

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Faculty of Physics  
AI in Physics Research Group

---

**PP**

---

**Group A**

Vyron Arvanitis  
Emanuele Poggi  
Paul Hofmann  
Moritz Heidtmann  
Angelos Aslanidis

**Supervisor**

Nikolai Krug

July 8, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Theory</b>	<b>2</b>
2.1	Deep Sets . . . . .	2
2.2	Graph Neural Networks . . . . .	3
<b>3</b>	<b>Methodology</b>	<b>4</b>
<b>4</b>	<b>Results</b>	<b>4</b>
<b>5</b>	<b>Conclusion</b>	<b>4</b>

# 1 Introduction

In this lab course, we work with simulated data from the Belle II experiment to develop a fast and efficient simulation pipeline. The goal is to train and compare machine learning models that can predict early in the simulation chain whether an event is likely to survive later selection stages. This allows us to discard unimportant background events early, saving computational resources typically spent on full detector simulation and reconstruction (processes which are far more computationally expensive). The strategy is illustrated in Fig. 1, where a neural network acts as a pre-filter to speed up the overall simulation process.

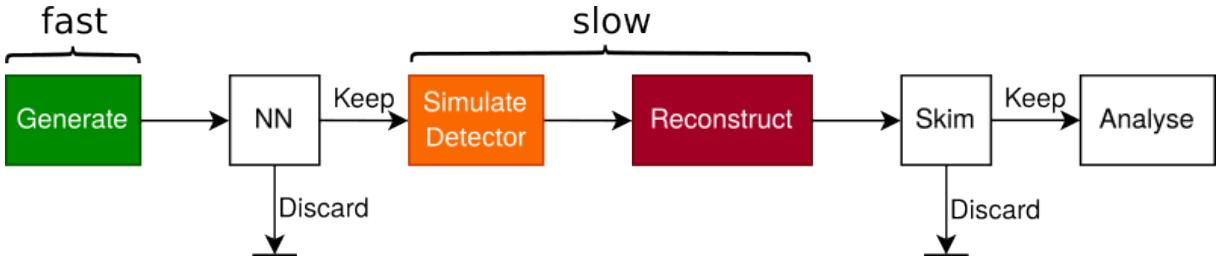


Figure 1: Smart simulation strategy: a neural network filters generated events before expensive detector simulation. Figure adapted from the official LMU AI Lab repository [1].

## 2 Theory

### 2.1 Deep Sets

In many physical systems, such as collections of particles or events in high-energy physics, the data can be naturally represented as unordered sets. Traditional neural networks, however, are not inherently permutation-invariant. This means that they assume an order in their input. To address this, Deep Sets provide a principled way to process such set-structured data while maintaining invariance to the order of elements. Figure 2 visualizes the structure of a Deep Set model.

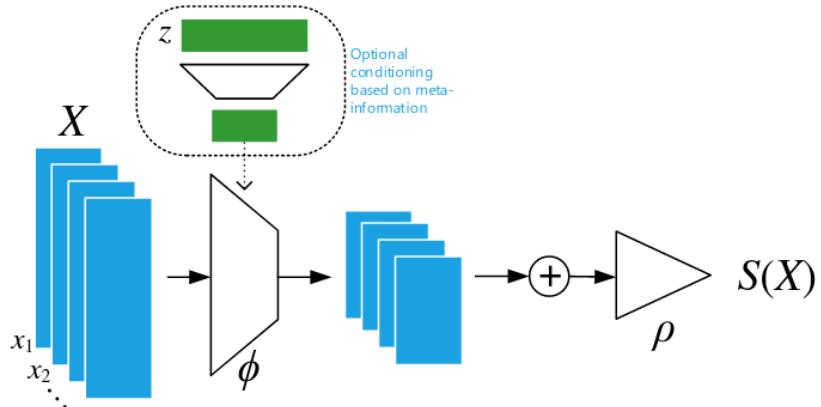


Figure 2: Schematic illustration of a Deep Set model.

A Deep Set model operates by applying a shared function  $\phi$  independently to each element in the

set. This produces a new set of embeddings, which are then aggregated using a permutation-invariant operation such as sum, mean, or max. The aggregated result is passed through a second function  $\rho$ , typically implemented as a multilayer perceptron (MLP), to produce the final output:

$$f(X) = \rho \left( \sum_{x \in X} \phi(x) \right) \quad (1)$$

This structure ensures that the model respects the symmetry of sets. The theoretical foundation of Deep Sets is discussed in Zaheer et al., 2017, where it is shown that any function on a set that is invariant to permutations can be decomposed in this form.

In our project, we use this architecture to model collections of particles, where the order of particles is physically irrelevant. The Deep Set model thus provides a natural and efficient baseline for learning from such data.

## 2.2 Graph Neural Networks

In contrast to Deep Set models, which treat the input as an unordered collection, Graph Neural Networks (GNNs) explicitly leverage relational information between data points. This is particularly important in our context, where particles are connected through a decay tree structure that naturally forms a graph. Each node in the graph represents a particle, and edges represent physical relationships (e.g., parent-child connections in the decay chain).

In terms of Deep Sets, graph convolutions can be seen as permutation-equivariant operations. As long as the aggregation function (e.g., sum, mean, or max) is symmetric, the update remains invariant to the ordering of neighbors.

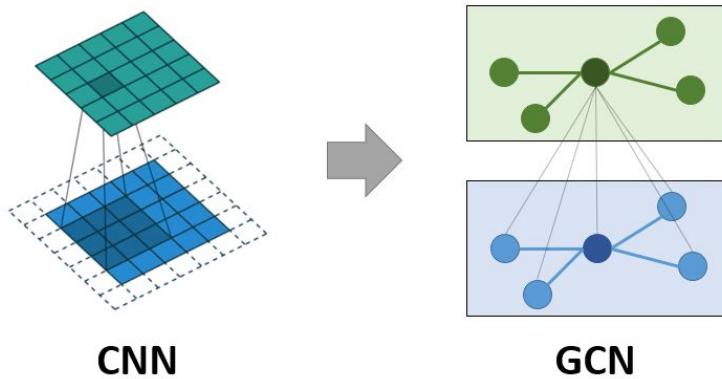


Figure 3: Analogy between CNNs and GCNs. In CNNs, pixels are updated based on spatial neighborhoods; in GCNs, nodes are updated based on graph neighborhoods.

Figure 3 illustrates the analogy between Convolutional Neural Networks (CNNs) and Graph Convolutional Networks (GCNs). In CNNs, pixels are updated based on their spatial neighborhoods, while in GCNs, nodes are updated based on their graph neighborhoods. This allows

GNNs to capture complex relationships and dependencies in the data, making them particularly suitable for tasks involving structured data like particle decays.

### **3 Methodology**

### **4 Results**

### **5 Conclusion**

## References

- [1] Nikolai Krug. Preparation materials for particle physics lab group a. <https://gitlab.physik.uni-muenchen.de/ai-lab/ss25/pp-lab-group-a/-/blob/main/preparation.md>, 2025. Accessed: July 8, 2025.