HIGH PERFORMANT SMART COMPUTATIONS FOR GRAPH ANALYSIS

RICHARD FORSTER

CERN

AGENDA

- Community Detection
- Communities with Deep Q-learning
- ForceAtlas
- Barnes-Hut Tree with RNN

COMMUNITY DETECTION

COMMUNITY DETECTION

- Used to reveal groups in real world data
- Louvain method
- Parallel heuristics

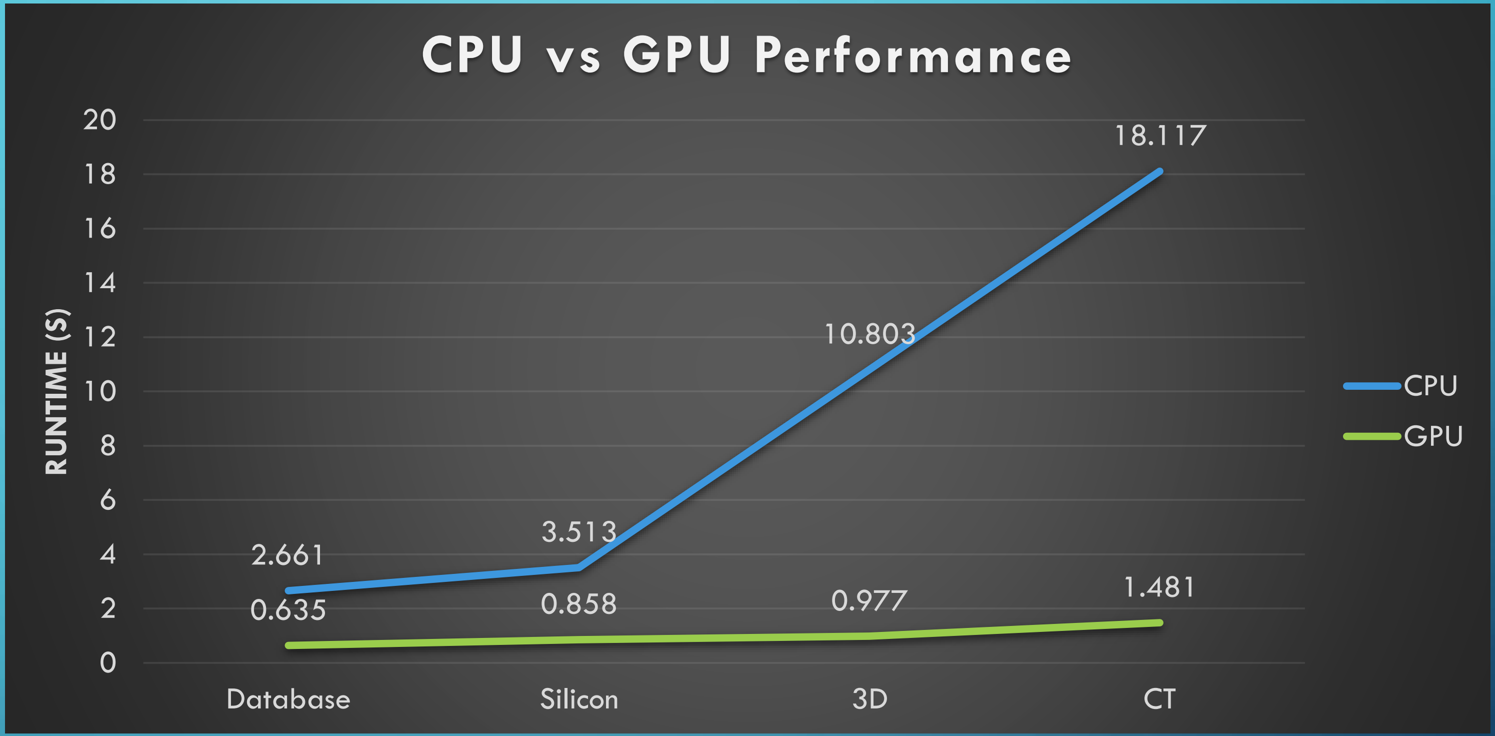
LOUVAIN METHOD

- Multi-phase, iterative, greedy algorithm
- Monotonically increasing modularity
- Inherently sequential

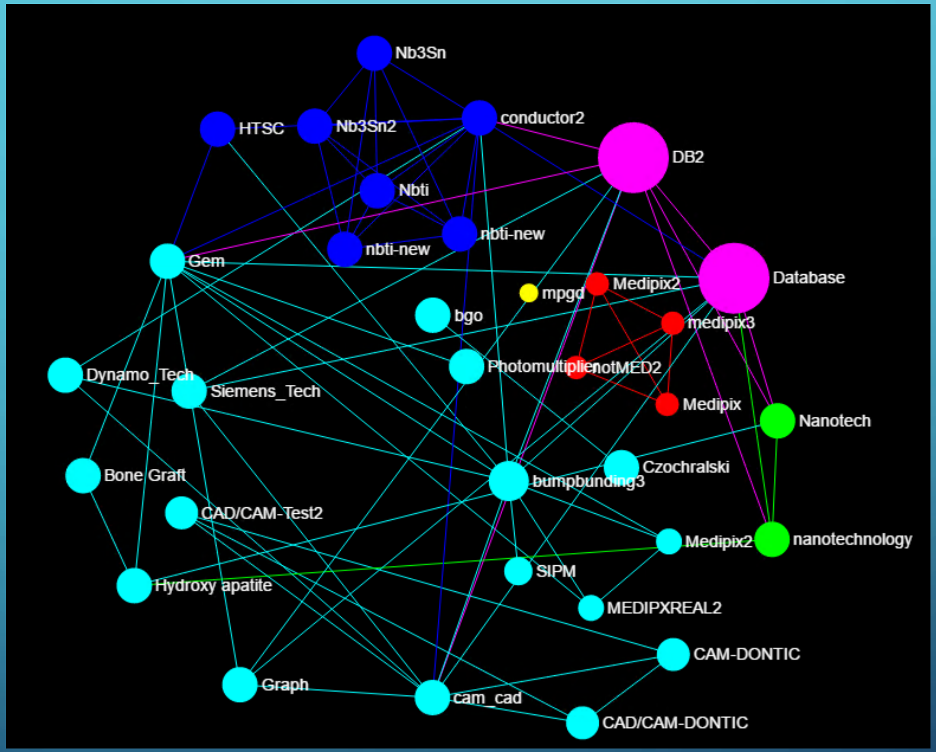
RESULTS

(COMMUNITY DETECTION)

- Database: 4x
- Silicon: 4x
- 3D: 11x
- CT: 12x



	Database	Silicon	3D	CT
Nodes	17832	24923	35763	53039
Edges	113622	185594	192336	325490

