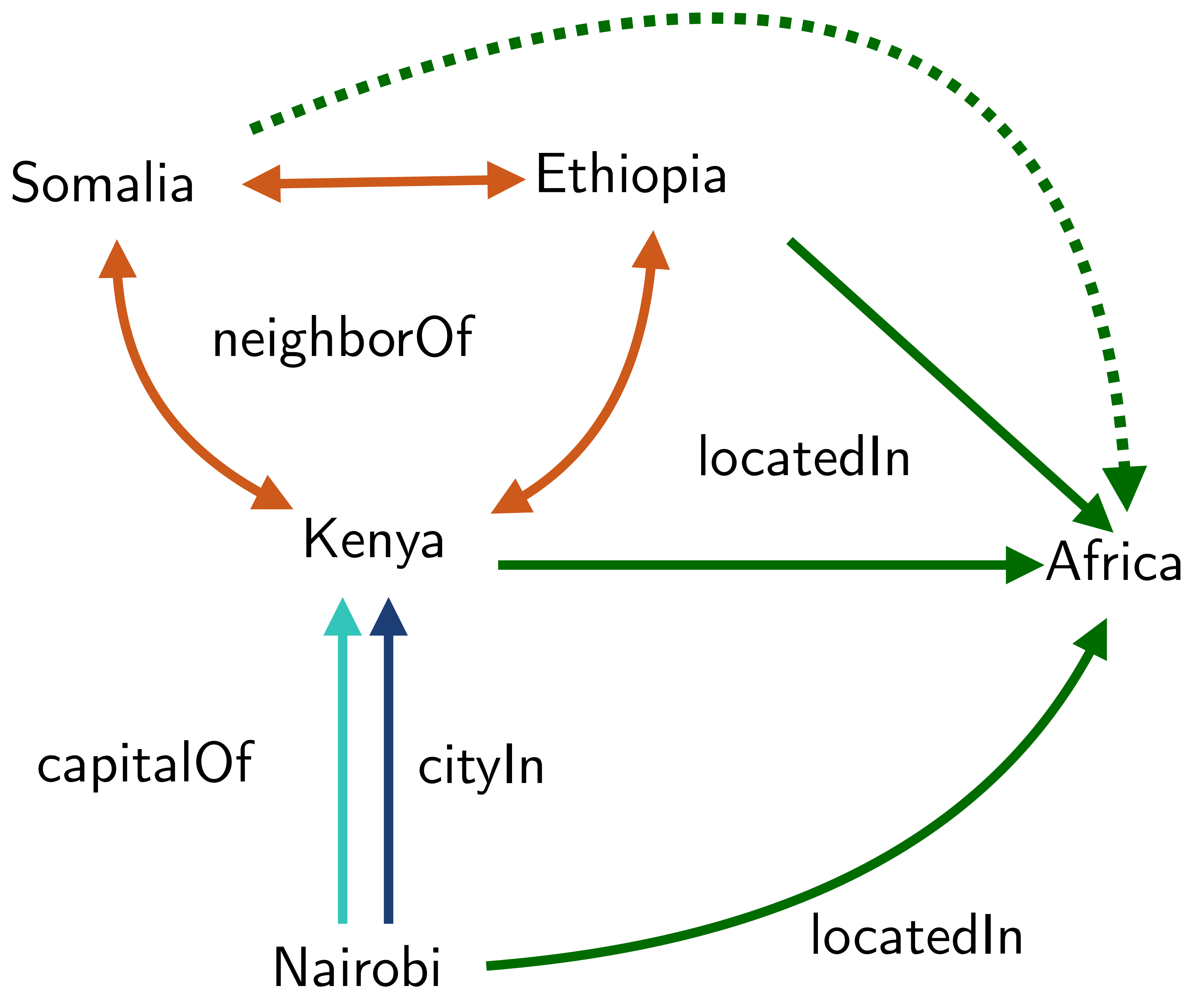
The set of negative facts sampled for a given true fact $r(h,t)$ is $N^{r(h,t)}$.

Various negative sampling techniques are used, e.g., uniform sampling, adversarial sampling, etc.

Remark: Negative sampling is not ideal, as random sampling can give a potentially correct fact as a negative fact, and require it to be ranked lower, misleadingly.

