# Advanced Mini-Batching

## Tutorial 4

Antonio Longa

https://antoniolonga.github.io/

# TABLE OF CONTENTS

01 Introduction to batch

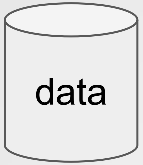
02 Batching

03 Batches and graphs

04 Advance mini-batching in PyTorch Geometric

05 Conclusion

# 01 Introduction to batching



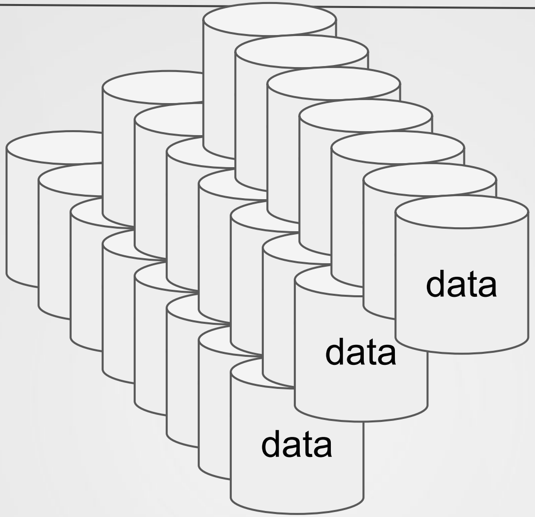
### TALKING ABOUT DATA..

- In DL more (data) is better…

# 01 Introduction to batching

**TALKING ABOUT DATA..**

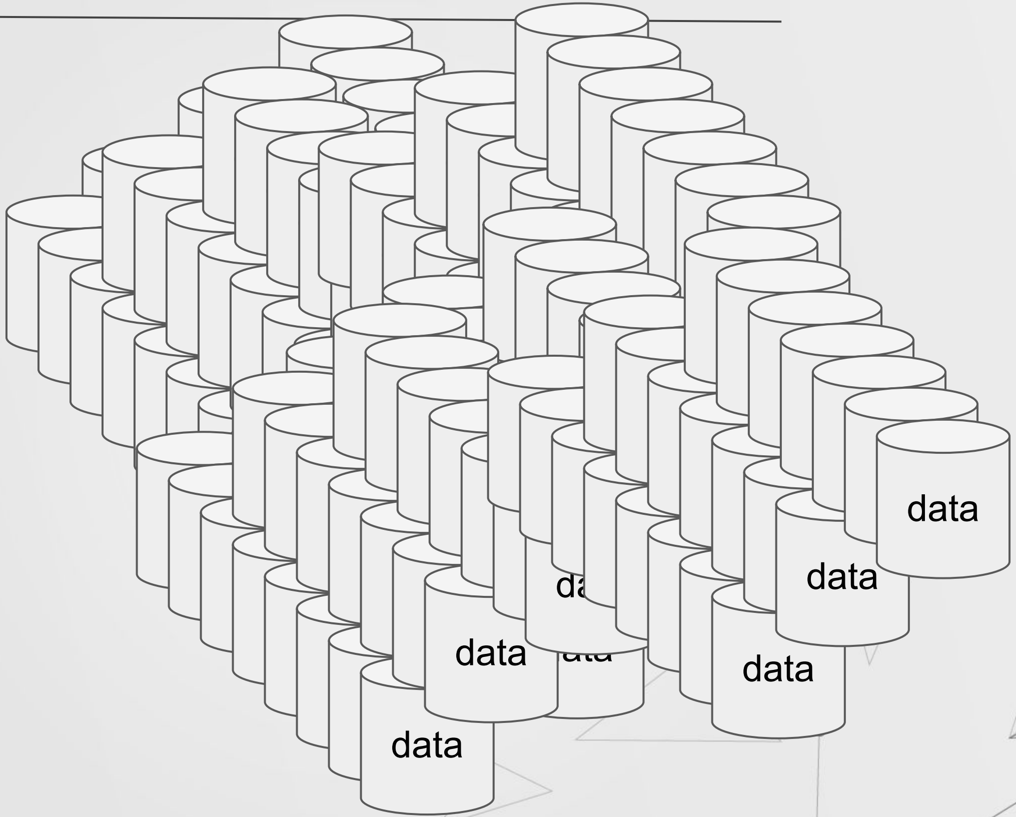
- In DL more (data) is better… **TRUE**



# 01 Introduction to batching

**TALKING ABOUT DATA..**

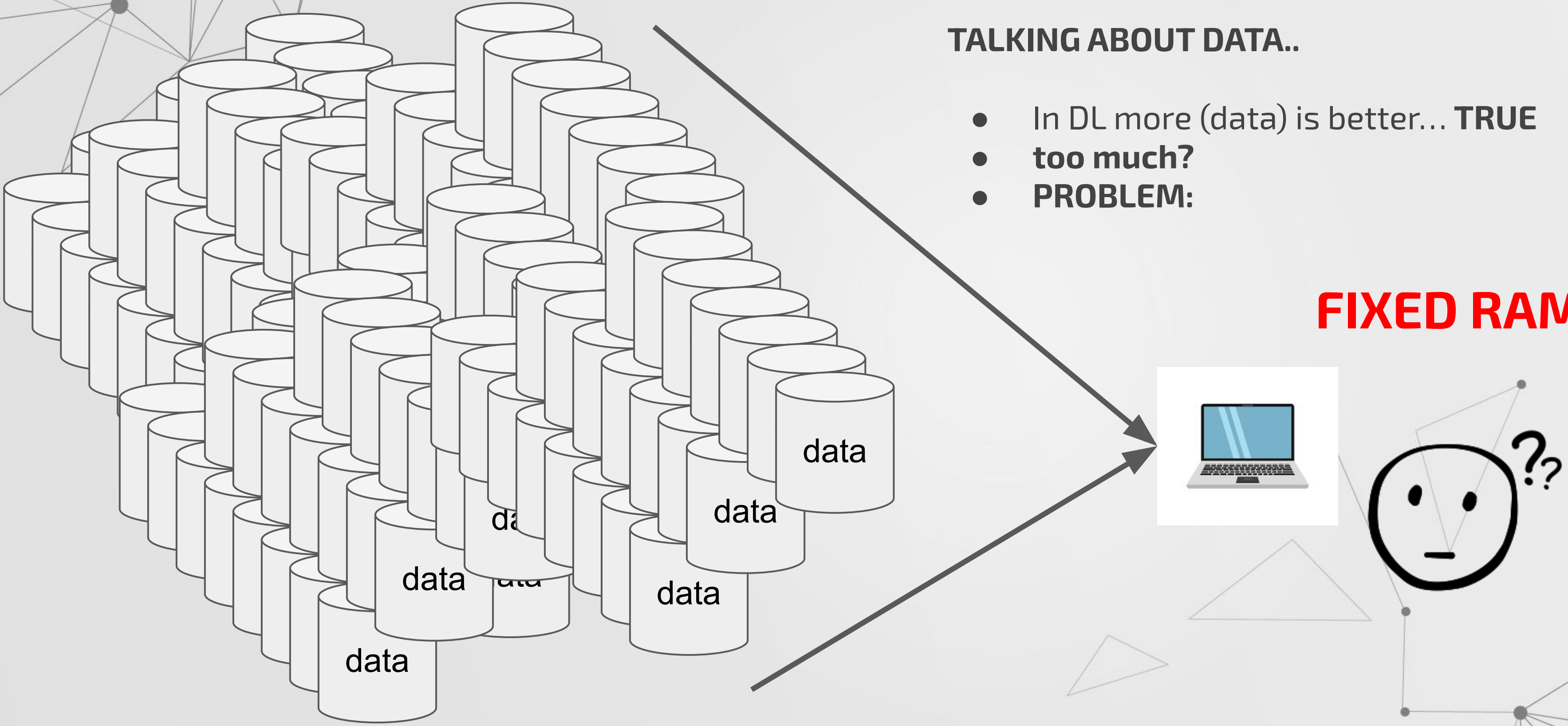
- In DL more (data) is better… **TRUE**
- **too much?**



