# Lecture 8: Applications of Graph Neural Networks

## Relational Learning

İsmail İlkan Ceylan

Advanced Topics in Machine Learning, University of Oxford

20.02.2021

## Overview

## Overview

- Life science applications
    - Biomedical data and drug discovery
    - Particle physics

## Overview

- Life science applications
  - Biomedical data and drug discovery
  - Particle physics
- Combinatorial optimisation & reasoning

## Overview

- Life science applications
  - Biomedical data and drug discovery
  - Particle physics
- Combinatorial optimisation & reasoning
- Computer vision: Scene graphs and question answering

## Overview

- Life science applications
  - Biomedical data and drug discovery
  - Particle physics
- Combinatorial optimisation & reasoning
- Computer vision: Scene graphs and question answering
- Knowledge graphs: Link prediction and beyond

## Overview

- Life science applications
  - Biomedical data and drug discovery
  - Particle physics
- Combinatorial optimisation & reasoning
- Computer vision: Scene graphs and question answering
- Knowledge graphs: Link prediction and beyond
- Summary of the lecture

## Overview

- Life science applications
  - Biomedical data and drug discovery
  - Particle physics
- Combinatorial optimisation & reasoning
- Computer vision: Scene graphs and question answering
- Knowledge graphs: Link prediction and beyond
- Summary of the lecture
- Summary of the relational learning theme

## Life Science Applications

## Biomedical Domain and Drug Discovery

## Biomedical Domain and Drug Discovery

- Biomedical data is inherently relational — typically represented as a graph

# Biomedical Domain and Drug Discovery

- Biomedical data is inherently relational — typically represented as a graph
    - **Molecular scale**: Proteins and other biomolecules can be represented as graphs capturing spatial and structural relationships between their amino acid residues. Small molecule drugs can similarly be represented as graphs relating their constituent atoms and chemical bonding structure.

# Biomedical Domain and Drug Discovery

- Biomedical data is inherently relational — typically represented as a graph
  - **Molecular scale**: Proteins and other biomolecules can be represented as graphs capturing spatial and structural relationships between their amino acid residues. Small molecule drugs can similarly be represented as graphs relating their constituent atoms and chemical bonding structure.
  - **Intermediary scale**: An interactome is a set of molecular interactions in a particular cell — They can be represented as graphs that capture specific types of interactions between biomolecular species, e.g., protein–protein interaction graphs.

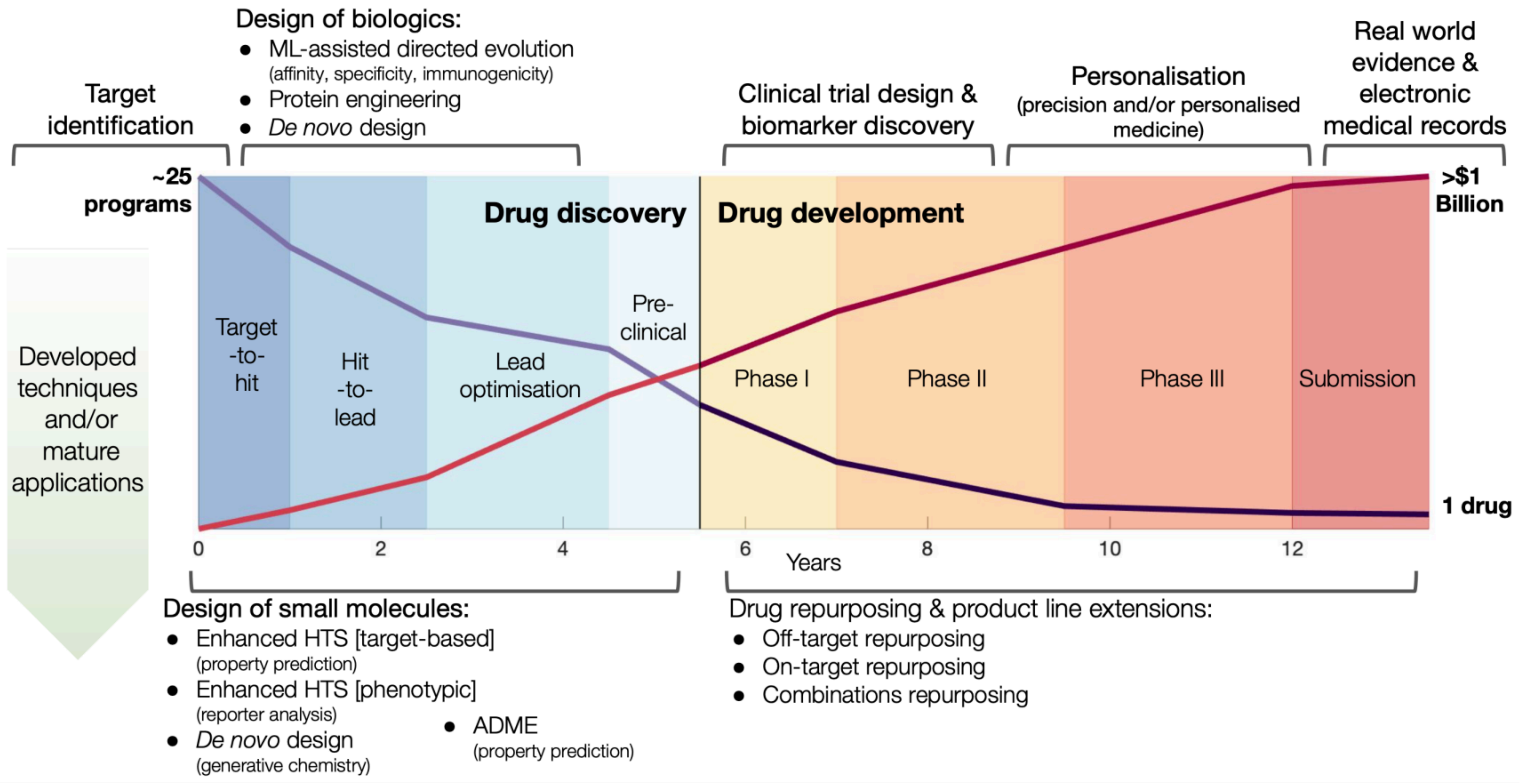
## Biomedical Domain and Drug Discovery

- Biomedical data is inherently relational — typically represented as a graph
  - **Molecular scale**: Proteins and other biomolecules can be represented as graphs capturing spatial and structural relationships between their amino acid residues. Small molecule drugs can similarly be represented as graphs relating their constituent atoms and chemical bonding structure.
  - **Intermediary scale**: An interactome is a set of molecular interactions in a particular cell — They can be represented as graphs that capture specific types of interactions between biomolecular species, e.g., protein–protein interaction graphs.
  - **Abstract scale**: Knowledge graphs can represent the complex relationships between drugs, side effects, diagnosis, associated treatments, and test results etc.

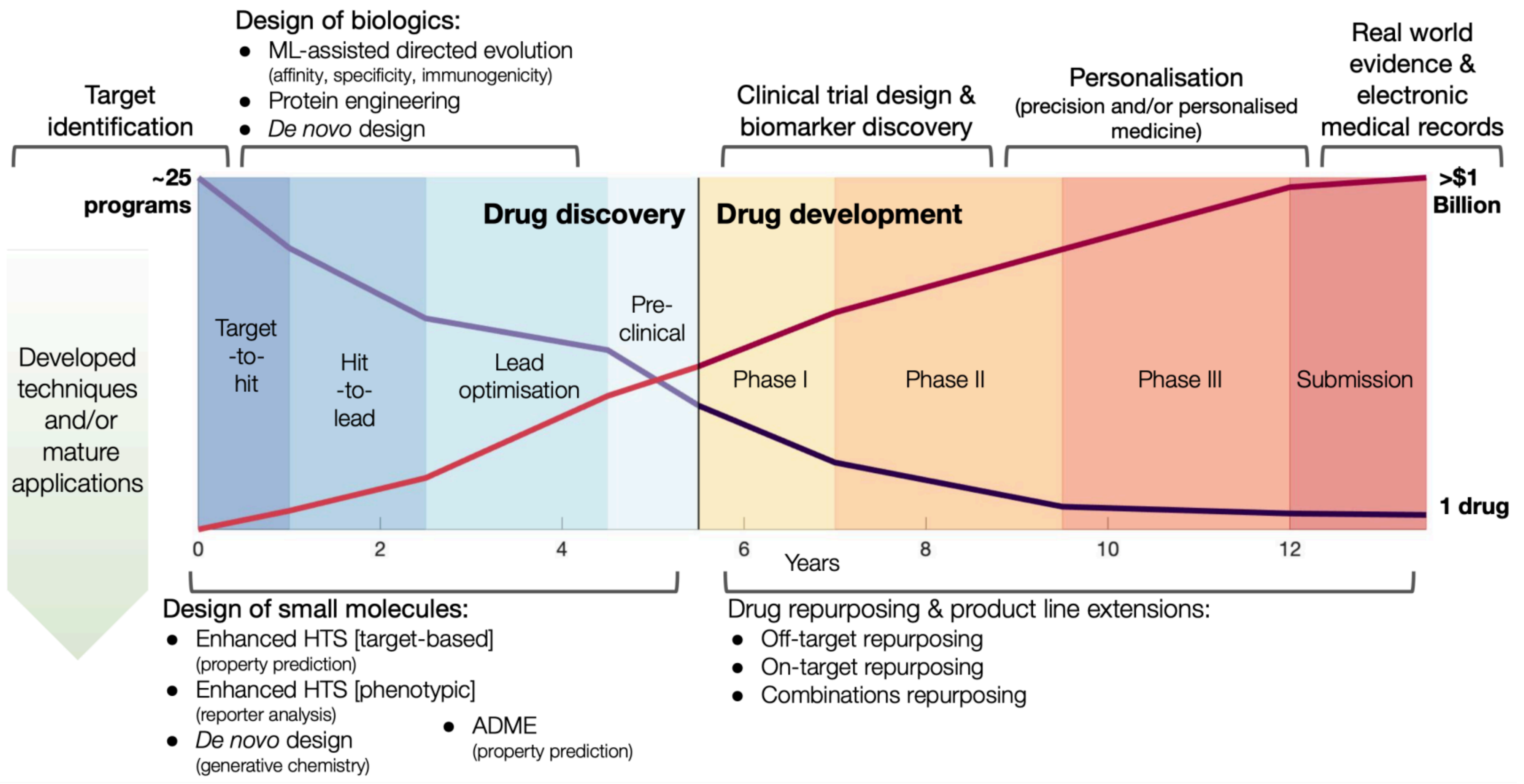
# Biomedical Domain and Drug Discovery

- Biomedical data is inherently relational — typically represented as a graph
    - **Molecular scale**: Proteins and other biomolecules can be represented as graphs capturing spatial and structural relationships between their amino acid residues. Small molecule drugs can similarly be represented as graphs relating their constituent atoms and chemical bonding structure.
    - **Intermediary scale**: An interactome is a set of molecular interactions in a particular cell — They can be represented as graphs that capture specific types of interactions between biomolecular species, e.g., protein–protein interaction graphs.
    - **Abstract scale**: Knowledge graphs can represent the complex relationships between drugs, side effects, diagnosis, associated treatments, and test results etc.
- Drug discovery is a long and expensive process — There is a greater interest in applying computational methodologies to enhance the drug discovery process and make it more efficient.

# Timeline of Drug Development



# Timeline of Drug Development



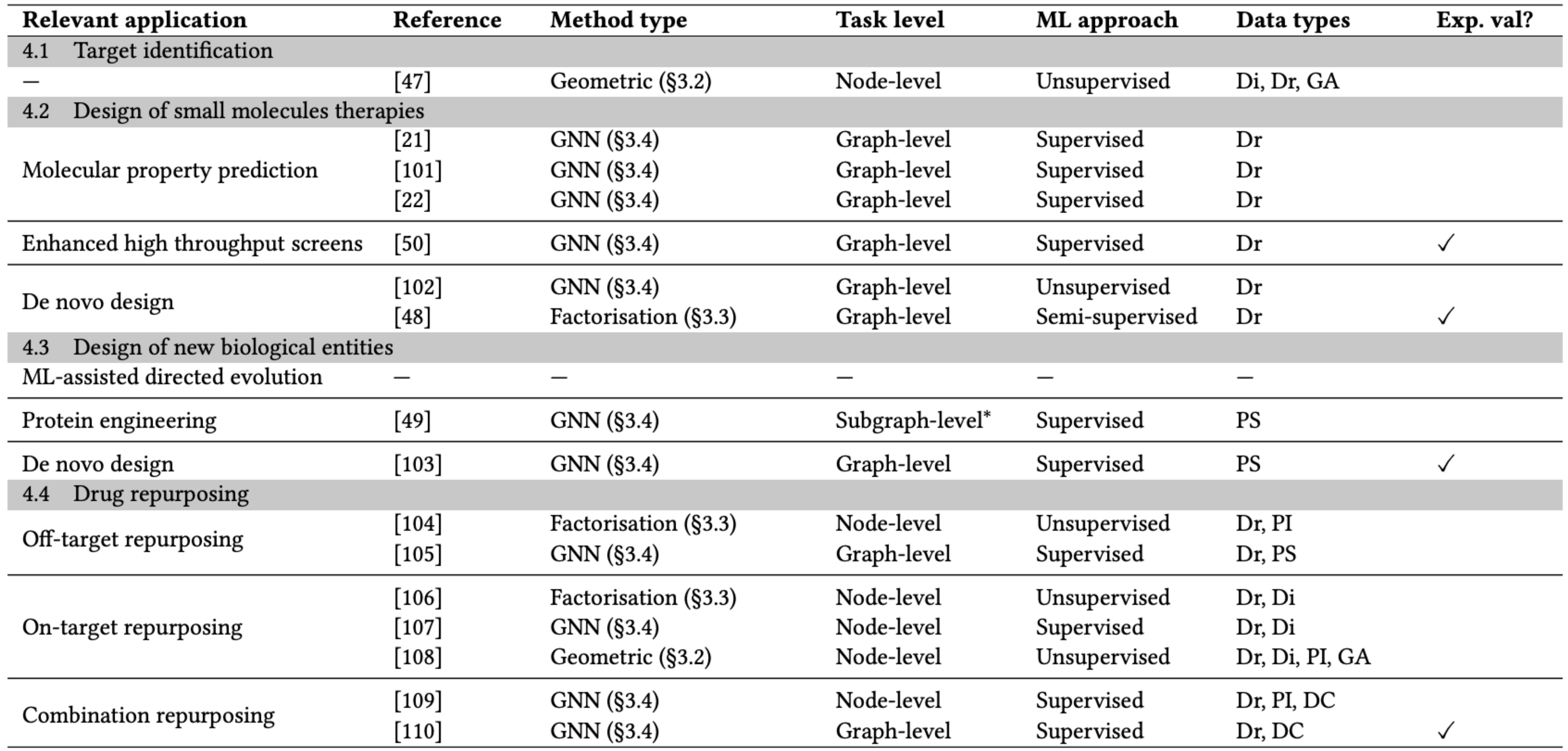
Target identification
Design of biologics:
ML-assisted directed evolution (affinity, specificity, immunogenicity)
Protein engineering
*De novo* design
Clinical trial design & biomarker discovery
Personalisation (precision and/or personalised medicine)
Real world evidence & electronic medical records
~25 programs
>$1 Billion
Drug discovery
Drug development
Developed techniques and/or mature applications
Target -to- hit
Hit -to- lead
Lead optimisation
Pre- clinical
Phase I
Phase II
Phase III
Submission
1 drug
0
2
4
6
8
10
12
Years
Design of small molecules:
Enhanced HTS [target-based] (property prediction)
Enhanced HTS [phenotypic] (reporter analysis)
*De novo* design (generative chemistry)
ADME (property prediction)
Drug repurposing & product line extensions:
Off-target repurposing
On-target repurposing
Combinations repurposing

Figure from (Gadoulet et al, 2021) showing the timeline of drug development linked to potential areas of application of GRL.