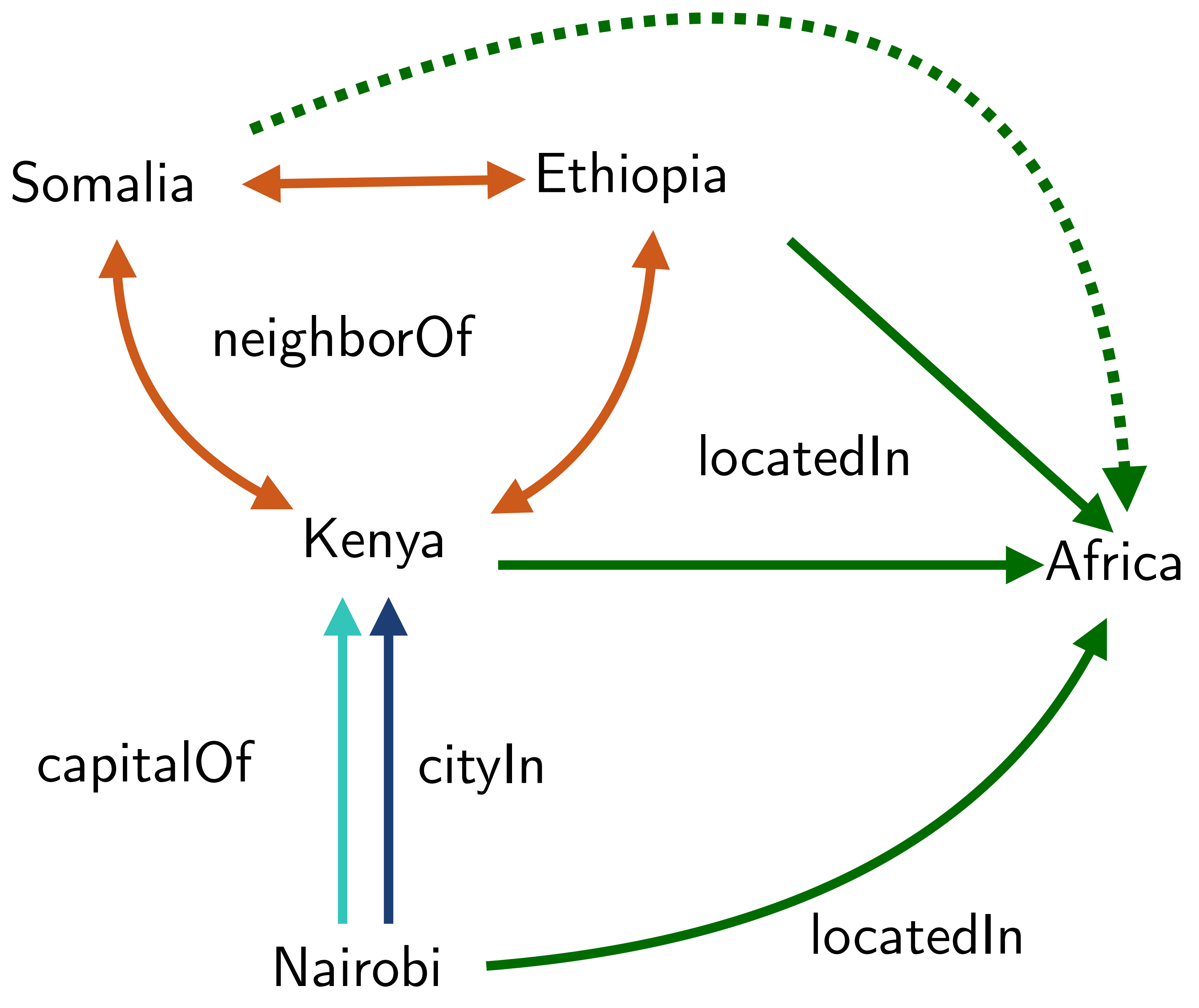
Model Expressiveness

Model Expressiveness



Model Expressiveness

A KGC model M is fully expressive if, for any given disjoint sets of true and false facts over a vocabulary (i.e., the ground truth of a set of facts), there exists a parameter configuration for M such that M accurately classifies all the given facts.

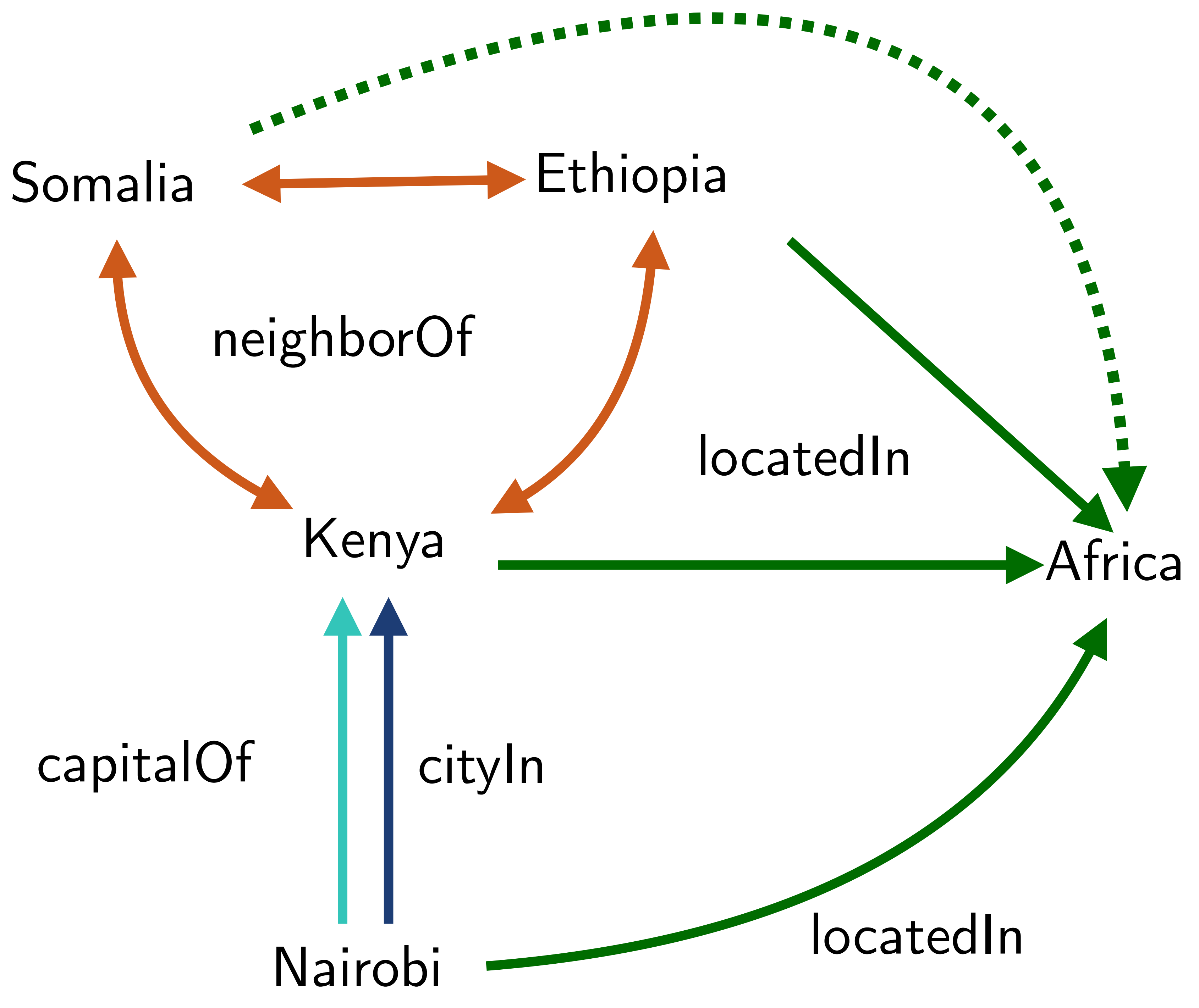


Model Expressiveness

Fully expressive models can capture any ground truth of a set of facts whereas inexpressive models can underfit.

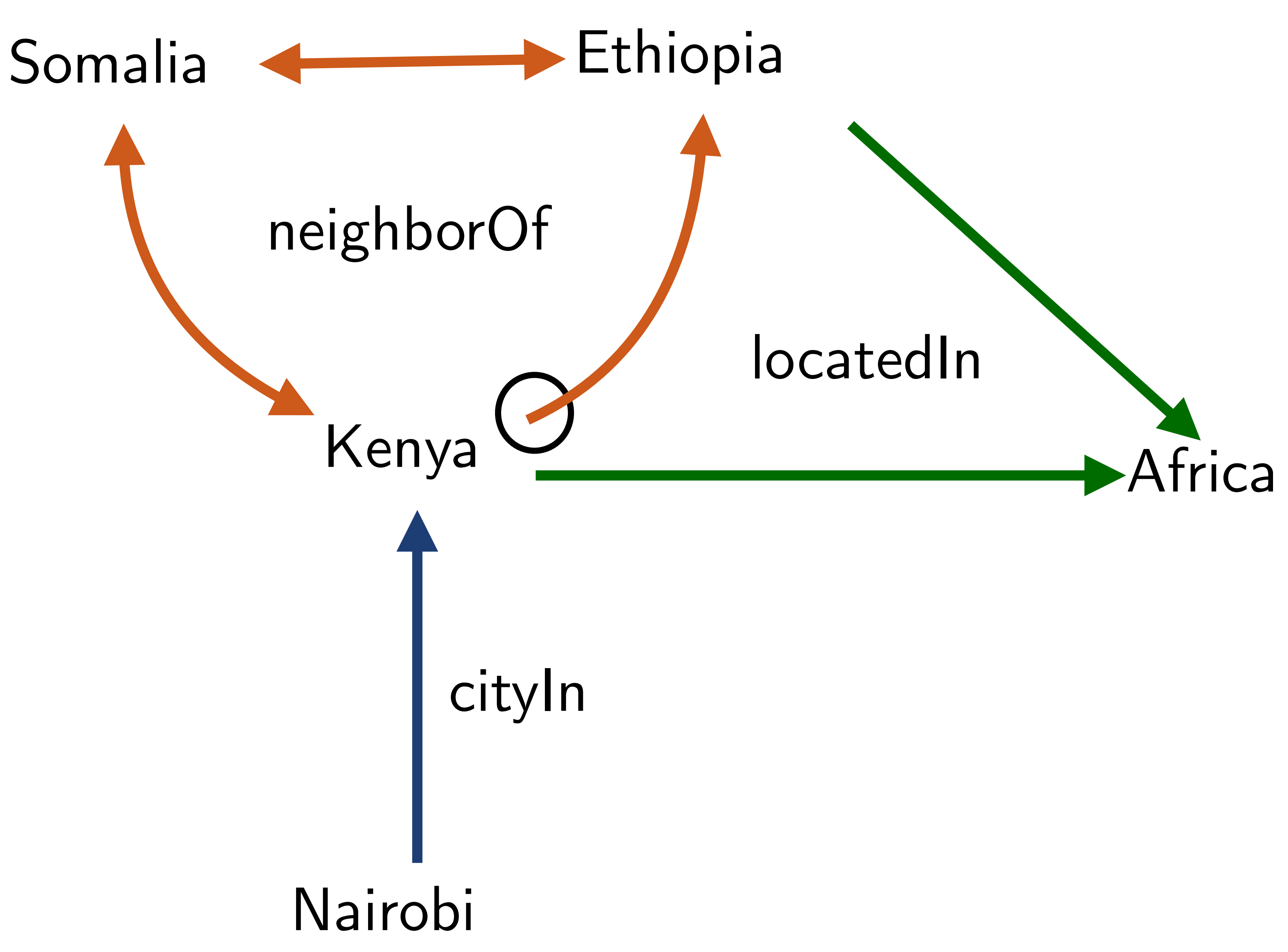
Theoretical inexpressivity of a model may not surface empirically, especially if the benchmark datasets are not very complex.

