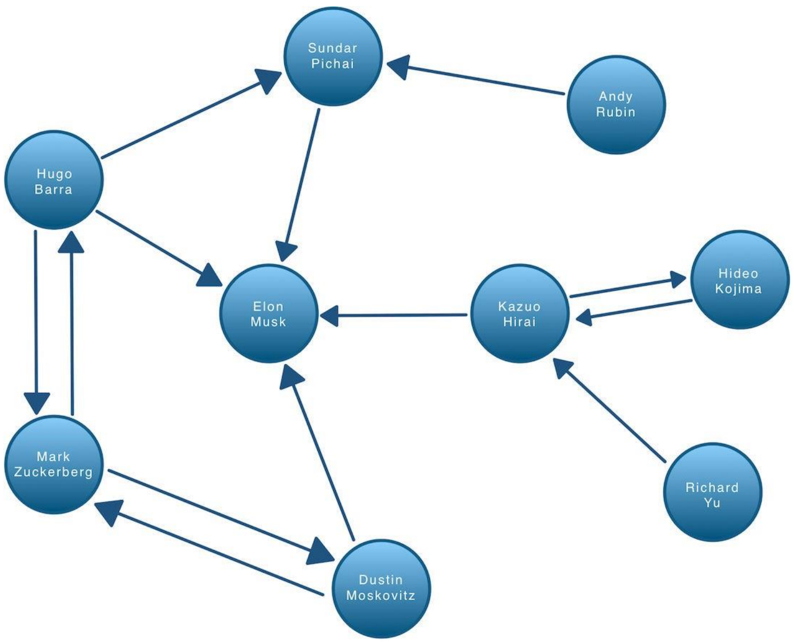
# A Deep Learning based Community Detection approach

**GIANCARLO SPERLI'**

**CINI – ITEM NATIONAL LABORATORY**

# Social Network Analysis




**Community Detection**

**Similar users over the networks, a subset of strictly connected nodes.**

**Why?**

- **Viral Marketing, brand awareness**
- **Advertising Targeting**
- **Recomendation**

**<u>Contribute</u>:**

**A novel algorithm to detect communities using deep learning approaches on large datasets**

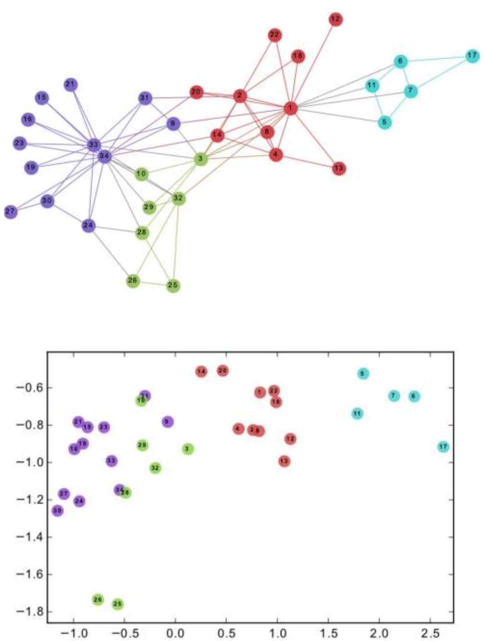
# Deep Learning

A Deep Learning based Community Detection approach

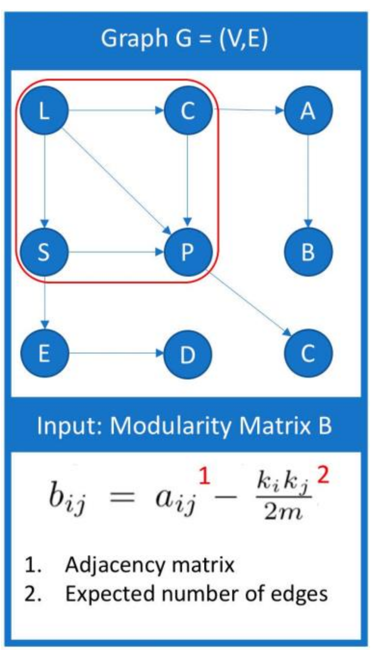
**Why?**

- ➢ **Train a network topology, predicting communities according to relationships that may change over the time.**
- ➢ **All existing approaches are limited**

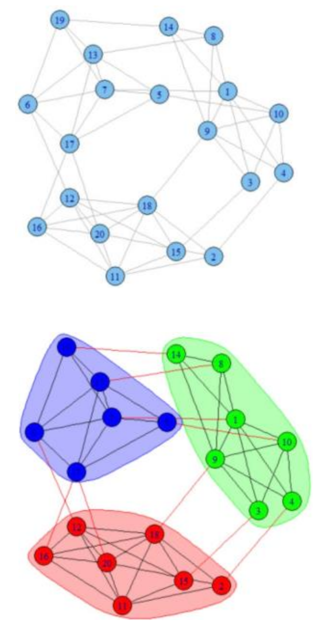
## DeepWalk



## Modularity Based




## K-means



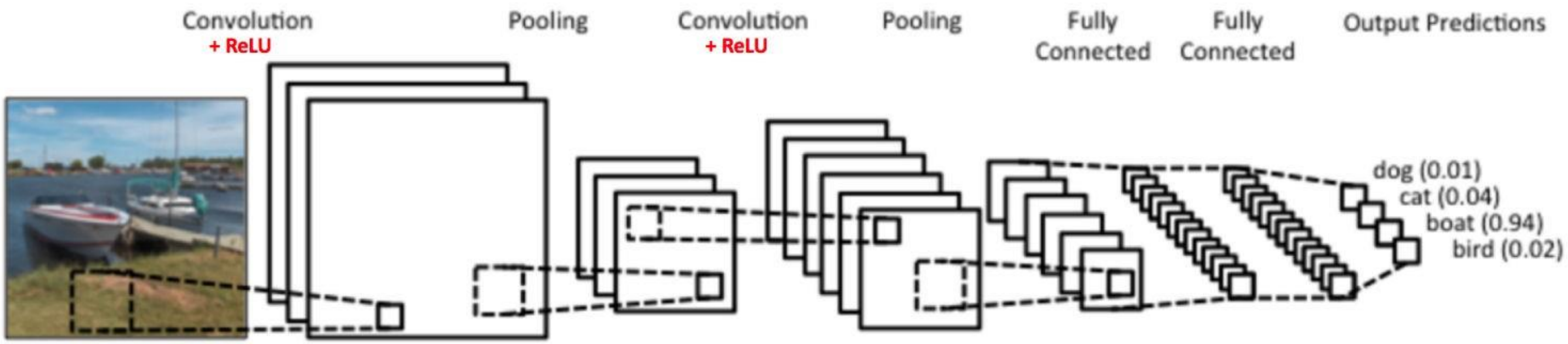
# Proposed Approach with CNN

**Traditional Application**

- **Object detection and image recognition**

**Goal**

- **Apply the same deep learning approach to social networking, for detecting communities**




**Input:**
- **Matrix with single pixel values**
- **3 sub-matrices for RGB images**
- **1 matrix for B/W image**

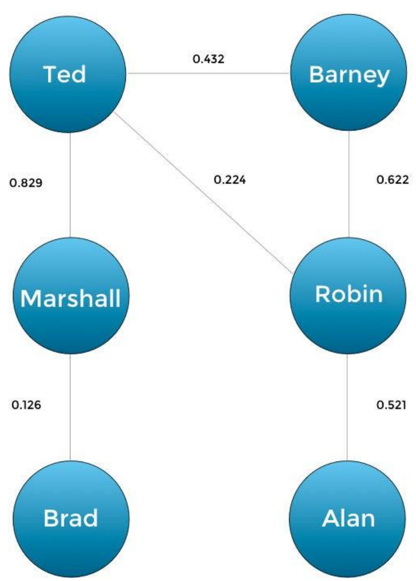
**Components:**
- **Convolution Layer**
- **Max-Pooling layer**
- **Fully Connected Layer**