## Drug Development Applications

| Relevant application | Reference | Method type | Task level | ML approach | Data types | Exp. val? |
|---|---|---|---|---|---|---|
| 4.1 Target identification | | | | | | |
| — | [47] | Geometric (§3.2) | Node-level | Unsupervised | Di, Dr, GA | |
| 4.2 Design of small molecules therapies | | | | | | |
| Molecular property prediction | [21] | GNN (§3.4) | Graph-level | Supervised | Dr | |
| | [101] | GNN (§3.4) | Graph-level | Supervised | Dr | |
| | [22] | GNN (§3.4) | Graph-level | Supervised | Dr | |
| Enhanced high throughput screens | [50] | GNN (§3.4) | Graph-level | Supervised | Dr | ✓ |
| De novo design | [102] | GNN (§3.4) | Graph-level | Unsupervised | Dr | |
| | [48] | Factorisation (§3.3) | Graph-level | Semi-supervised | Dr | ✓ |
| 4.3 Design of new biological entities | | | | | | |
| ML-assisted directed evolution | — | — | — | — | — | |
| Protein engineering | [49] | GNN (§3.4) | Subgraph-level* | Supervised | PS | |
| De novo design | [103] | GNN (§3.4) | Graph-level | Supervised | PS | ✓ |
| 4.4 Drug repurposing | | | | | | |
| Off-target repurposing | [104] | Factorisation (§3.3) | Node-level | Unsupervised | Dr, PI | |
| | [105] | GNN (§3.4) | Graph-level | Supervised | Dr, PS | |
| On-target repurposing | [106] | Factorisation (§3.3) | Node-level | Unsupervised | Dr, Di | |
| | [107] | GNN (§3.4) | Node-level | Supervised | Dr, Di | |
| | [108] | Geometric (§3.2) | Node-level | Unsupervised | Dr, Di, PI, GA | |
| Combination repurposing | [109] | GNN (§3.4) | Node-level | Supervised | Dr, PI, DC | |
| | [110] | GNN (§3.4) | Graph-level | Supervised | Dr, DC | ✓ |

# Drug Development Applications



| Relevant application | Reference | Method type | Task level | ML approach | Data types | Exp. val? |
|---|---|---|---|---|---|---|
| 4.1 Target identification | | | | | | |
| — | [47] | Geometric (§3.2) | Node-level | Unsupervised | Di, Dr, GA | |
| 4.2 Design of small molecules therapies | | | | | | |
| Molecular property prediction | [21] | GNN (§3.4) | Graph-level | Supervised | Dr | |
| | [101] | GNN (§3.4) | Graph-level | Supervised | Dr | |
| | [22] | GNN (§3.4) | Graph-level | Supervised | Dr | |
| Enhanced high throughput screens | [50] | GNN (§3.4) | Graph-level | Supervised | Dr | ✓ |
| De novo design | [102] | GNN (§3.4) | Graph-level | Unsupervised | Dr | |
| | [48] | Factorisation (§3.3) | Graph-level | Semi-supervised | Dr | ✓ |
| 4.3 Design of new biological entities | | | | | | |
| ML-assisted directed evolution | — | — | — | — | — | |
| Protein engineering | [49] | GNN (§3.4) | Subgraph-level* | Supervised | PS | |
| De novo design | [103] | GNN (§3.4) | Graph-level | Supervised | PS | ✓ |
| 4.4 Drug repurposing | | | | | | |
| Off-target repurposing | [104] | Factorisation (§3.3) | Node-level | Unsupervised | Dr, PI | |
| | [105] | GNN (§3.4) | Graph-level | Supervised | Dr, PS | |
| On-target repurposing | [106] | Factorisation (§3.3) | Node-level | Unsupervised | Dr, Di | |
| | [107] | GNN (§3.4) | Node-level | Supervised | Dr, Di | |
| | [108] | Geometric (§3.2) | Node-level | Unsupervised | Dr, Di, PI, GA | |
| Combination repurposing | [109] | GNN (§3.4) | Node-level | Supervised | Dr, PI, DC | |
| | [110] | GNN (§3.4) | Graph-level | Supervised | Dr, DC | ✓ |

Table from (Gadoulet et al, 2021) listing exemplar applications of GNNs in drug discovery. The acronyms stand for Dr: Drugs, DC: Drug combinations, PS: Protein, PI: Protein interactions, GA: Gene annotations, Di:Diseases, respectively.

# A Deep Learning Approach to Antibiotic Discovery

## A Deep Learning Approach to Antibiotic Discovery

In a very interesting work, (Stokes et al., 2020) use MPNNs for antibiotic discovery. Over the past few decades, very few new antibiotics have been developed, and most of those newly approved antibiotics are slightly different variants of existing drugs.

## A Deep Learning Approach to Antibiotic Discovery

In a very interesting work, (Stokes et al., 2020) use MPNNs for antibiotic discovery. Over the past few decades, very few new antibiotics have been developed, and most of those newly approved antibiotics are slightly different variants of existing drugs.

The overall approach:

## A Deep Learning Approach to Antibiotic Discovery

In a very interesting work, (Stokes et al., 2020) use MPNNs for antibiotic discovery. Over the past few decades, very few new antibiotics have been developed, and most of those newly approved antibiotics are slightly different variants of existing drugs.

The overall approach:

- Develop a message passing neural network model by building a molecular representation based on a specific property (e.g., the inhibition of the growth of E. coli), using a message passing approach.

## A Deep Learning Approach to Antibiotic Discovery

In a very interesting work, (Stokes et al., 2020) use MPNNs for antibiotic discovery. Over the past few decades, very few new antibiotics have been developed, and most of those newly approved antibiotics are slightly different variants of existing drugs.

The overall approach:

- Develop a message passing neural network model by building a molecular representation based on a specific property (e.g., the inhibition of the growth of E. coli), using a message passing approach.
- Train the model using a collection of 2,335 diverse molecules for those that inhibited the growth of E. coli, augmenting the model with a set of molecular features, hyperparameter optimization,etc.

# A Deep Learning Approach to Antibiotic Discovery

In a very interesting work, (Stokes et al., 2020) use MPNNs for antibiotic discovery. Over the past few decades, very few new antibiotics have been developed, and most of those newly approved antibiotics are slightly different variants of existing drugs.

The overall approach:

- Develop a message passing neural network model by building a molecular representation based on a specific property (e.g., the inhibition of the growth of E. coli), using a message passing approach.
- Train the model using a collection of 2,335 diverse molecules for those that inhibited the growth of E. coli, augmenting the model with a set of molecular features, hyperparameter optimization,etc.
- Apply the model to multiple chemical libraries, comprising $> 107$ million molecules (e.g., ZINC15 database), to identify potential lead compounds with activity against E. coli.

## A Deep Learning Approach to Antibiotic Discovery

In a very interesting work, (Stokes et al., 2020) use MPNNs for antibiotic discovery. Over the past few decades, very few new antibiotics have been developed, and most of those newly approved antibiotics are slightly different variants of existing drugs.

The overall approach:

- Develop a message passing neural network model by building a molecular representation based on a specific property (e.g., the inhibition of the growth of E. coli), using a message passing approach.
- Train the model using a collection of 2,335 diverse molecules for those that inhibited the growth of E. coli, augmenting the model with a set of molecular features, hyperparameter optimization,etc.
- Apply the model to multiple chemical libraries, comprising $> 107$ million molecules (e.g., ZINC15 database), to identify potential lead compounds with activity against E. coli.
- After ranking the candidates according to the model's predicted score, select a list of promising candidates (only, 23 compounds) that can potentially inhibit the growth of E. coli.

## A Deep Learning Approach to Antibiotic Discovery

## A Deep Learning Approach to Antibiotic Discovery

The results are impressive:

## A Deep Learning Approach to Antibiotic Discovery

The results are impressive:

- A new compound, named halicin (after HAL from "2001: A Space Odyssey"), is identified as a potent inhibitor of E. coli growth. Halicin is structurally divergent from conventional antibiotics.

## A Deep Learning Approach to Antibiotic Discovery

The results are impressive:

- A new compound, named halicin (after HAL from “2001: A Space Odyssey”), is identified as a potent inhibitor of E. coli growth. Halicin is structurally divergent from conventional antibiotics.
- Experimental investigations revealed that halicin displays growth inhibitory properties against a wide phylogenetic spectrum of pathogens. It has been tested against dozens of bacterial strains isolated from patients and grown in lab dishes, revealing that it was able to kill many that are resistant to treatment

## A Deep Learning Approach to Antibiotic Discovery

The results are impressive:

- A new compound, named halicin (after HAL from "2001: A Space Odyssey"), is identified as a potent inhibitor of E. coli growth. Halicin is structurally divergent from conventional antibiotics.
- Experimental investigations revealed that halicin displays growth inhibitory properties against a wide phylogenetic spectrum of pathogens. It has been tested against dozens of bacterial strains isolated from patients and grown in lab dishes, revealing that it was able to kill many that are resistant to treatment
- In addition to halicin, from a distinct set of 23 empirically tested predictions from $>$107 million molecules found in the ZINC15 database, eight additional antibacterial compounds are discovered. These are also structurally distant from known antibiotics.

## A Deep Learning Approach to Antibiotic Discovery

The results are impressive:

- A new compound, named halicin (after HAL from "2001: A Space Odyssey"), is identified as a potent inhibitor of E. coli growth. Halicin is structurally divergent from conventional antibiotics.
- Experimental investigations revealed that halicin displays growth inhibitory properties against a wide phylogenetic spectrum of pathogens. It has been tested against dozens of bacterial strains isolated from patients and grown in lab dishes, revealing that it was able to kill many that are resistant to treatment
- In addition to halicin, from a distinct set of 23 empirically tested predictions from >107 million molecules found in the ZINC15 database, eight additional antibacterial compounds are discovered. These are also structurally distant from known antibiotics.
- Remarkably, two of these molecules displayed potent broad-spectrum activity and could overcome an array of antibiotic-resistance determinants in E. coli.

## A Deep Learning Approach to Antibiotic Discovery

The results are impressive:

- A new compound, named halicin (after HAL from "2001: A Space Odyssey"), is identified as a potent inhibitor of E. coli growth. Halicin is structurally divergent from conventional antibiotics.
- Experimental investigations revealed that halicin displays growth inhibitory properties against a wide phylogenetic spectrum of pathogens. It has been tested against dozens of bacterial strains isolated from patients and grown in lab dishes, revealing that it was able to kill many that are resistant to treatment
- In addition to halicin, from a distinct set of 23 empirically tested predictions from >107 million molecules found in the ZINC15 database, eight additional antibacterial compounds are discovered. These are also structurally distant from known antibiotics.
- Remarkably, two of these molecules displayed potent broad-spectrum activity and could overcome an array of antibiotic-resistance determinants in E. coli.

"This work highlights the significant impact that machine learning can have on early antibiotic discovery efforts by simultaneously increasing the accuracy rate of lead compound identification and decreasing the cost of screening efforts."

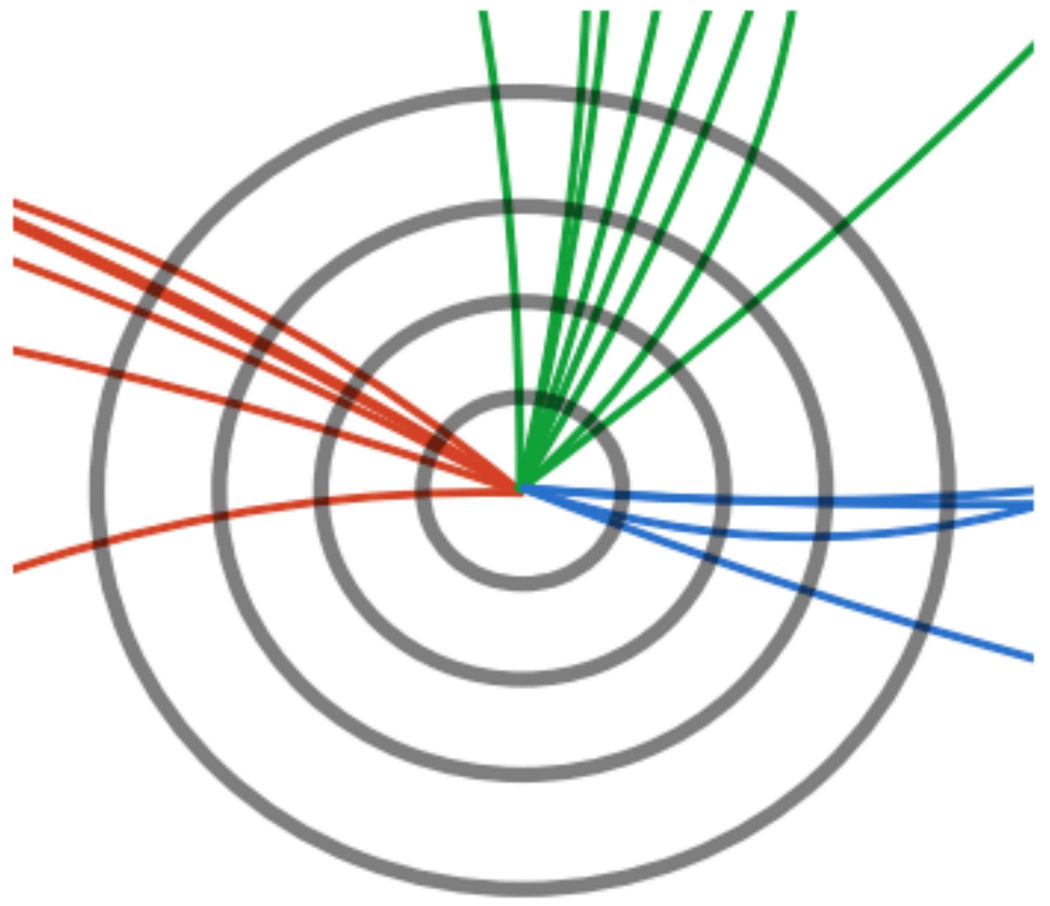
## A Deep Learning Approach to Antibiotic Discovery