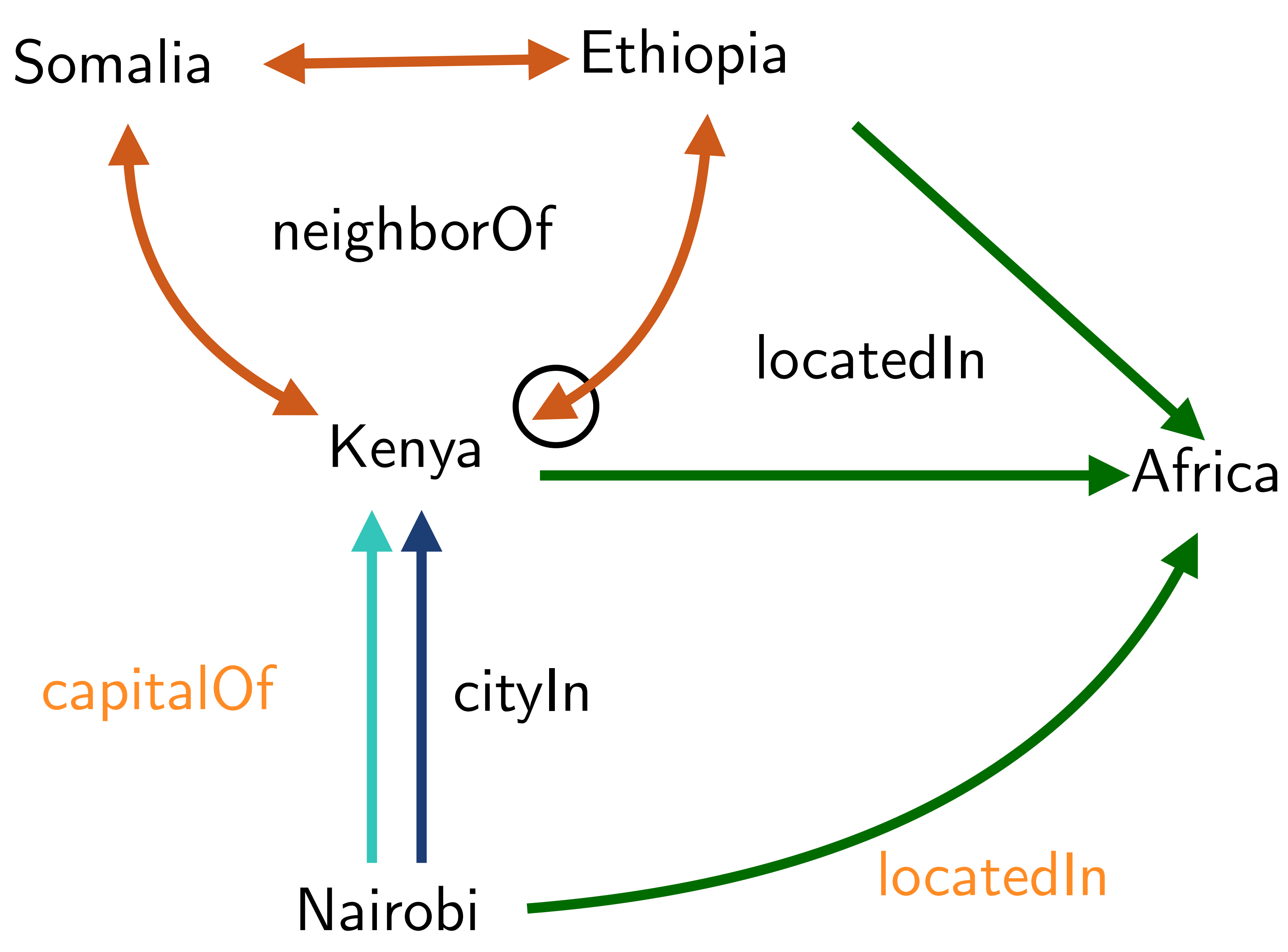
Knowing the expressive limitations of a model, however, it is easy to design datasets to empirically observe its limitations.