**Output:**
**Probability that input image belongs to a defined class**

A Deep Learning based Community Detection approach

# CNN and Social Networks

## Input Construction




Adjacency Matrix

| | Ted | Barney | Marshall | Robin | Brad | Alan |
|---|---|---|---|---|---|---|
| Ted | 0 | 0,432 | 0,829 | 0,224 | 0 | 0 |
| Barney | 0,432 | 0 | 0 | 0,622 | 0 | 0 |
| Marshall | 0,829 | 0 | 0 | 0 | 0,126 | 0 |
| Robin | 0,224 | 0,622 | 0 | 0 | 0 | 0,521 |
| Brad | 0 | 0 | 0,126 | 0 | 0 | 0 |
| Alan | 0 | 0 | 0 | 0,521 | 0 | 0 |

Node Ted Sliced

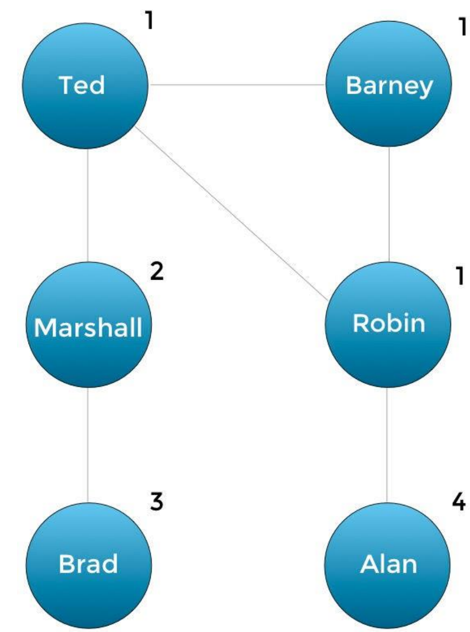
| | Ted | Barney | Marshall | Robin | Brad | Alan |
|---|---|---|---|---|---|---|
| Ted | 0 | 0,432 | 0,829 | 0,224 | 0 | 0 |

Node Ted Reshaped

| | | | |
|---|---|---|---|
| 0 | 0,432 | 0,829 | H |
| 0,224 | 0 | 0 | |

W

## Ground Truth

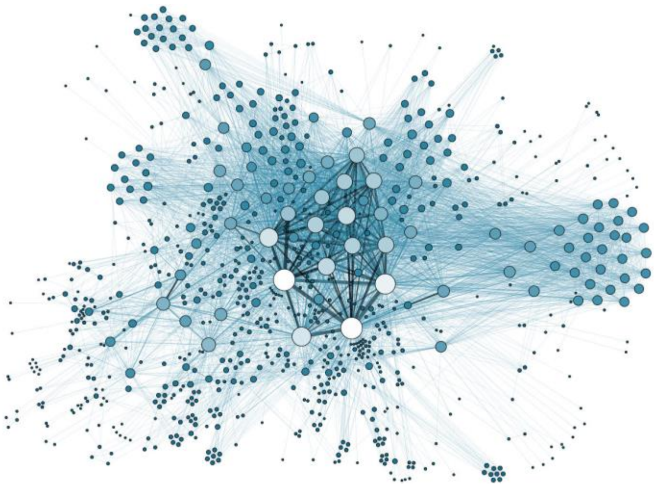



**Output** **Probability that input node belongs to a community**

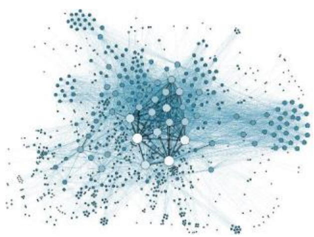
# The dimensional challenge

- **Image size:**
  **1024 x 1024 pixel**
- **Matrix values:**
  **1.048.576 x 32 bit = 4,2 MB**

- **Graph Size:**
**1.000.000 nodes**
- **Matrix values:**
**1.000.000.000.000 x 32 bit = 4000 GB**



# Sparsity Approach



- Graph Size:
  1.000.000 nodes

- Non-Zero values with
  $10^{-5}$ *sparsity grade*:

| Dense | Sparse |
|---|---|
| 4000 GB | 120 MB |

**1.000.000 x 32 bit = 120 MB**

**Primitives for sparse Matrices:**
- Storing
- Slicing
- Densifying

**Conv2D doesn't work with Sparse Matrices**

## Proposed Algorithm

**Re-implementing tf.Conv2D**
**in a SparseConv2D**

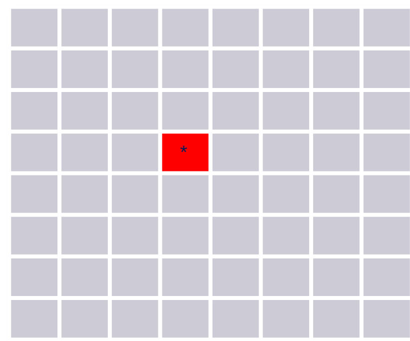
**exploiting sparse structure of network adjacency matrices**