# 01 Introduction to batching

## TALKING ABOUT DATA..

- In DL more (data) is better… **TRUE**
- **too much?**
- **PROBLEM:**

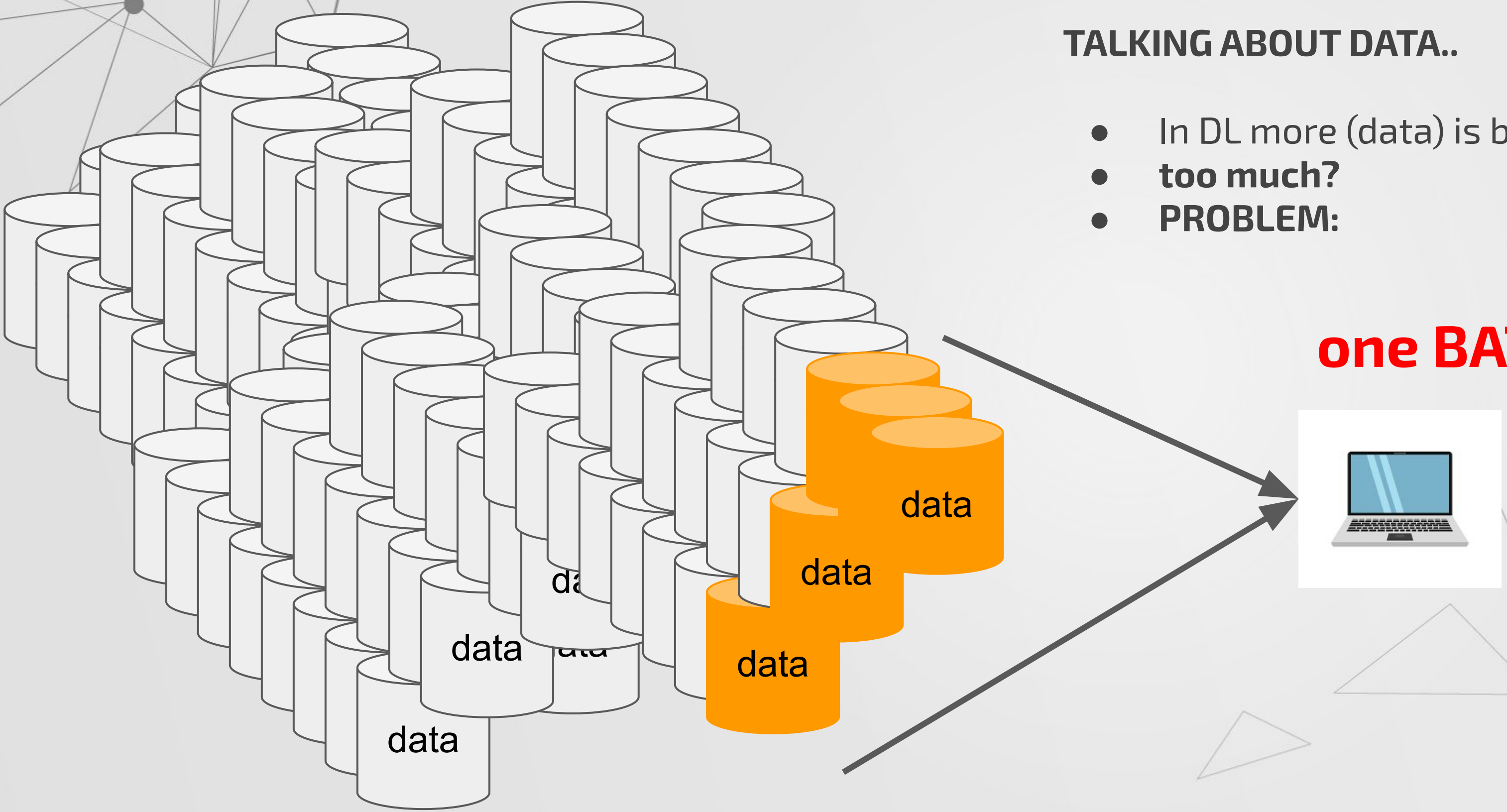


**FIXED RAM**

# 01 Introduction to batching

**TALKING ABOUT DATA..**

- In DL more (data) is better… **TRUE**
- **too much?**
- **PROBLEM:**

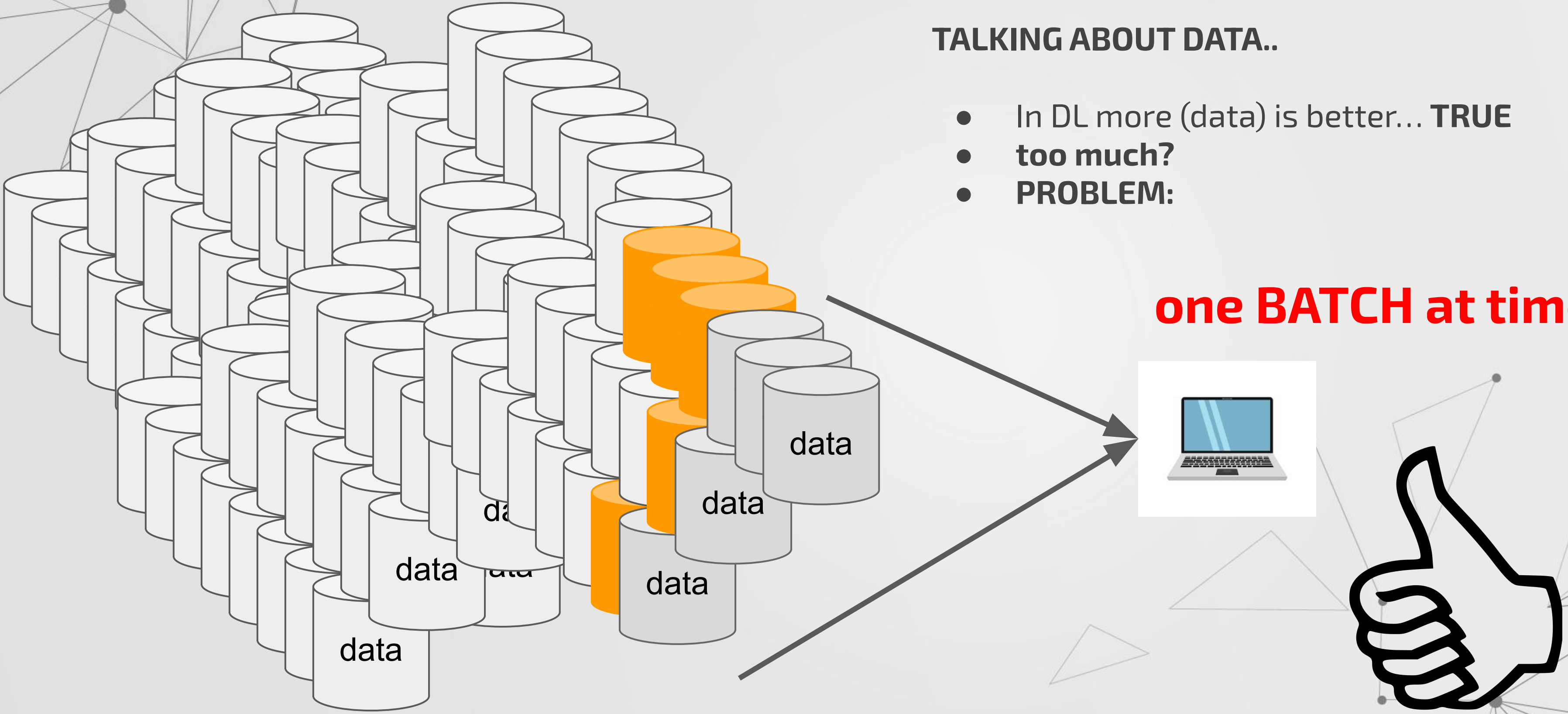


**one BATCH at time**

# 01 Introduction to batching

## TALKING ABOUT DATA..

- In DL more (data) is better… **TRUE**
- **too much?**
- **PROBLEM:**



**one BATCH at time**

# 02 Batching

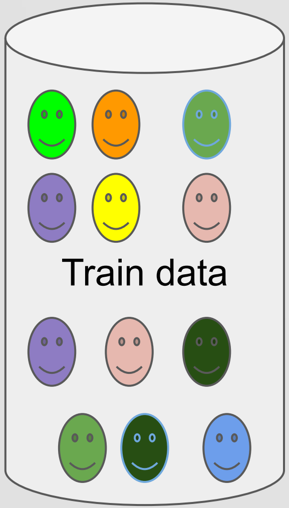
A **Batch** refers to the set of training samples used in one iterations.

# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

**EXAMPLE:**

The Train data contains 12 samples ( ☺ ).

