SE4AI: Design of Neural Network Architectures to Support AI and Machine Learning Formalisms working Side-by-Side as a Team

Sponsor: OUSD(R&E) | CCDC

By
UMD Team: Dr. Mark Austin and Maria Coelho
Stevens Team: Dr. Mark Blackburn (Presenter)
October 20-21, 2021

AI4SE/SE4AI Workshop

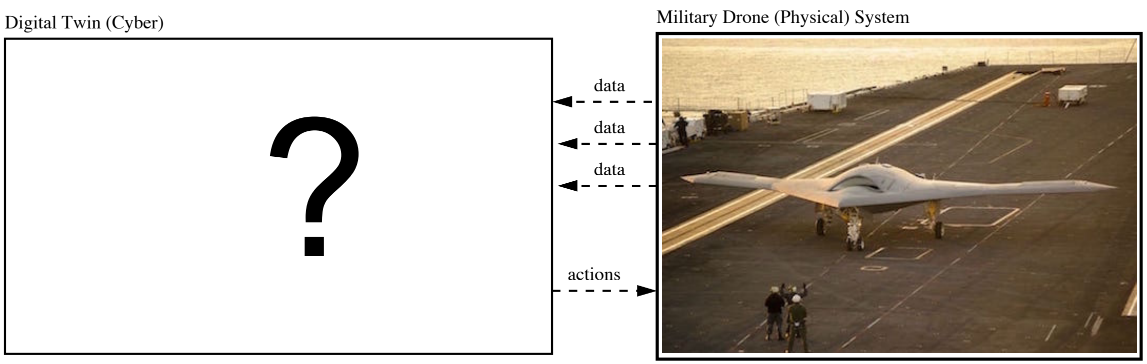
www.sercuarc.org

Cleared for Public Release

Motivation: Digital Twins

Definition (2000 – today)

- Virtual representation of a physical object or system that operates across the system lifecycle (not just front end).



Required Functionality

- Mirror implementation of physical world through real-time-monitoring and synchronization of data with events.
- Provide algorithms and software for observation, reasoning and physical systems control.

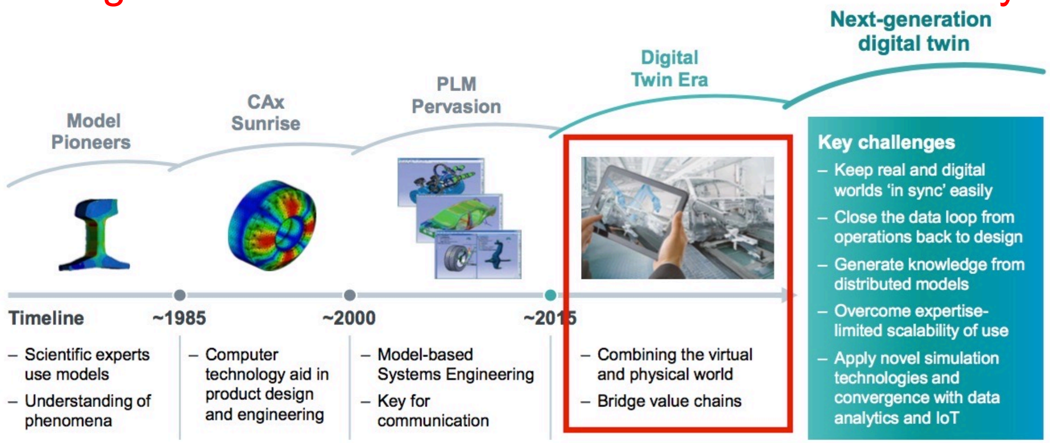
Many Application Domains

- NASA, manufacturing processes, building operations, personalized medicine, smart cities, among others.

Importance and Timeliness (Why?)

Business Drivers (Why this project is timely?)

Siemens, IBM, now see Digital Twin Era as the successor to MBSE with SysML



Digital Twin Era (Business Spin)

- New methods and tools for model-centric engineering.
- New operating system environments for observation, reasoning and physical systems control.
- Superior levels of system performance, agility, economy, etc.

Technical Implementation (2020, Google, Apple, Amazon, Siemens, IBM …)

- AI and ML will be deeply embedded in new software and algorithms.

Proposed Approach (Why?)

Definition of AI and ML

- AI: Knowledge representation and reasoning with ontologies and rules. Construction of semantic graphs, executable event-based processing, multi-domain reasoning.
- ML: Modern neural networks (closely related to signal processing of data streams). Data Mining. Input-to-output prediction, Learn structure and sequence. Identify objects, events, anomalies. Remember stuff.

AI/ML Strengths and Weaknesses

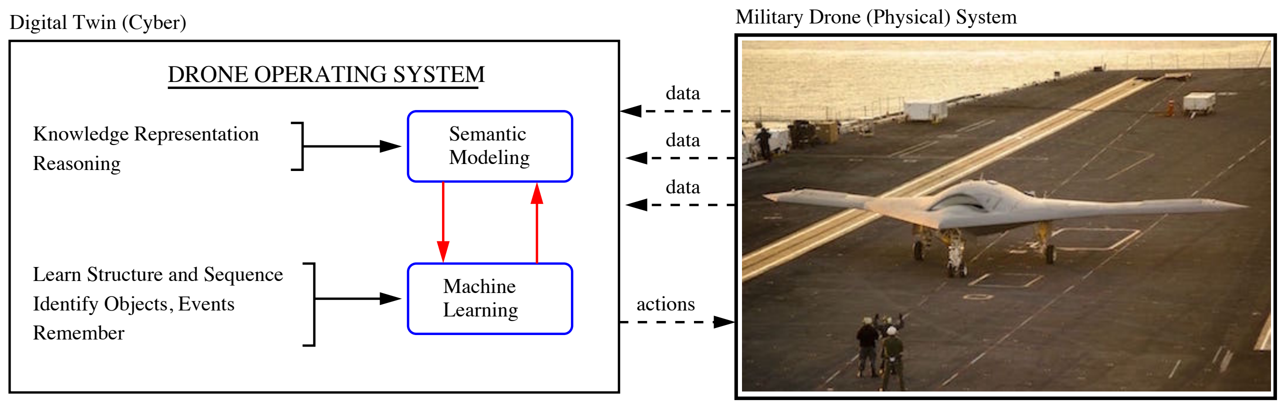
State-of-the-art AI and ML technologies are fragmented in their capability:

- AI provides a broad view of concepts needed for reasoning. Decision making processes are transparent; semantic graphs are flexible.
- Semantic reasoning is decision making in-the-moment (no memory).
- Data mining algorithms can organize information from large data sources.
- ML procedures developed to solve very specific tasks.
- ML decision making procedures lack transparency.
- ML procedures can identify anomalies (events) in streams of data.

Proposed Approach (What's New?)

Digital Twins (What's New?)

- Explore design of digital twin architectures that support AI and ML formalisms working side-by-side as a team.