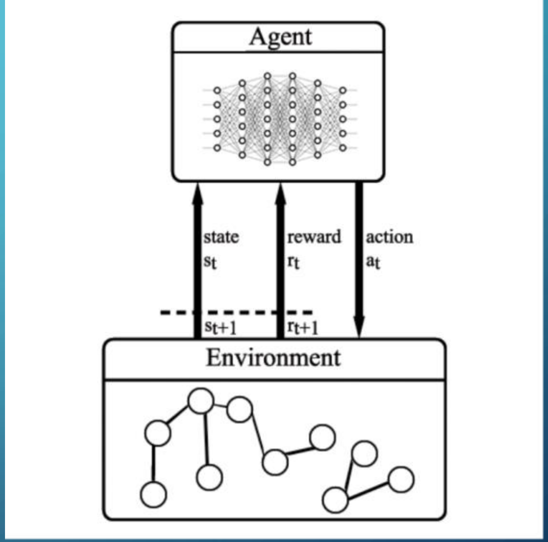

Nb3Sn
HTSC
Nb3Sn2
conductor2
DB2
Nbti
nbti-new
nbti-new
Gem
Database
Medipix2
mpgd
bgo
medipix3
Photomultiplier
notMED2
Dynamo_Tech
Siemens_Tech
Medipix
Nanotech
Czochralski
Bone Graft
bumpbunding3
CAD/CAM-Test2
Medipix2
nanotechnology
SIPM
Hydroxy apatite
MEDIPXREAL2
CAM-DONTIC
Graph
cam_cad
CAD/CAM-DONTIC

COMMUNITIES WITH DEEP Q-LEARNING

COMMUNITIES WITH DEEP Q-LEARNING

Reinforcement learning problem:

- Interactions between the environment and agent
    - Action-reward
- Goal is to maximize the reward, find appropriate π policy
- Finite action space in every timestep



COMMUNITIES WITH DEEP Q-LEARNING

Extensions for deep Q-learning:

1. Calculate Q for all possible actions in state st,
2. Make prediction for Q in st+1, get action at+1 = maxa a ϵ A(st+1),
3. Q[a] = r+γQ(st+1,at+1),
4. Update the network

Current limitations:

- Maximum 4 neighbors are explored

Processing:

- Training data contains graphs with various size and structure.
- The environment computes the real Louvain modularity to get the next community. Reward for correct prediction 10000, for wrong -1000
- The agent makes a prediction for every node

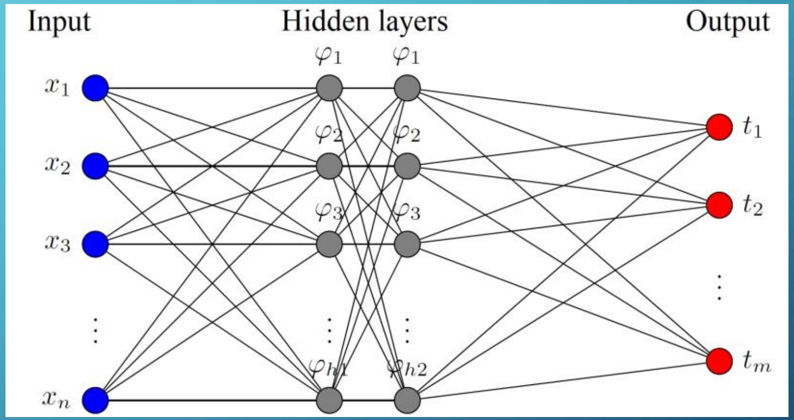
COMMUNITIES WITH DEEP Q-LEARNING

The agent:

- n: input size
- h1: size of hidden layer 1
- h2: size of hidden layer 2
- m: size of the action space
- Input size is state_size*action_size
- h1 = h2 = 128
- m = 4

To prevent overfitting:

- 50% dropout on both hidden layers
- Multiple independent representation



83% precision with early model after 70 epochs

FORCEATLAS

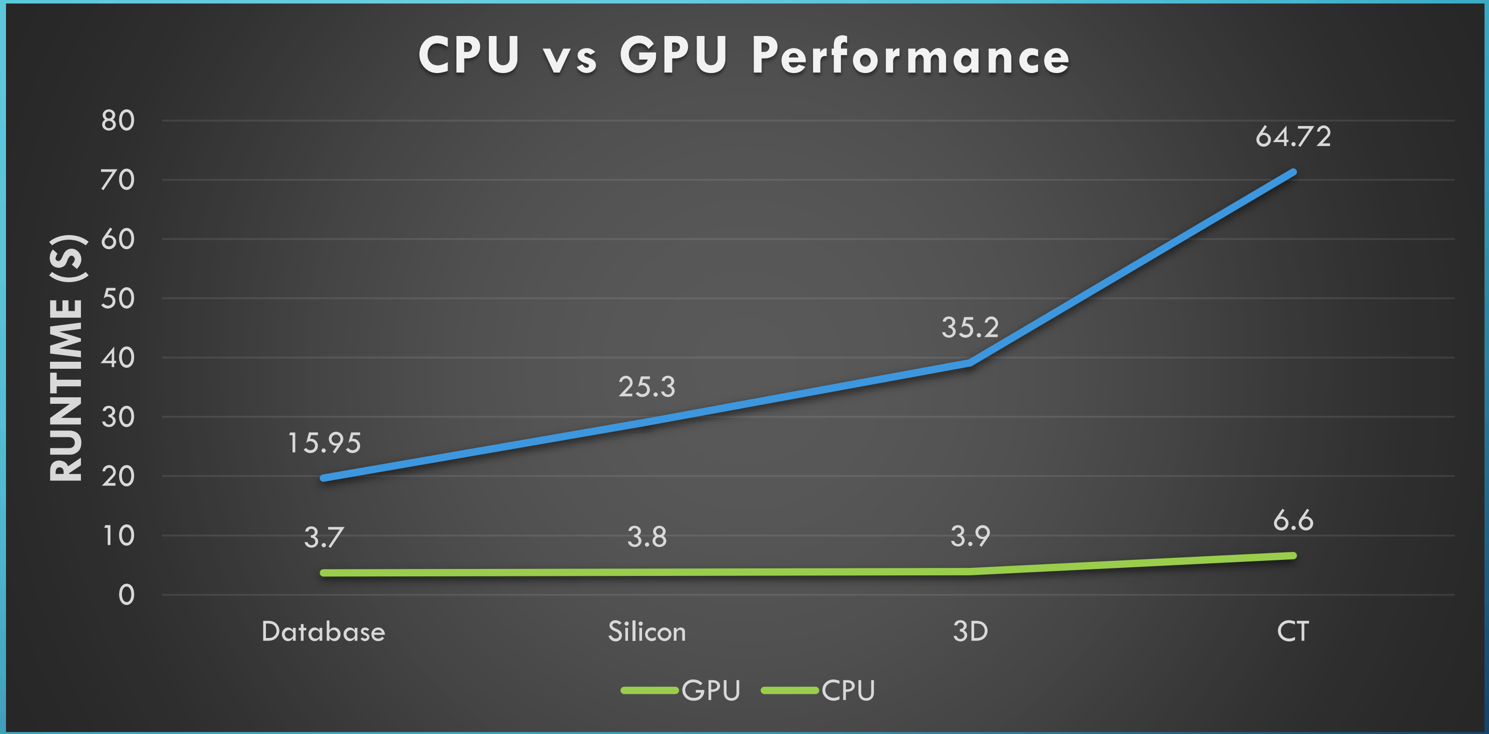
FORCEATLAS

- Force-directed layout based on n-body simulation
- Repulsion-attraction
- Makes visual interpretation easier
- Result depends on starting state