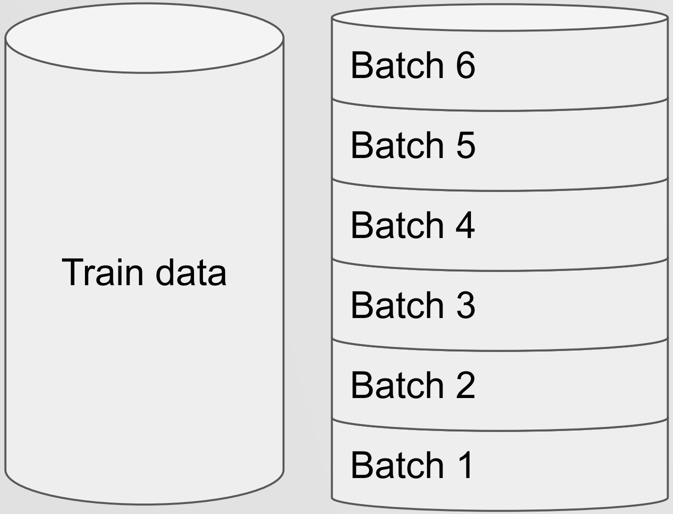


# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

**EXAMPLE:**

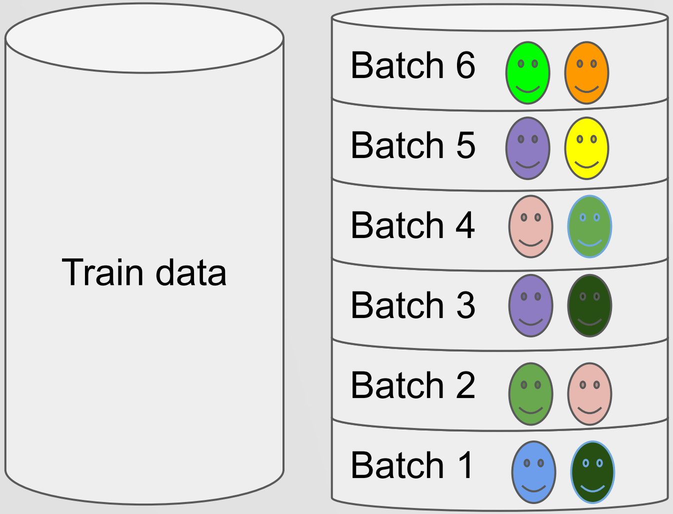
The Train data contains 12 samples ( 🙂 ). We split the dataset into 6 batches ( ⬭ ).



# 02 Batching

A **Batch** refers to the  set of training samples used in one iterations.

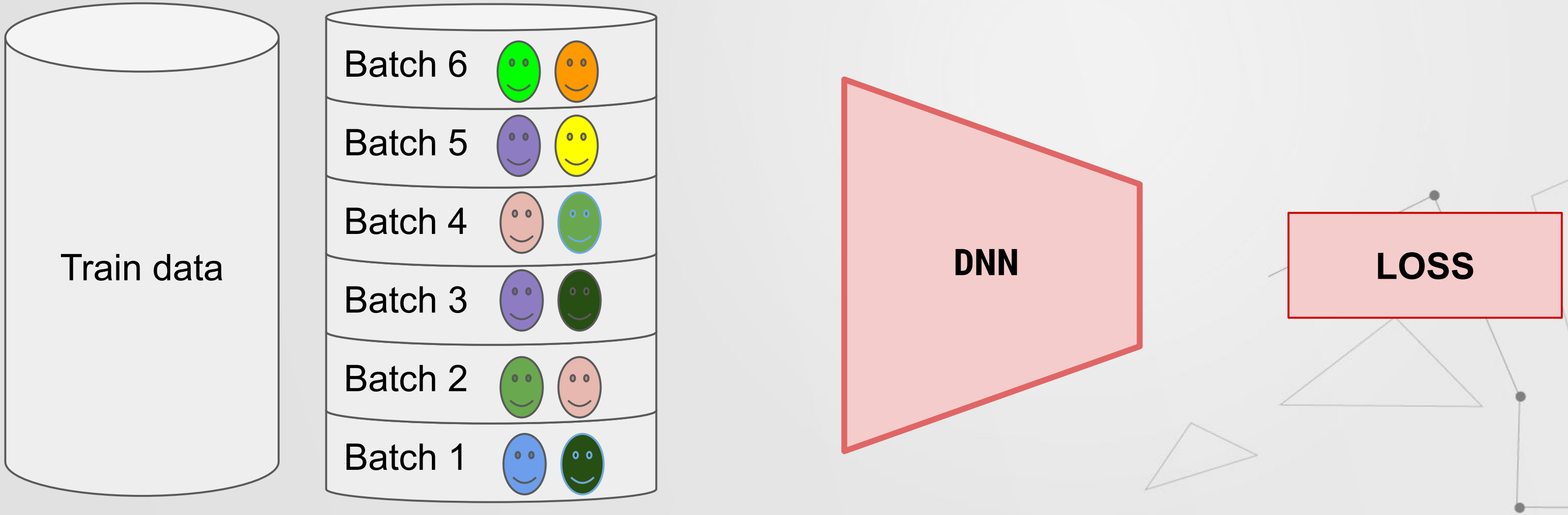
**EXAMPLE:**
The Train data contains 12 samples ( ☺ ). We split the dataset into 6 batches ( ▭ ).
Each batch contains 2 samples.



# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

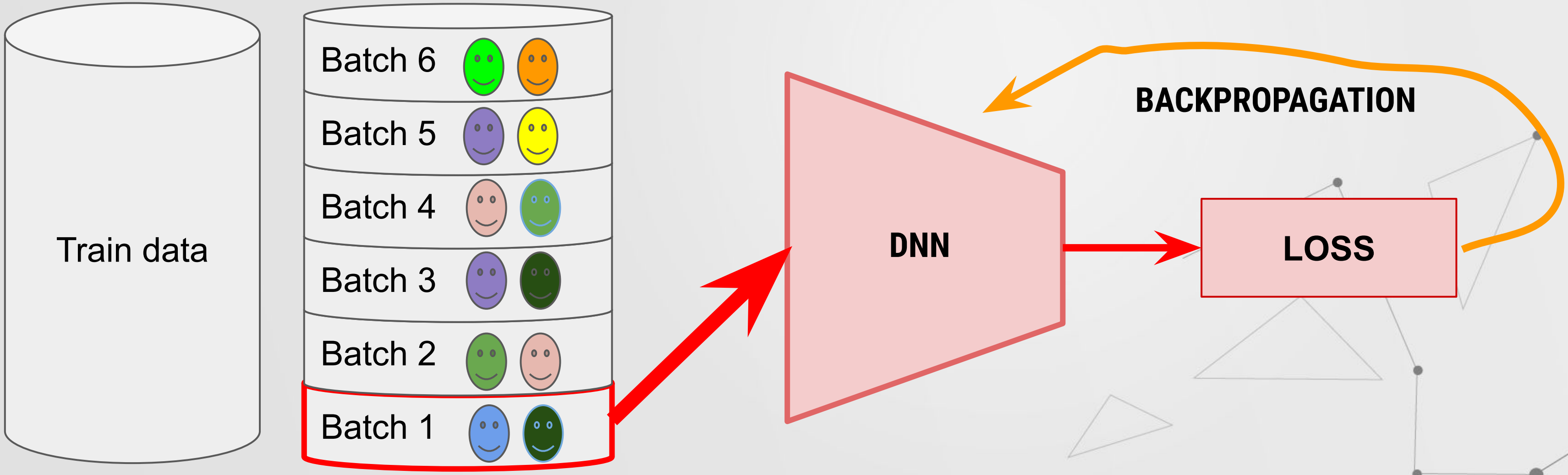
**EXAMPLE:**
The Train data contains 12 samples ( ☺ ). We split the dataset into 6 batches ( ▭ ).
Each batch contains 2 samples.



# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

**EXAMPLE:**
The Train data contains 12 samples ( ☺ ). We split the dataset into 6 batches ( ▭ ).
Each batch contains 2 samples.