# Sparse Convolution

## Kernel Initialization

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

## Sliced and Reshaped Node “n”

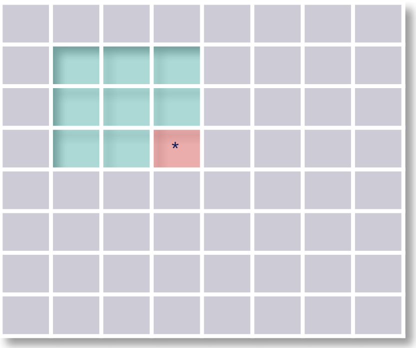


# Sparse Convolution

**Kernel Initialization**

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

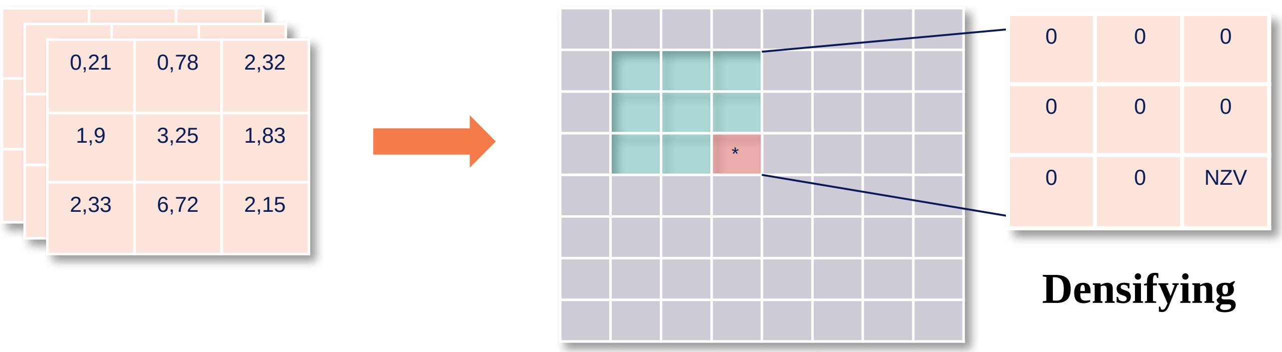
**Sliced and Reshaped Node "n"**



# Sparse Convolution

**Kernel Initialization**

**Sliced and Reshaped Node "n"**



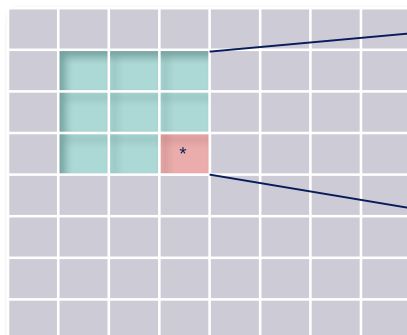

**Densifying**

# Sparse Convolution

## Kernel Initialization

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

## Sliced and Reshaped Node “n”




| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | NZV |

**Densifying**

X

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

**Kernel**

| | | |
|---|---|---|
| 0*0,21 | 0*0,78 | 0*2,32 |
| 0*1,9 | 0*3,25 | 0*1,83 |
| 0*2,32 | 0*6,72 | NZV*2,15 |

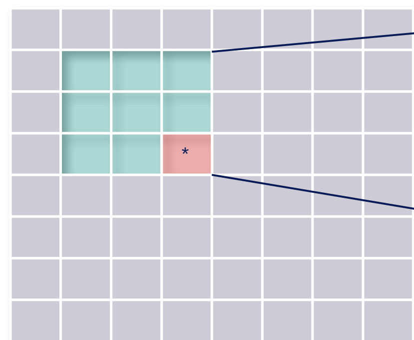
**Feature Value**

# Sparse Convolution

## Kernel Initialization

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

## Sliced and Reshaped Node “n”




| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | NZV |

Densifying

X

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

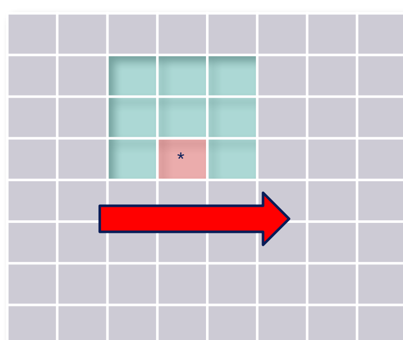
Kernel

| | | |
|---|---|---|
| 0*0,21 | 0*0,78 | 0*2,32 |
| 0*1,9 | 0*3,25 | 0*1,83 |
| 0*2,32 | 0*6,72 | NZV*2,15 |

Feature Value

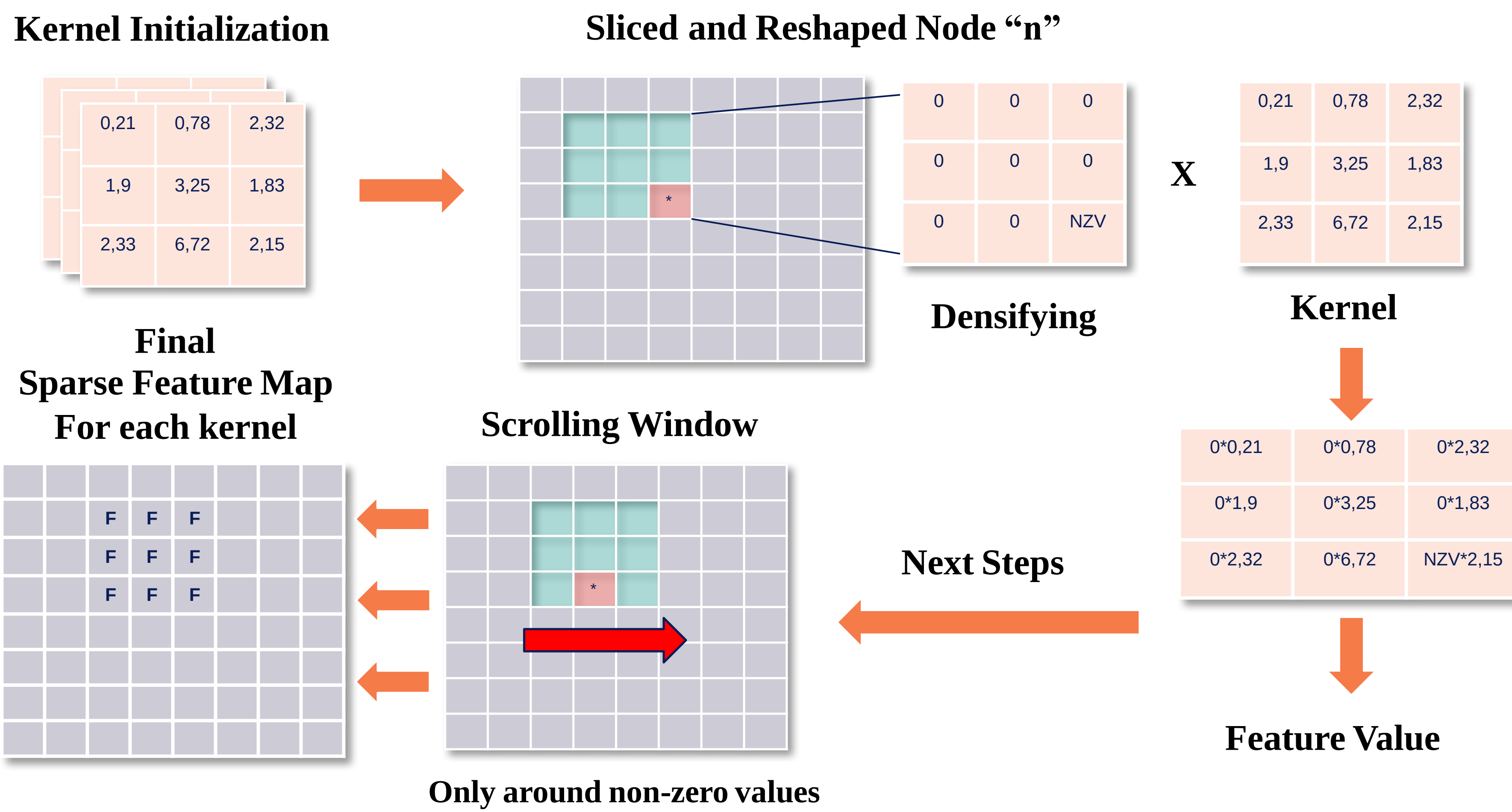
Next Steps

## Scrolling Window




Only around non-zero values

# Sparse Convolution

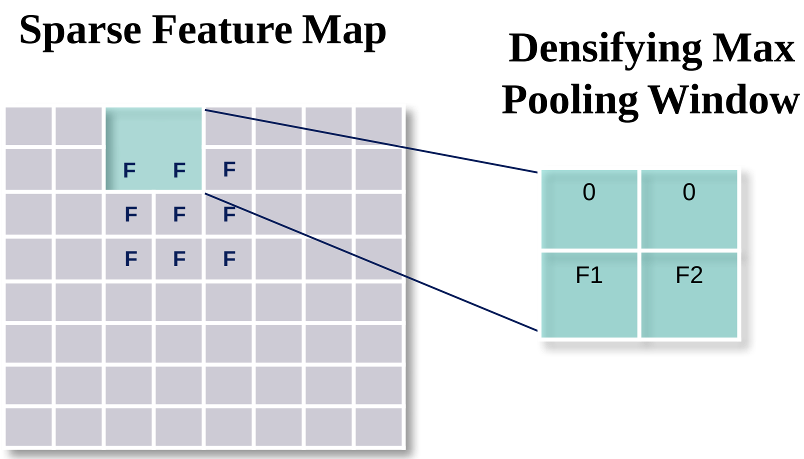


# Sparse Max Pooling

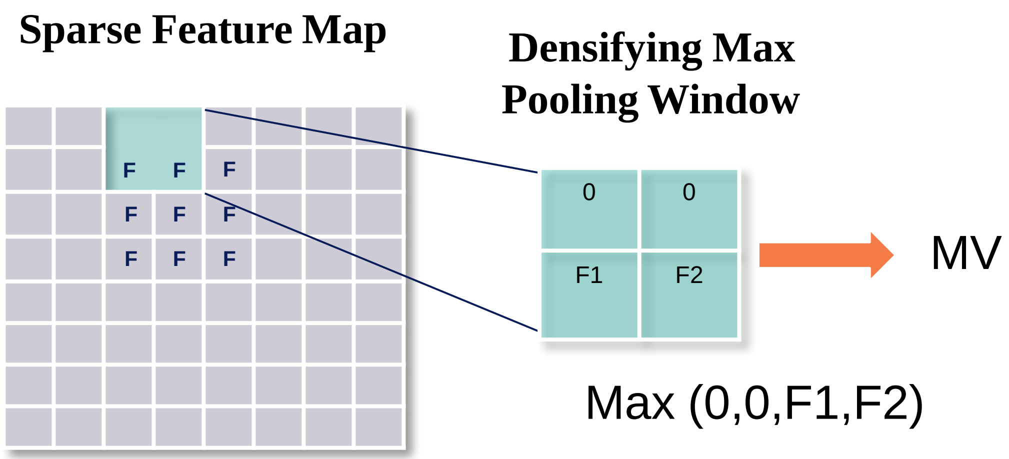
**Sparse Feature Map**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | F | F | F | | | |
| | | F | F | F | | | |
| | | F | F | F | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