Key Research Challenge

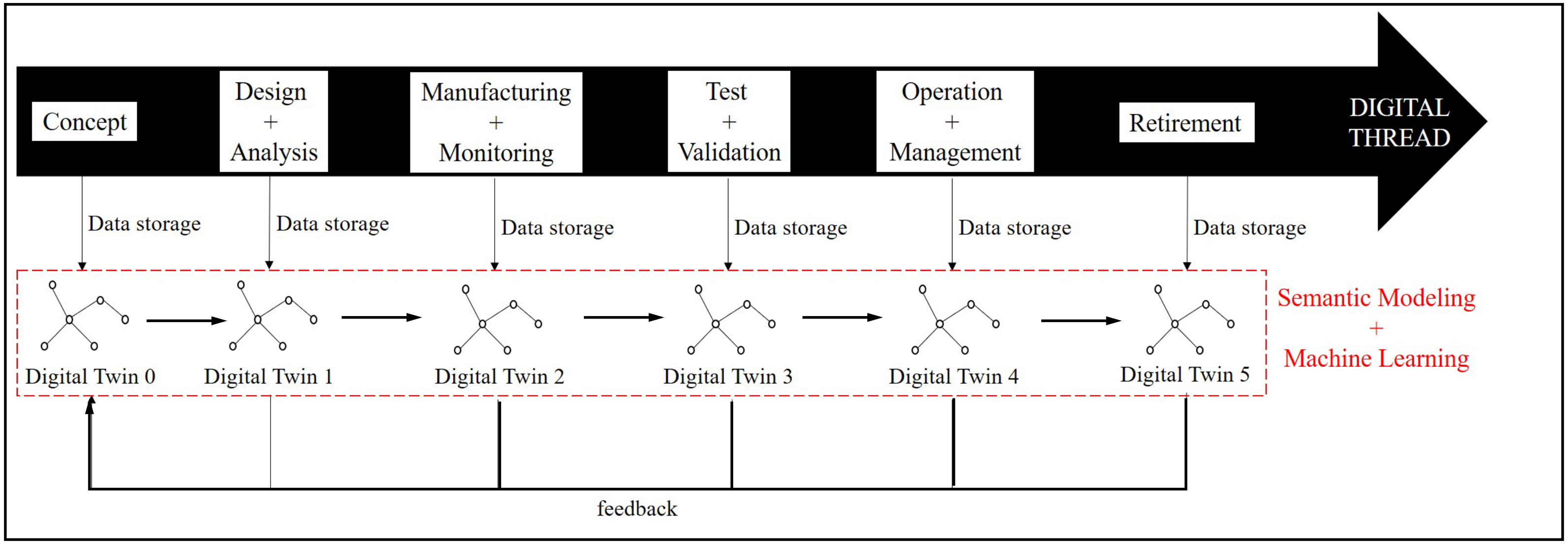
- How to design digital twin elements and their interactions to support: (1) methods and tools for model-centric engineering, and (2) digital twin operating system environments for observation, reasoning, control.

Project Success (What does it look like?)

- Knowledge to guide architectural development of future digital twins enabled by AI / ML technology.

Digital Twins → Digital Threads (What?)

Cradle-to-Grave Lifecycle Support (Digital Threads)



Observation: A lot of model-centric engineering boils down to representation of systems as graphs and sequences of graph transformations punctuated by decision making and work / actions.

Reasonable Starting Point: Understand the range of possibilities for which machine learning of graphs and their attributes support and enhance activities in model-centric engineering and systems operation.

Presentation Objectives

1. Identify neural network architectures and strategies of learning for variety of graph structures and their attributes.

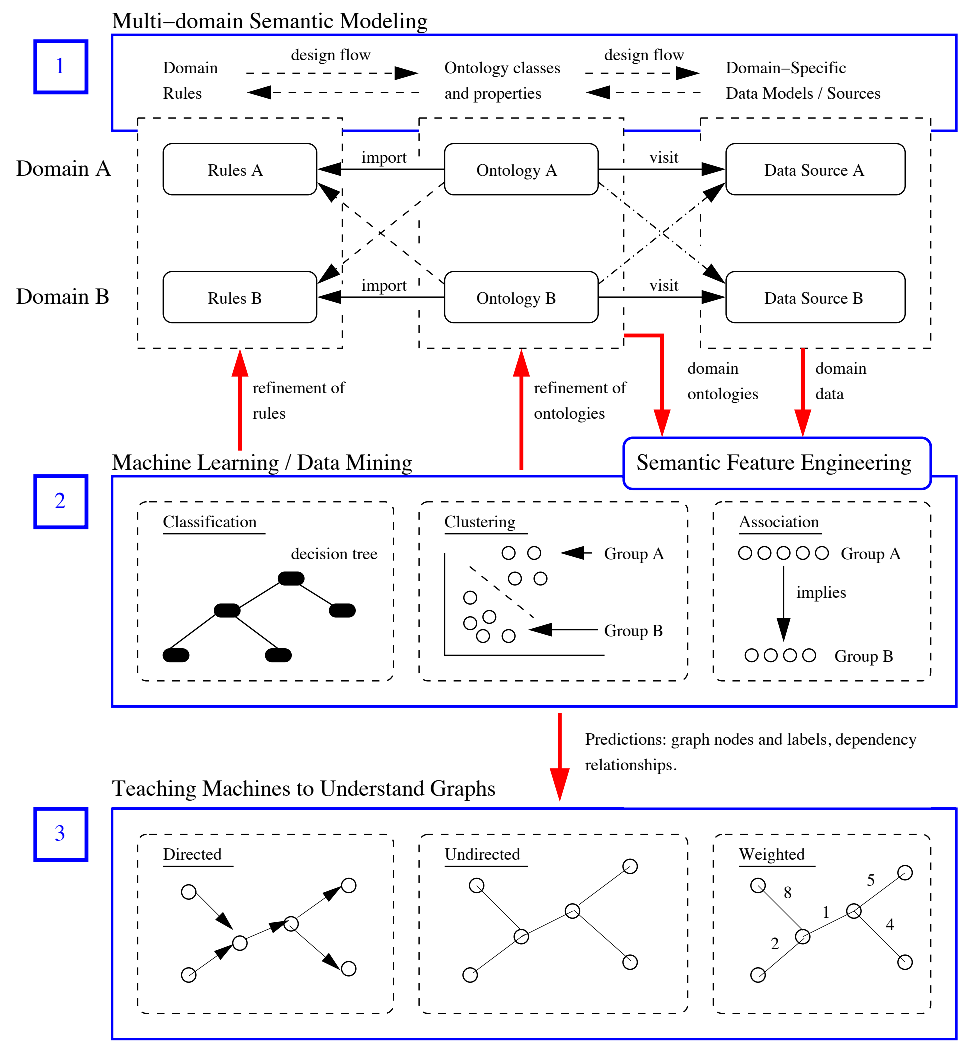
2. Exercise machine architecture and strategies of learning on case study problems:

3. Briefly show Maria's applications: water distribution system and urban metrorail system.

4. Explore the effects of graph size on learning performance.

Digital Twin Architecture

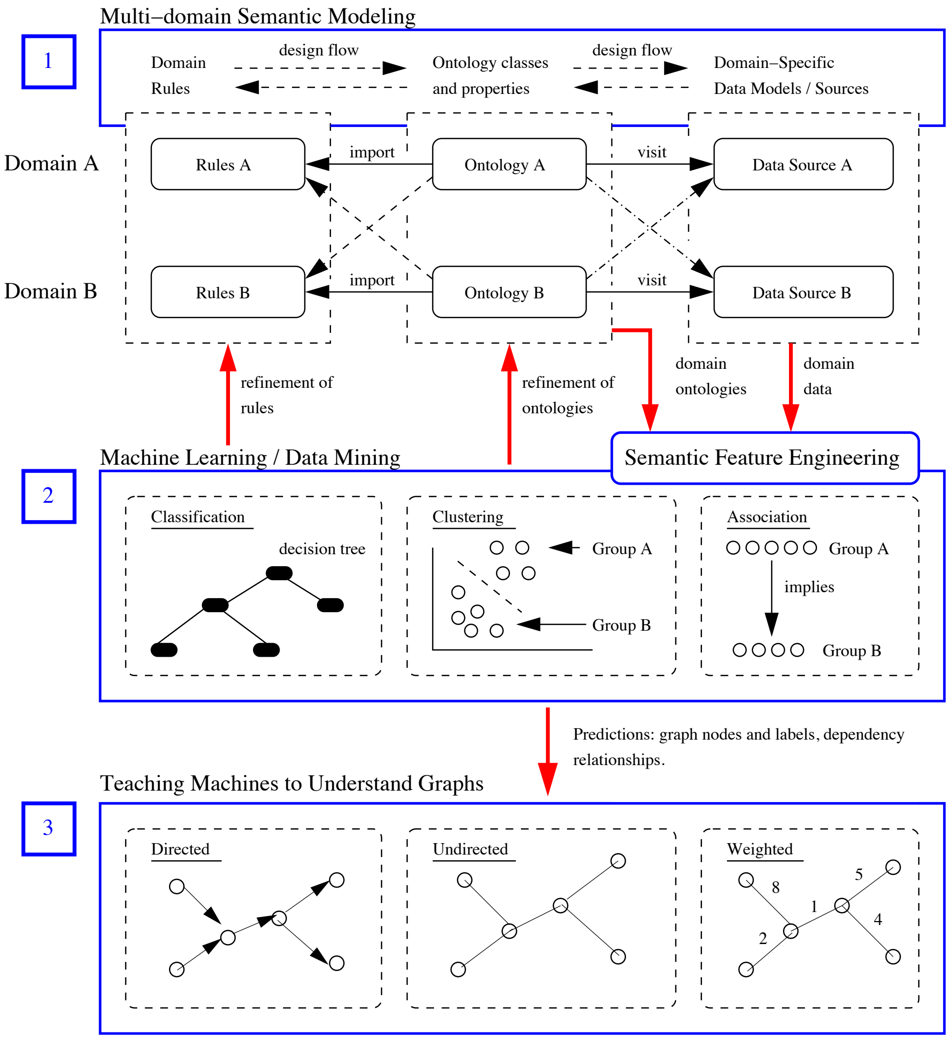
- Business Drivers
- Post Incubator Project
- Real-World Considerations
- **Step 1:** Multi-Domain Semantic Modeling
- **Step 2:** Semantic Modeling + Data Mining
- **Step 3:** Teaching Machines to Understand Graphs
- Opportunities and Extensions
- Plan of Work