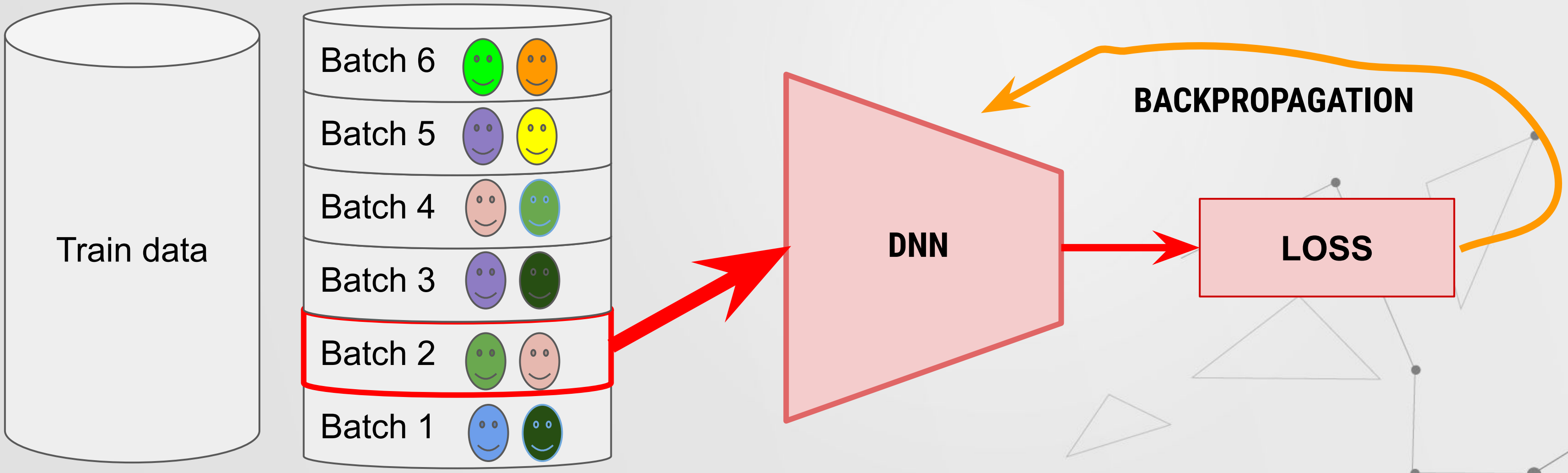


# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

**EXAMPLE:**

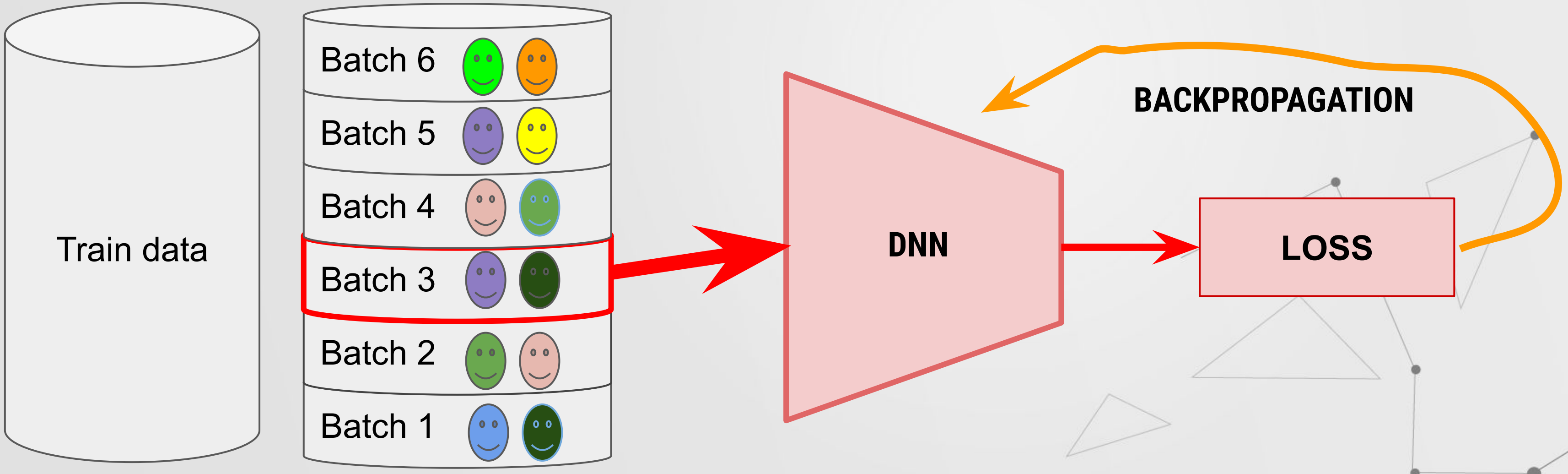
The Train data contains 12 samples ( ☺ ). We split the dataset into 6 batches ( ▭ ).
Each batch contains 2 samples.



# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

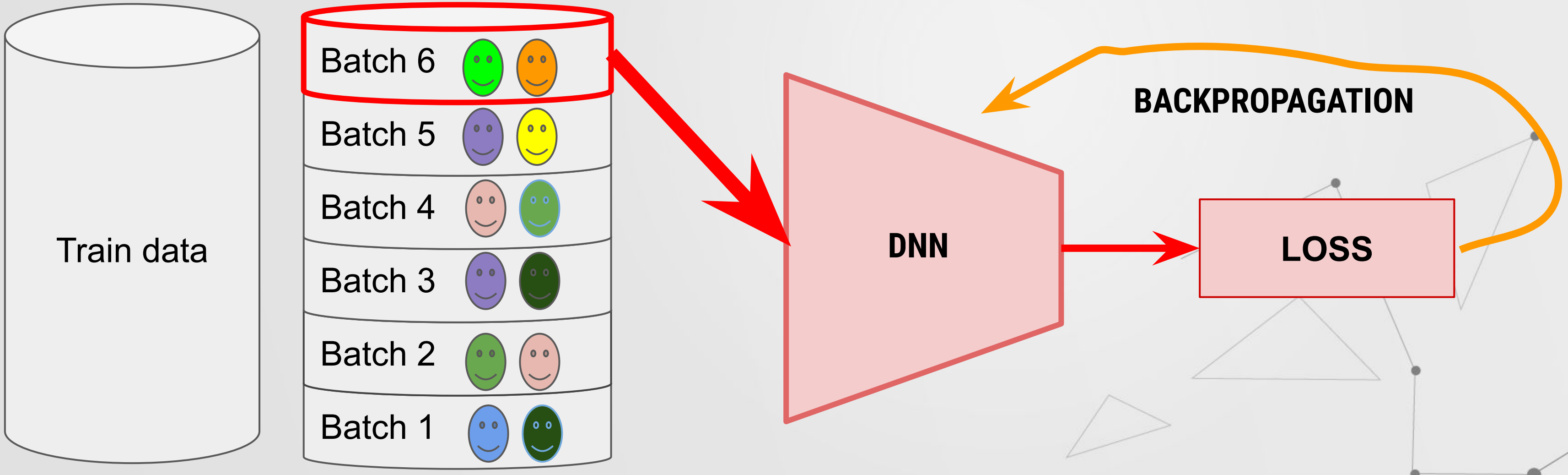
**EXAMPLE:**
The Train data contains 12 samples ( ☺ ). We split the dataset into 6 batches ( ⬭ ).
Each batch contains 2 samples.



# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

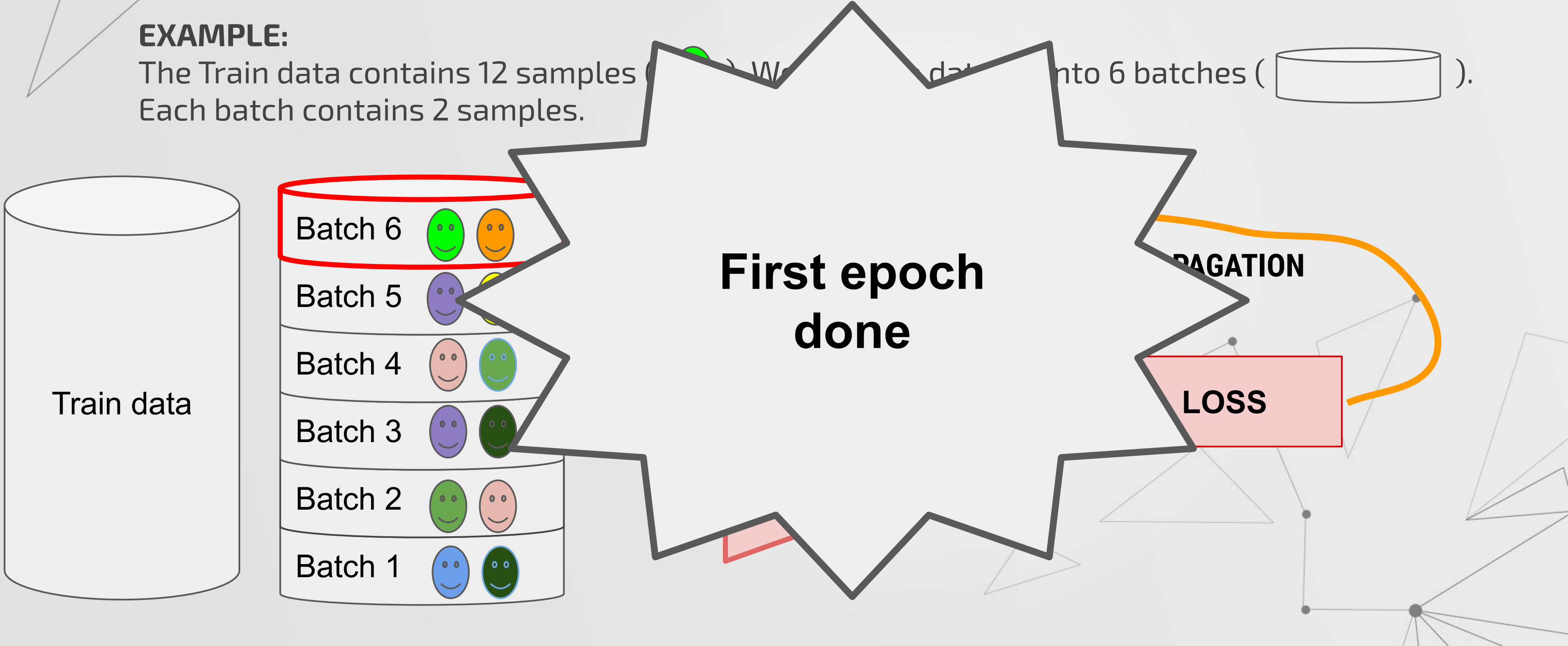
**EXAMPLE:**
The Train data contains 12 samples ( ☺ ). We split the dataset into 6 batches ( ⬭ ).
Each batch contains 2 samples.