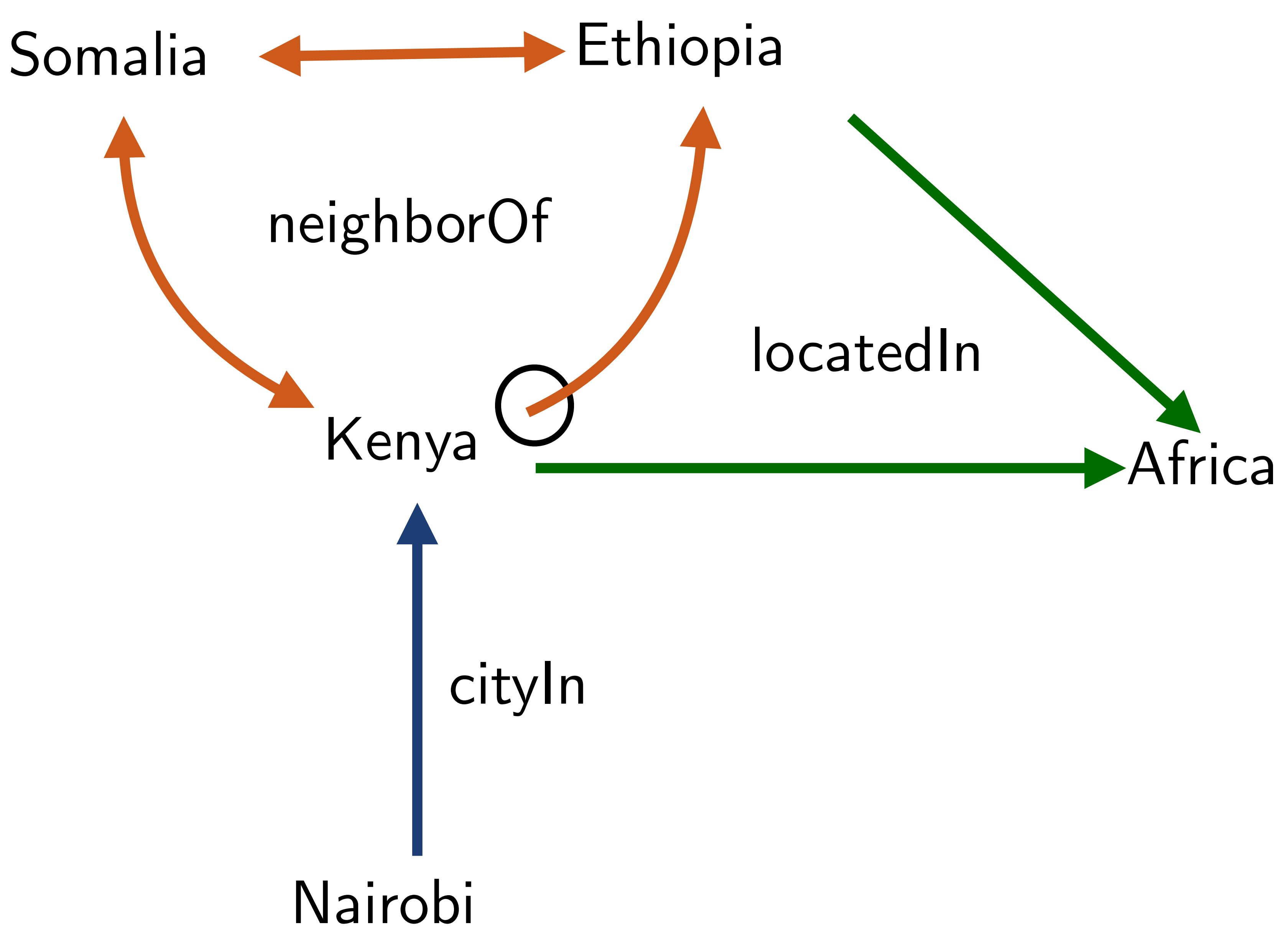
Model Inductive Capacity

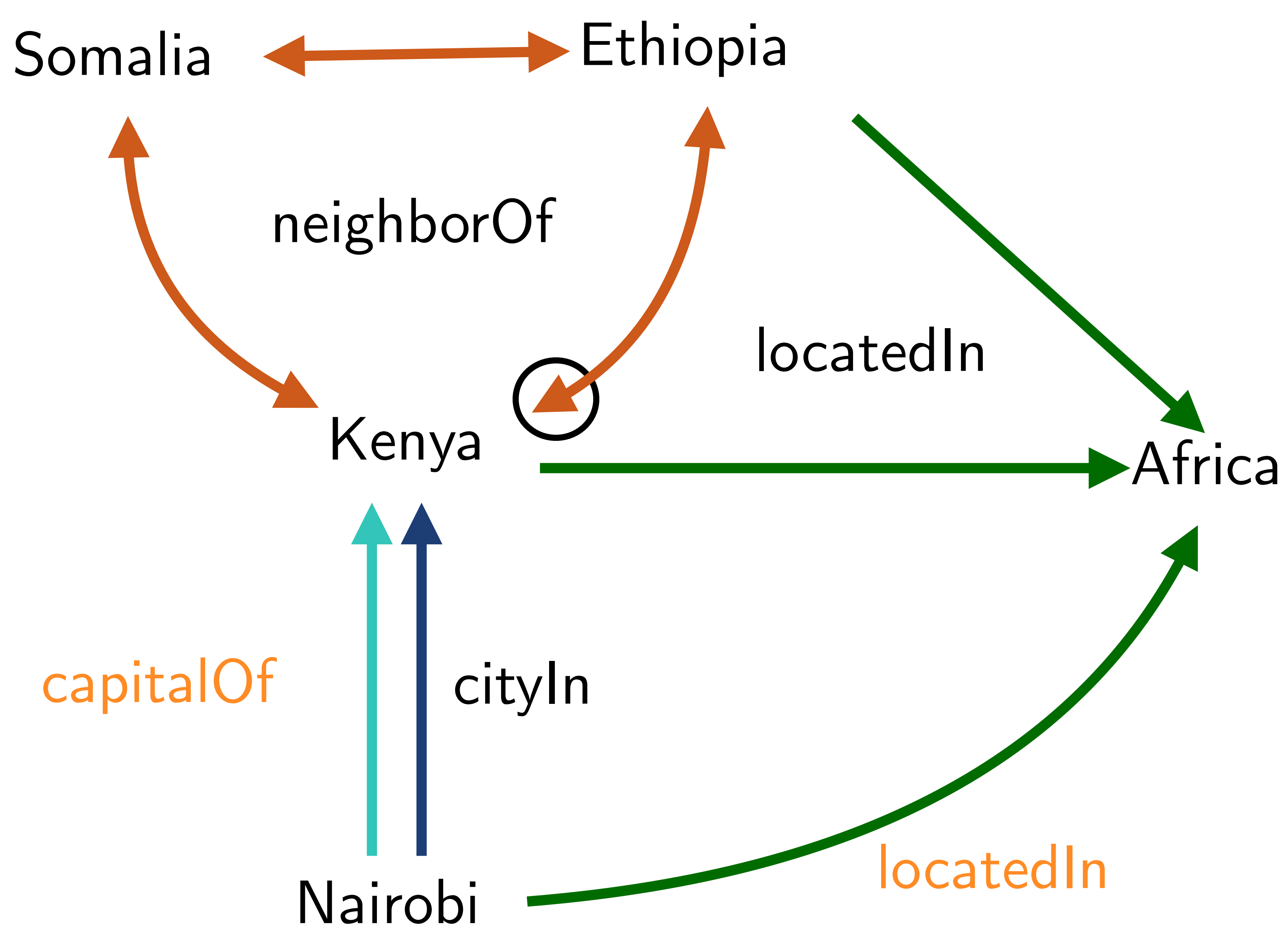
Model Inductive Capacity





Model Inductive Capacity

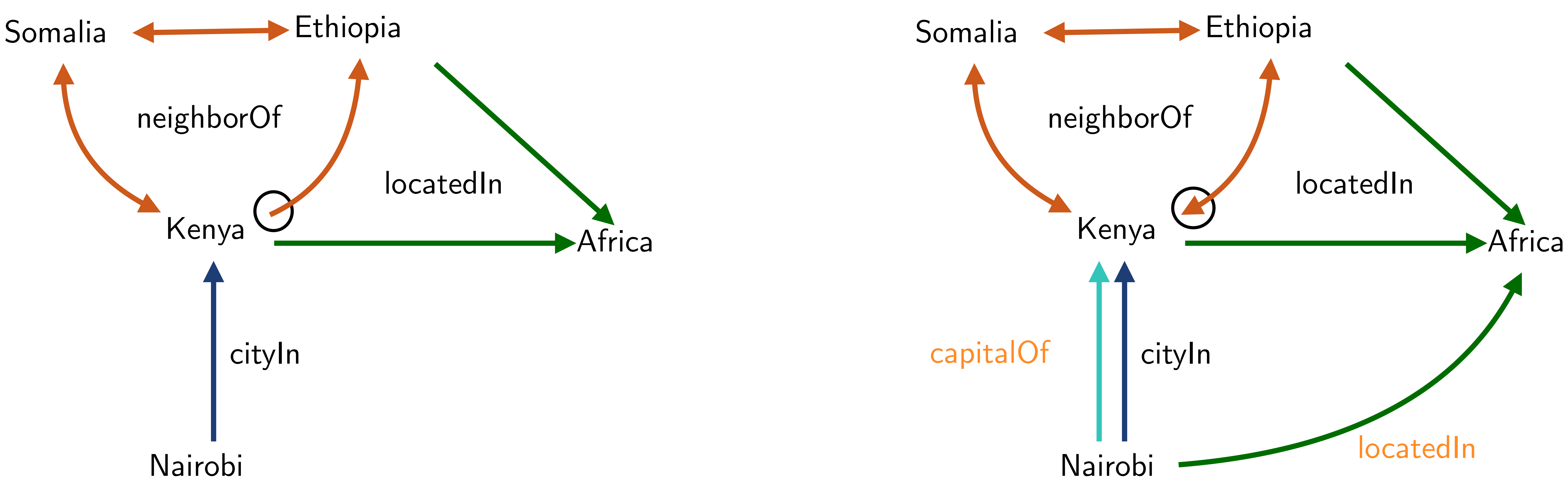




Model inductive capacity is the generalization capacity of a model, i.e., the quality of the predictions of the model over incomplete datasets.

Full expressiveness does not necessarily correlate with inductive capacity: Fully expressive models can merely memorize training data and generalize poorly.

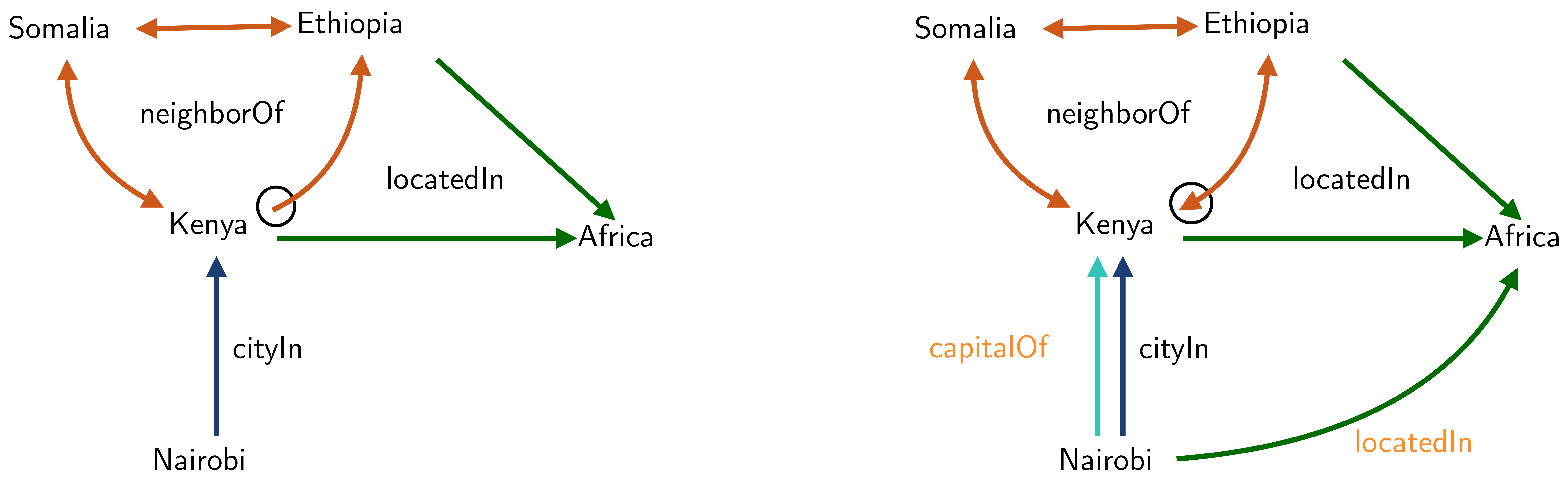
Model Inductive Capacity



How can model inductive capacity be studied?

Inference patterns are specifications of logical properties that may exist in a KG, which, if learned, enable further principled inferences from existing KG facts.

Model Inductive Capacity



Example: A relation $r \in R$ is symmetric if, for any choice of entities $e_1, e_2 \in E$, whenever a fact $r(e_1, e_2)$ holds, then so does $r(e_2, e_1)$.

If a model learns a symmetry pattern for a relation r, then it can infer facts in the symmetric closure of r, thus providing a strong inductive bias.

Inference Patterns

Inference Patterns

An inference pattern specifies a logical property over a KG

Consider an extended relational vocabulary over E and R with a set V of variables.

A first-order atom is an expression of the form $r(x_i, x_j)$, where $r \in R$, and $x_i, x_j \in V$.

Inference Patterns

An inference pattern specifies a logical property over a KG

Consider an extended relational vocabulary over E and R with a set V of variables.

A first-order atom is an expression of the form $r(x_i, x_j)$, where $r \in R$, and $x_i, x_j \in V$.

A Boolean combination of first-order atoms is defined inductively using logical constructors $\neg, \wedge, \vee$:

$$\phi_1(x_1, x_3) = r_1(x_1, x_2) \wedge r_2(x_2, x_2) \qquad \phi_2(x_3, x_4) = r_2(x_3, x_4) \vee \neg r_3(x_4, x_3)$$

Inference Patterns

An inference pattern specifies a logical property over a KG

Consider an extended relational vocabulary over E and R with a set V of variables.

A first-order atom is an expression of the form $r(x_i, x_j)$, where $r \in R$, and $x_i, x_j \in V$.