So what will the machine learning do?

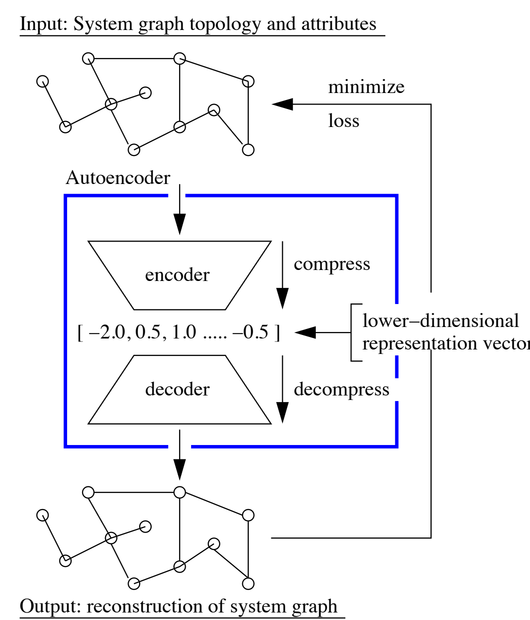
Presentation Objectives

Focus on Machine Learning of Graphs and Model-Centric Engineering.



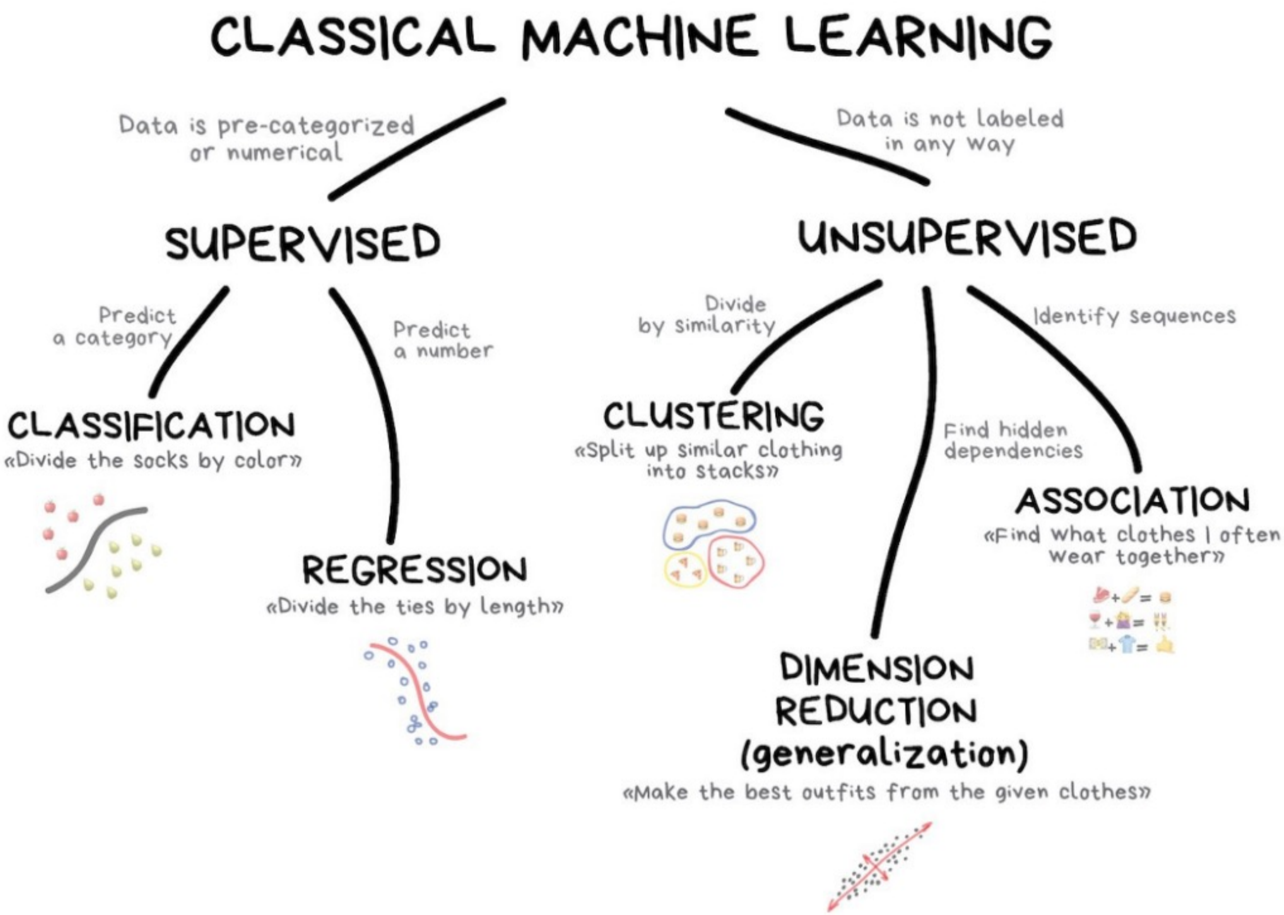
Observation: A lot of model-centric engineering boils down to representation of systems as graphs and sequences of graph transformations punctuated by decision making and work / actions.

Hence: Explore opportunities for training machines to understand graphs.



Machine Learning

Algorithms that use statistics to **learn patterns and hidden insights in data** without being explicitly programmed for it.



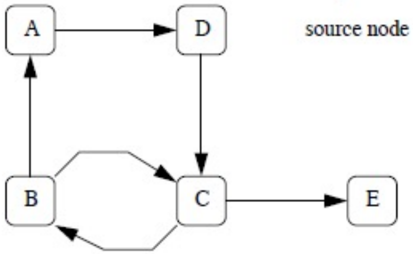
SYSTEMS ENGINEERING RESEARCH CENTER

Graphs and Graph Analysis

A **graph** is defined as G = (V, E), where V is a set of vertices (i.e. nodes), E = set of edges, and each edge is formed from pair of distinct vertices in V.

Traditional Approach to Graph Analysis

- Traditional approaches to graph modeling employ adjacency matrices.
- Topology properties can then be extracted through **graph analysis** tasks:
  - Connectivity analysis, traceability analysis, cycle detection, shortest path identification, etc.