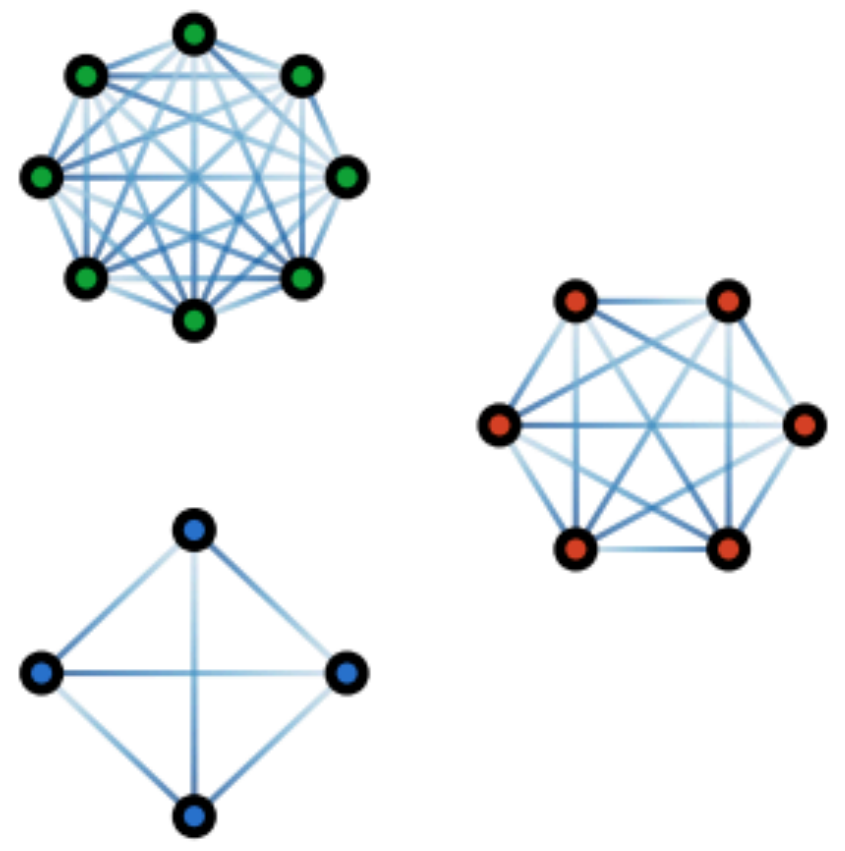
## A Deep Learning Approach to Antibiotic Discovery

“Indeed, modern neural molecular representations have the potential to: (1) decrease the cost of lead molecule identification because screening is limited to gathering appropriate training data, (2) increase the true positive rate of identifying structurally novel compounds with the desired bioactivity, and (3) decrease the time and labor required to find these ideal compounds from months or years to weeks.”

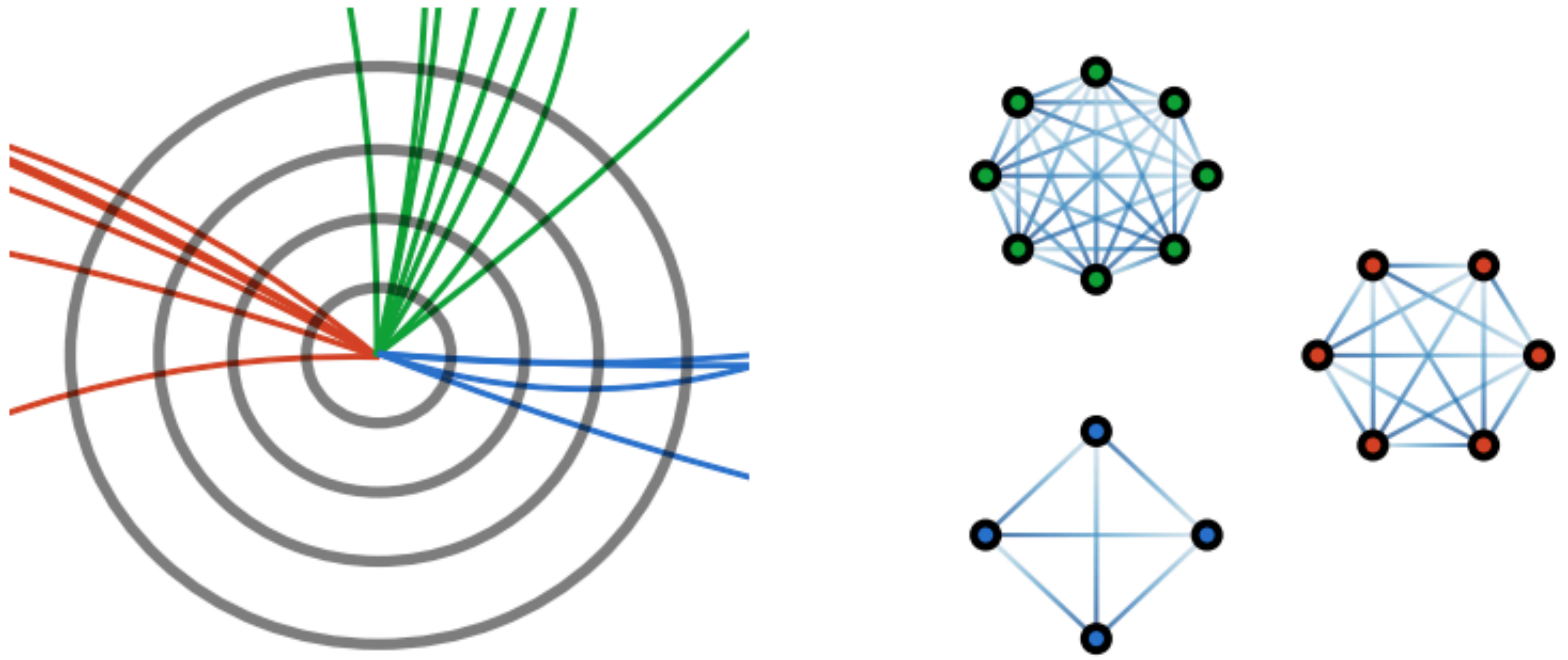
(Stokes et al., 2020)

# Particle Physics



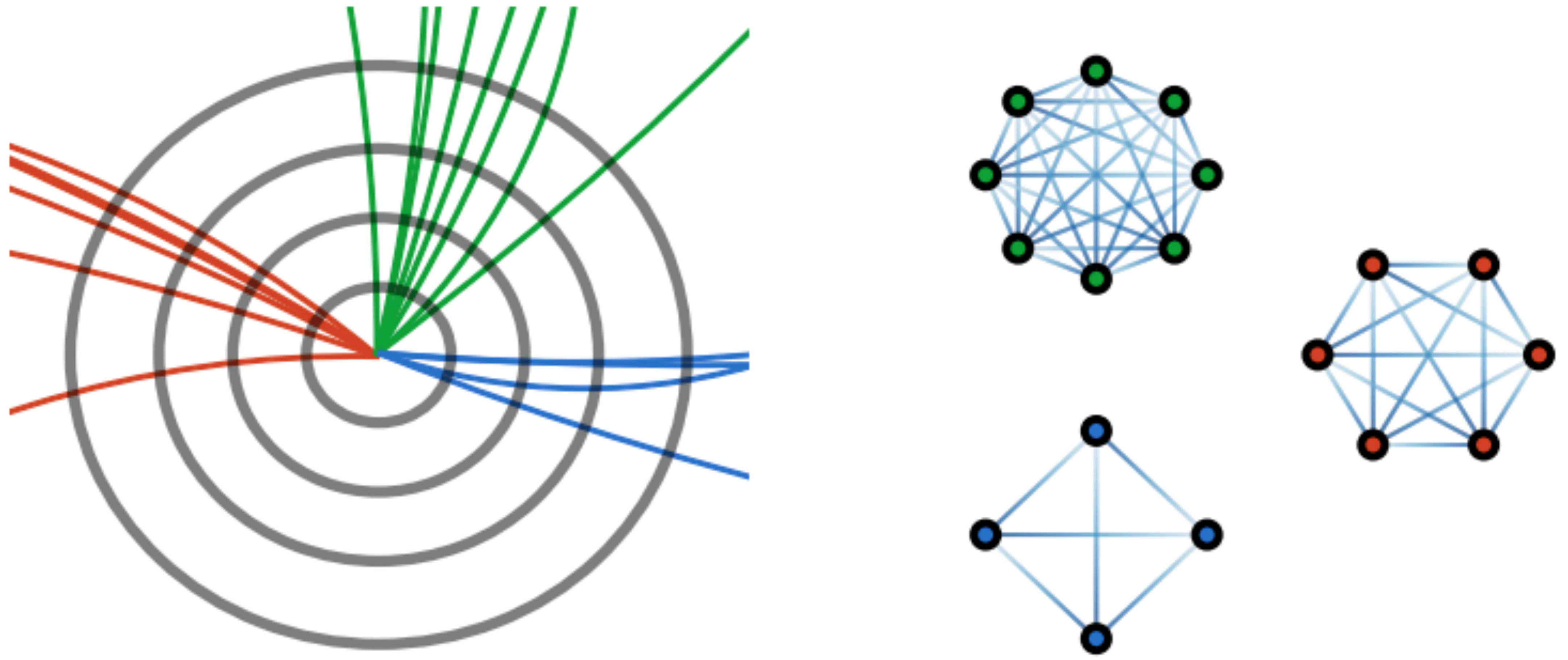


# Particle Physics



The data in particle physics are often represented by sets and graphs and as such, graph neural networks offer key advantages.

# Particle Physics



The data in particle physics are often represented by sets and graphs and as such, graph neural networks offer key advantages.

We follow the survey by (Shlomi et al., 2021), and briefly highlight some applications of graph representation learning in particle physics.

# Particle Physics: Jet Classification

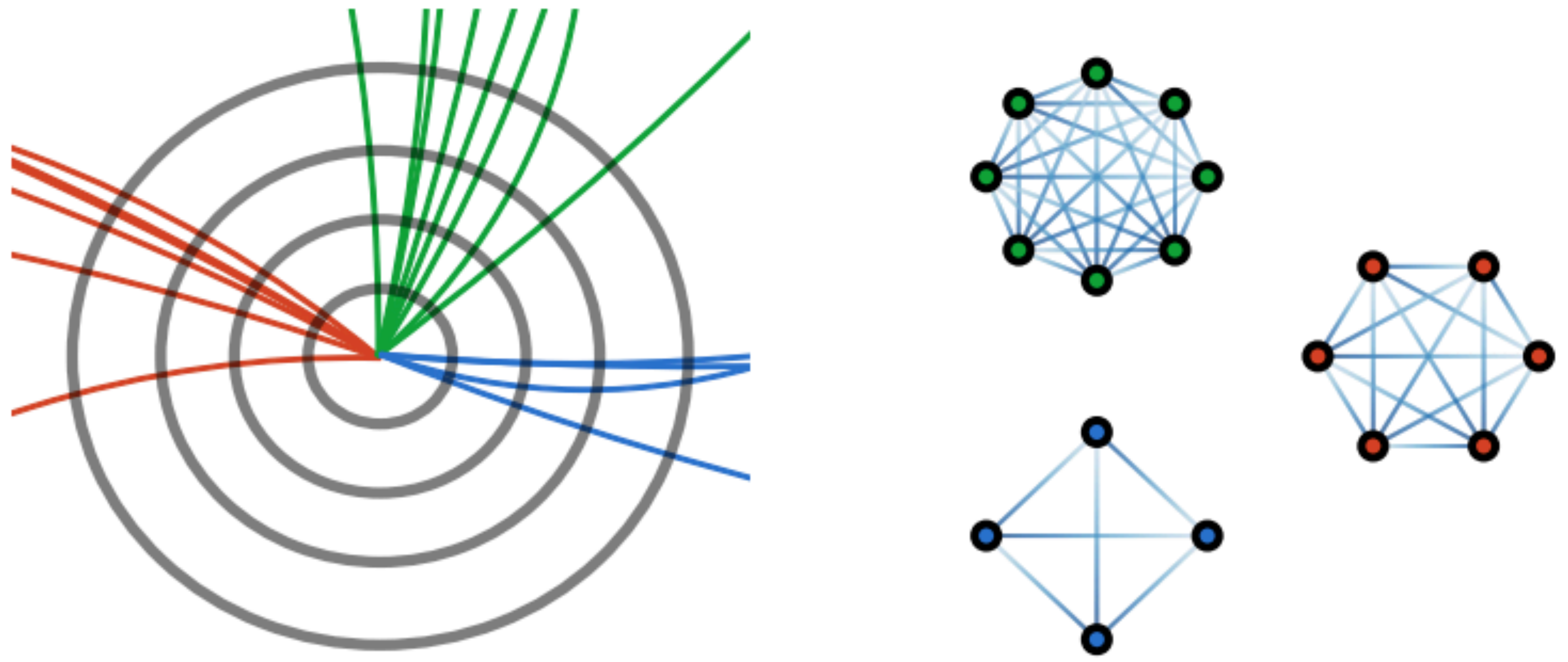


Figure from (Shlomi et al., 2021), depicting jet classification based on the particles associated to the jet.

# Particle Physics: Jet Classification

**Jet classification**. Jets are sprays of stable particles that stem from multiple successive interaction and decays of particles, originating from a single initial object, i.e., quark, gluon, W-boson, top-quark, or Higgs boson.