A Boolean combination of first-order atoms is defined inductively using logical constructors $\neg, \wedge, \vee$:

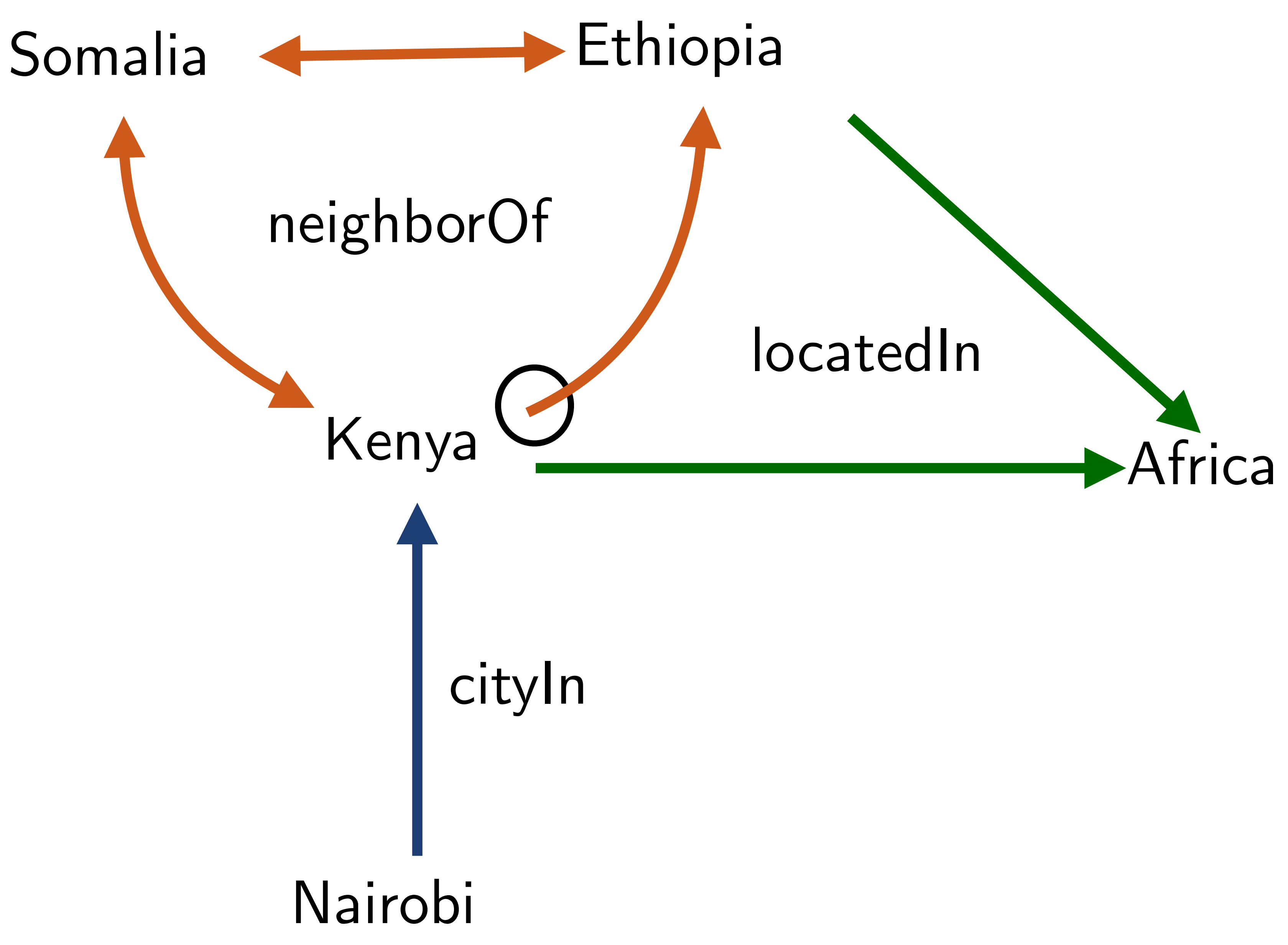
$$\phi_1(x_1, x_3) = r_1(x_1, x_2) \wedge r_2(x_2, x_2) \qquad \phi_2(x_3, x_4) = r_2(x_3, x_4) \vee \neg r_3(x_4, x_3)$$

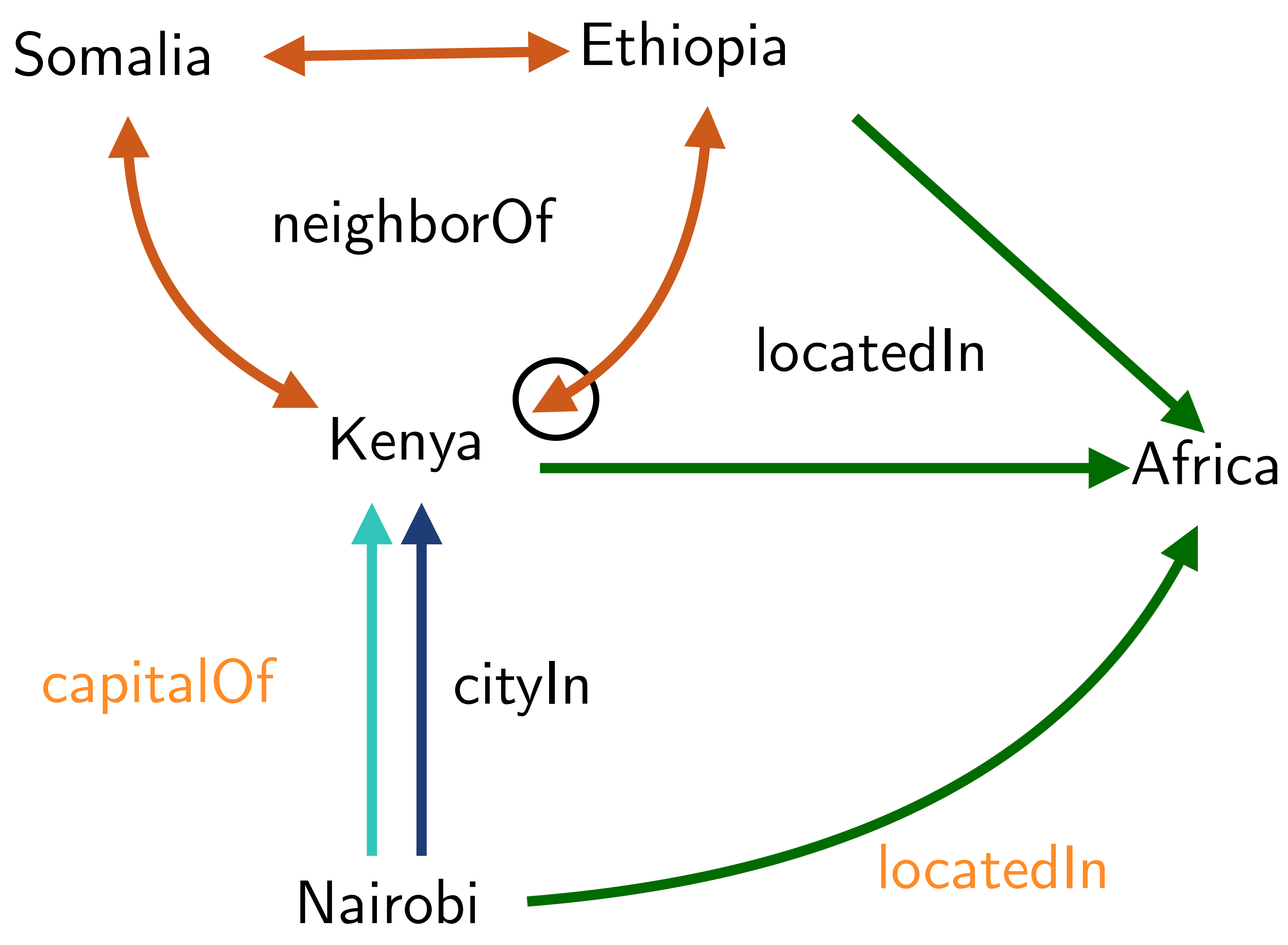
We are interested in universally quantified first-order rules of the form:

$$\forall x_1 \ldots x_k \; \phi(x_1 \ldots x_k) \Rightarrow \psi(x_1 \ldots x_l),$$