# Sparse Max Pooling



# Sparse Max Pooling



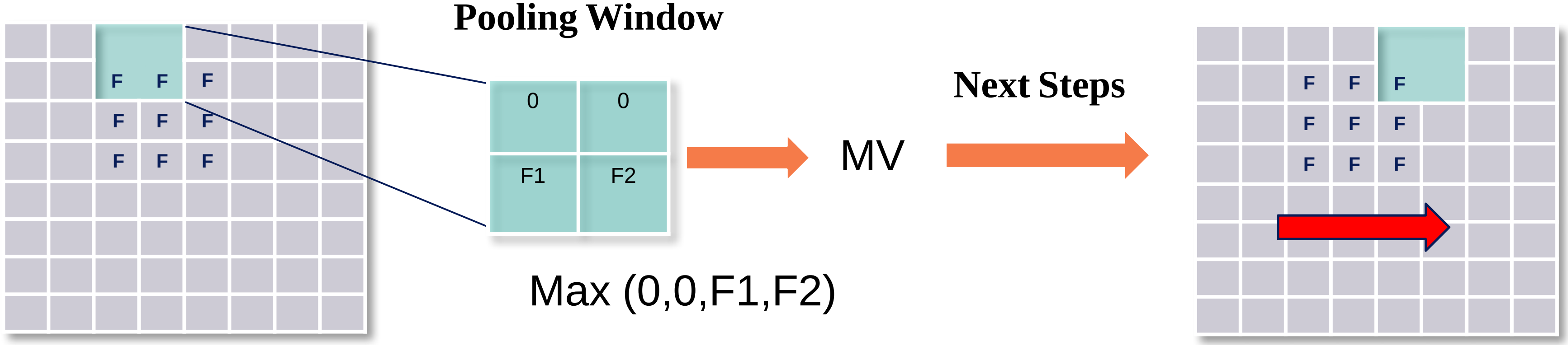
# Sparse Max Pooling

**Sparse Feature Map**

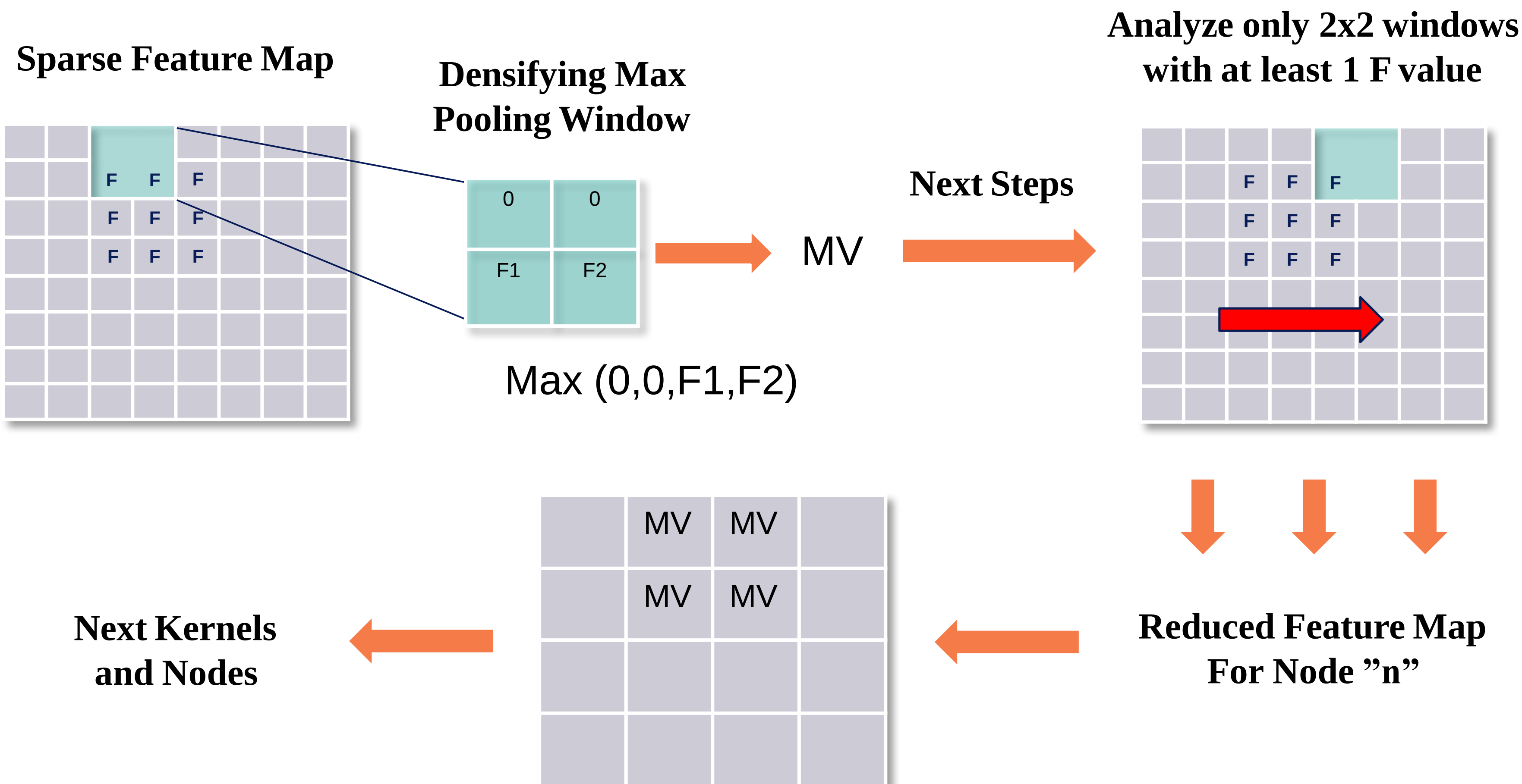
**Densifying Max Pooling Window**

**Next Steps**

**Analyze only 2x2 windows with at least 1 F value**



# Sparse Max Pooling

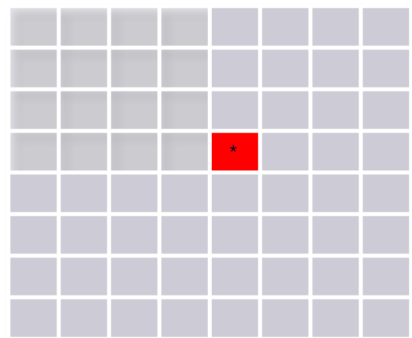


# Another Approach: Divide et impera

**Kernel Initialization**

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

**Sliced and Reshaped Node "n"**

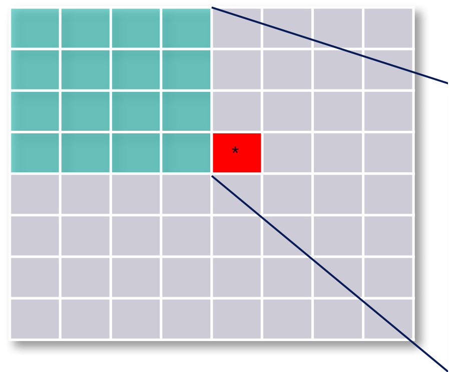


# Another Approach: Divide et impera

**Kernel Initialization**

| | | |
|---|---|---|
| 0,21 | 0,78 | 2,32 |
| 1,9 | 3,25 | 1,83 |
| 2,33 | 6,72 | 2,15 |