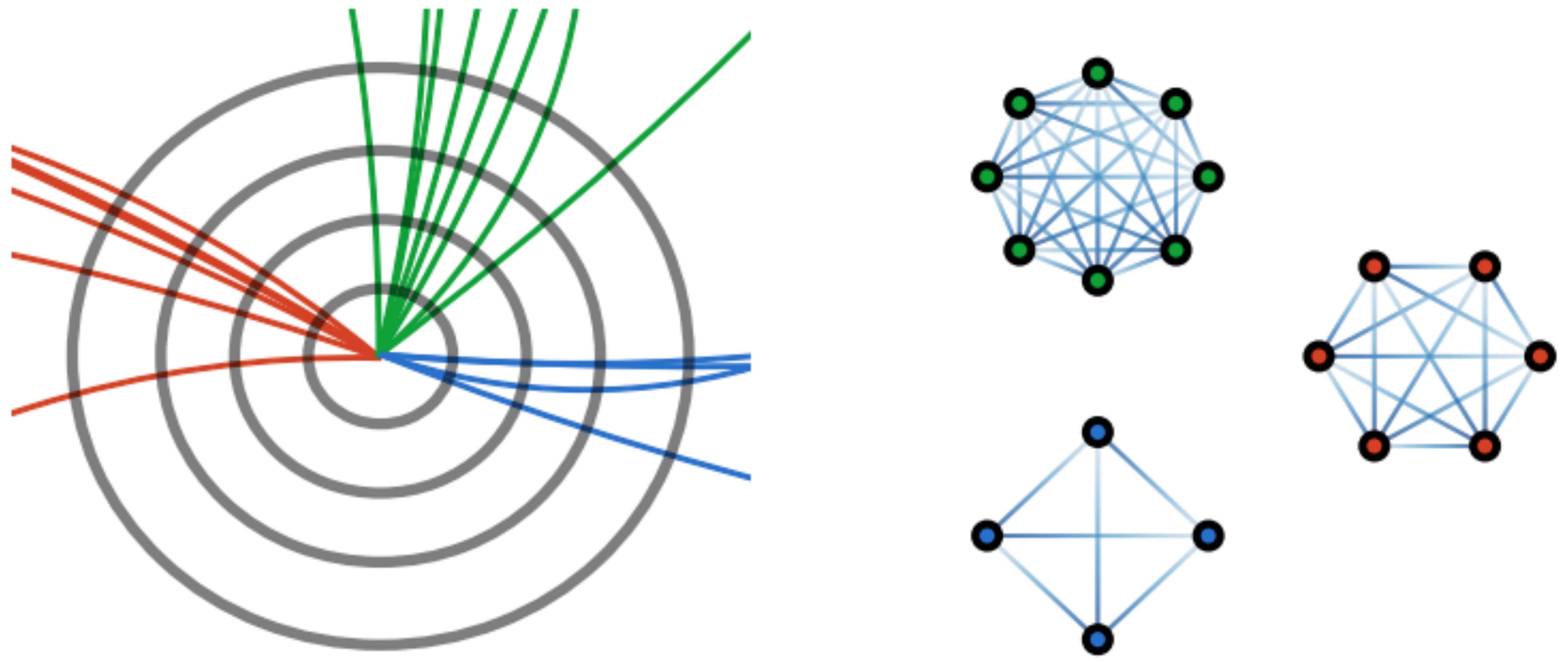


Figure from (Shlomi et al., 2021), depicting jet classification based on the particles associated to the jet.

# Particle Physics: Jet Classification

**Jet classification**. Jets are sprays of stable particles that stem from multiple successive interaction and decays of particles, originating from a single initial object, i.e., quark, gluon, W-boson, top-quark, or Higgs boson.

The task is then to identify the original object that gave rise to the jet — a very important task in particle physics.

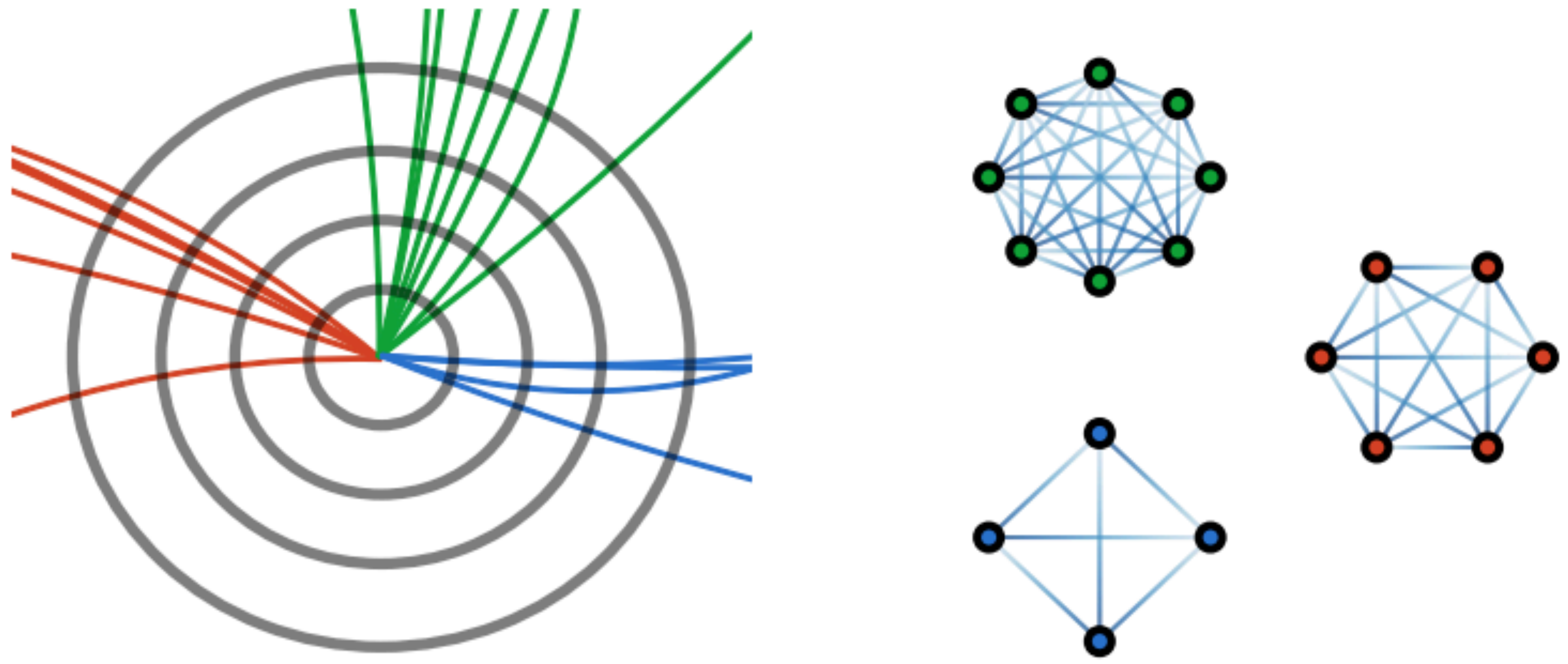


Figure from (Shlomi et al., 2021), depicting jet classification based on the particles associated to the jet.

# Particle Physics: Jet Classification

**Jet classification**. Jets are sprays of stable particles that stem from multiple successive interaction and decays of particles, originating from a single initial object, i.e., quark, gluon, W-boson, top-quark, or Higgs boson.

The task is then to identify the original object that gave rise to the jet — a very important task in particle physics.

One approach is to view a jet as a graph, where nodes are particles (with features) and edges represent interactions, and apply graph classification.

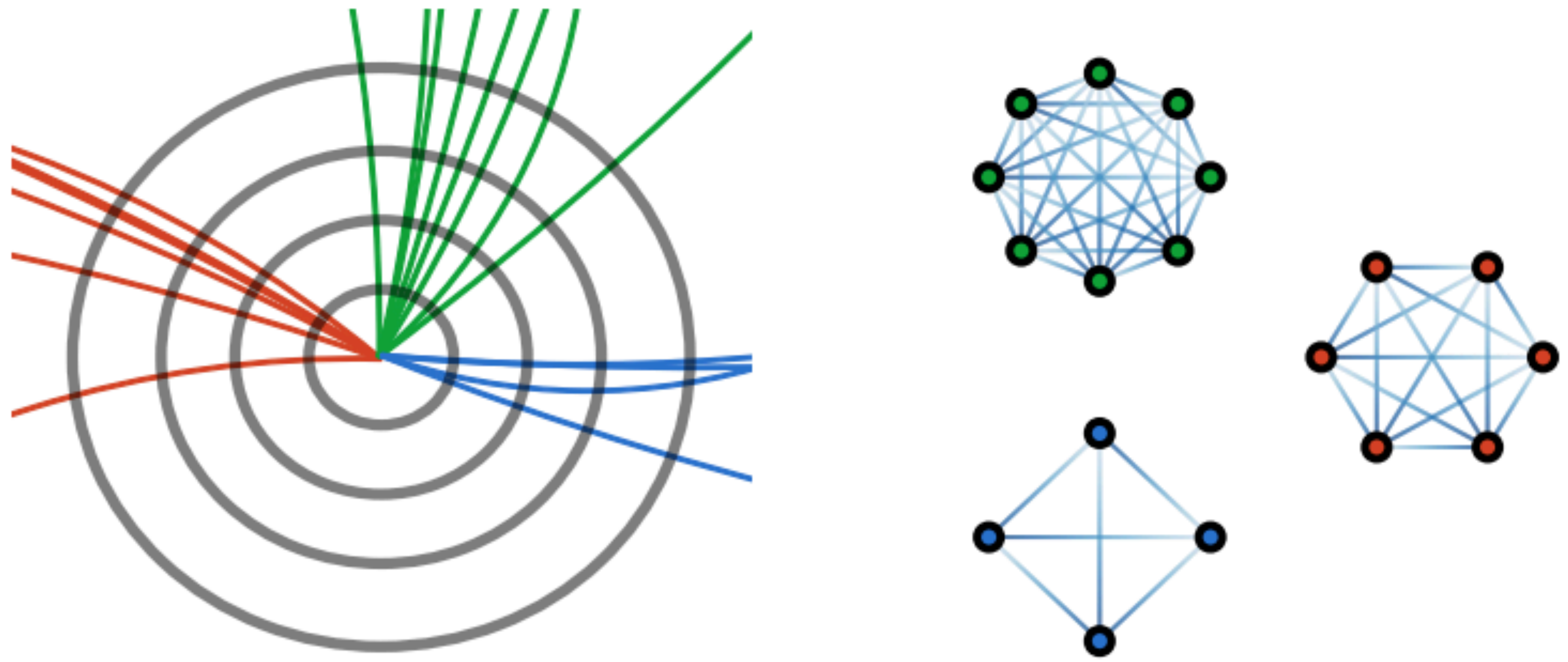


Figure from (Shlomi et al., 2021), depicting jet classification based on the particles associated to the jet.

## Particle Physics: Event Classification

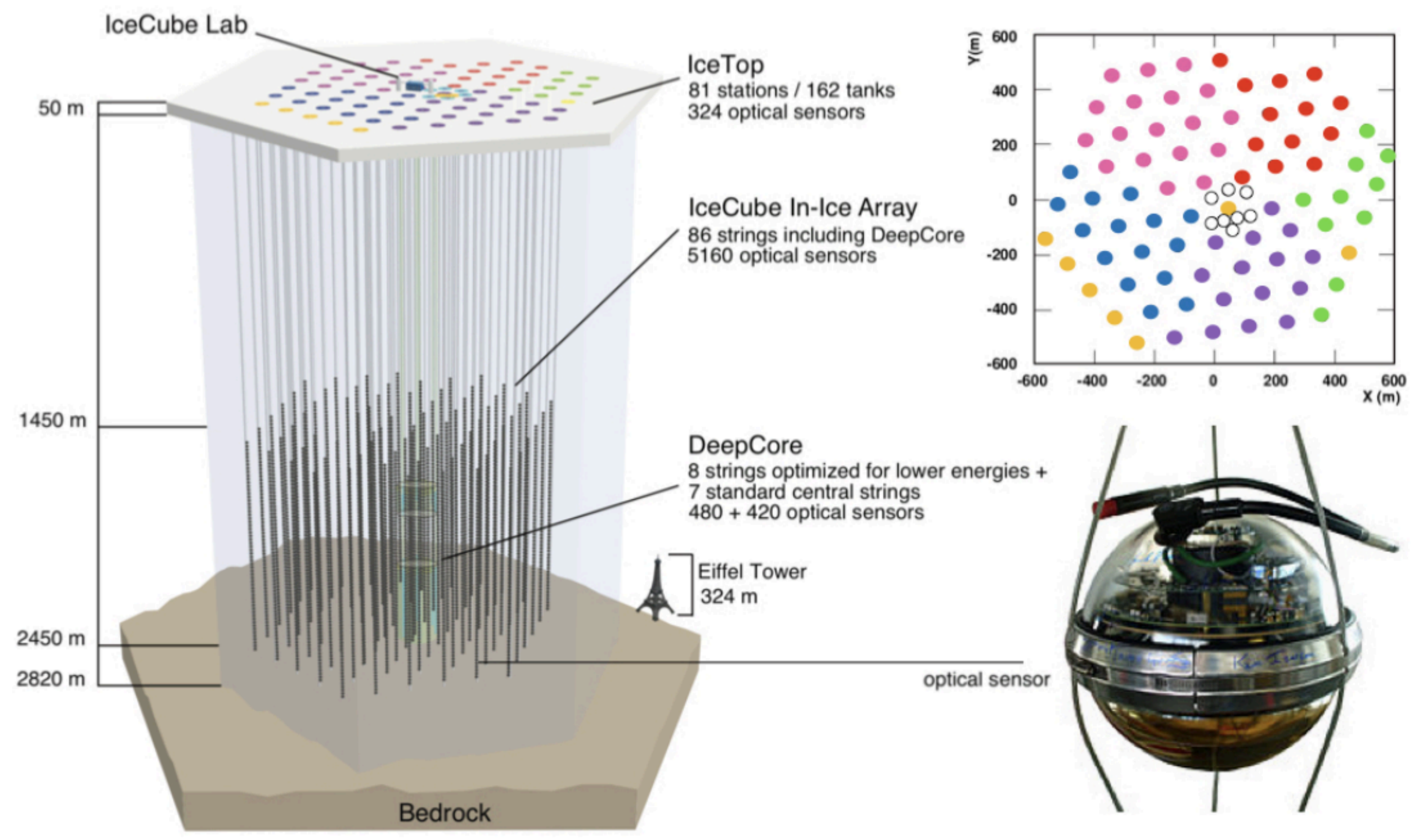


Figure from (Choma et al., 2018) depicting the IceCube Neutrino Observatory with the in-ice array, its subarray DeepCore, and the cosmic-ray air shower array IceTop.