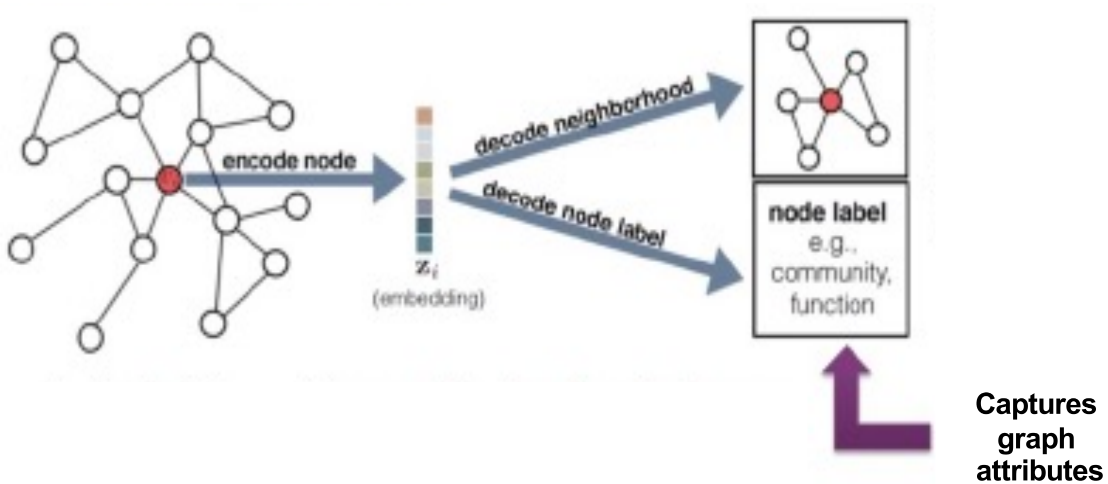
target node → A B C D E; source node → A

source node \ target node	A	B	C	D	E
A	0	0	0	1	0
B	1	0	1	0	0
C	0	1	0	0	1
D	0	0	1	0	0
E	0	0	0	0	0

Graph Analytics

Machine Learning Approach to Graph Analytics

- Adjacency matrices suffer from data sparsity, high-dimensionality, and a lack of support for capturing graph attributes.
- Surge in graph embedding approaches.
- Output vectors are statistical, should be interpreted as **graph analytics**.
- Learned embeddings could advance various downstream learning tasks:
  - Node Classification
  - Node Clustering
  - Anomaly Prediction
  - Attribute Prediction
  - Link Prediction
  - Recommendation
  - Etc.



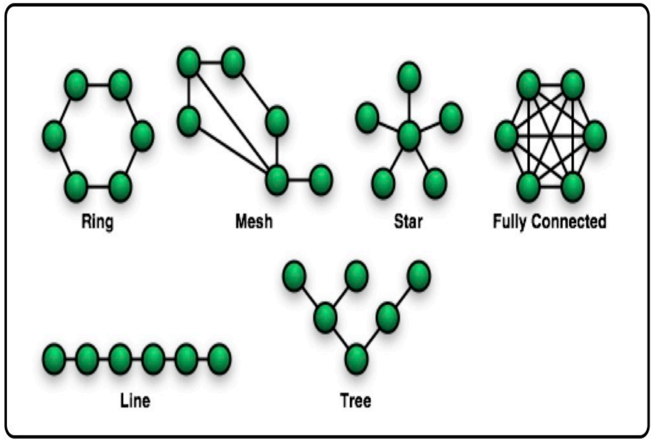
Preliminary Research at UMD, 2019-2020

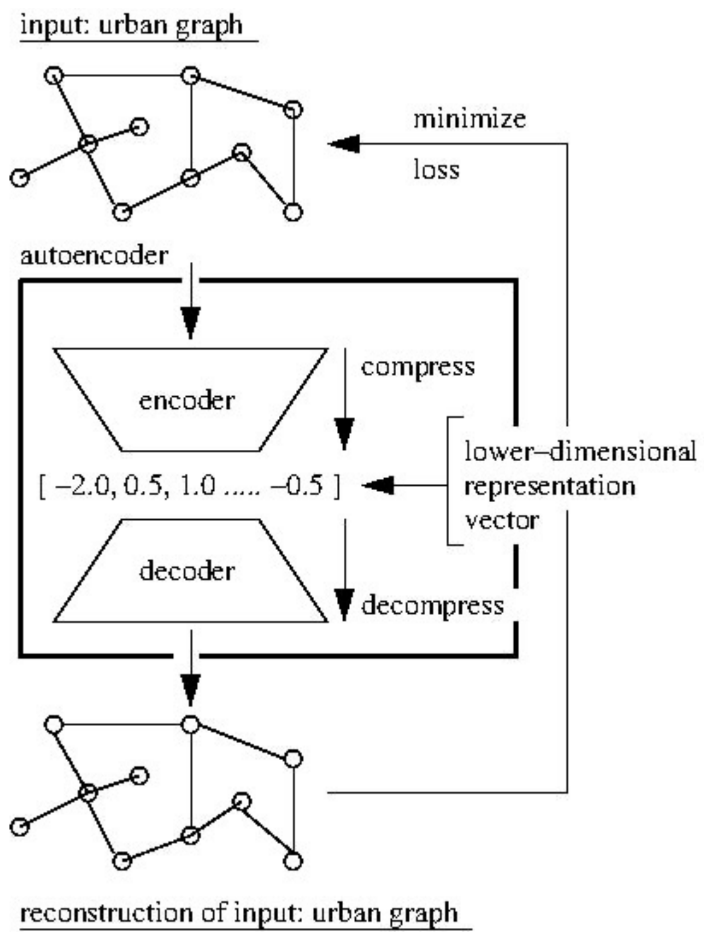
Graph AutoEncoder Approach

Graph AutoEncoders (GAE)

Deep neural network architectures trained to reconstruct their original graph input.