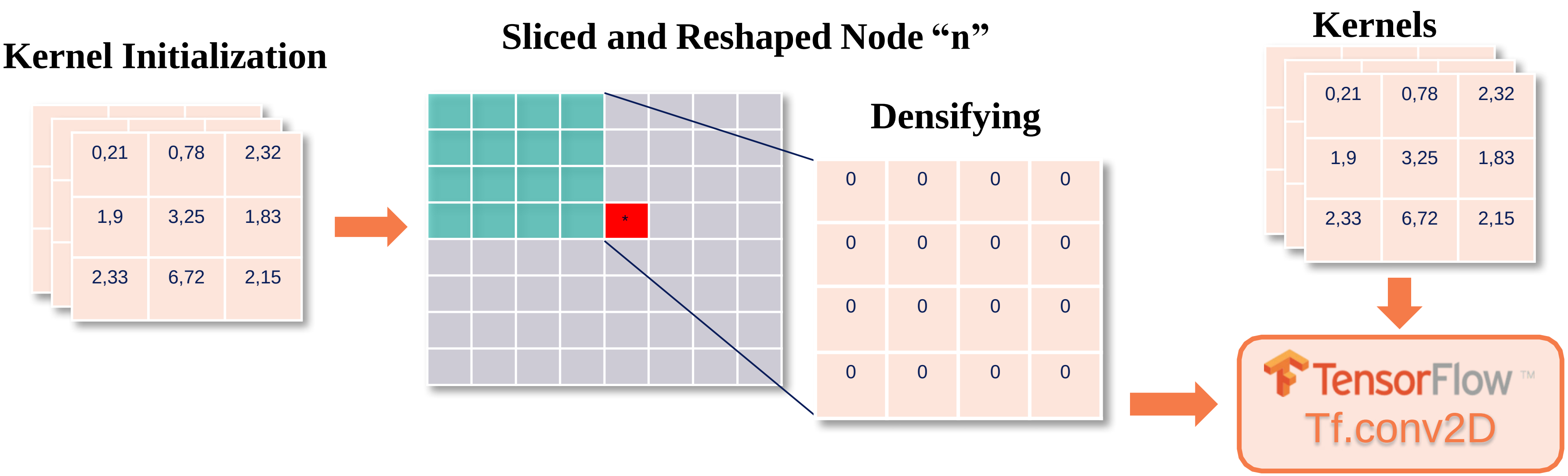
**Sliced and Reshaped Node "n"**



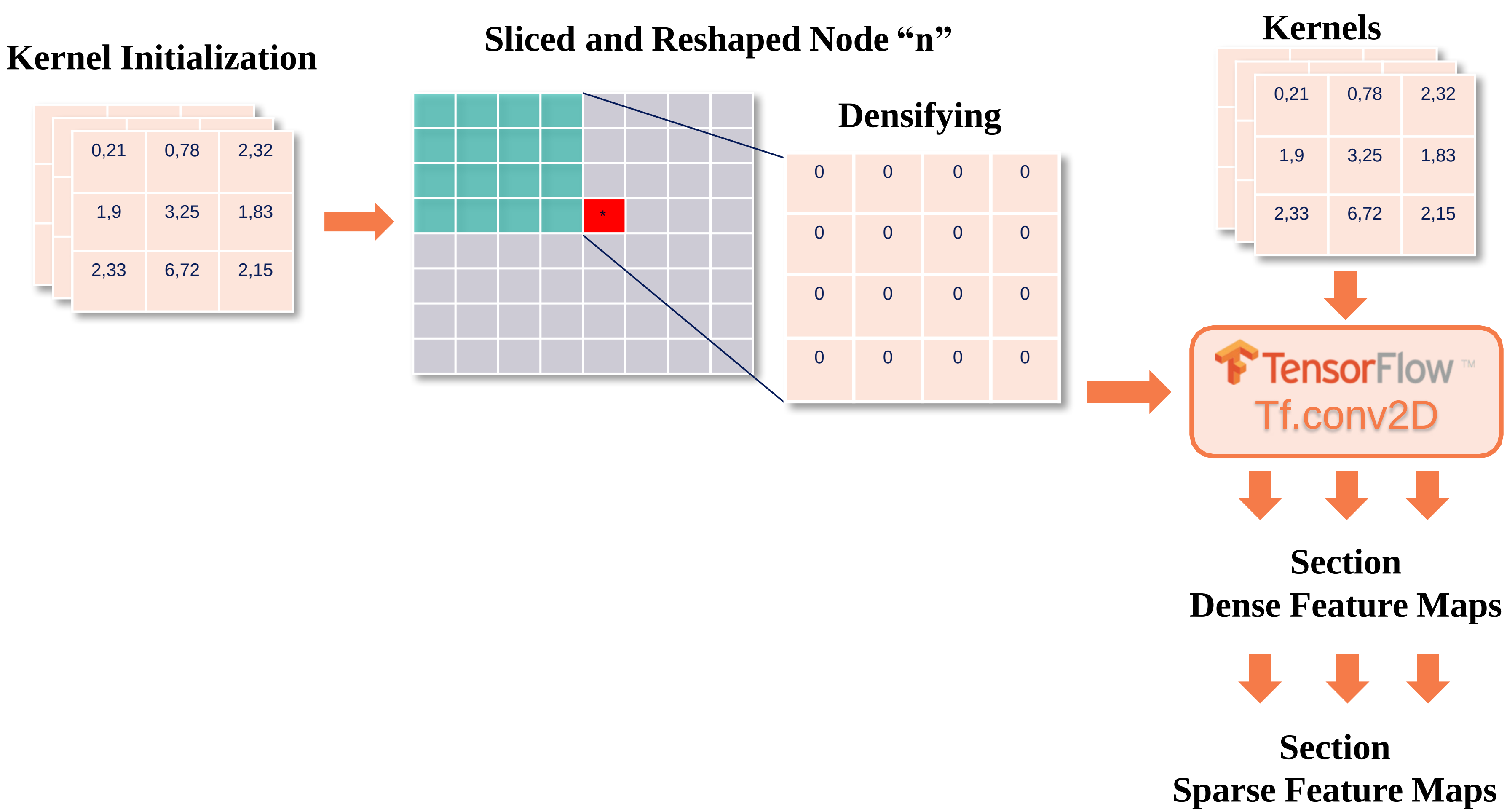

**Densifying**

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

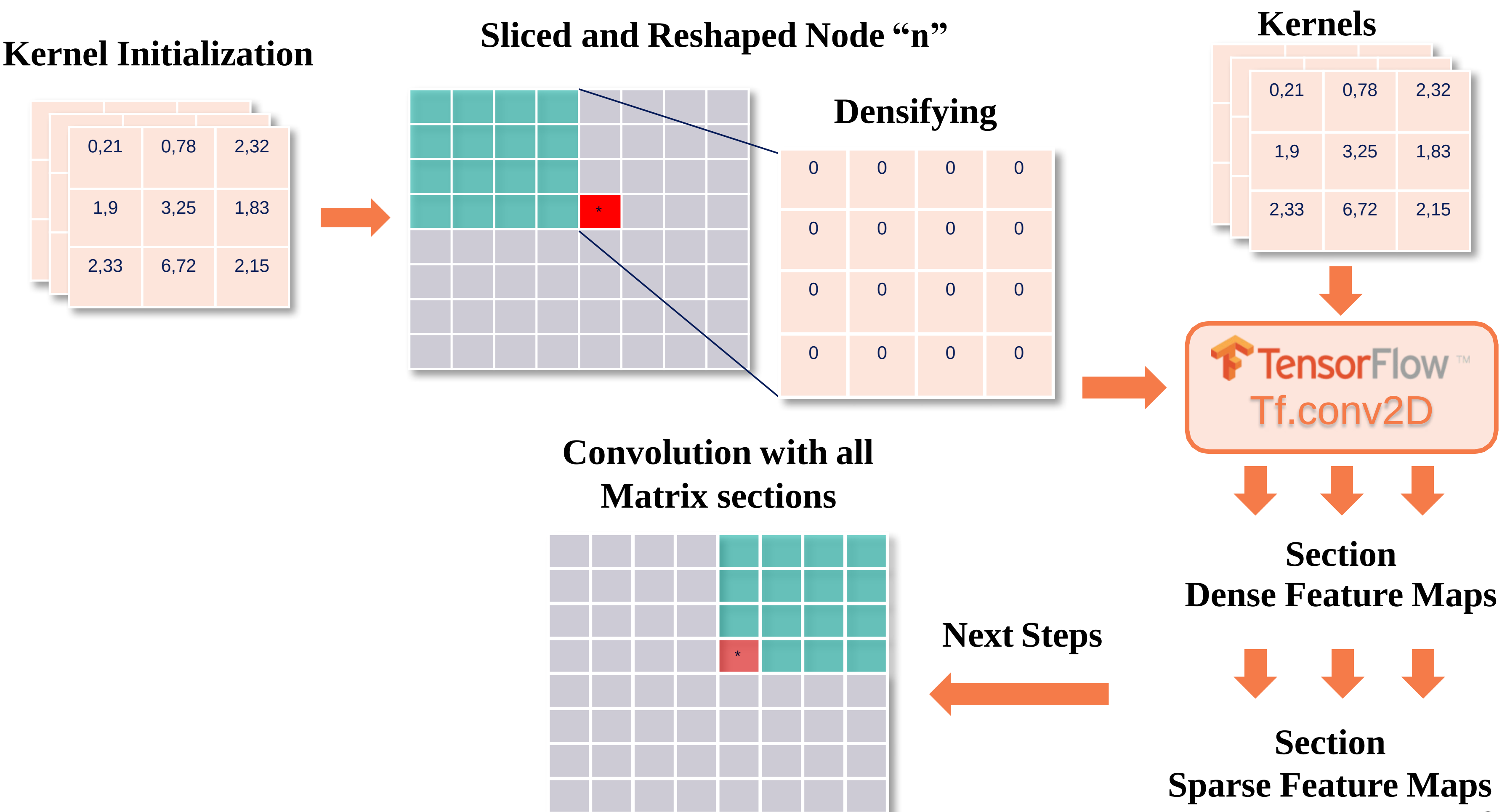
# Another Approach: Divide et impera



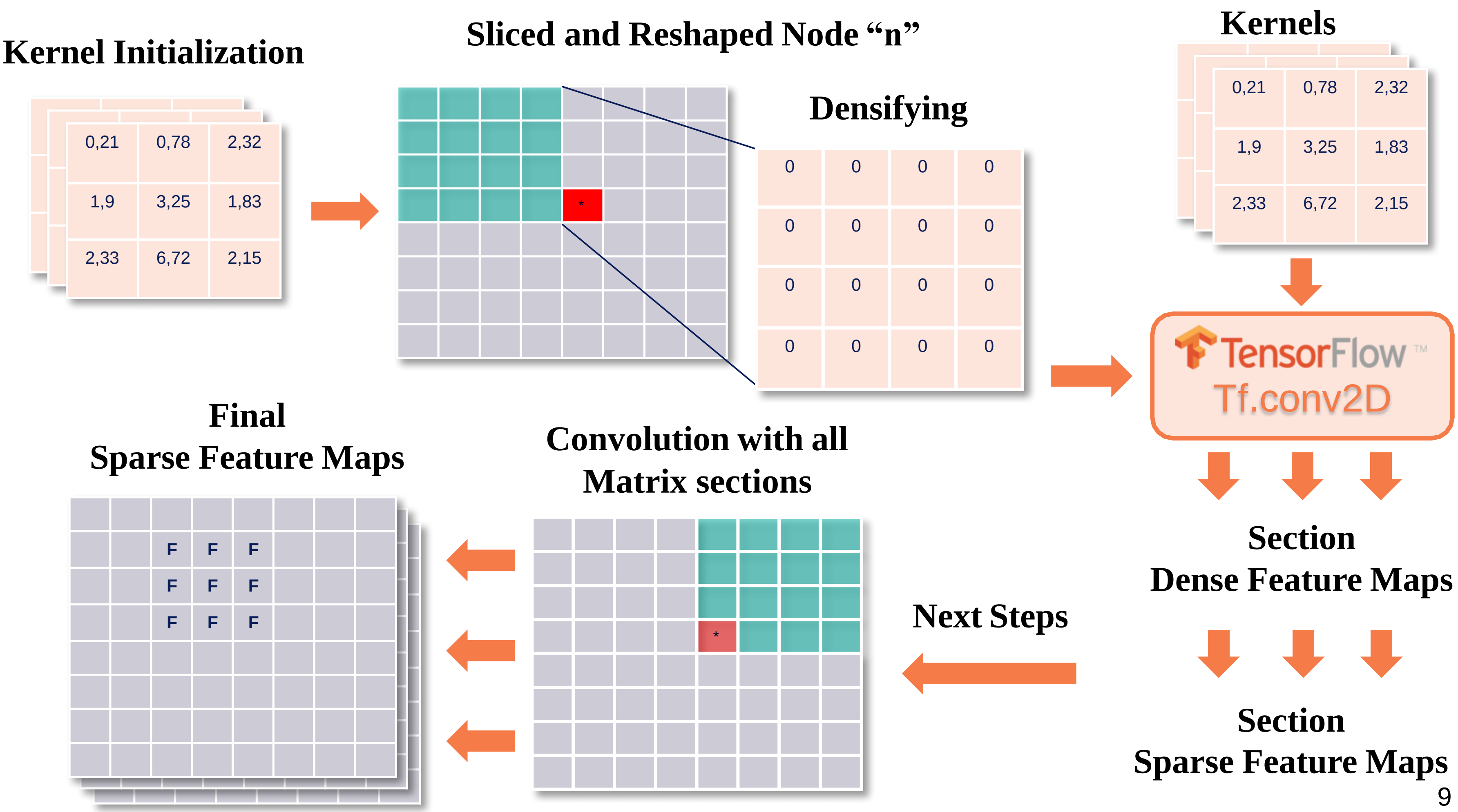
# Another Approach: Divide et impera



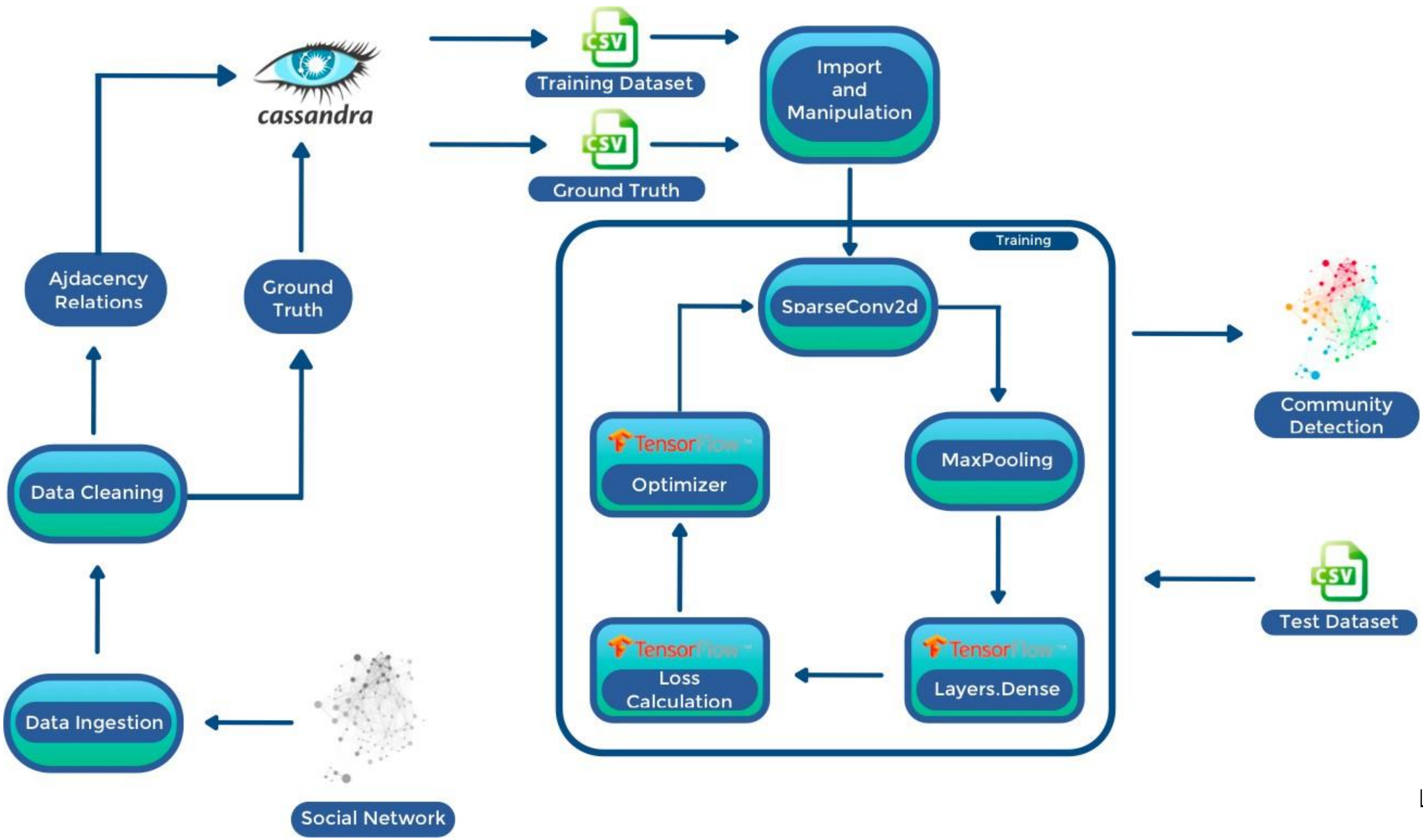