RESULTS

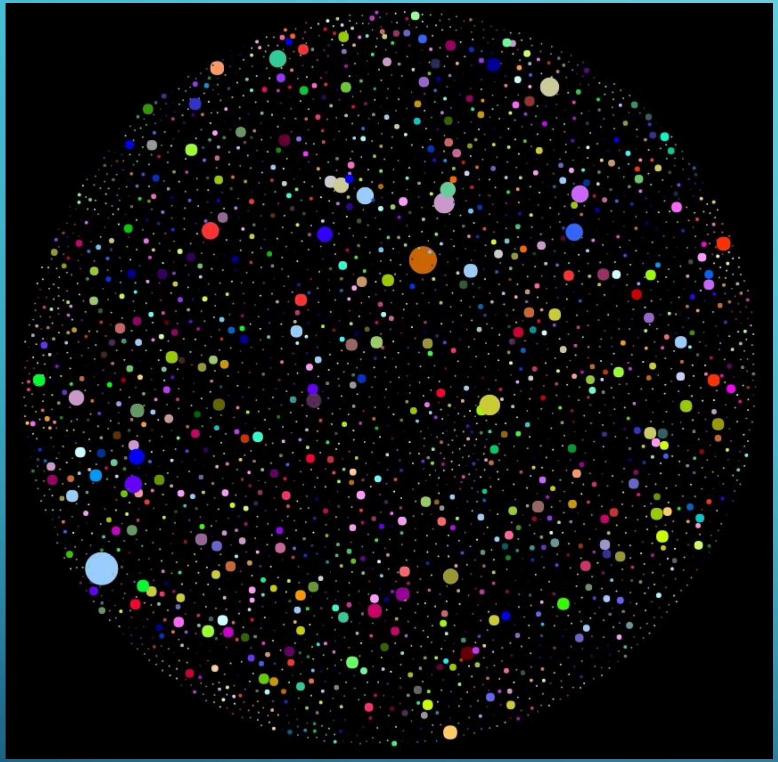
(FORCEATLAS)

- Database: 4,31x
- Silicon: 6,65x
- 3D: 9x
- CT: 9,8x

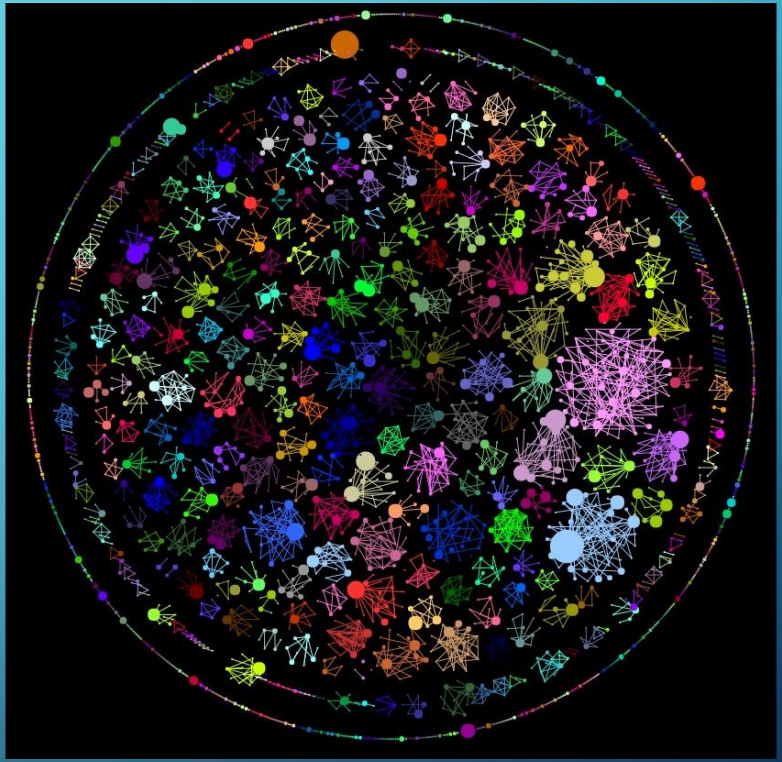


	Database	Silicon	3D	CT
Nodes	17832	24923	35763	53039
Edges	113622	185594	192336	325490