# Particle Physics: Event Classification

**Event Classification**. The task of predicting the physics process at the origin of the recorded data.

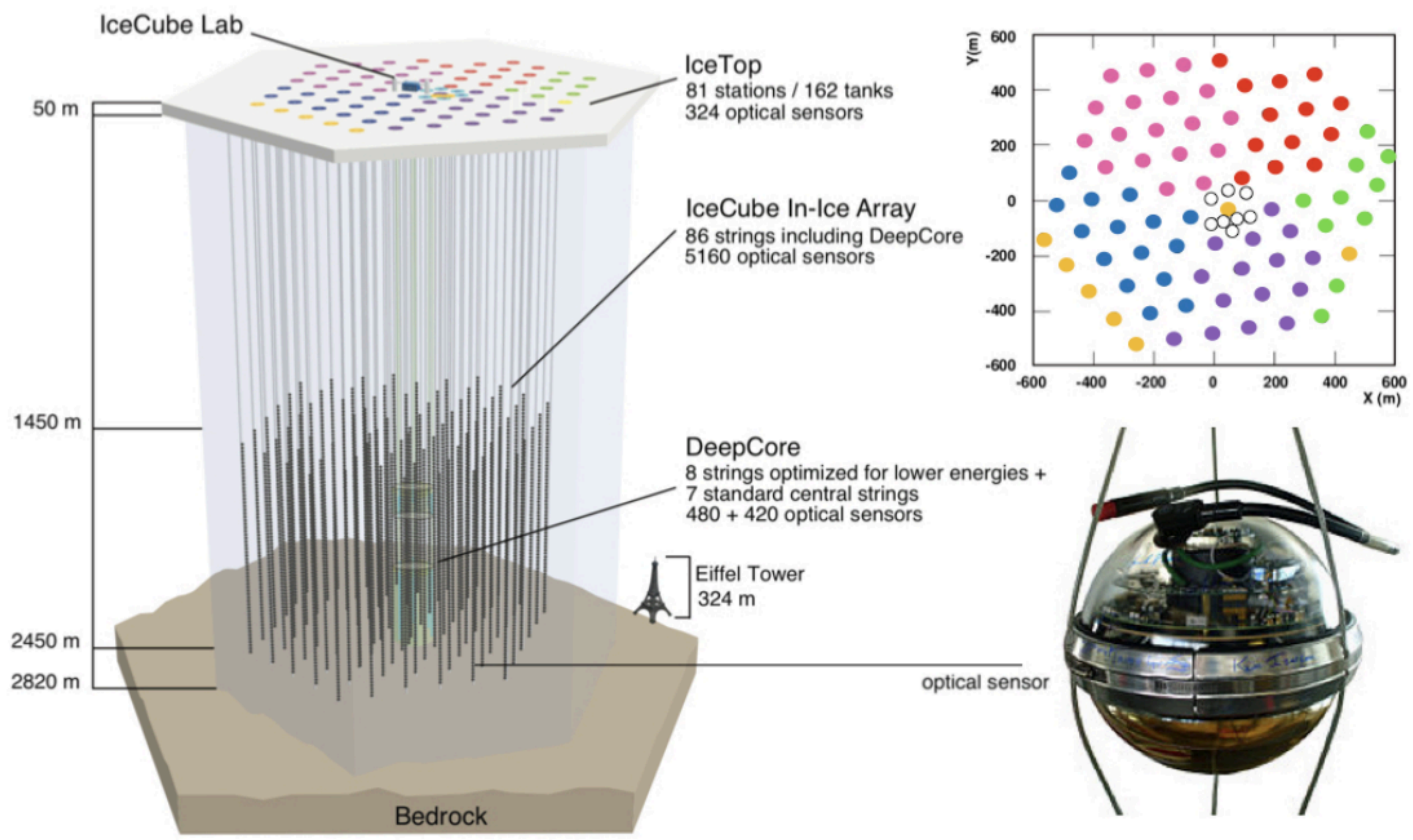


Figure from (Choma et al., 2018) depicting the IceCube Neutrino Observatory with the in-ice array, its subarray DeepCore, and the cosmic-ray air shower array IceTop.

## Particle Physics: Event Classification

**Event Classification**. The task of predicting the physics process at the origin of the recorded data.

**Example:** Graph neural networks are leveraged to improve signal detection in the IceCube neutrino observatory (Choma et al., 2018).

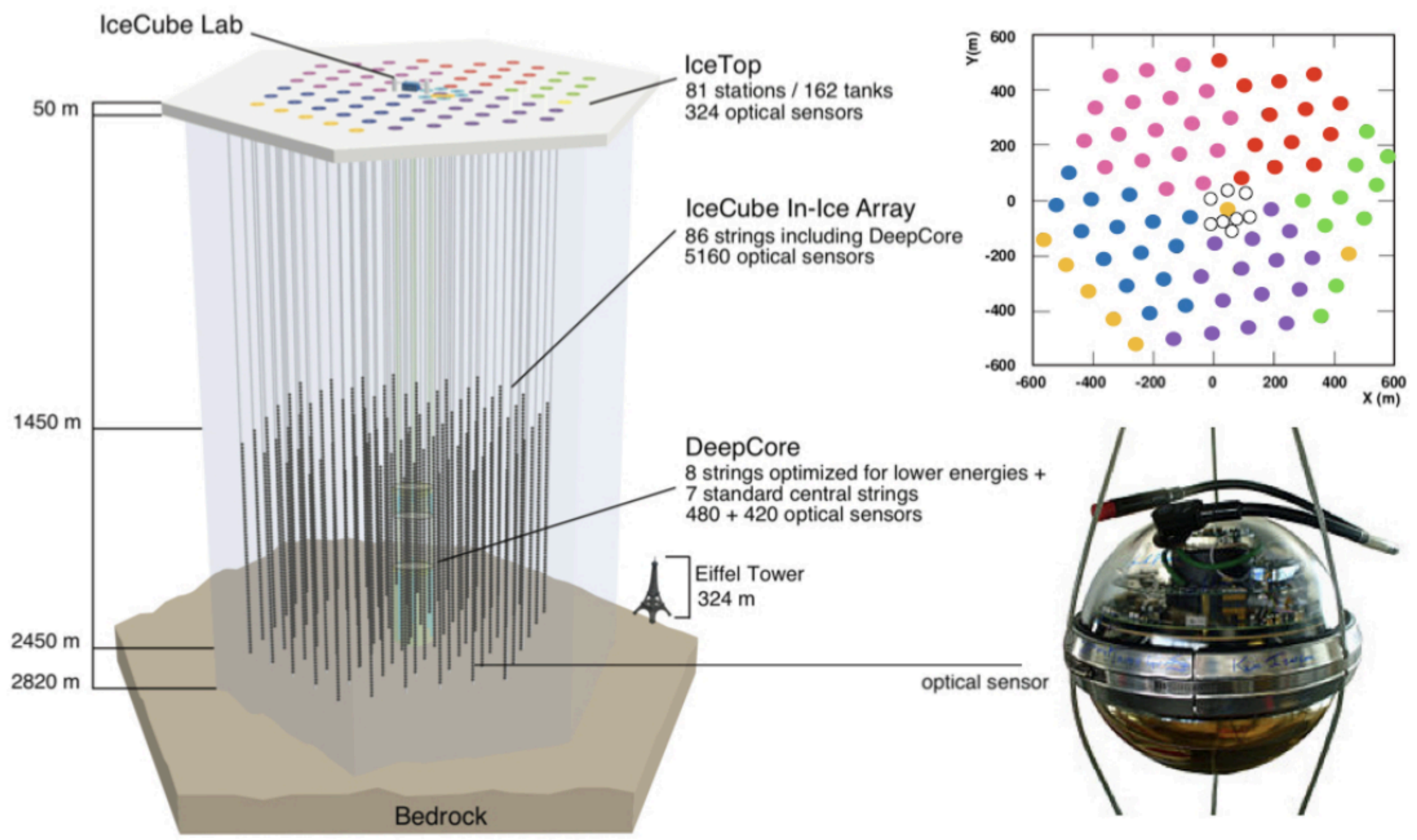


Figure from (Choma et al., 2018) depicting the IceCube Neutrino Observatory with the in-ice array, its subarray DeepCore, and the cosmic-ray air shower array IceTop.

# Particle Physics: Event Classification

**Event Classification**. The task of predicting the physics process at the origin of the recorded data.

**Example:** Graph neural networks are leveraged to improve signal detection in the IceCube neutrino observatory (Choma et al., 2018).

The IceCube detector array is modelled as a graph, where vertices are sensors and edges are a learned function of the sensors' spatial coordinates.

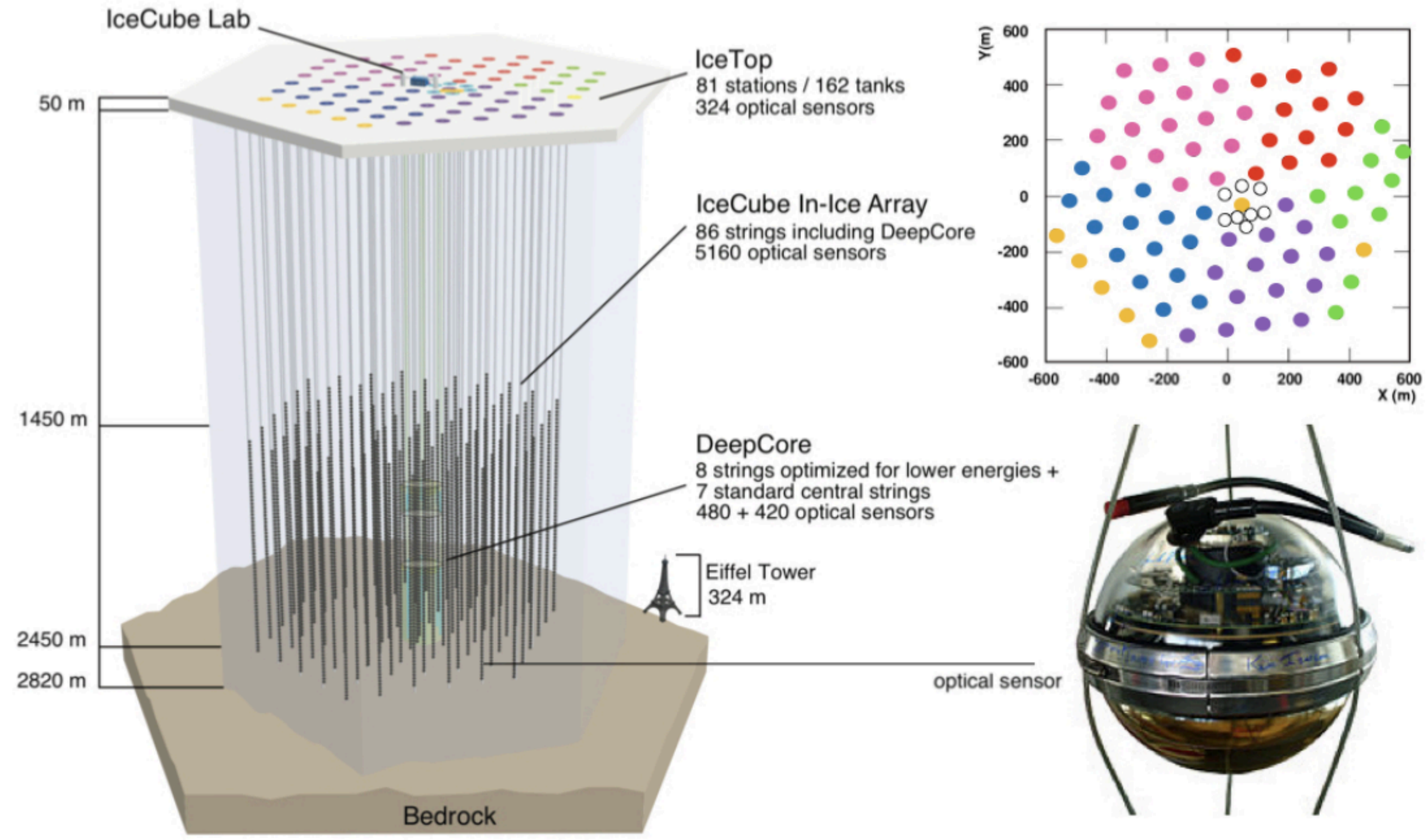


Figure from (Choma et al., 2018) depicting the IceCube Neutrino Observatory with the in-ice array, its subarray DeepCore, and the cosmic-ray air shower array IceTop.

## Particle Physics: Event Classification

**Event Classification**. The task of predicting the physics process at the origin of the recorded data.

**Example:** Graph neural networks are leveraged to improve signal detection in the IceCube neutrino observatory (Choma et al., 2018).

The IceCube detector array is modelled as a graph, where vertices are sensors and edges are a learned function of the sensors' spatial coordinates.

The goal is the classification of the signal in the IceCube detector, to determine if a muon originated from a cosmic neutrino, or from a cosmic ray showering in the earth atmosphere.

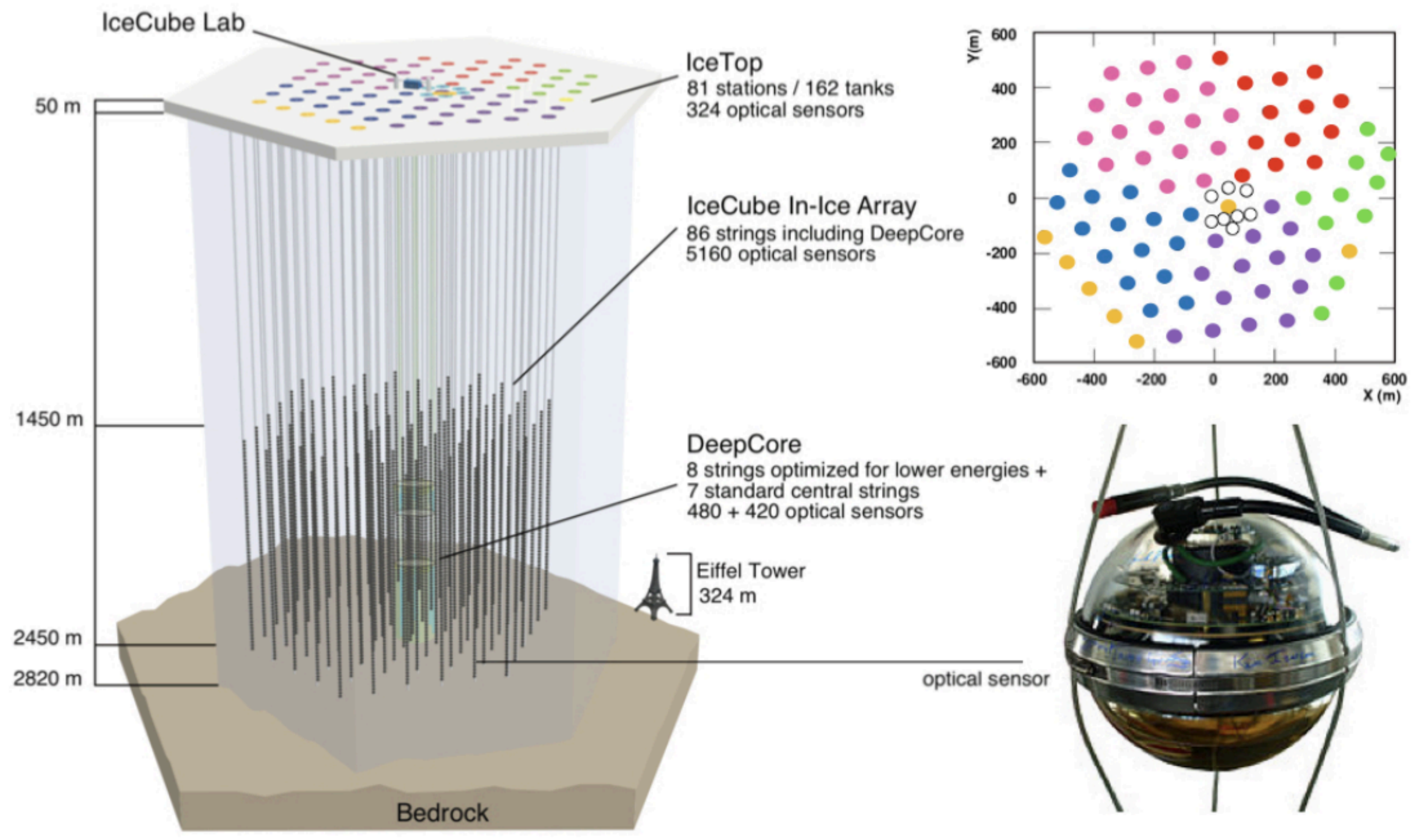


Figure from (Choma et al., 2018) depicting the IceCube Neutrino Observatory with the in-ice array, its subarray DeepCore, and the cosmic-ray air shower array IceTop.