# 02 Batching

A **Batch** refers to the set of training samples used in one iterations.

**EXAMPLE:**
The Train data contains 12 samples ( ). W data nto 6 batches ( ).
Each batch contains 2 samples.



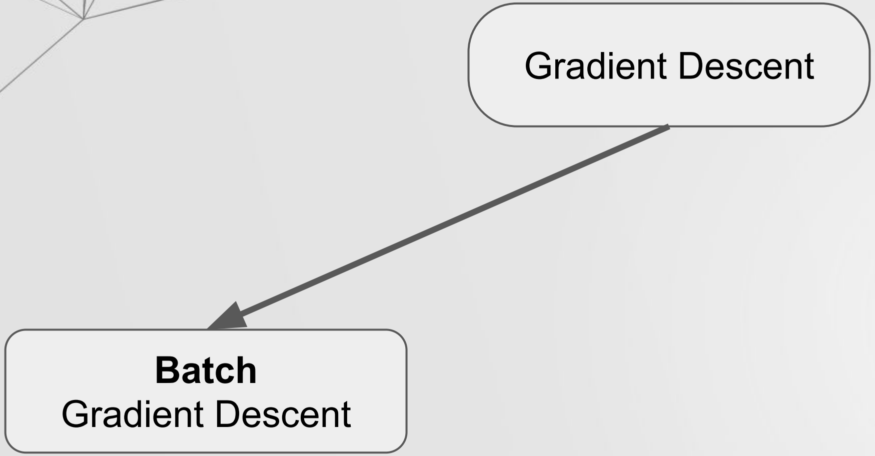
# 02 Batching

Gradient Descent

Optimization algorithm to train machine learning algorithms.

# 02 Batching

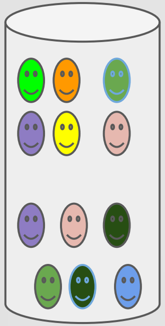
Optimization algorithm to train machine learning algorithms.



Gradient Descent

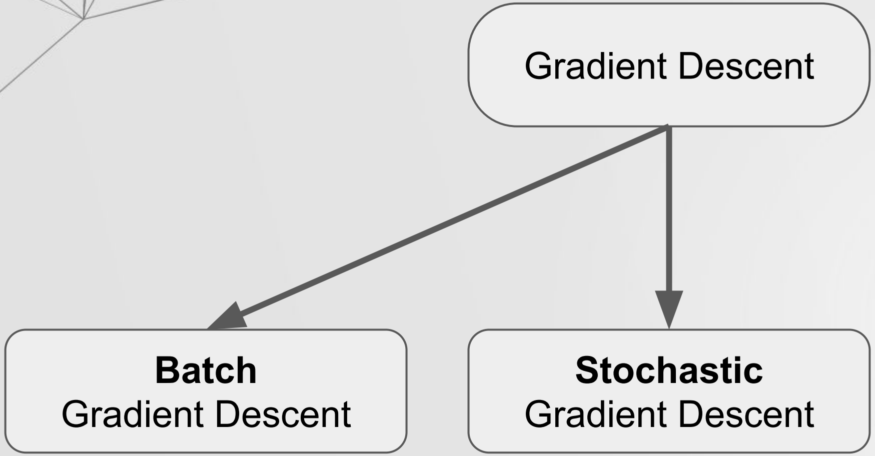
**Batch**
Gradient Descent

Batch size equal to train set size. all the samples in one unique batch



# 02 Batching

Optimization algorithm to train machine learning algorithms.

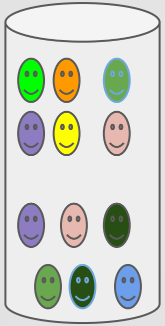


Gradient Descent

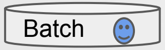
**Batch** Gradient Descent

**Stochastic** Gradient Descent

Batch size equal to train set size. all the samples in one unique batch



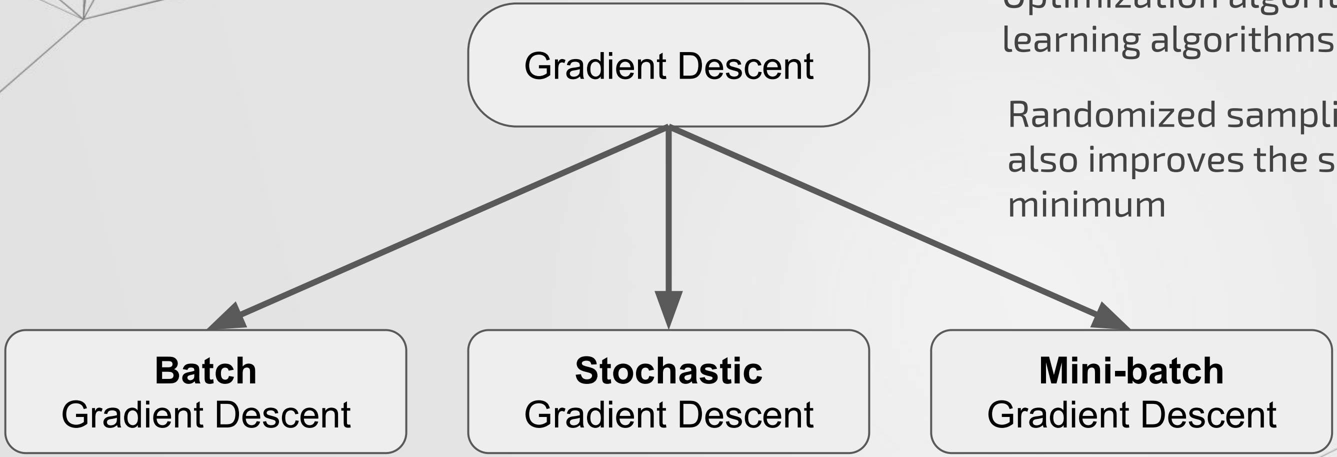
Batch size equal to 1. 1 sample for each batch



# 02 Batching

Optimization algorithm to train machine learning algorithms.

Randomized sampling over the training set also improves the search for a good minimum



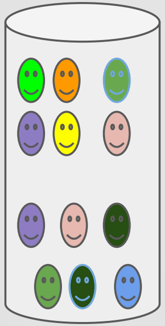
Gradient Descent

**Batch** Gradient Descent

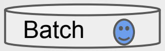
**Stochastic** Gradient Descent

**Mini-batch** Gradient Descent

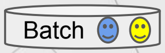
Batch size equal to train set size. all the samples in one unique batch



Batch size equal to 1. 1 sample for each batch



Batch size equal to X. X samples for each batch

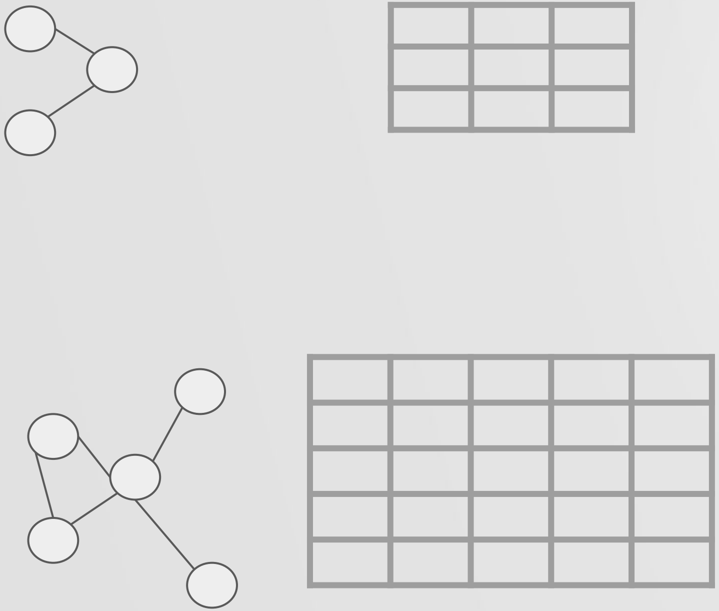


# 03 Batches and graphs

- Each graph has a different number of nodes.