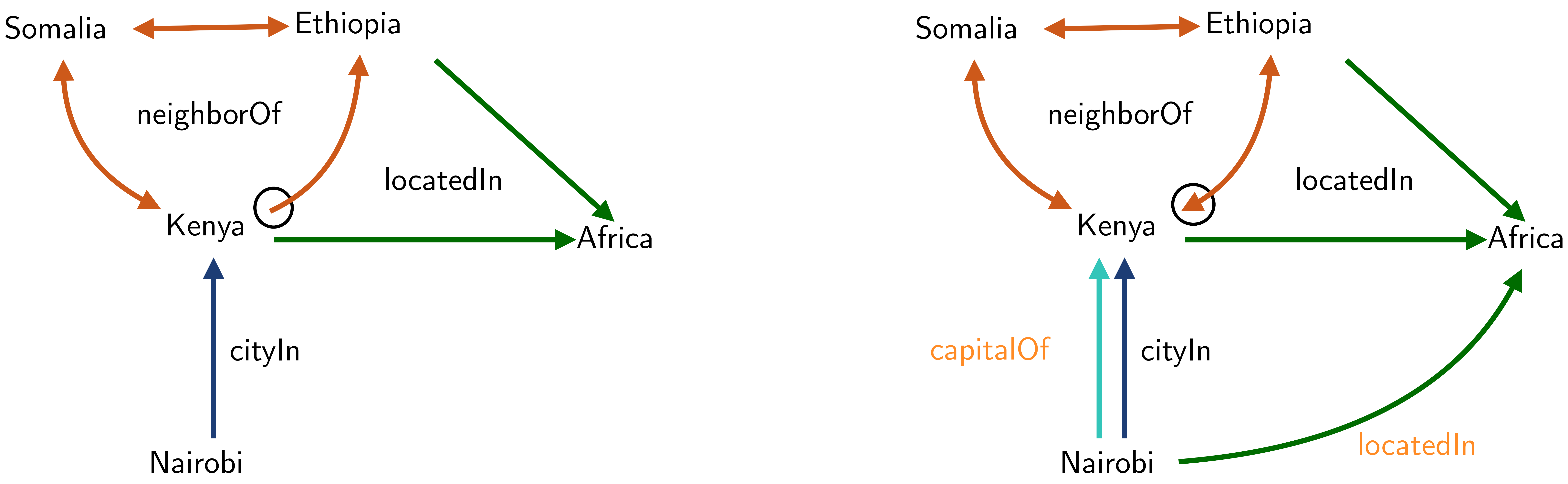
with $k \geq l$. The semantics of such rules is that of first-order logic, restricted to a finite domain.

Inference Patterns





Inference Patterns

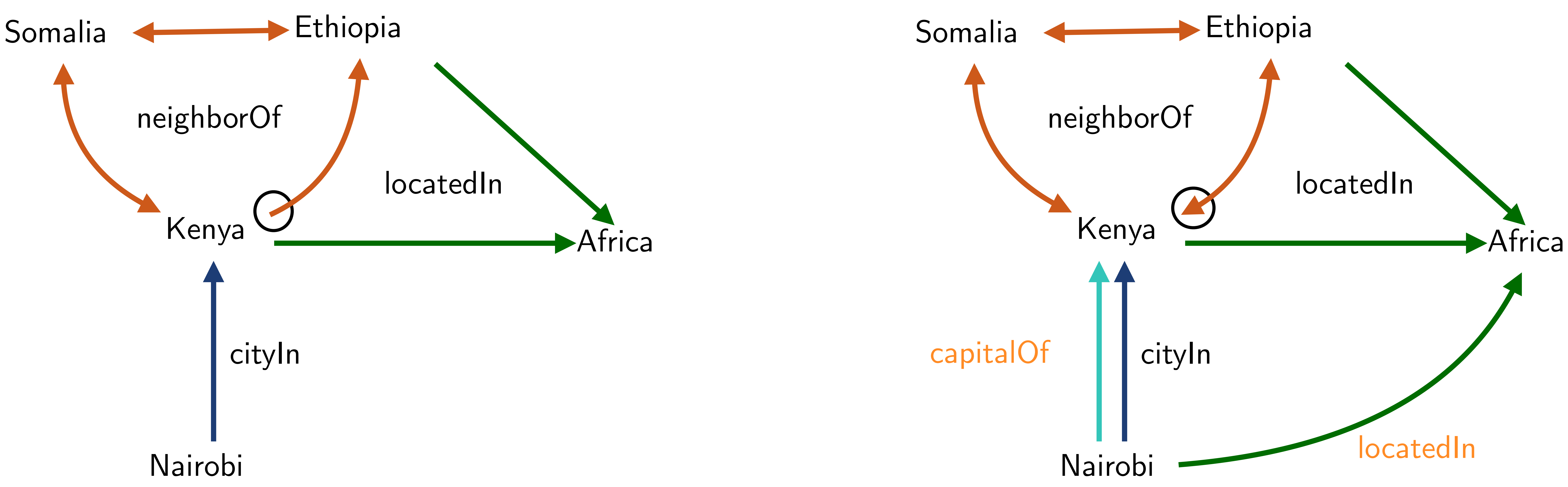


We can express the symmetry inference pattern for a relation $r \in R$:

$$\forall x, y \; r(x, y) \Rightarrow r(y, x),$$

which holds if and only if the relation r is symmetric, e.g., neighborOf relation should be symmetric.

Inference Patterns



Similarly, we can express that the relations $r_1, r_2 \in R$ are the inverse of each other in terms of two rules:

$$\forall x, y \; r_1(x, y) \Rightarrow r_2(y, x) \text{ and } \forall x, y \; r_2(x, y) \Rightarrow r_1(y, x).$$

...and abbreviate as $\forall x, y \; r_1(x, y) \Leftrightarrow r_2(y, x)$.

Inference Patterns

Inference pattern	Inference rule
Symmetry	$\forall x, y \; r(x, y) \Rightarrow r(y, x)$
Anti-symmetry	$\forall x, y \; r(x, y) \Rightarrow \neg r(y, x)$
Inversion	$\forall x, y \; r_1(x, y) \Leftrightarrow r_2(y, x)$
Composition	$\forall x, y, z \; r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z)$
Hierarchy	$\forall x, y \; r_1(x, y) \Rightarrow r_2(x, y)$
Intersection	$\forall x, y \; r_1(x, y) \wedge r_2(x, y) \Rightarrow r_3(x, y)$
Mutual exclusion	$\forall x, y \; r_1(x, y) \Rightarrow \neg r_2(x, y)$

List of inference patterns commonly used in the literature and the corresponding logical rules. It is assumed that $r_1 \neq r_2 \neq r_3$.

These patterns are prominent in datasets. While these patterns and the corresponding rules are not very expressive, they already are a challenge for KGE models, as it is already hard for existing systems to capture these patterns.

Empirical Evaluation

Empirical Evaluation: Ranking

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*.