## Combinatorial Optimisation and Reasoning

## Reasoning Capacity of Graph Neural Networks

## Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

## Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

## Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

It is not really plausible to expect GNNs to solve NP-hard problems beyond small instances.

# Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

It is not really plausible to expect GNNs to solve NP-hard problems beyond small instances.

**Example**: Can GNNs learn to solve (small) SAT instances with single-bit supervision (Selsam et al., 2018)?

## Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

It is not really plausible to expect GNNs to solve NP-hard problems beyond small instances.

**Example**: Can GNNs learn to solve (small) SAT instances with single-bit supervision (Selsam et al., 2018)?

- This is an interesting exercise to understand the reasoning capacity of graph neural networks:

## Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

It is not really plausible to expect GNNs to solve NP-hard problems beyond small instances.

**Example**: Can GNNs learn to solve (small) SAT instances with single-bit supervision (Selsam et al., 2018)?

- This is an interesting exercise to understand the reasoning capacity of graph neural networks:
    - Represent each propositional formula as a graph.

# Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

It is not really plausible to expect GNNs to solve NP-hard problems beyond small instances.

**Example**: Can GNNs learn to solve (small) SAT instances with single-bit supervision (Selsam et al., 2018)?

- This is an interesting exercise to understand the reasoning capacity of graph neural networks:
  - Represent each propositional formula as a graph.
  - Produce labelled training data, based on existing SAT solvers, i.e., label graphs with 0/1 reflecting the satisfiability of the formula the graph represents.

# Reasoning Capacity of Graph Neural Networks

There is a series of work trying to identify the reasoning capacity of GNNs.

NP-hard problems are of interest due to their combinatorial nature.

It is not really plausible to expect GNNs to solve NP-hard problems beyond small instances.

**Example**: Can GNNs learn to solve (small) SAT instances with single-bit supervision (Selsam et al., 2018)?

- This is an interesting exercise to understand the reasoning capacity of graph neural networks:
    - Represent each propositional formula as a graph.
    - Produce labelled training data, based on existing SAT solvers, i.e., label graphs with 0/1 reflecting the satisfiability of the formula the graph represents.
    - Train the graph neural network, and predict satisfiability status of novel formulas, given as graphs.

## Graph Neural Networks and Combinatorial Problems

## Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

## Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.

## Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.
- Permutation-invariance, e.g., $(x \land y) \lor (\neg x \land \neg y) \equiv (\neg x \land \neg y) \lor (x \land y)$.

# Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.
- Permutation-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (x \wedge y)$.
- Naming-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg z \wedge \neg u) \vee (z \wedge u)$,

## Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.
- Permutation-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (x \wedge y)$.
- Naming-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg z \wedge \neg u) \vee (z \wedge u)$,
- Strong inductive bias, given by formula distinguishability.

## Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.
- Permutation-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (x \wedge y)$.
- Naming-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg z \wedge \neg u) \vee (z \wedge u)$,
- Strong inductive bias, given by formula distinguishability.
- Separate representations for logical operators $\wedge$ , $\vee$ .

# Graph Neural Networks and Combinatorial Problems

GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.
- Permutation-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (x \wedge y)$.
- Naming-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg z \wedge \neg u) \vee (z \wedge u)$,
- Strong inductive bias, given by formula distinguishability.
- Separate representations for logical operators $\wedge$ , $\vee$ .

Gated graph neural networks have shown promising generalisation performance on very small problem instances. Clearly, this is approach is nowhere near actual SAT solvers, but it shows that GNNs can simulate logical reasoning at a very small scale.

## Graph Neural Networks and Combinatorial Problems

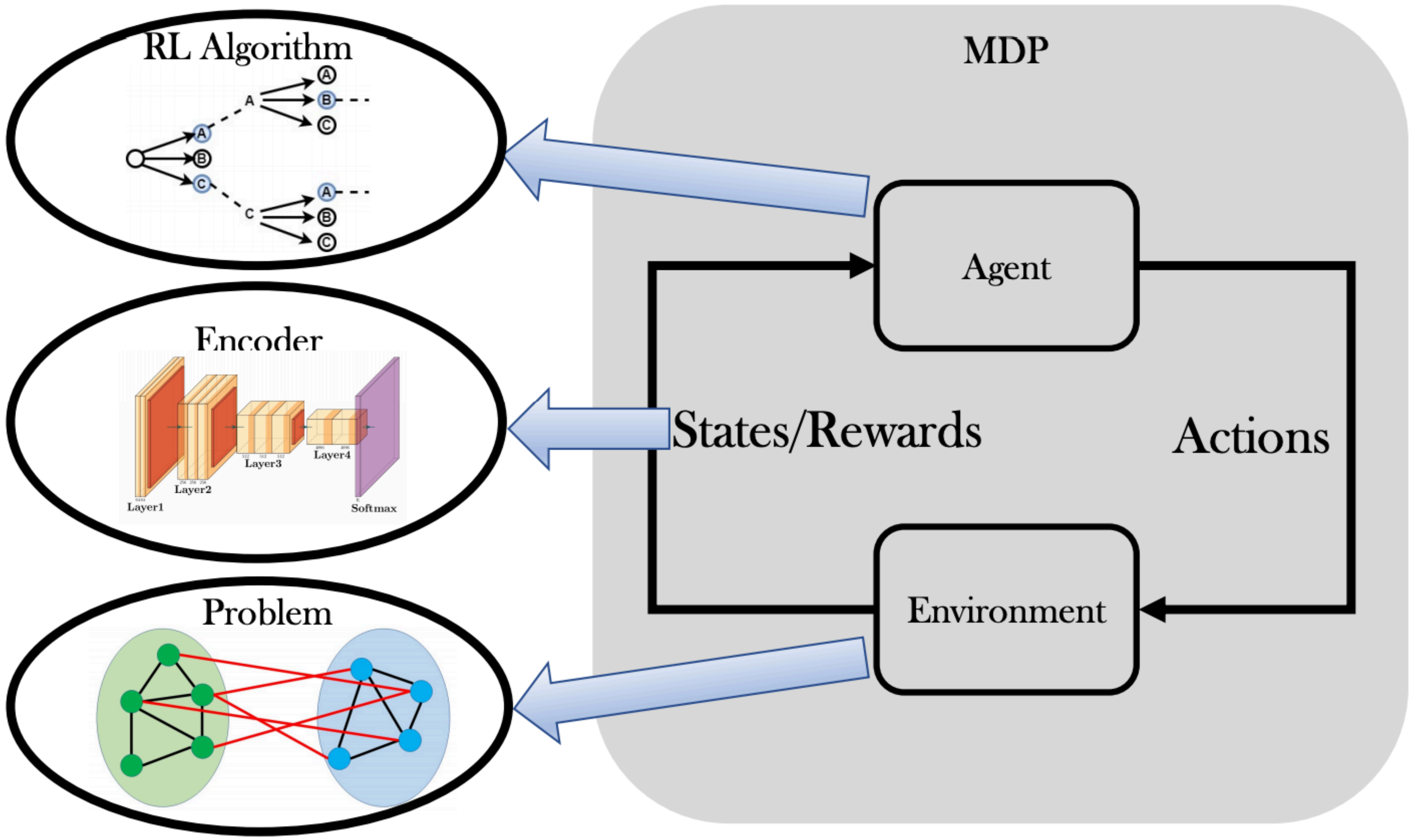
GNNs are obvious candidates for such tasks in the context of neural networks:

- Explicit structural encoding of an input formula.
- Permutation-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg x \wedge \neg y) \vee (x \wedge y)$.
- Naming-invariance, e.g., $(x \wedge y) \vee (\neg x \wedge \neg y) \equiv (\neg z \wedge \neg u) \vee (z \wedge u)$,
- Strong inductive bias, given by formula distinguishability.
- Separate representations for logical operators $\wedge$, $\vee$.

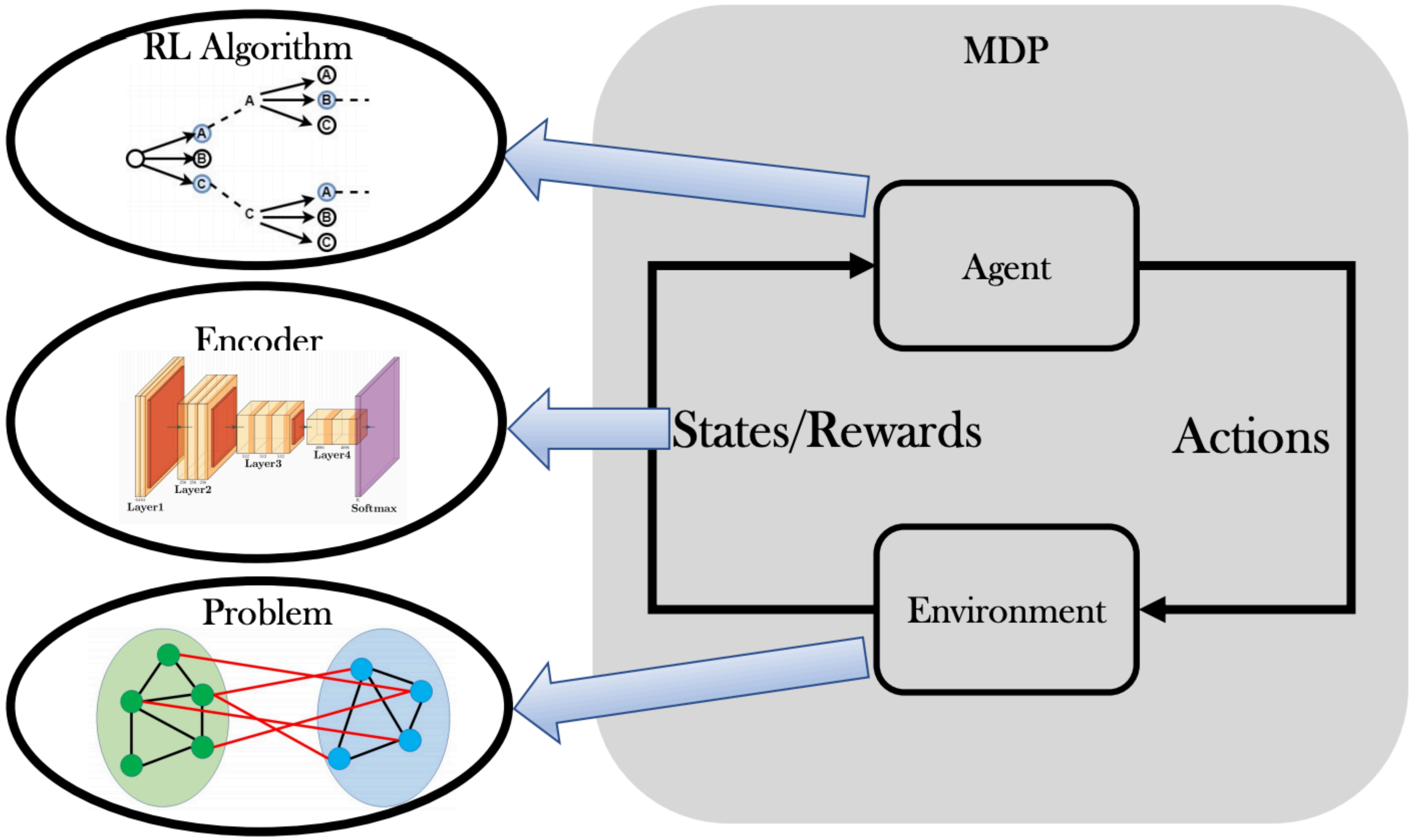
Gated graph neural networks have shown promising generalisation performance on very small problem instances. Clearly, this is approach is nowhere near actual SAT solvers, but it shows that GNNs can simulate logical reasoning at a very small scale.

Many other problems, beyond SAT, such as TSP, #SAT, etc. are investigated in this context.

## Graph Neural Networks and Combinatorial Problems



# Graph Neural Networks and Combinatorial Problems



Graph neural networks alone are quite limited to attack such problems, and a line of work combines the power of graph neural networks with reinforcement learning for solving combinatorial optimisation problems. Figure is taken from a survey paper (Mazyavkina et al., 2020) and shows the pipeline.

## Graph Neural Networks and Combinatorial Problems

## Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

## Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

## Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

**Encoder:** States of a Markov Decision Process are mapped to the actions' values, using an encoder.

## Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

**Encoder:** States of a Markov Decision Process are mapped to the actions' values, using an encoder.

A typical run is as follows:

## Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

**Encoder:** States of a Markov Decision Process are mapped to the actions' values, using an encoder.

A typical run is as follows:

- Formulate the combinatorial problem, e.g., Max-Cut problem, as a Markov Decision Process.

# Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

**Encoder:** States of a Markov Decision Process are mapped to the actions' values, using an encoder.

A typical run is as follows:

- Formulate the combinatorial problem, e.g., Max-Cut problem, as a Markov Decision Process.
- Encode states with a GNN model, i.e., every node has a vector representation encoded by a GNN.

# Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

**Encoder:** States of a Markov Decision Process are mapped to the actions' values, using an encoder.

A typical run is as follows:

- Formulate the combinatorial problem, e.g., Max-Cut problem, as a Markov Decision Process.
- Encode states with a GNN model, i.e., every node has a vector representation encoded by a GNN.
- The agent is driven by an reinforcement learning algorithm, e.g., Monte-Carlo Tree Search, and makes decisions that move the environment to the next state, e.g., removing a vertex from a solution set.

## Graph Neural Networks and Combinatorial Problems

**Idea**: Model the problem as a sequential decision-making process, e.g., Markov Decision Process, where the agent interacts with the environment by performing a sequence of actions in order to find a solution.

**Goal:** An agent acting in Markov Decision Process tries to find a policy function that maps states into actions, while maximising the expected cumulative discounted sum of rewards, i.e., finding an optimal policy.