# Another Approach: Divide et impera



# Another Approach: Divide et impera



# System Overview



# Experimental Protocol

**Convolution Time Tests**
- ➢ **SparseConv2D vs Divide et Impera**
- ➢ **Artificial Large Datasets with different dimensions and sparsity**

**Training Tests**
- ➢ **SparseConv2D loss trend (kernels, levels, optimizer and rate)**
- ➢ **Email Dataset to test whole network**

**Accuracy Tests**
- ➢ **Using 3 different training**
- ➢ **Evaluation datasets generated from Email Dataset**

## Artificial Datasets

- **Sparse Random Matrices**
- **Variable Dimensions**
- **Variable Sparsity**

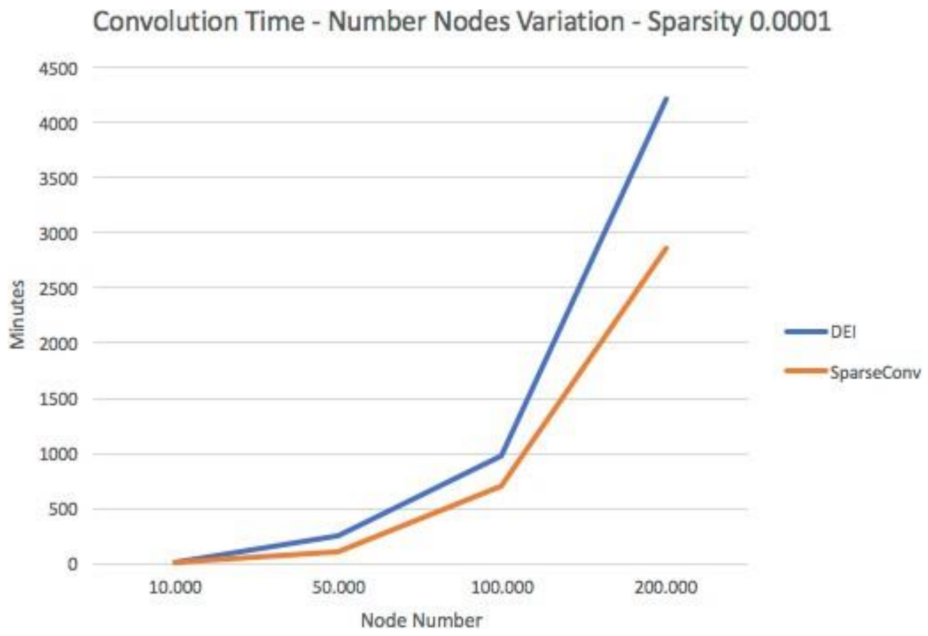
## Large Dataset

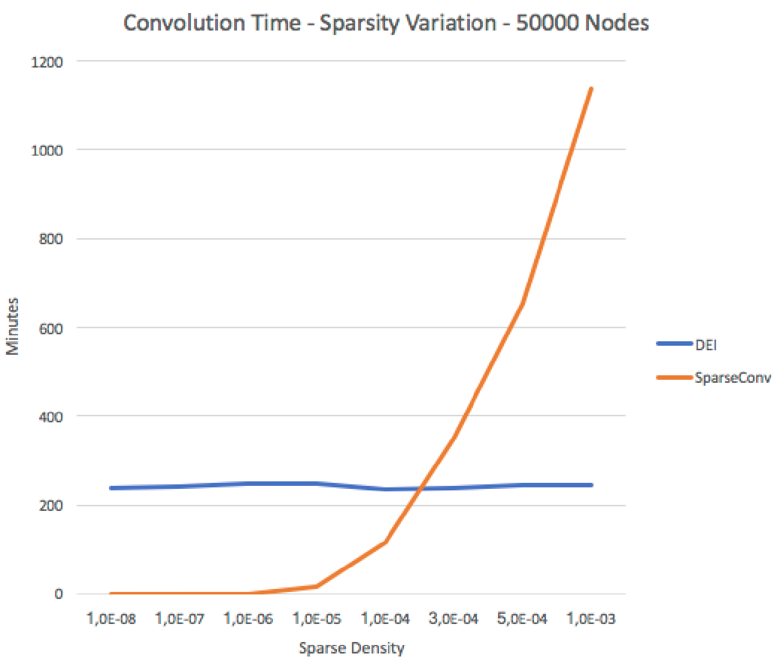
- **Dimension: 138.000 nodes**
- **Sparsity: $7,1 * 10^{-6}$**