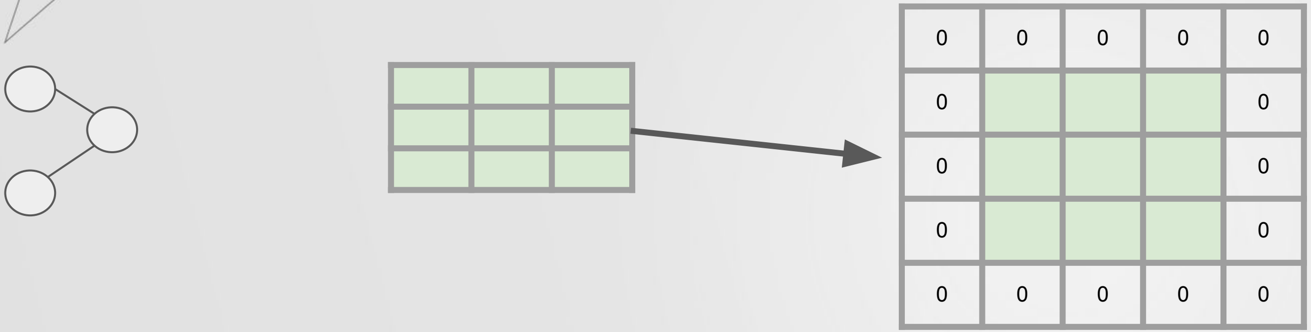
# 03 Batches and graphs

- Each graph has a different number of nodes.

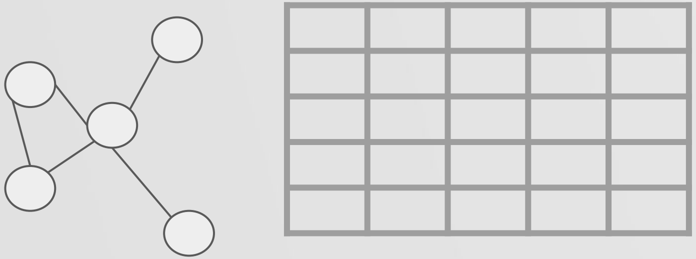


# 03 Batches and graphs

- Each graph has a different number of nodes.
  → pad our adj



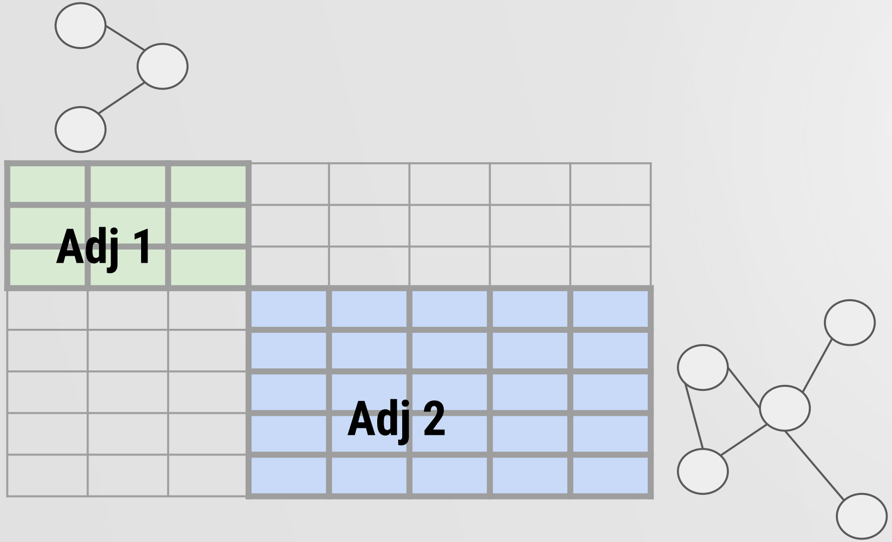
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | | | | 0 |
| 0 | | | | 0 |
| 0 | | | | 0 |
| 0 | 0 | 0 | 0 | 0 |



**not good!**
**too unnecessary memory consumption.**

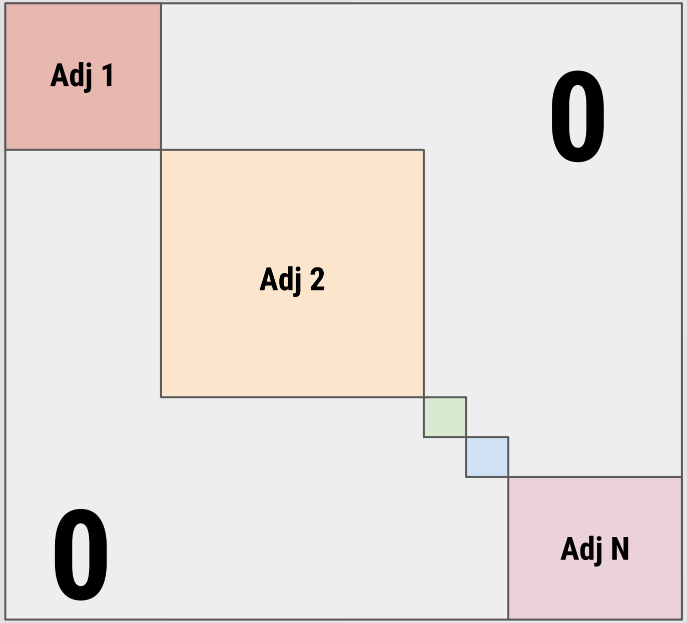
# 03 Batches and graphs

- Each graph has a different number of nodes.
  → pad our adj
  → **build a “giant” matrices**



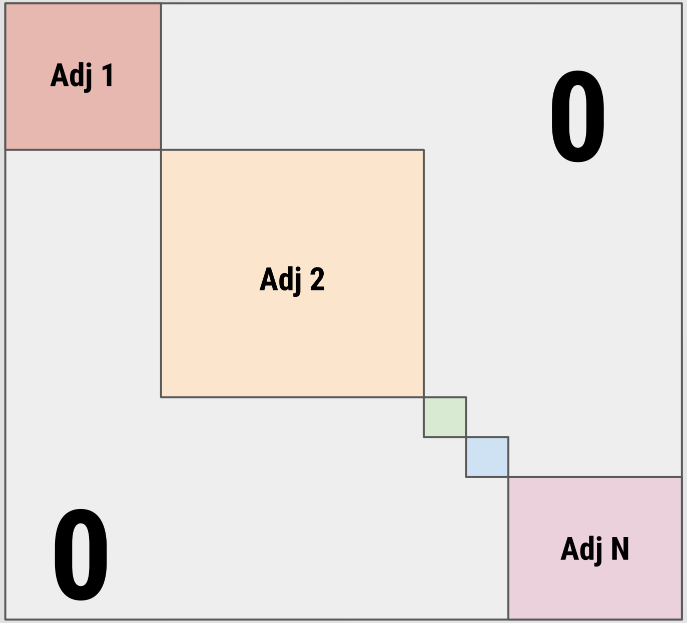
# 03 Batches and graphs

- Each graph has a different number of nodes.
  → pad our adj
  → **build a "giant" matrices**



# 03 Batches and graphs

- Each graph has a different number of nodes.
  → pad our adj
  → **build a “giant” matrices**



Adj 1

0

Adj 2

0

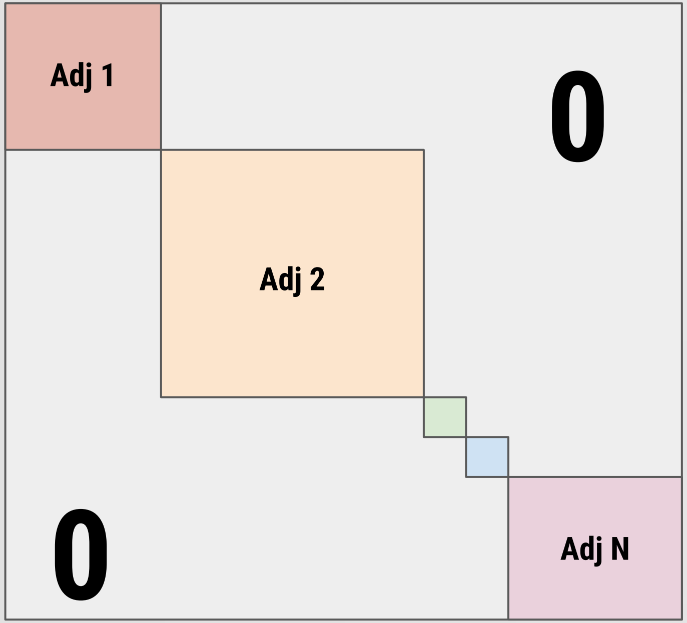
Adj N

**PRO:**

- Message passing do not require changes (disconnected graphs, in a giant graph)