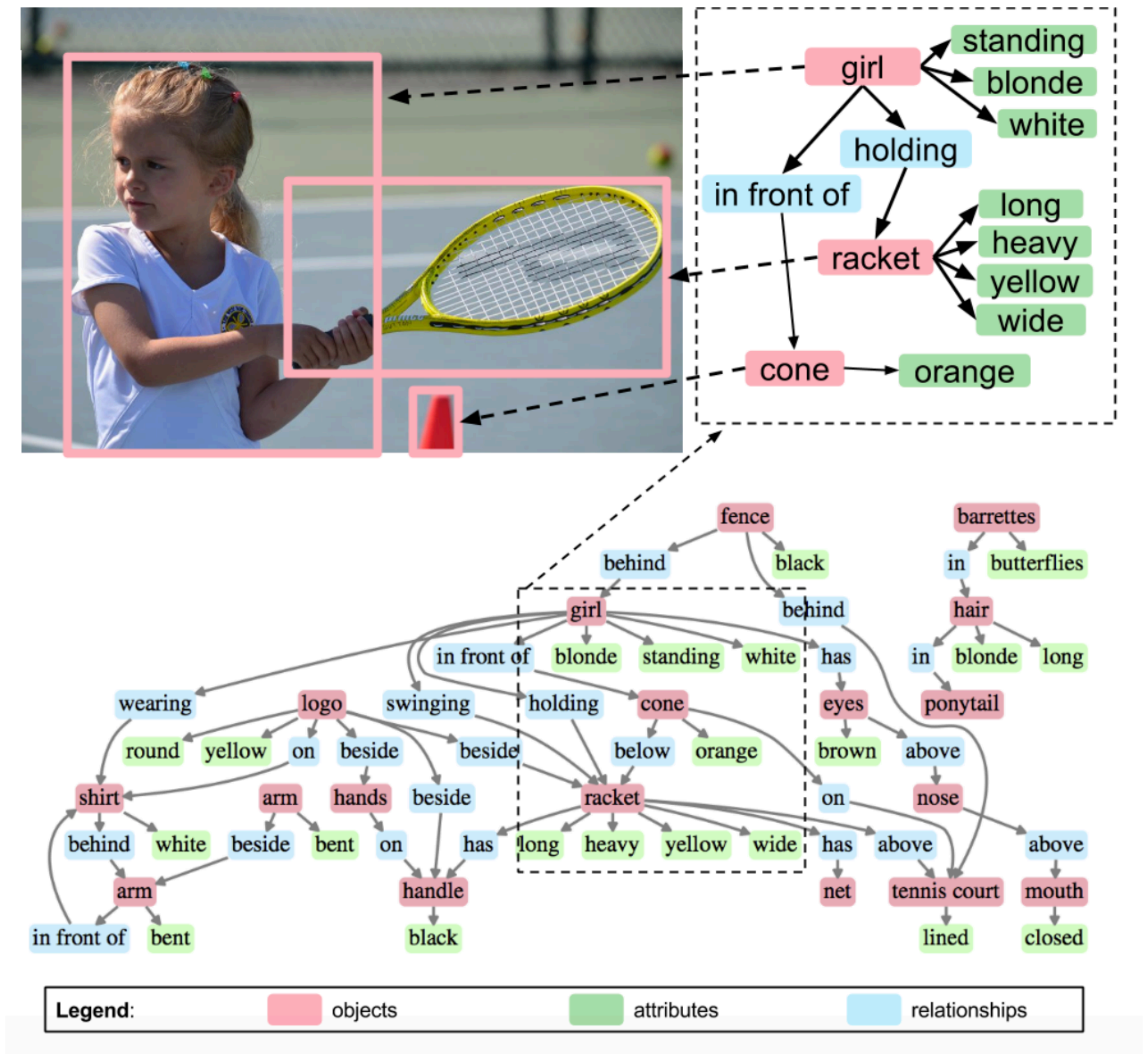
**Encoder:** States of a Markov Decision Process are mapped to the actions' values, using an encoder.

A typical run is as follows:

- Formulate the combinatorial problem, e.g., Max-Cut problem, as a Markov Decision Process.
- Encode states with a GNN model, i.e., every node has a vector representation encoded by a GNN.
- The agent is driven by an reinforcement learning algorithm, e.g., Monte-Carlo Tree Search, and makes decisions that move the environment to the next state, e.g., removing a vertex from a solution set.
- Once the parameters of the model have been trained, the agent is capable of searching the solutions for unseen instances of the problem.

## Computer Vision: Scene Graphs and Question Answering

# Scene Graphs



Scene graph from (Johnson et al., 2015).

## Scene Graphs

## Scene Graphs

Retrieving images/videos by describing their contents is an exciting application of computer vision.

## Scene Graphs

Retrieving images/videos by describing their contents is an exciting application of computer vision.

**Example**: A system may allow people to search for images by specifying not only objects (“man”, “boat”) but also structured relationships (“man on boat”) and attributes (“boat is white”) involving these objects.

## Scene Graphs

Retrieving images/videos by describing their contents is an exciting application of computer vision.

**Example**: A system may allow people to search for images by specifying not only objects (“man”, “boat”) but also structured relationships (“man on boat”) and attributes (“boat is white”) involving these objects.

These are structured queries! To solve this problem, a system must explicitly represent and reason about the objects, attributes, and relationships in images.

## Scene Graphs

Retrieving images/videos by describing their contents is an exciting application of computer vision.

**Example**: A system may allow people to search for images by specifying not only objects (“man”, “boat”) but also structured relationships (“man on boat”) and attributes (“boat is white”) involving these objects.

These are structured queries! To solve this problem, a system must explicitly represent and reason about the objects, attributes, and relationships in images.

A scene graph is a data structure that describes the contents of a scene. A scene graph encodes object instances, attributes of objects, and relationships between objects.

## Scene Graphs

Retrieving images/videos by describing their contents is an exciting application of computer vision.

**Example**: A system may allow people to search for images by specifying not only objects (“man”, “boat”) but also structured relationships (“man on boat”) and attributes (“boat is white”) involving these objects.

These are structured queries! To solve this problem, a system must explicitly represent and reason about the objects, attributes, and relationships in images.

A scene graph is a data structure that describes the contents of a scene. A scene graph encodes object instances, attributes of objects, and relationships between objects.

Scene graphs yield a rich representation of the given scene in an image.

## Scene Graphs

Retrieving images/videos by describing their contents is an exciting application of computer vision.

**Example**: A system may allow people to search for images by specifying not only objects (“man”, “boat”) but also structured relationships (“man on boat”) and attributes (“boat is white”) involving these objects.

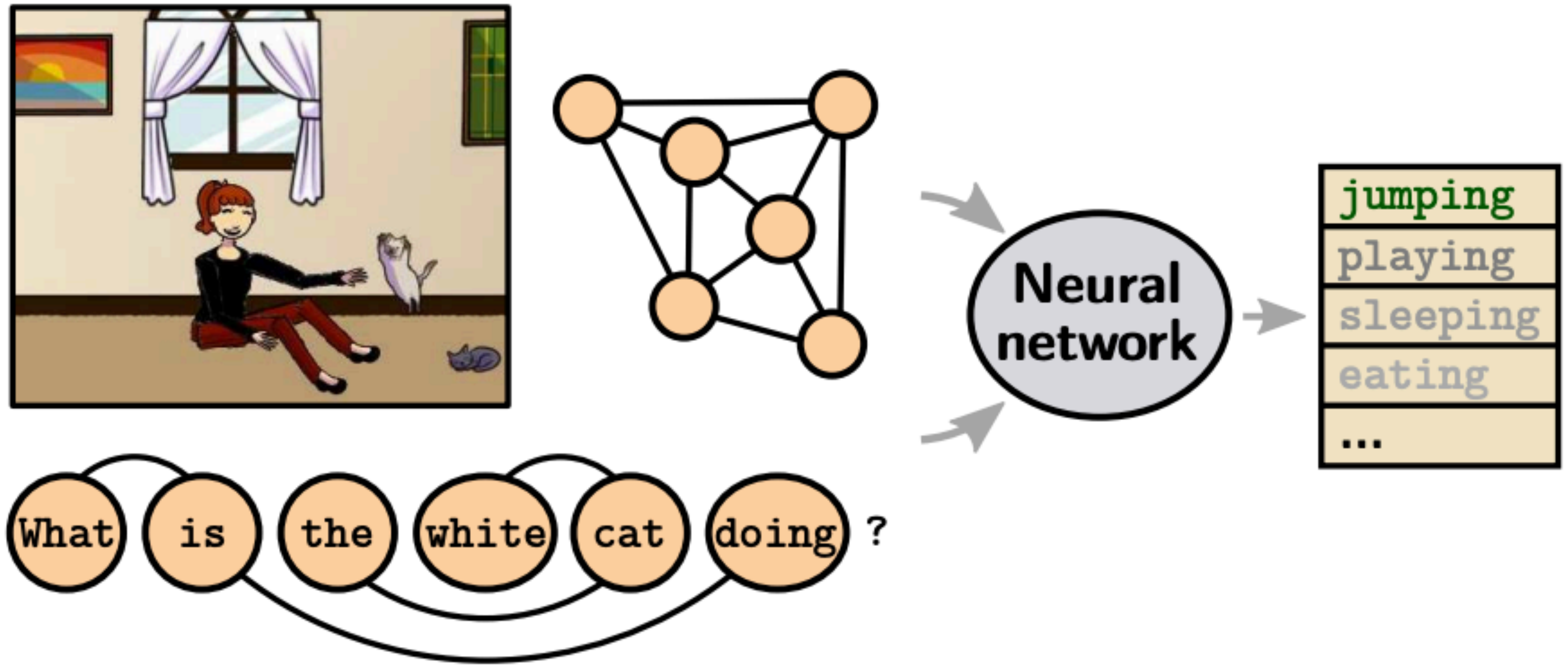
These are structured queries! To solve this problem, a system must explicitly represent and reason about the objects, attributes, and relationships in images.

A scene graph is a data structure that describes the contents of a scene. A scene graph encodes object instances, attributes of objects, and relationships between objects.

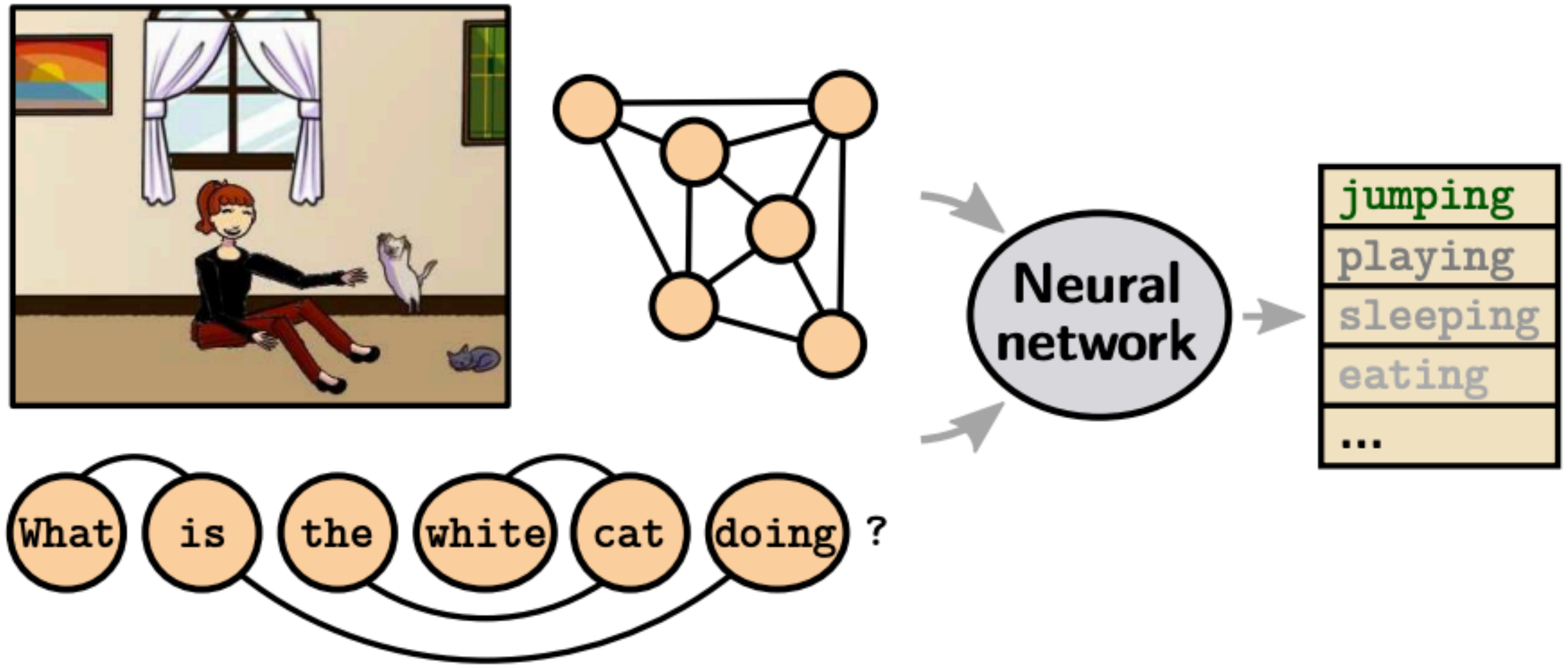
Scene graphs yield a rich representation of the given scene in an image.

Graph neural networks are used in both generating scene graphs and for high-level tasks that one would be interested in performing on them, e.g., visual question answering.

## Visual Question Answering



# Visual Question Answering



Encode the input scene as a graph representing the objects and their spatial arrangement. Encode the input question as a graph representing words and their syntactic dependencies. Train a neural network to reason over these representations, and to produce a suitable answer as a prediction. (Tenet et al., 2016)

# Visual Question Answering

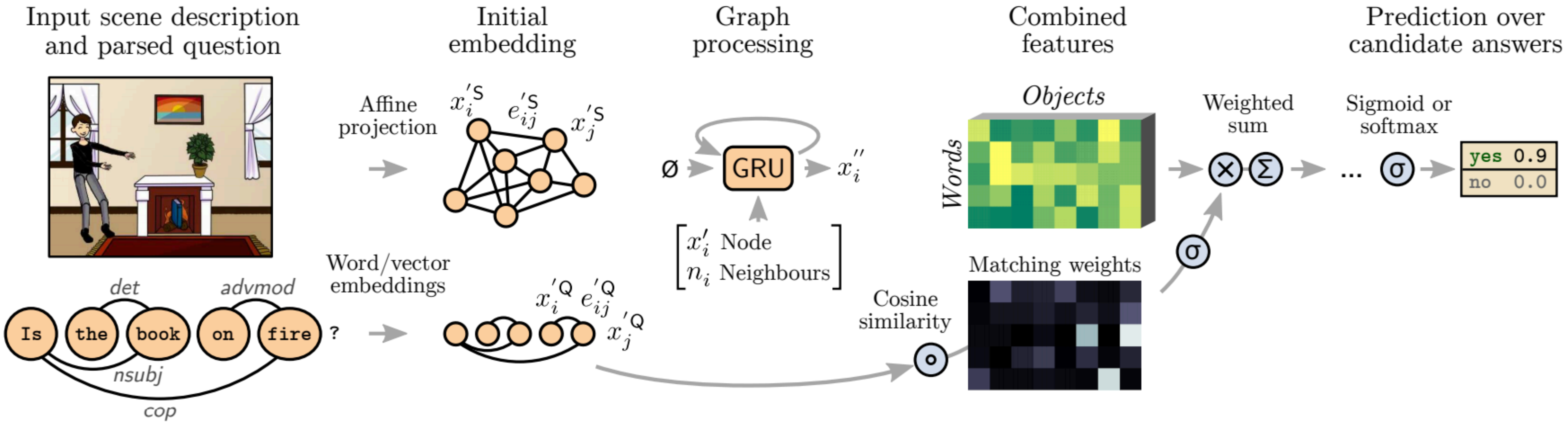


Figure from (Tenet et al., 2016) illustrating a pipeline for visual question answering using gated GNNs.