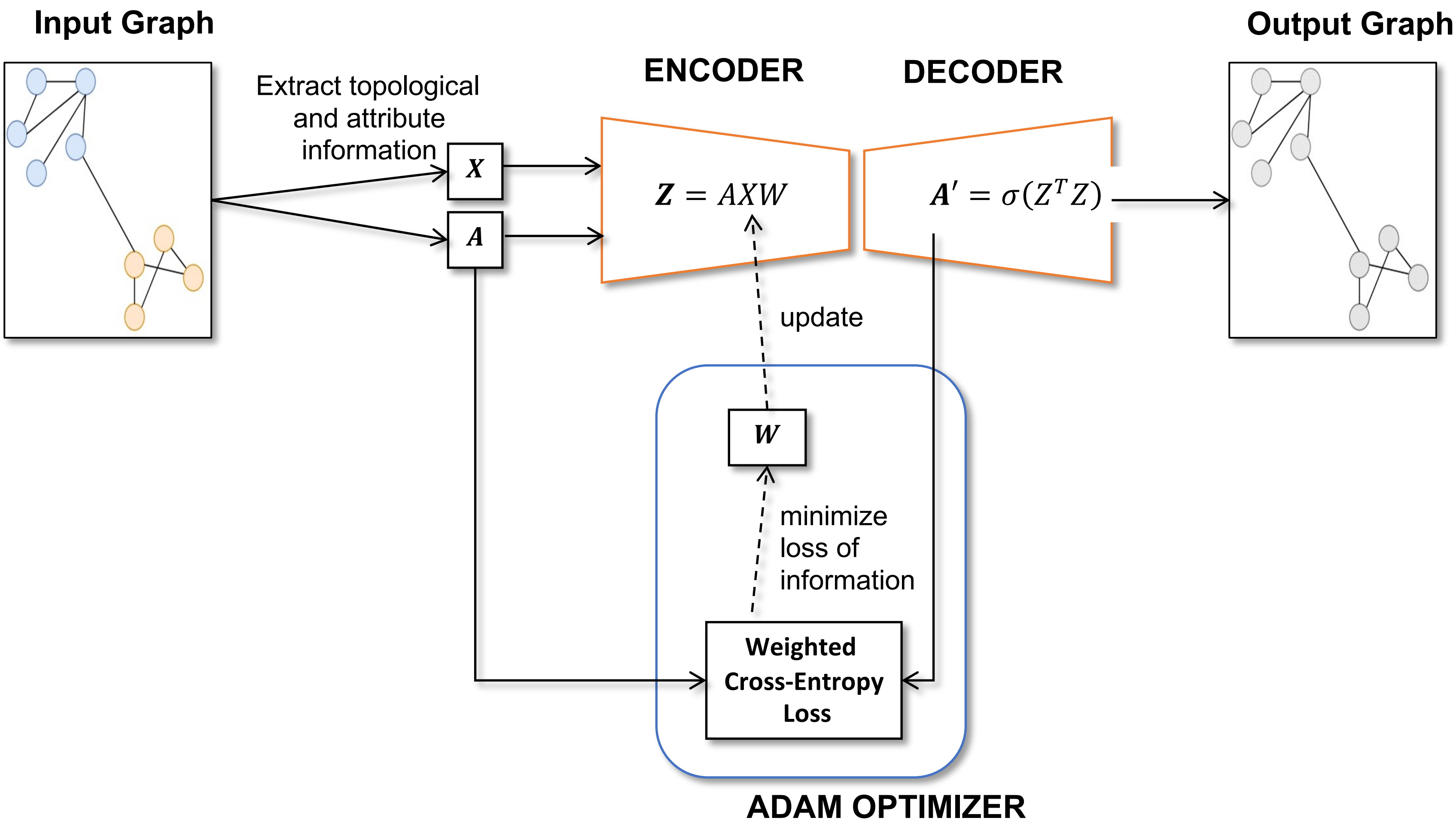
Common Graph Topologies



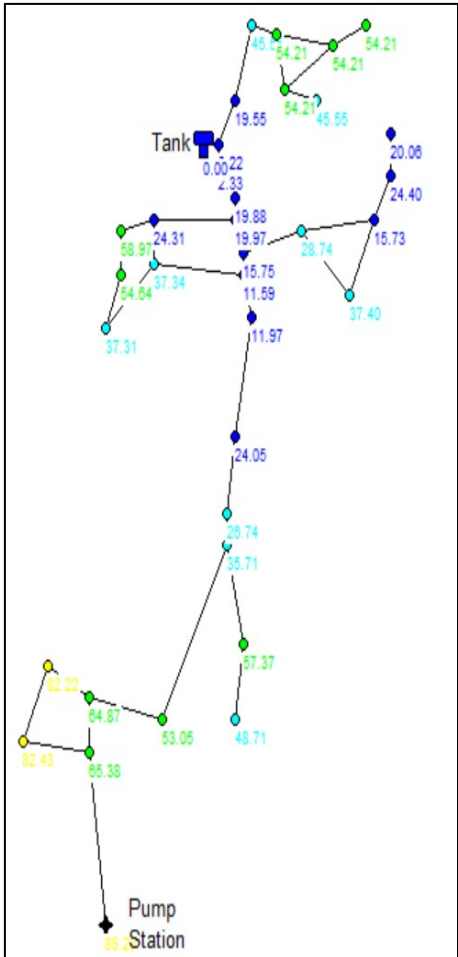


Graph AutoEncoders (GAE)

Mathematical Procedure: Designed to ensure numerical optimization will converge (play nicely).



AutoEncoding an Urban Graph

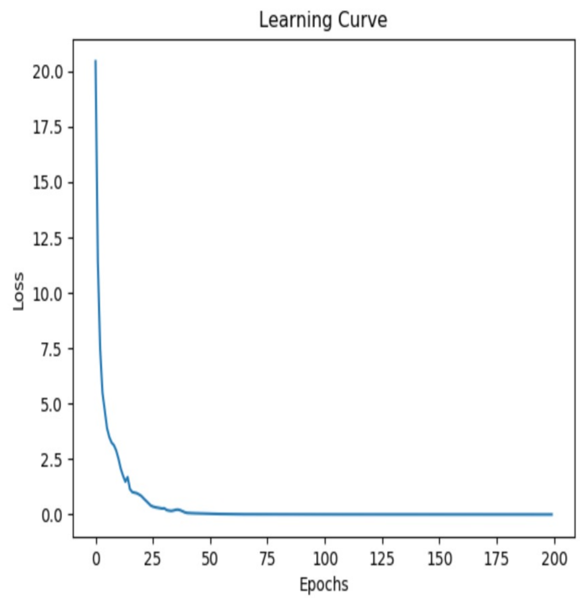


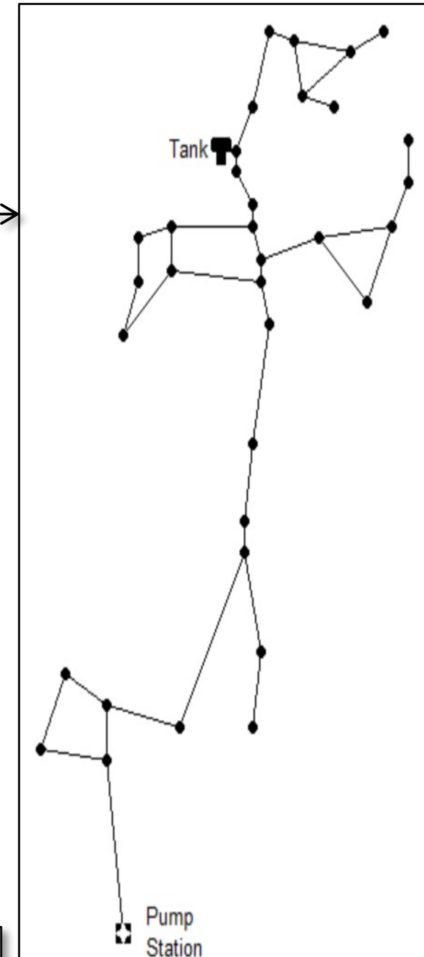
Input Graph

ENCODER **DECODER**

$$\boldsymbol{Z} = AXW$$

$$\boldsymbol{A'} = \sigma(Z^T Z)$$



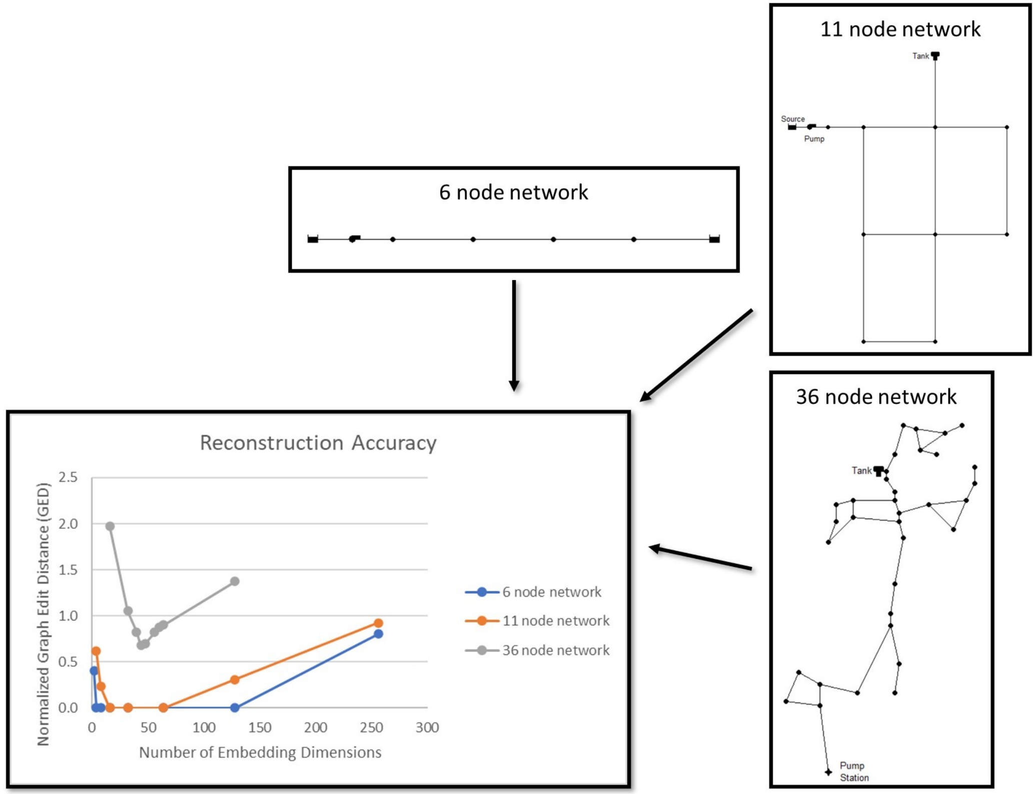


Output Graph

Number of embedding layer neurons = 74
Minimum Loss = 0.00196
Input/Output isomorphism = True