# 03 Batches and graphs

- Each graph has a different number of nodes.
  → pad our adj
  → **build a "giant" matrices**



**PRO:**

- Message passing do not require changes (disconnected graphs, in a giant graph)
- too many zeros → **SPARSE MATRICES**

# 03 Batches and graphs

### DENSE

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

# 03 Batches and graphs

### DENSE

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

### SPARSE

| | | | |
|---|---|---|---|
| **ROW** | 1 | 2 | 3 |
| **COLUMN** | 2 | 4 | 0 |
| **VALUE** | 1 | 1 | 1 |

# 03 Batches and graphs

### DENSE

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

**25 elements stored**

### SPARSE

| | | | |
|---|---|---|---|
| **ROW** | 1 | 2 | 3 |
| **COLUMN** | 2 | 4 | 0 |
| **VALUE** | 1 | 1 | 1 |

**9 elements stored**

# 04 Advance mini-batching in PyTorch Geometric

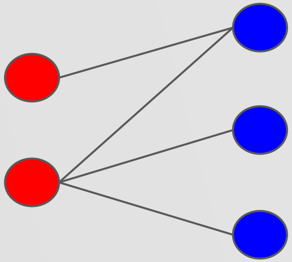
Don’t worry, all the work it is done by:

```
torch_geometric.loader.DataLoader
```

let see this basics on Jupyter :)

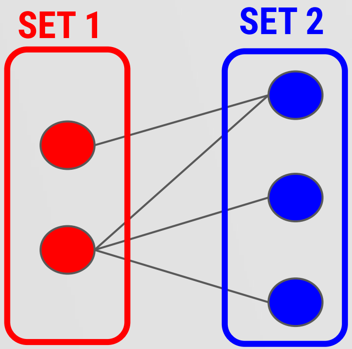
# 04 Advance mini-batching in PyTorch Geometric

BIPARTITE GRAPHS



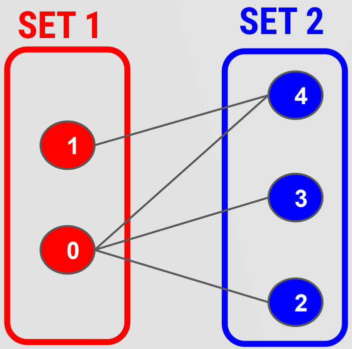
# 04 Advance mini-batching in PyTorch Geometric

BIPARTITE GRAPHS



# 04 Advance mini-batching in PyTorch Geometric

## BIPARTITE GRAPHS



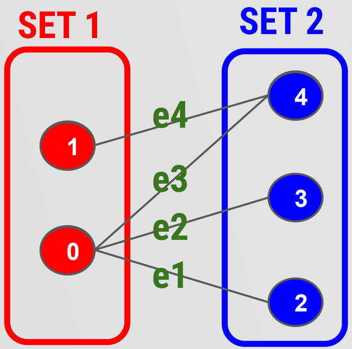
**Adj**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

5x5

# 04 Advance mini-batching in PyTorch Geometric

BIPARTITE GRAPHS



**Adj**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

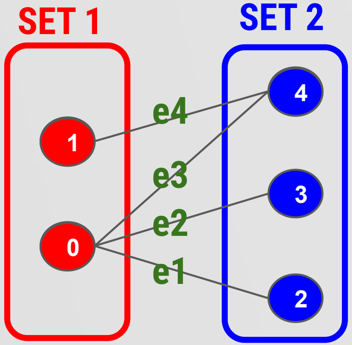
5x5

**edge list**

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 2 | 3 | 4 | 4 |

# 04 Advance mini-batching in PyTorch Geometric

BIPARTITE GRAPHS



**Adj**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

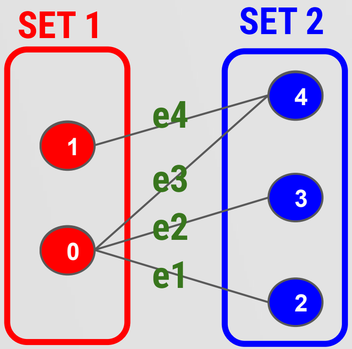
5x5

**edge list**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | SET 1 |
| 2 | 3 | 4 | 4 | SET 2 |

# 04 Advance mini-batching in PyTorch Geometric

BIPARTITE GRAPHS



**Adj**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

5x5

**edge list**

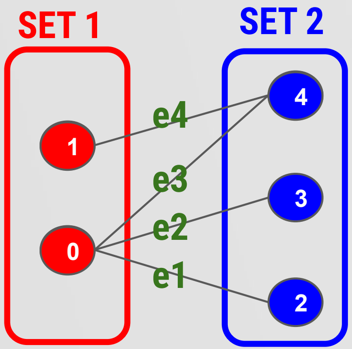
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | SET 1 |
| 2 | 3 | 4 | 4 | SET 2 |

**edge_index**

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | | | |

# 04 Advance mini-batching in PyTorch Geometric

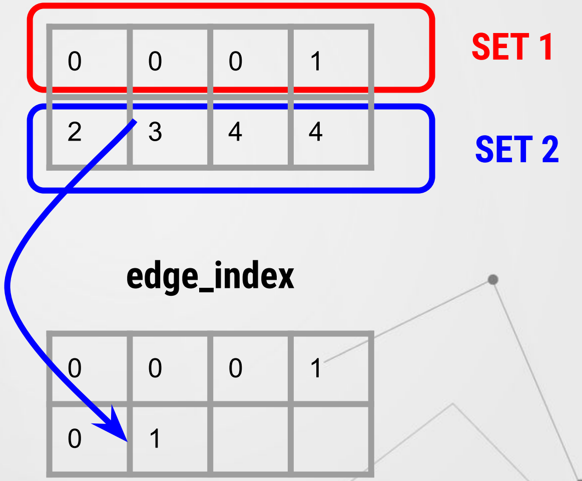
BIPARTITE GRAPHS



**Adj**

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

5x5



**edge list**

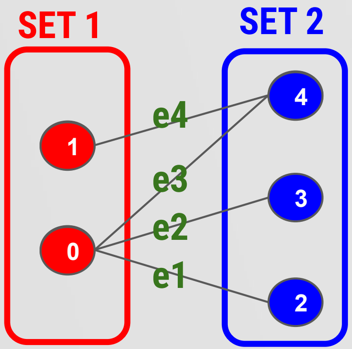
|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | SET 1 |
| 2 | 3 | 4 | 4 | SET 2 |

**edge_index**

|   |   |   |   |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 |   |   |

# 04 Advance mini-batching in PyTorch Geometric

## BIPARTITE GRAPHS



**Adj**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

5x5

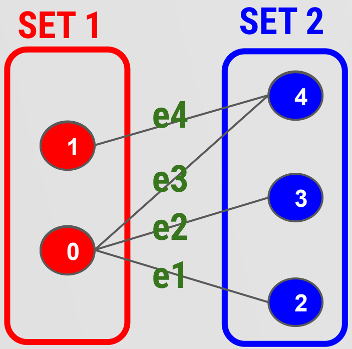
**edge list**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | SET 1 |
| 2 | 3 | 4 | 4 | SET 2 |

**edge_index**

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 2 | |

BIPARTITE GRAPHS



**Adj**

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

5x5

**edge list**

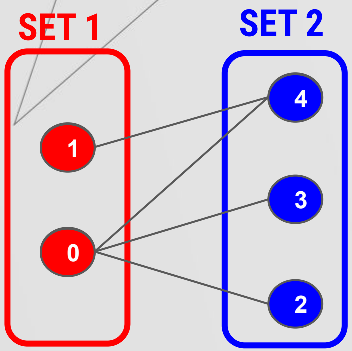
|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | SET 1 |
| 2 | 3 | 4 | 4 | SET 2 |