FORCEATLAS LAYOUT TYPES



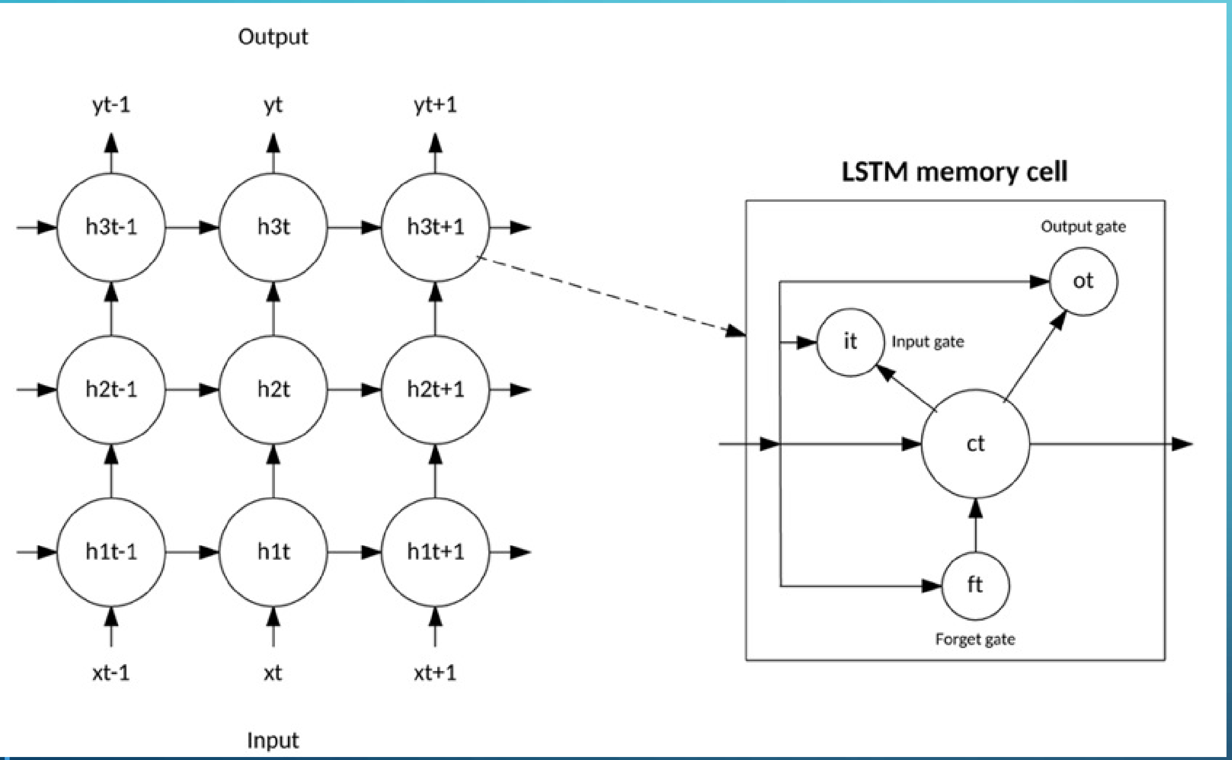
Original layout



Community based layout

BARNES-HUT TREE WITH RNN

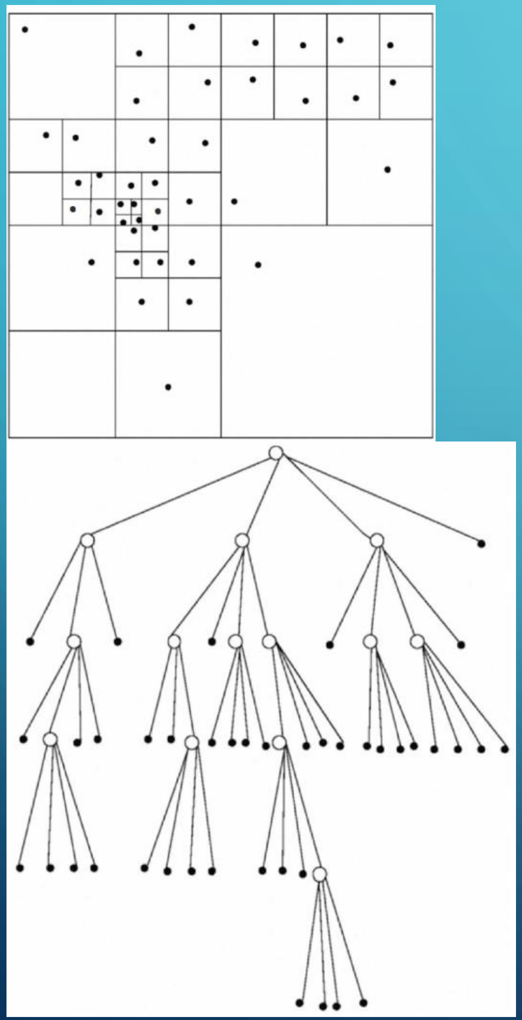
BARNES-HUT TREE WITH RNN



$$h_t = \phi(Wx_t + Uh_{t-1})$$

- h_t : hidden state in timestep t
- x_t : input in timestep t
- h_{t-1} : hidden state in timestep t-1
- U: transformation matrix
- Φ: activation function(logistic sigmoid, tanh)

BARNES-HUT TREE WITH RNN



- Generation provides a quadratic tree
- Plane is halved until only 1 node remains in a region.
- For RNN system is provided as a time series

BARNES-HUT TREE WITH RNN

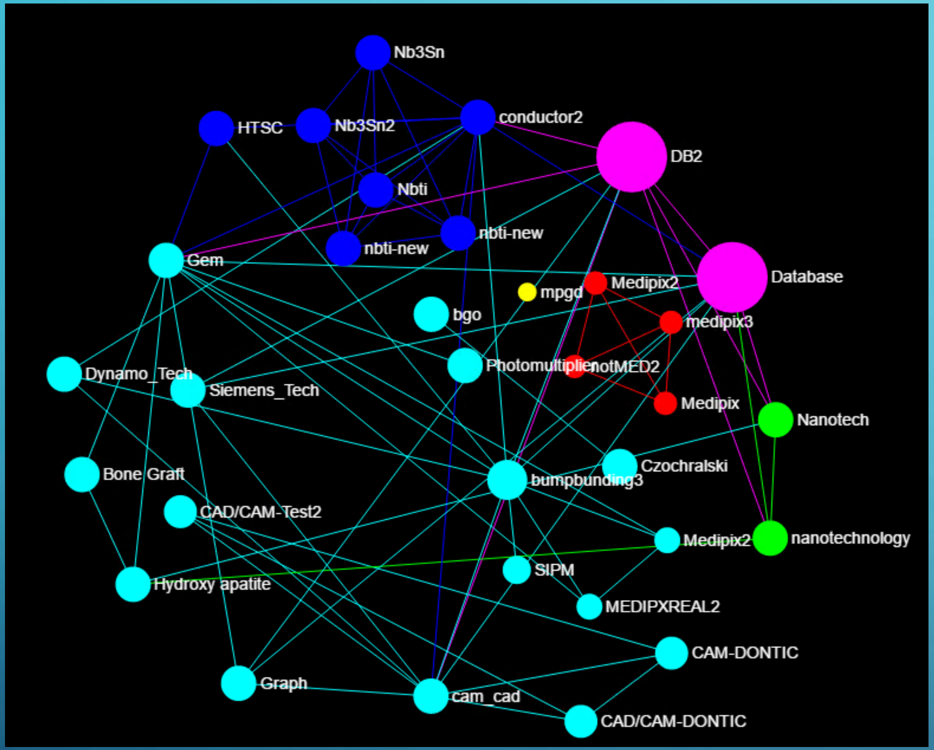
t, x, y, old_n_1, old_n_2, old_n_3, …, old_n_N, new_n_1, new_n_2, …, new_n_N

- t: timestep
- x, y: coordinates of node in timestep t
- old_n_N: state generated in timestep t-1 (initially -1)
- new_n_N: state generated in timestep t (no -1 at the end)
- N: number of nodes
- Model trained for graphs with 1000 nodes: 94% accuracy.