Experiments with Graph AutoEncoder

- Explored different architectures and their effect on reconstruction accuracy.
- Window of convergence where good reconstruction can be achieved.
- Need to understand **neural network architectural requirements** for accurate reconstruction of graph structures.

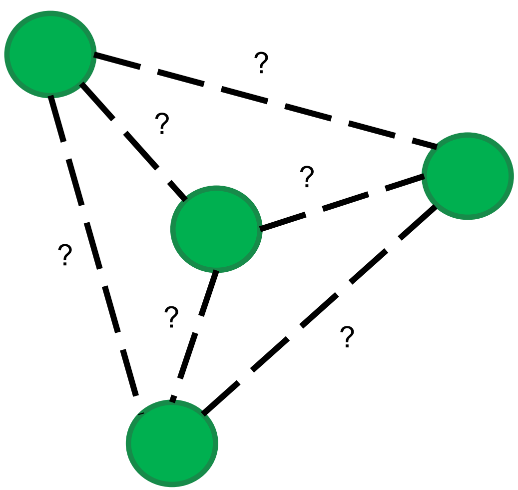


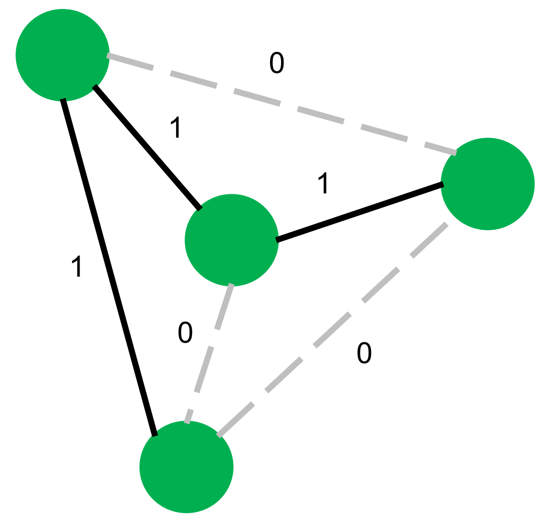
Key Takeaways: Mathematical procedures for graph learning with autoencoders work, but only over a limited range. Are the underlying algorithmic assumptions really needed? Can we improve the whole process?

Frame Graph Learning as a Binary Classification Problem

Learning Graph Structure

- Learning the structure of a graph can be framed as a **binary classification problem**.
- If a connection between nodes exists → output 1.
- If a connection between nodes does not exists → output 0.

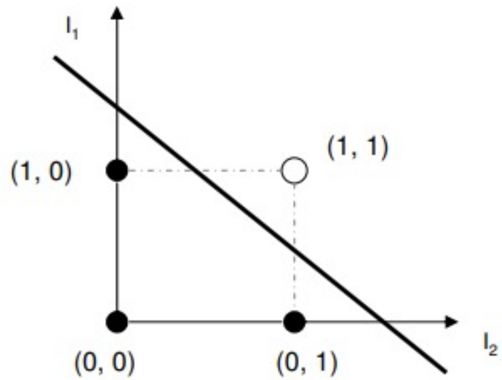




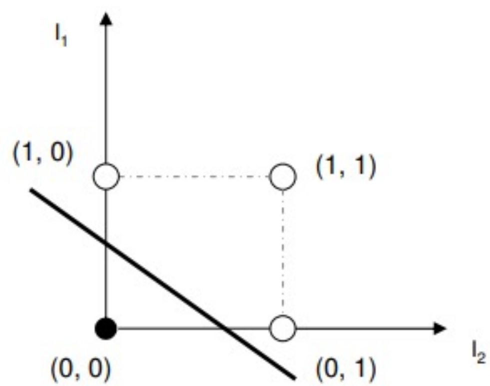
Classic Binary Classification Problem

Binary Classification (XOR Problem)

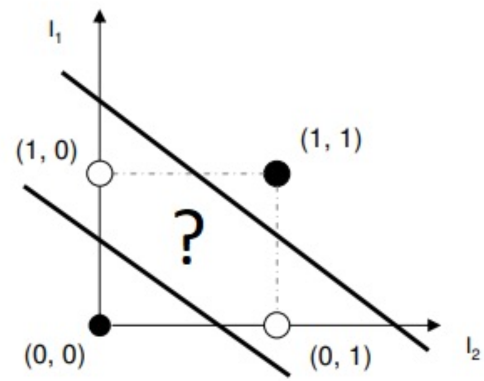
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