The KG G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

For a test fact $r(h,t) \in G_{test}$, we define:

$$\begin{aligned} r(\_,t) = &\ \{r(e,t) \mid e \in \mathbf{E}, r(e,t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}, \\ r(h,\_) = &\ \{r(h,e) \mid e \in \mathbf{E}, r(h,e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}\,. \end{aligned}$$

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*.

The KG G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

For a test fact $r(h,t) \in G_{test}$, we define:

$$\begin{aligned} r(\_,t) = & \ \{r(e,t) \mid e \in \mathbf{E}, r(e,t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}, \\ r(h,\_) = & \ \{r(h,e) \mid e \in \mathbf{E}, r(h,e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}\,. \end{aligned}$$

Remark: All facts from the training, validation, or test data are filtered out from these sets (except the test fact itself) to ensure that facts known to be true do not affect the ranking (Bordes et al., 2013).

Empirical Evaluation: Ranking

The most common empirical evaluation task for KGE methods is based on entity *ranking*.

The KG G is partitioned into a set of training (G_{tr}), validation (G_v), and test facts (G_{test}).

For a test fact $r(h,t) \in G_{test}$, we define:

$$
\begin{aligned}
r(\_,t) = & \ \{r(e,t) \mid e \in \mathbf{E}, r(e,t) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}, \\
r(h,\_) = & \ \{r(h,e) \mid e \in \mathbf{E}, r(h,e) \notin G_{tr} \cup G_v \cup G_{test}\} \cup \{r(h,t)\}\,.
\end{aligned}
$$

Remark: All facts from the training, validation, or test data are filtered out from these sets (except the test fact itself) to ensure that facts known to be true do not affect the ranking (Bordes et al., 2013).

Every fact in these sets is ranked in accordance to a scoring function of the model in descending order.

The rank of e relative to the facts in $r(\_,t)$, denoted $rank(e \mid r(\_,t))$, is the rank of $r(e,t)$ in $r(\_,t)$.

The rank of e relative to the facts in $r(h,\_)$, denoted $rank(e \mid r(h,\_))$, is the rank of $r(h,e)$ in $r(h,\_)$.

Empirical Evaluation: Metrics

Empirical Evaluation: Metrics

Mean rank (MR) is the average rank of true facts against their corrupted counterparts:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \big(rank(h \mid r(\_, t)) + rank(t \mid r(h, \_))\big)$$

Empirical Evaluation: Metrics

Mean rank (MR) is the average rank of true facts against their corrupted counterparts:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \left(rank(h \mid r(\_, t)) + rank(t \mid r(h, \_)) \right)$$

Mean reciprocal rank (MRR) is the inverse average rank of true facts against their corrupted counterparts:

$$\frac{1}{2 \mid G_{test} \mid} \sum_{r(h,t) \in G_{test}} \left(\frac{1}{rank(h \mid r(\_, t))} + \frac{1}{rank(t \mid r(h, \_))} \right)$$

Empirical Evaluation: Metrics

Mean rank (MR) is the average rank of true facts against their corrupted counterparts:

$$\frac{1}{2\mid G_{test}\mid}\sum_{r(h,t)\in G_{test}}\big(rank(h\mid r(\_,t))+rank(t\mid r(h,\_))\big)$$

Mean reciprocal rank (MRR) is the inverse average rank of true facts against their corrupted counterparts:

$$\frac{1}{2\mid G_{test}\mid}\sum_{r(h,t)\in G_{test}}\left(\frac{1}{rank(h\mid r(\_,t))}+\frac{1}{rank(t\mid r(h,\_))}\right)$$

Hits@k is the proportion of true facts with rank at most k:

$$\frac{1}{2\mid G_{test}\mid}\sum_{r(h,t)\in G_{test}}\Big(\mathbf{1}(rank(h\mid r(\_,t))\leq k)+\mathbf{1}(rank(t\mid r(h,\_))\leq k)\Big),$$

where $\mathbf{1}(c)$ is the indicator function that returns 1, if c is true, and 0, otherwise.

Empirical Evaluation: Datasets

FB15k (Bordes et al., 2013): A subset of Freebase (Bollacker et al., 2008), where a large part of the test facts $r(x, y)$ can be directly inferred via an inverse relation $r'(y, x)$, which makes the inversion very prominent (Toutanova & Chen, 2015). Other patterns on FB15k are symmetry/antisymmetry and composition patterns.

FB15K-237 (Toutanova & Chen, 2015): A subset of FB15k , where inverse relations are deleted. The prominent patterns are composition and symmetry/antisymmetry patterns.

WN18 (Bordes et al., 2013): A subset of WordNet (Miller, 1995), featuring lexical relations between words. It contains many inverse relations, and the main inference patterns are symmetry/antisymmetry and inversion.

WN18RR (Dettmers et al., 2017): A subset of WN18, where inverse relations are deleted. The prominent inference patterns are symmetry/antisymmetry and composition.

YAGO3-10: A subset of the YAGO3 (Mahdisoltani et al., 2015), where all entities appear in at least 10 facts.

Empirical Evaluation: Datasets

Dataset	|E|	|R|	Training facts	Validation facts	Test facts
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,034
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Empirical Evaluation: Datasets

Dataset	|E|	|R|	Training facts	Validation facts	Test facts
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,034
YAGO3-10	123,182	37	1,079,040	5,000	5,000

Datasets with their respective #entities ($|\mathbf{E}|$), #relations ($|\mathbf{R}|$), and #facts.

Summary

- Relational data is prominent in real-world applications!
- Machine learning on graph: encoder-decoder framework
- Shallow KG embedding models through the lens of the KG completion task
- The families of translational, bilinear, and neural models
- Established evaluation criteria for different models:
  - Model expressiveness
  - Model inductive capacity and inference patterns
  - Empirical evaluation: Datasets and metrics
- We have not introduced or discussed any specific model: **Lecture 2**!

References

- Ganapathiraju, M. K., Thahir, M., Handen, A., Sarkar, S. N., Sweet, R. A., Nimgaonkar, V. L., Loscher, C. E., Bauer, E. M., & Chaparala, S. (2016). Schizophrenia interactome with 504 novel protein-protein interactions. *NPJ schizophrenia*, *2*, 16012.
- Rao, P.P., Kabir, S.N., & Mohamed, T.S. (2010). Nonsteroidal Anti-Inflammatory Drugs (NSAIDs): Progress in Small Molecule Drug Development. *Pharmaceuticals, 3*, 1530 - 1549.
- J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, L. Fei-Fei. Image Retrieval using Scene Graphs, *CVPR*, 2015.
- K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. *MOD*, 2008.
- Allamanis, Miltiadis, Graph Neural Networks on Program Analysis. 2021.
- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *NIPS*, 2013.
- M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond Euclidean data, *IEEE Signal Processing Magazine,* 2017.

References

- A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *NIPS*, 2013.
- G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.
- K. Toutanova and D. Chen. Observed versus latent features for knowledge base and text inference. Proc. of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015.
- F. Mahdisoltani, J. Biega, and F. Suchanek. YAGO3: A Knowledge Base from Multilingual Wikipedias. *CIDR,* 2015.
- T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel. Convolutional 2D knowledge graph embeddings. *AAAI*, 2018.