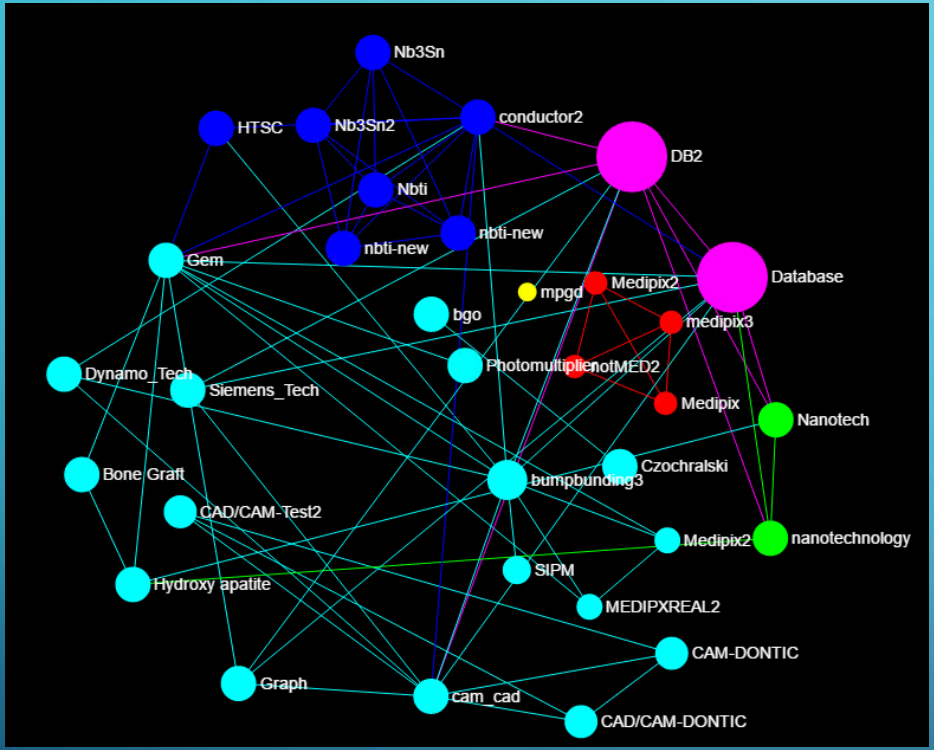
BARNES-HUT TREE WITH RNN

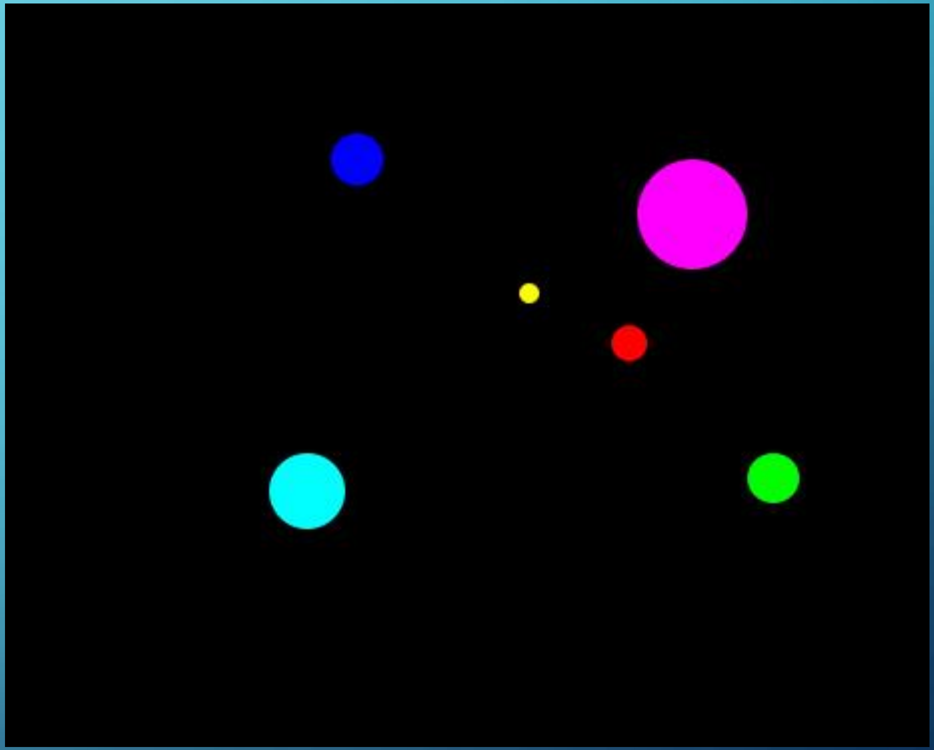
How to use this approach to increase the visual experience?