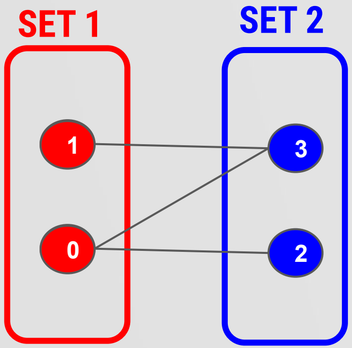
**edge_index**

|   |   |   |   |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 2 | 2 |

# 04 Advance mini-batching in PyTorch Geometric

Multiple bipartite graphs



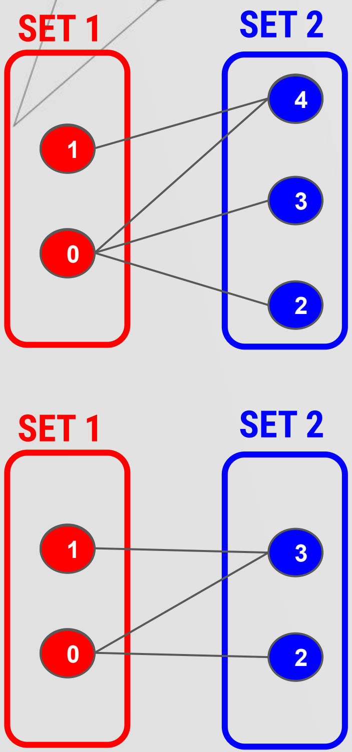


**edge_index**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

# 04 Advance mini-batching in PyTorch Geometric

Multiple bipartite graphs



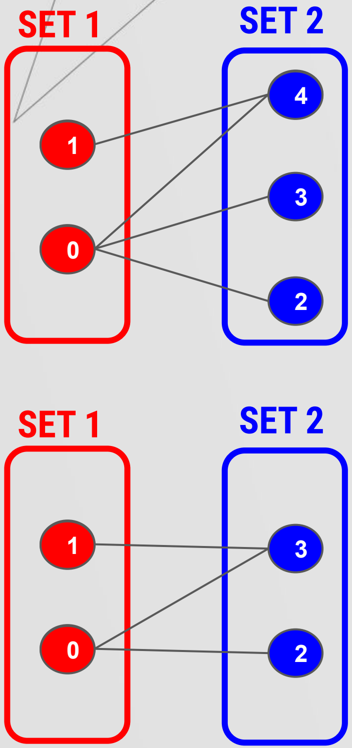
**edge_index**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

**We have to scale SET1 and SET2 differently!!!**

# 04 Advance mini-batching in PyTorch Geometric

Multiple bipartite graphs



**edge_index**

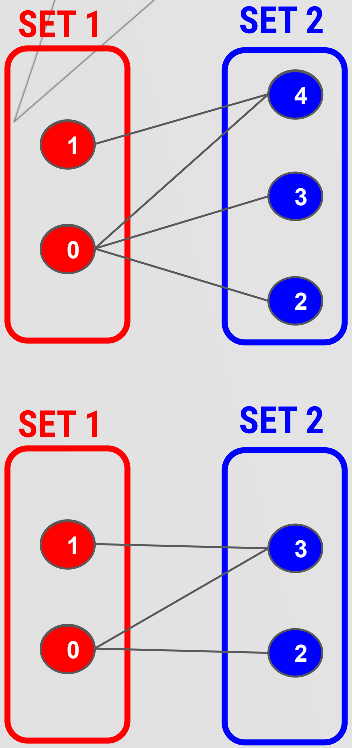
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 2 | | |
| SET 2 | 0 | 1 | 2 | 2 | | | |

**We have to scale SET1 and SET2 differently!!!**

# 04 Advance mini-batching in PyTorch Geometric

Multiple bipartite graphs



**edge_index**

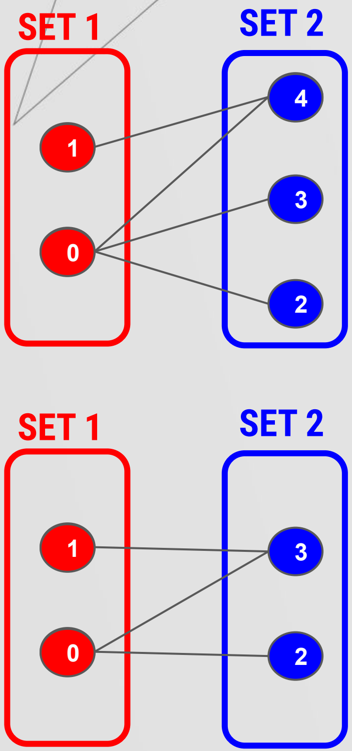
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 2 | 2 | 3 |
| SET 2 | 0 | 1 | 2 | 2 | | | |

**We have to scale SET1 and SET2 differently!!!**

# 04 Advance mini-batching in PyTorch Geometric

Multiple bipartite graphs



**edge_index**

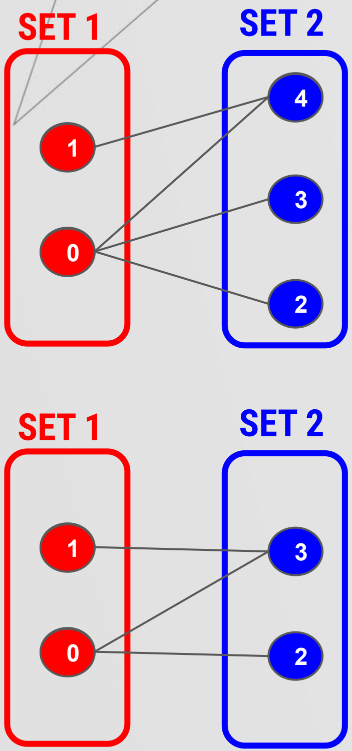
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 2 | 2 | 3 |
| SET 2 | 0 | 1 | 2 | 2 | 3 | | |

**We have to scale SET1 and SET2 differently!!!**

# 04 Advance mini-batching in PyTorch Geometric

Multiple bipartite graphs



**edge_index**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 2 | 2 | 3 |
| SET 2 | 0 | 1 | 2 | 2 | 3 | 4 | 4 |

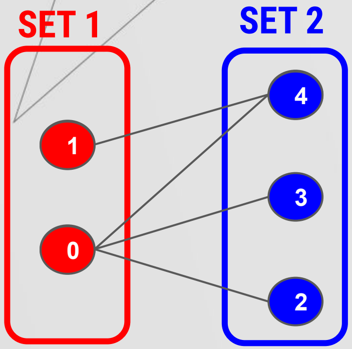
**We have to scale SET1 and SET2 differently!!!**

# 04 Advance mini-batching in PyTorch Geometric

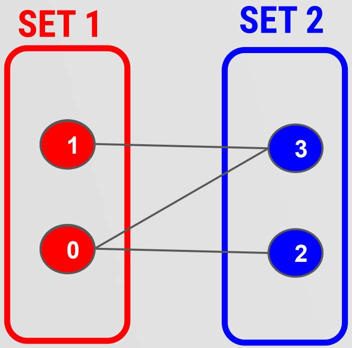
Multiple bipartite graphs

**We have to scale SET1 and SET2 differently!!!**

**edge_index**



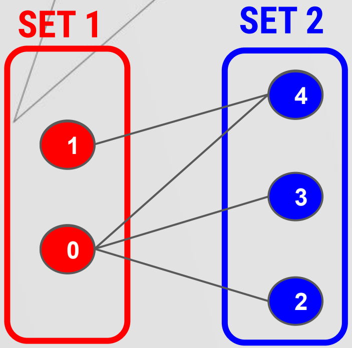
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 2 | 2 | 3 |
| SET 2 | 0 | 1 | 2 | 2 | 3 | 4 | 4 |

# 04 Advance mini-batching in PyTorch Geometric

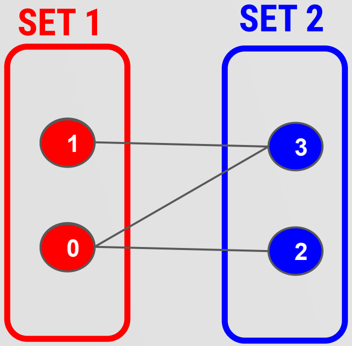
Multiple bipartite graphs



**edge_index**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| SET 2 | 0 | 1 | 2 | 2 | 2 | 3 | 3 |

Let's see this in practice!



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| SET 1 | 0 | 0 | 0 | 1 | 2 | 2 | 3 |
| SET 2 | 0 | 1 | 2 | 2 | 3 | 4 | 4 |

# Conclusion

- Batching allow us to scale our NN to larger datasets
- PyTorch Geometric handle simple batching
- PyG allow to modify batching to special purposes (like bipartite graphs)

# THANKS

Does anyone have any questions?

longaantonio@gmail.com
https://antoniolonga.github.io/