## Knowledge Graph Completion

## Beyond Unlabelled Graphs

## Beyond Unlabelled Graphs

Recall the base MPNN model:

$$\mathbf{h}_u^{(t)} = \sigma\left(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)}\sum_{v\in N(u)}\mathbf{h}_v^{(t-1)}\right),$$

where all neighbours are aggregated using a unique weight matrix.

## Beyond Unlabelled Graphs

Recall the base MPNN model:

$$\mathbf{h}_u^{(t)} = \sigma\left(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)} \sum_{v \in N(u)} \mathbf{h}_v^{(t-1)}\right),$$

where all neighbours are aggregated using a unique weight matrix.

This works if we focus on undirected graphs without edge labels, but knowledge graphs are more general, as contain a multiple types of relations between pairs of entities.

## Beyond Unlabelled Graphs

Recall the base MPNN model:

$$\mathbf{h}_u^{(t)} = \sigma\left(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)}\sum_{v\in N(u)}\mathbf{h}_v^{(t-1)}\right),$$

where all neighbours are aggregated using a unique weight matrix.

This works if we focus on undirected graphs without edge labels, but knowledge graphs are more general, as contain a multiple types of relations between pairs of entities.

GNNs are extended to the multi-relational setting to deal with multi-relational graphs.

## Beyond Unlabelled Graphs

Recall the base MPNN model:

$$\mathbf{h}_u^{(t)} = \sigma\Big(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)} \sum_{v \in N(u)} \mathbf{h}_v^{(t-1)}\Big),$$

where all neighbours are aggregated using a unique weight matrix.

This works if we focus on undirected graphs without edge labels, but knowledge graphs are more general, as contain a multiple types of relations between pairs of entities.

GNNs are extended to the multi-relational setting to deal with multi-relational graphs.

A natural way to encode multiple distinct relations in a graph is to transform neighbours based on which relation connects them to the original node.

## Beyond Unlabelled Graphs

Recall the base MPNN model:

$$\mathbf{h}_u^{(t)} = \sigma\Big(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)} \sum_{v \in N(u)} \mathbf{h}_v^{(t-1)}\Big),$$

where all neighbours are aggregated using a unique weight matrix.

This works if we focus on undirected graphs without edge labels, but knowledge graphs are more general, as contain a multiple types of relations between pairs of entities.

GNNs are extended to the multi-relational setting to deal with multi-relational graphs.

A natural way to encode multiple distinct relations in a graph is to transform neighbours based on which relation connects them to the original node.

This is the primary intuition behind rGCNs (Schlichtkrull et al., 2018).

## Knowledge Graph Completion with GNNs

# Knowledge Graph Completion with GNNs

Specifically, rGCNs (Schlichtkrull et al., 2018) builds on GNNs and incorporate relation-specific weight matrices in aggregation:

$$\mathbf{h}_u^{(t)} = \sigma\Big(\sum_{r \in \mathbf{R}} \sum_{v \in N(u)} \big(\frac{1}{c_{u,r}}\big)\mathbf{h}_v^{(t-1)}\mathbf{W}_r^{(t)} + \mathbf{h}_u^{(t-1)}\mathbf{W}_{self}^{(t)}\Big),$$

where $r \in \mathbf{R}$ is a relation, and $c_{u,r}$ is a problem-specific normalisation constant that can either be learned or chosen in advance.

# Knowledge Graph Completion with GNNs

Specifically, rGCNs (Schlichtkrull et al., 2018) builds on GNNs and incorporate relation-specific weight matrices in aggregation:

$$\mathbf{h}_u^{(t)} = \sigma\Big(\sum_{r \in \mathbf{R}} \sum_{v \in N(u)} (\frac{1}{c_{u,r}})\mathbf{h}_v^{(t-1)}\mathbf{W}_r^{(t)} + \mathbf{h}_u^{(t-1)}\mathbf{W}_{self}^{(t)}\Big),$$

where $r \in \mathbf{R}$ is a relation, and $c_{u,r}$ is a problem-specific normalisation constant that can either be learned or chosen in advance.

Note that rGCN applies to both for node/graph classification, as with standard GCNs, but also link prediction, i.e., KG completion.

# Knowledge Graph Completion with GNNs

Specifically, rGCNs (Schlichtkrull et al., 2018) builds on GNNs and incorporate relation-specific weight matrices in aggregation:

$$\mathbf{h}_u^{(t)} = \sigma\Big( \sum_{r \in \mathbf{R}} \sum_{v \in N(u)} (\frac{1}{c_{u,r}}) \mathbf{h}_v^{(t-1)} \mathbf{W}_r^{(t)} + \mathbf{h}_u^{(t-1)} \mathbf{W}_{self}^{(t)} \Big),$$

where $r \in \mathbf{R}$ is a relation, and $c_{u,r}$ is a problem-specific normalisation constant that can either be learned or chosen in advance.

Note that rGCN applies to both for node/graph classification, as with standard GCNs, but also link prediction, i.e., KG completion.

To do this, the node representations following message passing in rGCN are used as the entity embeddings of a KGC model, referred to as a ''decoder''. In this paper, rGCN uses DistMult.

# Knowledge Graph Completion with GNNs

Specifically, rGCNs (Schlichtkrull et al., 2018) builds on GNNs and incorporate relation-specific weight matrices in aggregation:

$$\mathbf{h}_u^{(t)} = \sigma\Big(\sum_{r\in\mathbf{R}}\sum_{v\in N(u)} (\frac{1}{c_{u,r}})\mathbf{h}_v^{(t-1)}\mathbf{W}_r^{(t)} + \mathbf{h}_u^{(t-1)}\mathbf{W}_{self}^{(t)}\Big),$$

where $r \in \mathbf{R}$ is a relation, and $c_{u,r}$ is a problem-specific normalisation constant that can either be learned or chosen in advance.

Note that rGCN applies to both for node/graph classification, as with standard GCNs, but also link prediction, i.e., KG completion.

To do this, the node representations following message passing in rGCN are used as the entity embeddings of a KGC model, referred to as a ''decoder''. In this paper, rGCN uses DistMult.

Note that rGCNs combine many aspects of this course: shallow KGC models and GNNs!

## Knowledge Graph Completion with GNNs

## Knowledge Graph Completion with GNNs

In practice, rGCNs performs usually worse than shallow tools. This is likely due to its embeddings incorporating information from across the entire knowledge graph, whereas existing models have dedicated embeddings for every entity.

# Knowledge Graph Completion with GNNs

In practice, rGCNs performs usually worse than shallow tools. This is likely due to its embeddings incorporating information from across the entire knowledge graph, whereas existing models have dedicated embeddings for every entity.

KBGAT introduces attention on top of multi-relational aggregation to allow nodes to attend to more relevant neighbours when making predictions, as opposed to aggregating from all neighbours, which is potentially harmful.

# Knowledge Graph Completion with GNNs

In practice, rGCNs performs usually worse than shallow tools. This is likely due to its embeddings incorporating information from across the entire knowledge graph, whereas existing models have dedicated embeddings for every entity.

KBGAT introduces attention on top of multi-relational aggregation to allow nodes to attend to more relevant neighbours when making predictions, as opposed to aggregating from all neighbours, which is potentially harmful.

GrAIL performs KGC by sampling a local subgraph around the 2 nodes of the link being predicted, and labels the nodes in the subgraphs based on their roles.

# Knowledge Graph Completion with GNNs

In practice, rGCNs performs usually worse than shallow tools. This is likely due to its embeddings incorporating information from across the entire knowledge graph, whereas existing models have dedicated embeddings for every entity.

KBGAT introduces attention on top of multi-relational aggregation to allow nodes to attend to more relevant neighbours when making predictions, as opposed to aggregating from all neighbours, which is potentially harmful.

GrAIL performs KGC by sampling a local subgraph around the 2 nodes of the link being predicted, and labels the nodes in the subgraphs based on their roles.

GrAIL then performs message passing in this subgraph to make predictions. Since it relies on a local subgraph, and introduces its own labels, it can be applied to new unseen entities.

## Summary

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin
- Particle physics: jet classification, event classification

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin
- Particle physics: jet classification, event classification
- Combinatorial optimisation & reasoning:

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin
- Particle physics: jet classification, event classification
- Combinatorial optimisation & reasoning:
  - Reasoning capacity of GNNs

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin
- Particle physics: jet classification, event classification
- Combinatorial optimisation & reasoning:
  - Reasoning capacity of GNNs
  - Reinforcement learning and GNNs

## Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin
- Particle physics: jet classification, event classification
- Combinatorial optimisation & reasoning:
  - Reasoning capacity of GNNs
  - Reinforcement learning and GNNs
- Computer vision: Scene graphs and visual question answering

# Summary

- Drug discovery: target identification, small molecule therapies, drug repurposing
- MPNNs and AI-assisted antibiotic discovery: Halicin
- Particle physics: jet classification, event classification
- Combinatorial optimisation & reasoning:
    - Reasoning capacity of GNNs
    - Reinforcement learning and GNNs
- Computer vision: Scene graphs and visual question answering
- Knowledge graphs: relation-specific message passing

## Summary of the Relational Learning Theme

## Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.

## Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.

# Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.

## Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.
- Lecture 4. **Message passing neural network architectures**: GCN, GGNN, GIN, GAT.

# Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.
- Lecture 4. **Message passing neural network architectures**: GCN, GGNN, GIN, GAT.
- Lecture 5. **Expressive power of message passing neural networks**: graph isomorphism, 1-WL equivalence, logical characterisation, limitations.

# Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.
- Lecture 4. **Message passing neural network architectures**: GCN, GGNN, GIN, GAT.
- Lecture 5. **Expressive power of message passing neural networks**: graph isomorphism, 1-WL equivalence, logical characterisation, limitations.
- Lecture 6. **Higher-order graph neural networks**: k-GNN, invariant/equivariant graph networks, PPGNs, homophile, heterophily.

# Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.
- Lecture 4. **Message passing neural network architectures**: GCN, GGNN, GIN, GAT.
- Lecture 5. **Expressive power of message passing neural networks**: graph isomorphism, 1-WL equivalence, logical characterisation, limitations.
- Lecture 6. **Higher-order graph neural networks**: k-GNN, invariant/equivariant graph networks, PPGNs, homophile, heterophily.
- Lecture 7. **Message passing neural networks and randomisation**: universality, permutation-invariance, evaluation.

## Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.
- Lecture 4. **Message passing neural network architectures**: GCN, GGNN, GIN, GAT.
- Lecture 5. **Expressive power of message passing neural networks**: graph isomorphism, 1-WL equivalence, logical characterisation, limitations.
- Lecture 6. **Higher-order graph neural networks**: k-GNN, invariant/equivariant graph networks, PPGNs, homophile, heterophily.
- Lecture 7. **Message passing neural networks and randomisation**: universality, permutation-invariance, evaluation.
- Lecture 8. **Applications of graph neural networks**: life sciences, combinatorial optimisation, computer vision, KGs.

## Summary of the Relational Learning Theme

- Lecture 1. **Relational data & node embeddings**: Model properties, inductive capacity, expressiveness, evaluation.
- Lecture 2. **Knowledge graph embedding models**: translational, bilinear, box embedding models, and beyond.
- Lecture 3. **Graph neural networks**: motivation, permutation-invariance, permutation-equivariance, message passing neural networks, generalisations, graph representation learning tasks.
- Lecture 4. **Message passing neural network architectures**: GCN, GGNN, GIN, GAT.
- Lecture 5. **Expressive power of message passing neural networks**: graph isomorphism, 1-WL equivalence, logical characterisation, limitations.
- Lecture 6. **Higher-order graph neural networks**: k-GNN, invariant/equivariant graph networks, PPGNs, homophile, heterophily.
- Lecture 7. **Message passing neural networks and randomisation**: universality, permutation-invariance, evaluation.
- Lecture 8. **Applications of graph neural networks**: life sciences, combinatorial optimisation, computer vision, KGs.
- Lecture 9. **Guest Lecture**: William Hamilton.

## Thanks!

## Good luck with your projects...

# References

- T. Gaudelet , B. Day , A. Jamasb, J. Soman , C. Regep , G. Liu1 , J. B. R. Hayter , R. Vickers , C. Roberts, J. Tang, D. Roblin, T. L. Blundell, M. M. Bronstein, and J. P. Taylor-King. Utilising Graph Machine Learning within Drug Discovery and Development, *arXiv*, 2021.
- Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackerman, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, James J. Collins. A Deep Learning Approach to Antibiotic Discovery. *Cell*, 2020; 180 (4): 688
- J. Shlomi, P. Battaglia, J. Vlimant. Graph Neural Networks in Particle Physics. *Machine Learning: Science and Technology*, 2021.
- A. Agarwal, A. Mangal,Vipul. Visual Relationship Detection using Scene Graphs: A Survey, *arXiv*, 2020.
- N. Mazyavkina , S. Sviridov, S. Ivanov and E. Burnaev. Reinforcement Learning for Combinatorial Optimization: A Survey, *arXiv*, 2020.

# References

- N. Choma, F. Monti, L. Gerhardt, T. Palczewski, Z. Ronaghi, M. Prabhat, W. Bhimji, M. Bronstein, S. Klein, J. Bruna. Graph Neural Networks for IceCube Signal Classification. *ICMLA*, 2018.
- D. Selsam, M. Lamm, B. Bunz, P. Liang, D.L. Dill, L. de Moura. Learning a SAT solver from single-bit supervision, *ICLR*, 2019.
- J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, L. Fei-Fei. Image Retrieval using Scene Graphs, *CVPR*, 2015.
- D. Teney, L. Liu, A. Hengel. Graph-Structured Representations for Visual Question Answering, *CVPR,* 2016.
- M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van den Berg, I. Titov, M. Welling, Modeling Relational Data with Graph Convolutional Networks, *ESWC*, 2018.