BARNES-HUT TREE WITH RNN

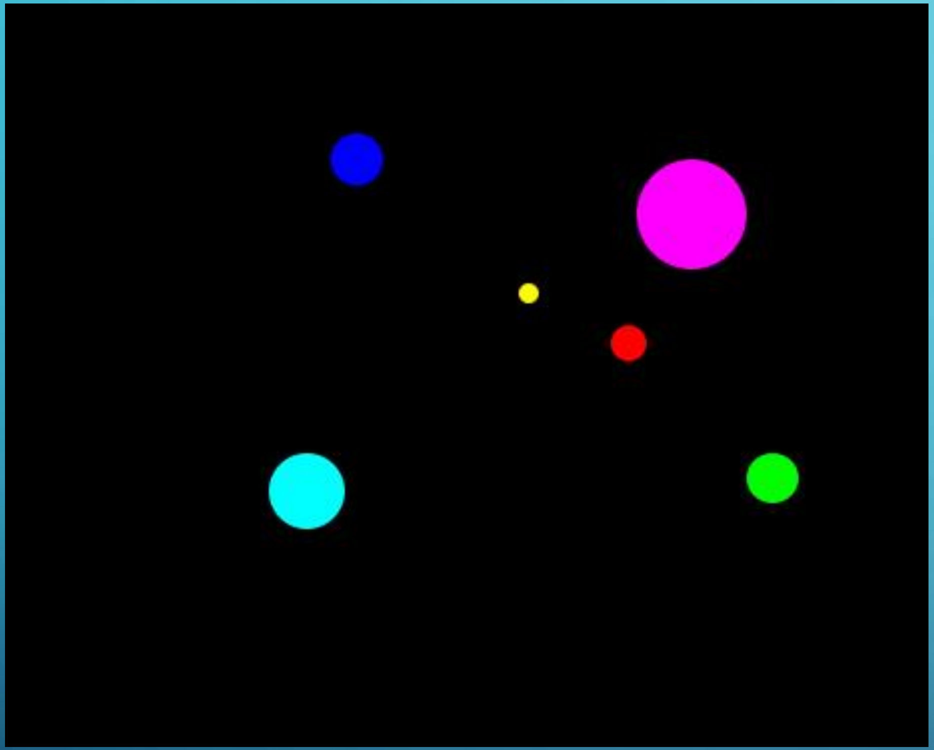


BARNES-HUT TREE WITH RNN

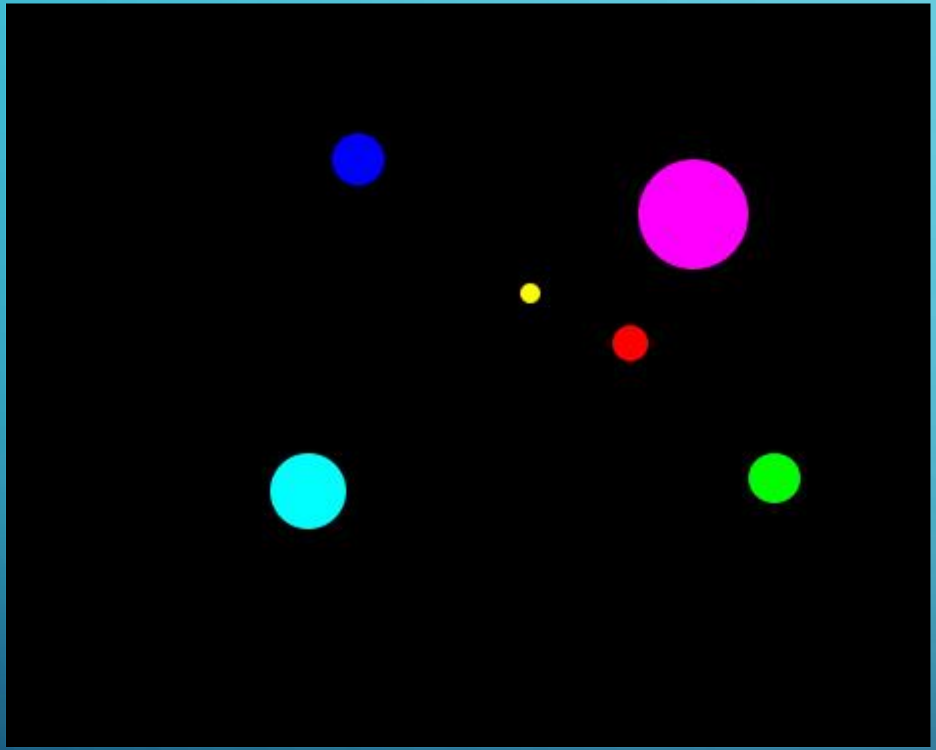


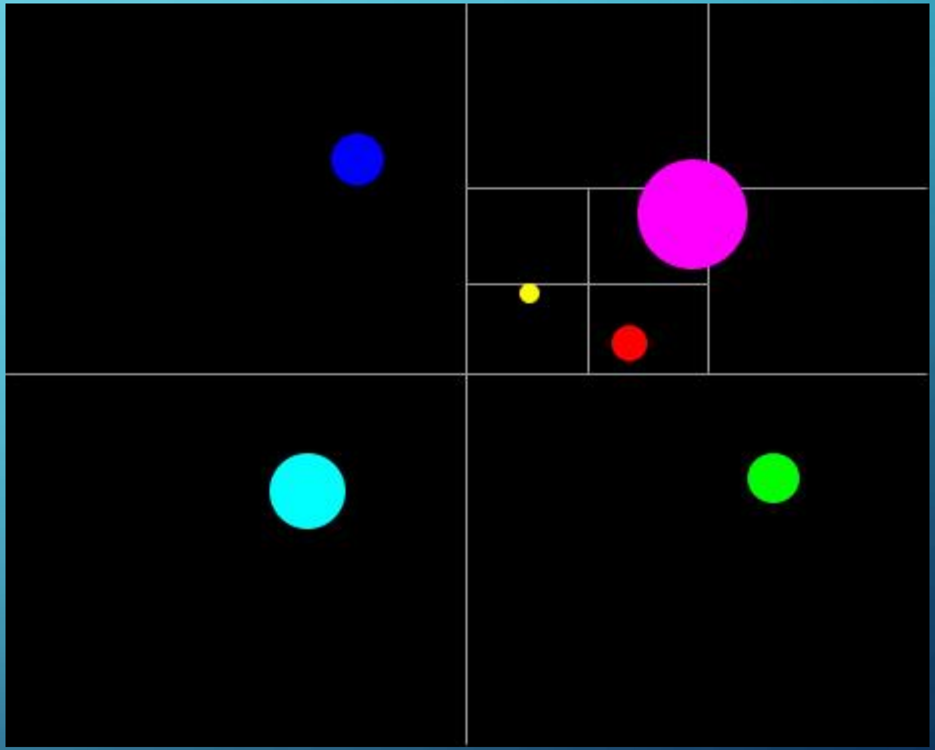


BARNES-HUT TREE WITH RNN



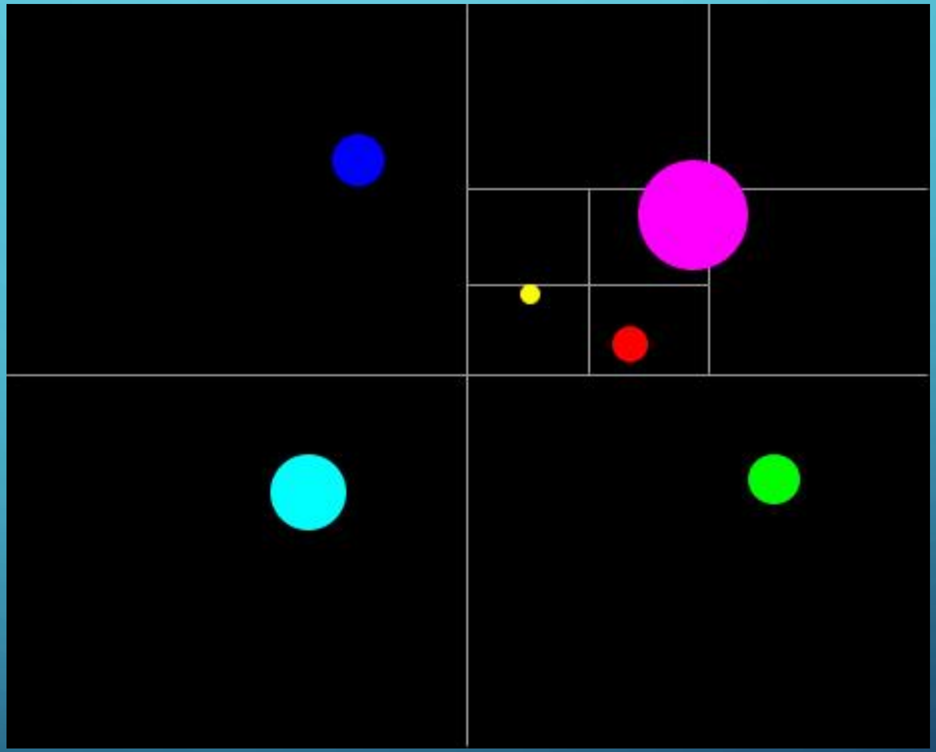
BARNES-HUT TREE WITH RNN





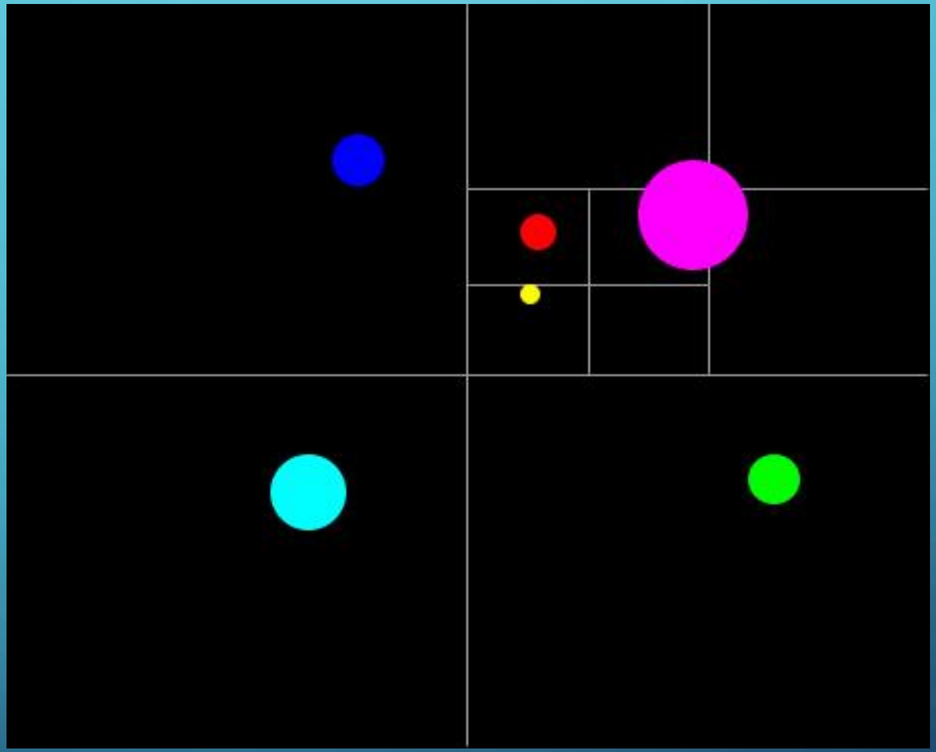
BARNES-HUT TREE WITH RNN

- We would like to analyse Medipix (red node)
- This time the top right corner is the hot zone