## Email Dataset

- **Dimension: 1000 nodes**
- **Edges: 25000**

# Convolution Time Tests

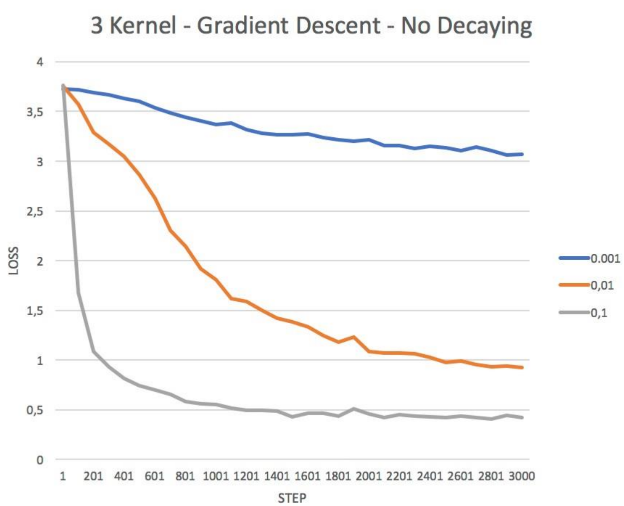


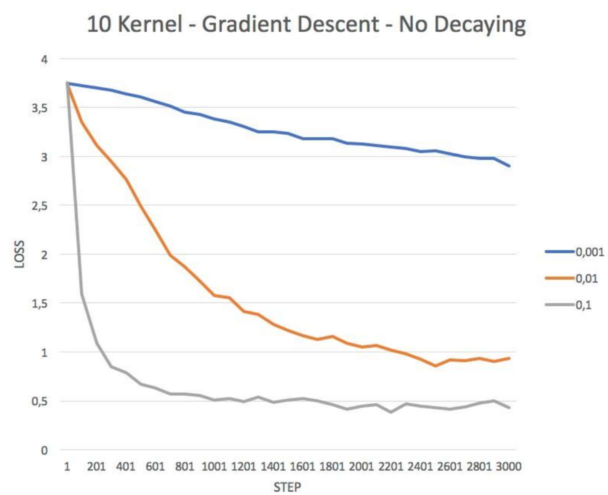



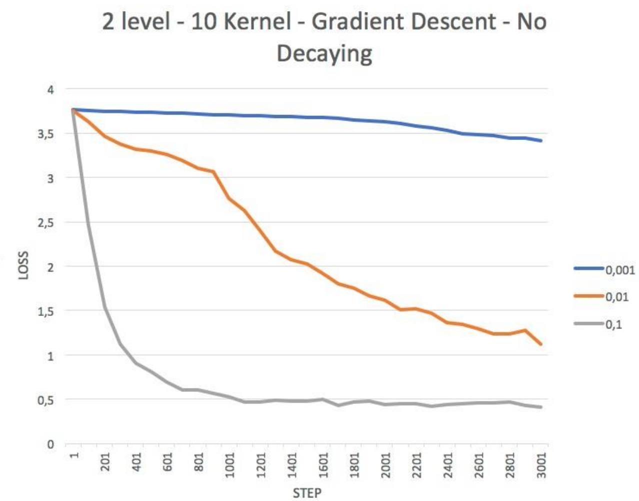

- **138.000 nodes, $10^{-6}$**
- **SparseConv: 12 minutes**
- **Divide et impera: 602 minutes**

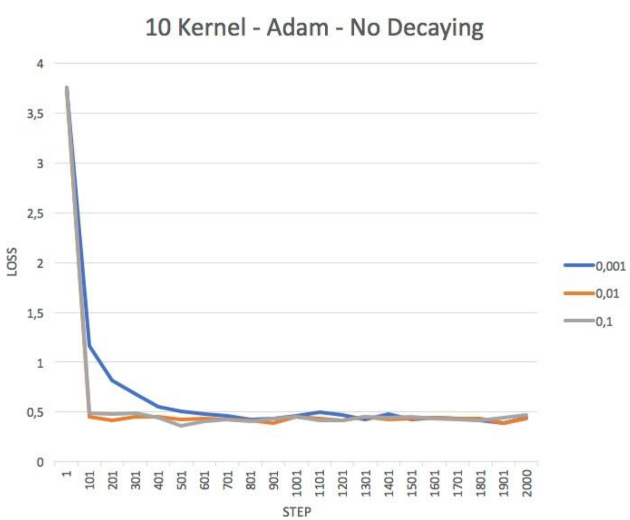
# Training

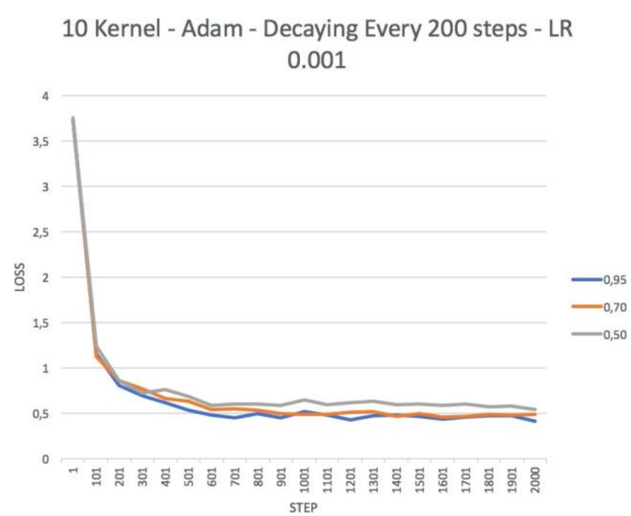
**A Deep Learning based Community Detection approach**

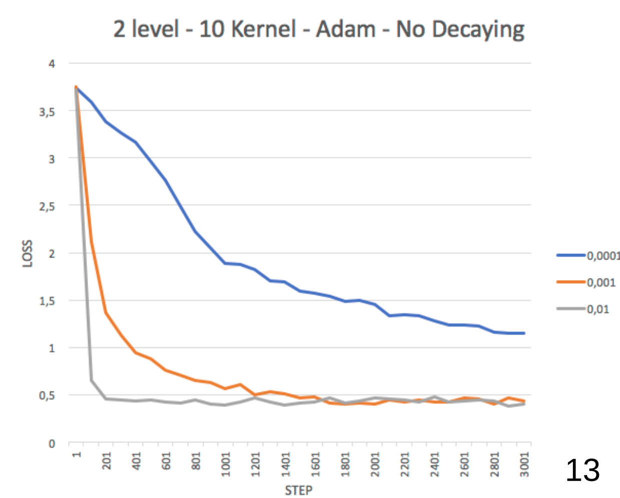


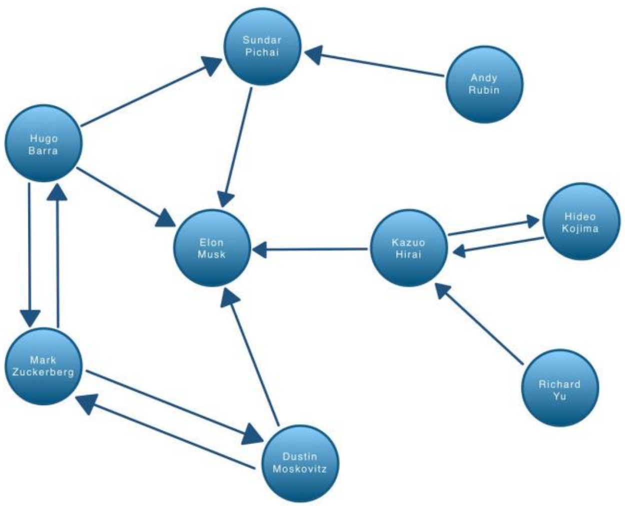



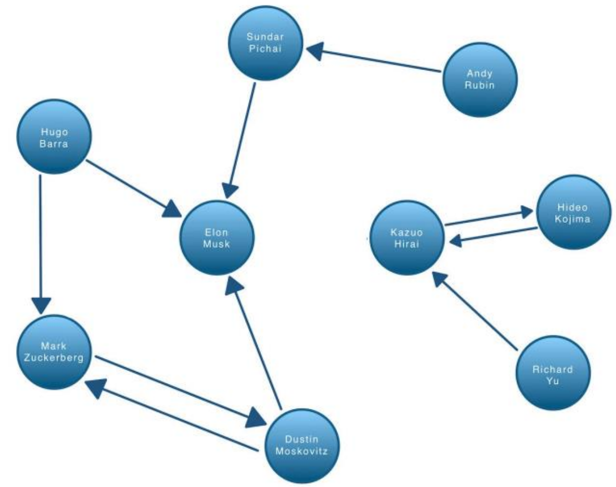
# Evaluation Datasets

**Training Set**




**Deleting Edges**

**Test Set**




# Entries with Hop Count

$$entry = e^{\sigma \cdot (1-s)}$$

- **s: hop count of node n ′ to node n**
- **σ: attenuation factor**
- **s0: hop count threshold**

A Deep Learning based Community Detection approach

# Evaluation Tests