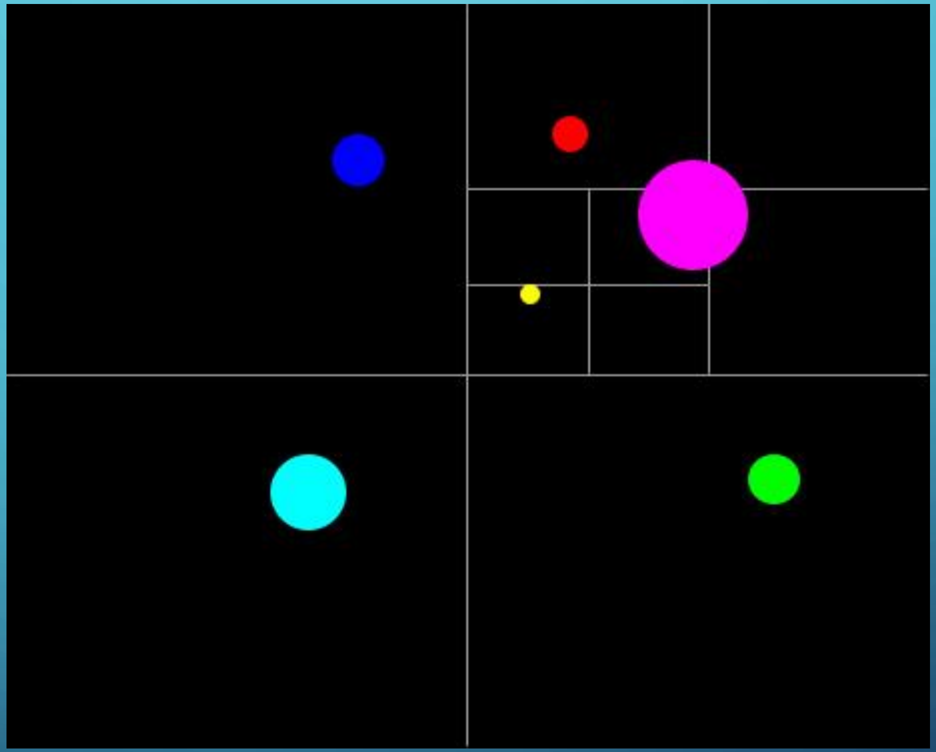
BARNES-HUT TREE WITH RNN

We move the red node, recompute the tree and store the state for future training



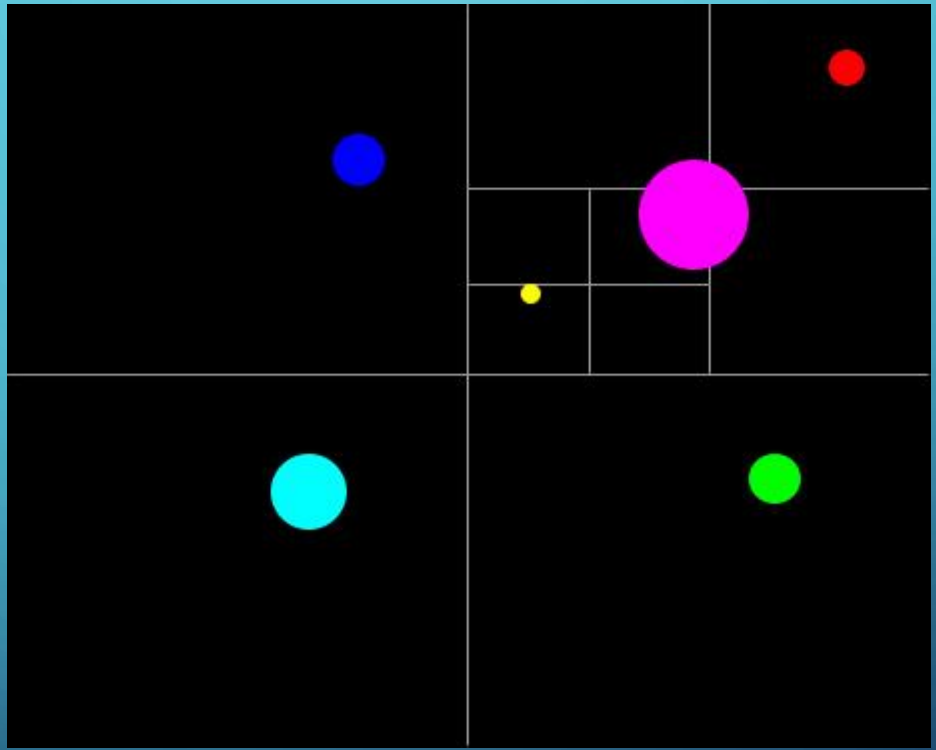
BARNES-HUT TREE WITH RNN

Red node advancing



BARNES-HUT TREE WITH RNN

Red node advancing to final position

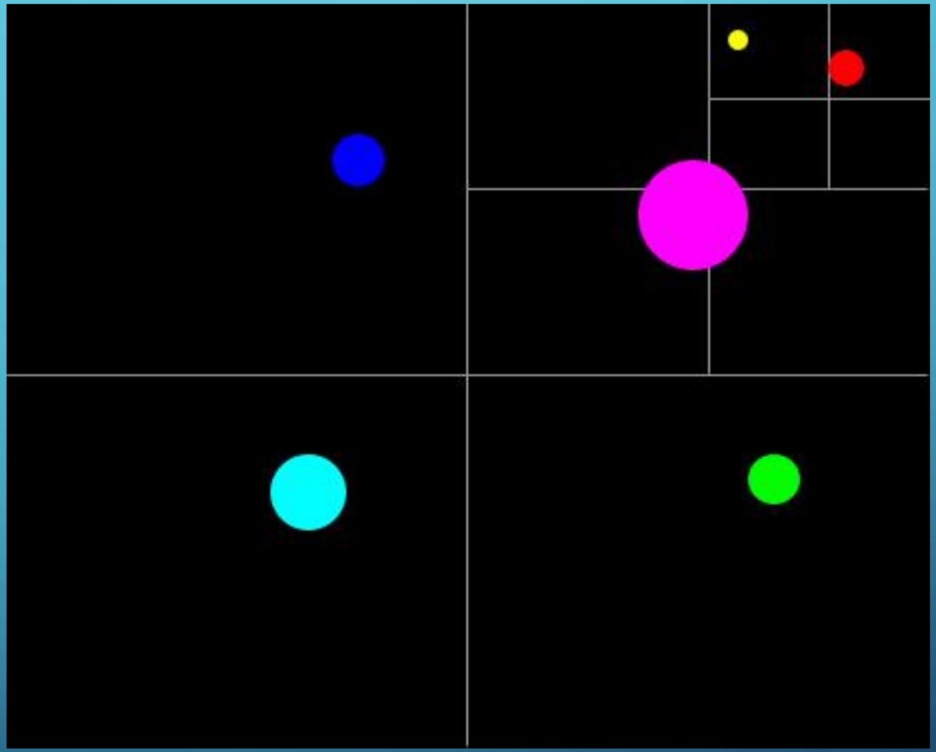


BARNES-HUT TREE WITH RNN

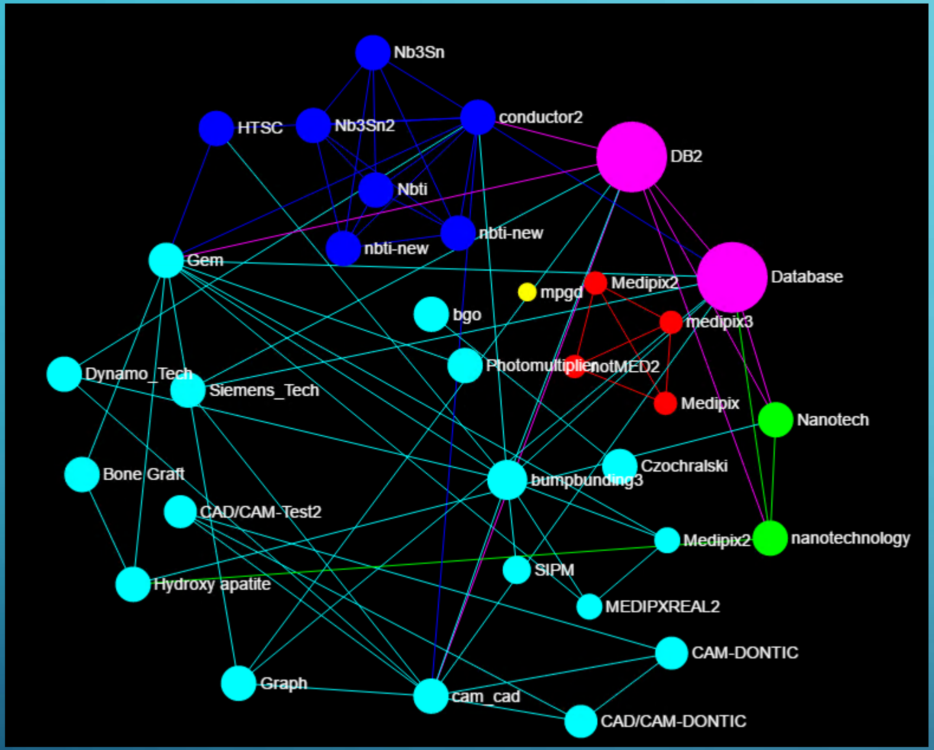
- States stored for one node (Medipix)
- Train a model for Medipix
- The model will move the node through the regions

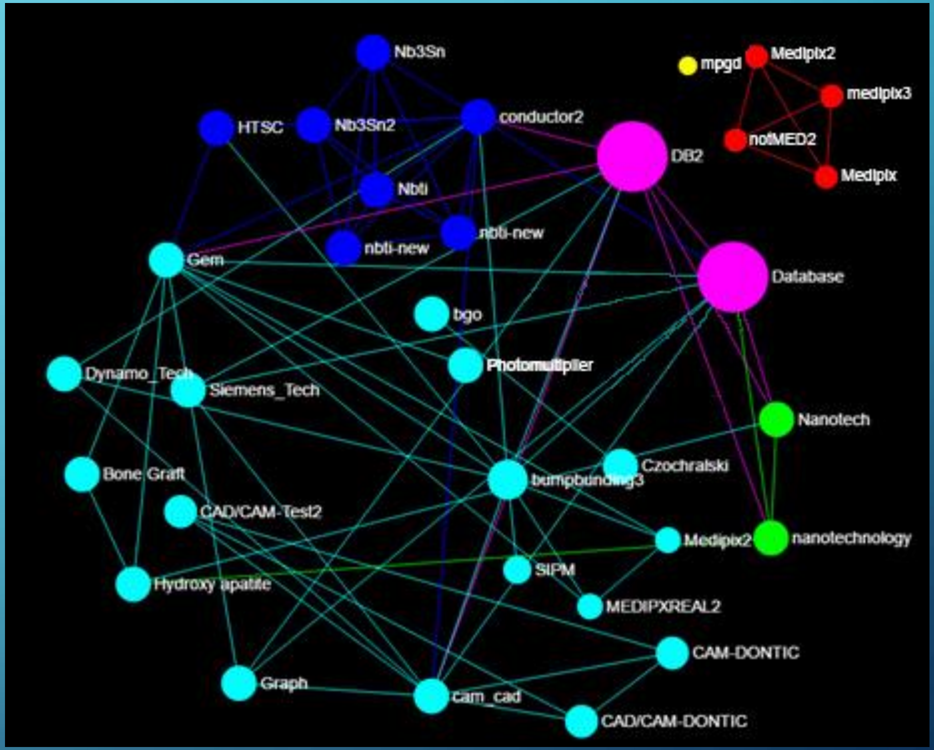
BARNES-HUT TREE WITH RNN

Graph preprocessed based on Medipix (red) and mpgd (yellow)



BARNES-HUT TREE WITH RNN





THANK YOU