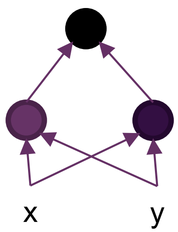


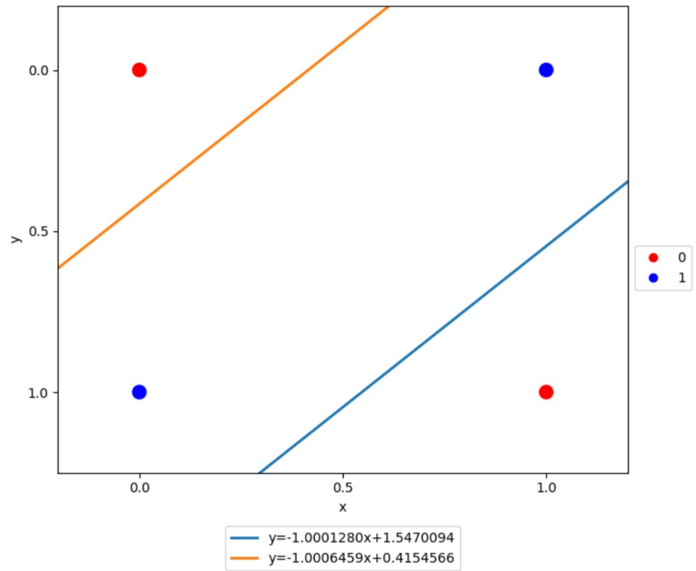
Trivial Implementation in TensorFlow

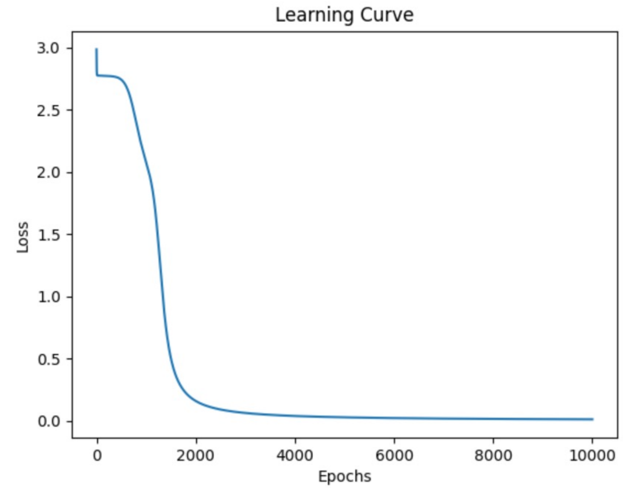
<u>Topology:</u>

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

<u>Architecture:</u>



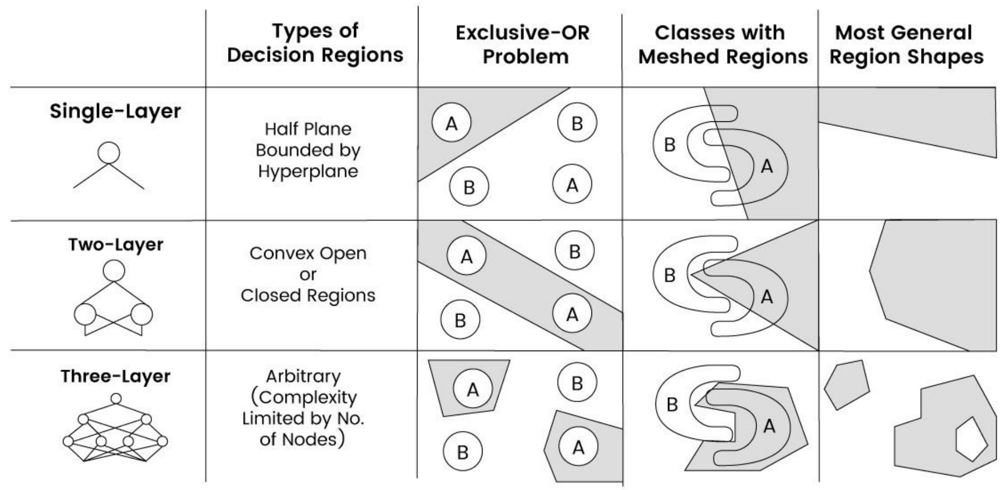




Neural Network Architecture for Classification

One Region

- One Hidden Layer
- Hidden Layer Size = number of hyperplanes required to form region
- Output neuron



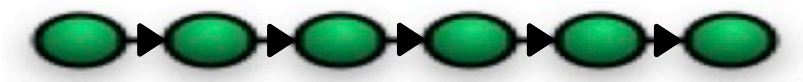
Source: Lippmann, R., 1987

Many Regions

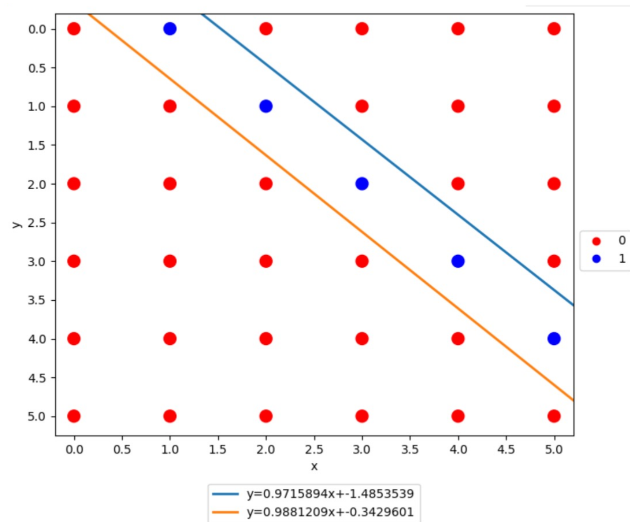
- Two Hidden Layers
- Hidden Layer 1 Size = number of hyperplanes required to form regions
- Hidden Layer 2 Size = number of regions
- Output neuron

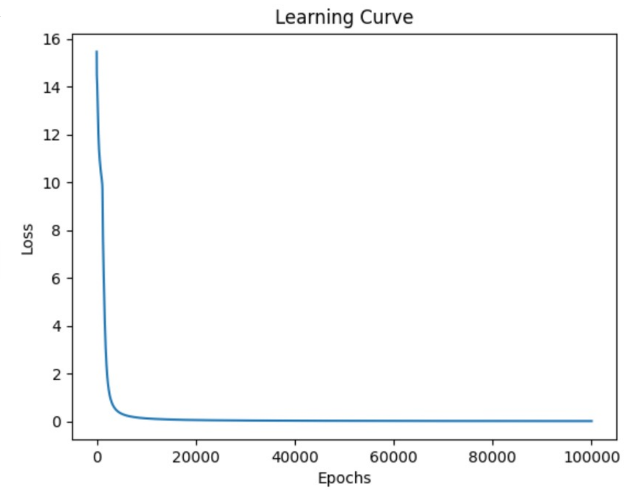
Directed Line Problem (One Region)

Topology:

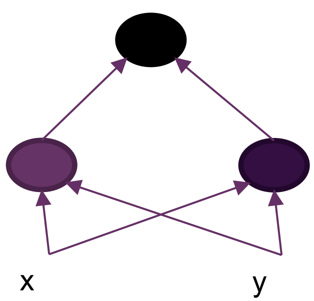


$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



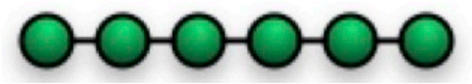


Architecture:

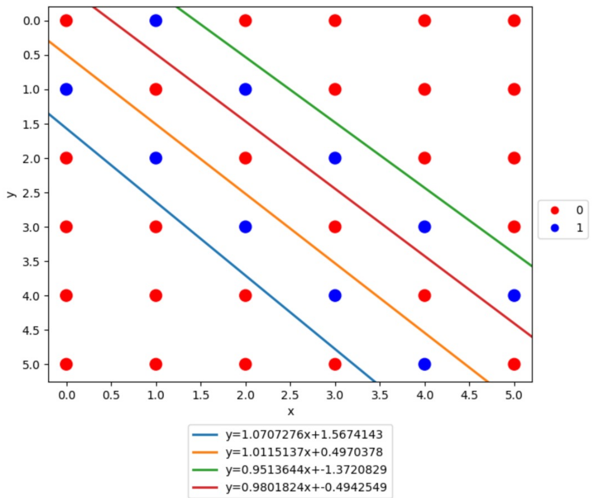


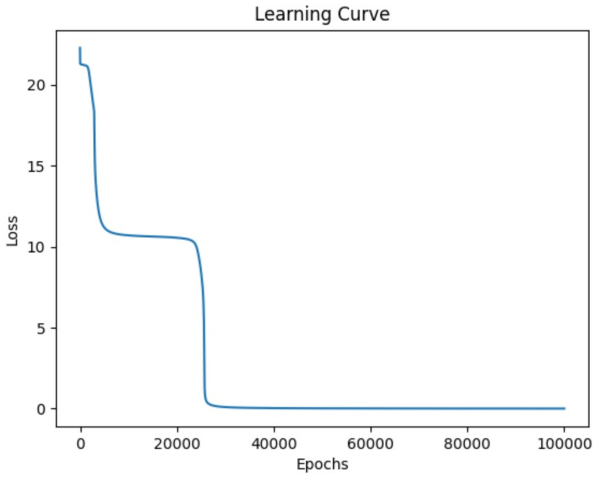
Line Problem (Multiple Regions)

Topology:

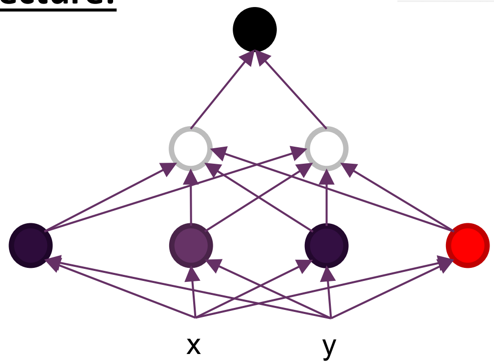


$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$





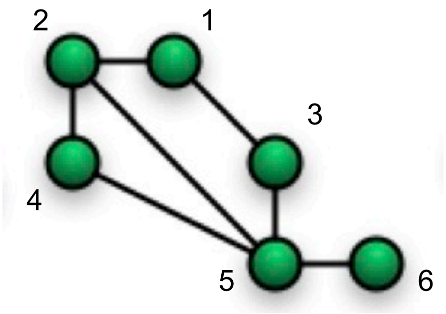
Architecture:



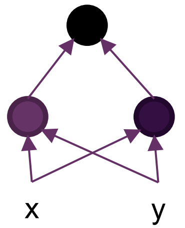
Graph Mesh Problem

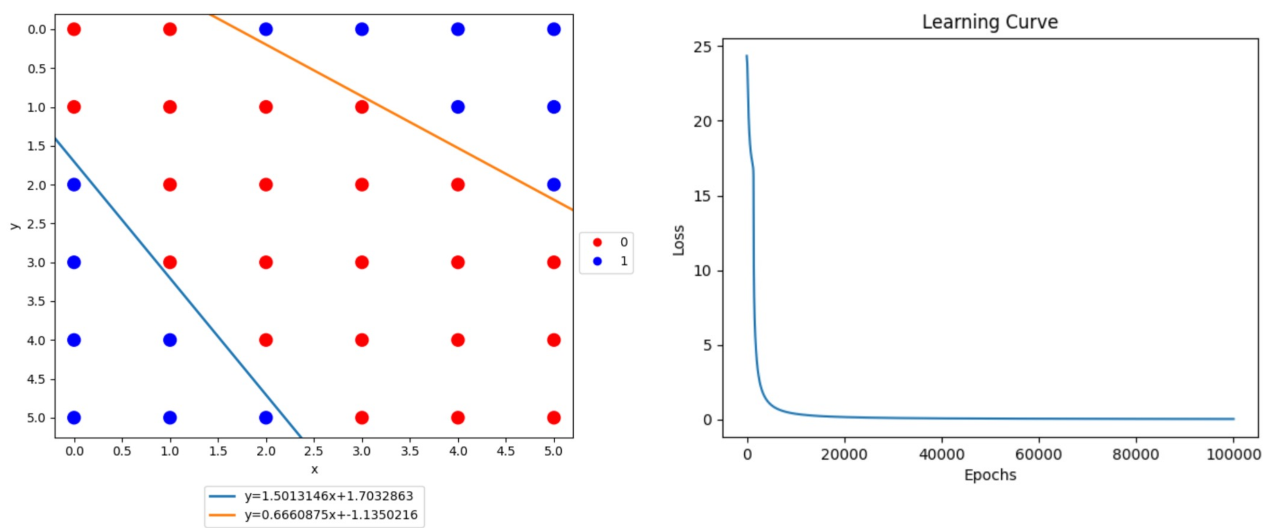
Topology:

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$



Architecture:

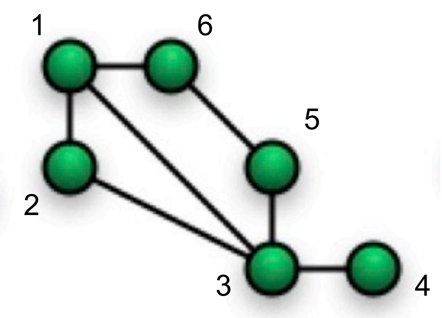




Graph Mesh Problem

Topology:

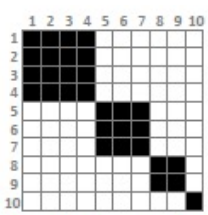
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



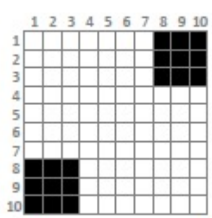
Architecture:

Visually hard to determine required architecture, **need for matrix reordering approach.**

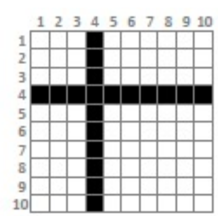
Matrix Reordering: Automation to Reveal Visual Patterns



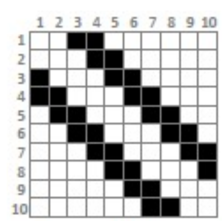
Block Pattern



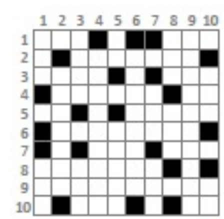
Off-diagonal Block Pattern



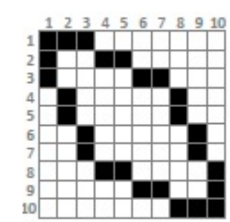
Line/Star Pattern



Bands Pattern



Noise Anti-Pattern



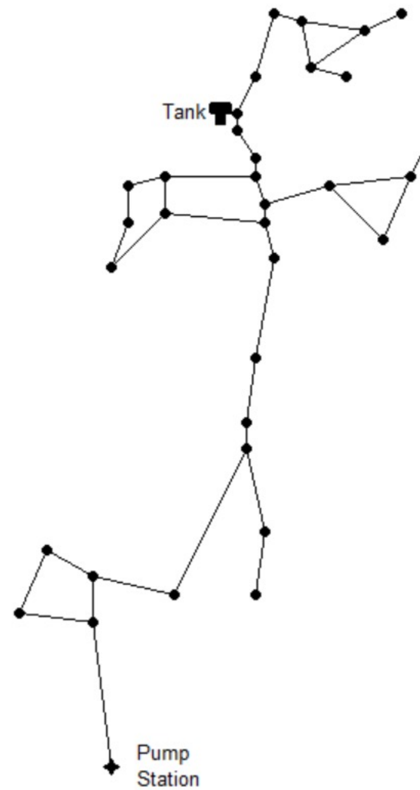
Bandwidth Anti-Pattern

Matrix Reordering for Graph Learning

Example: Water Distribution Network

<u>**Topology:**</u>

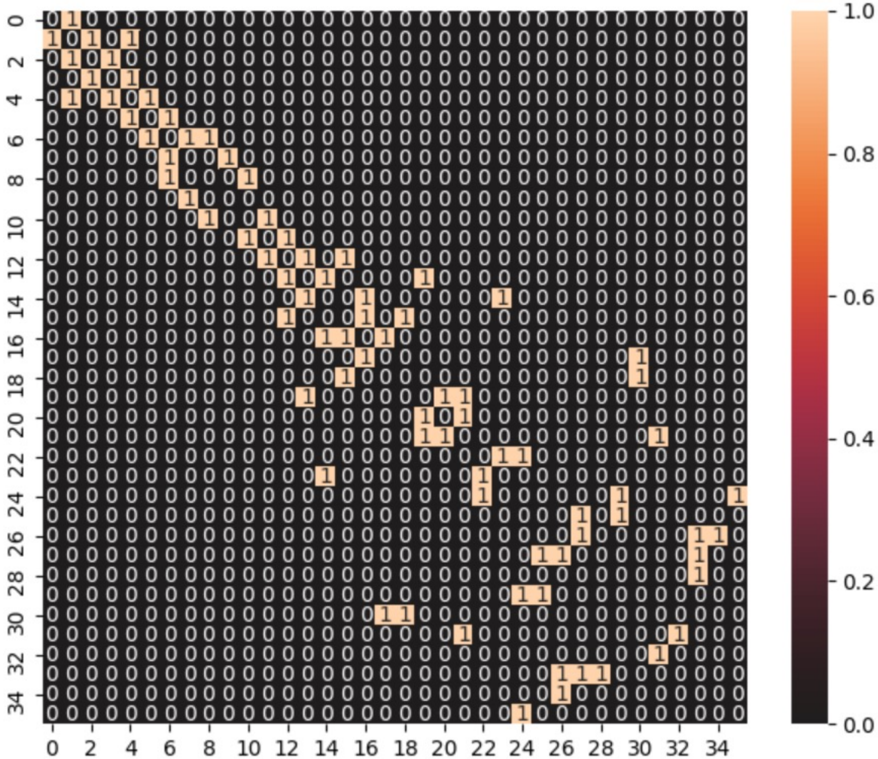
$$A = \left[\begin{array}{*{36}{c}}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

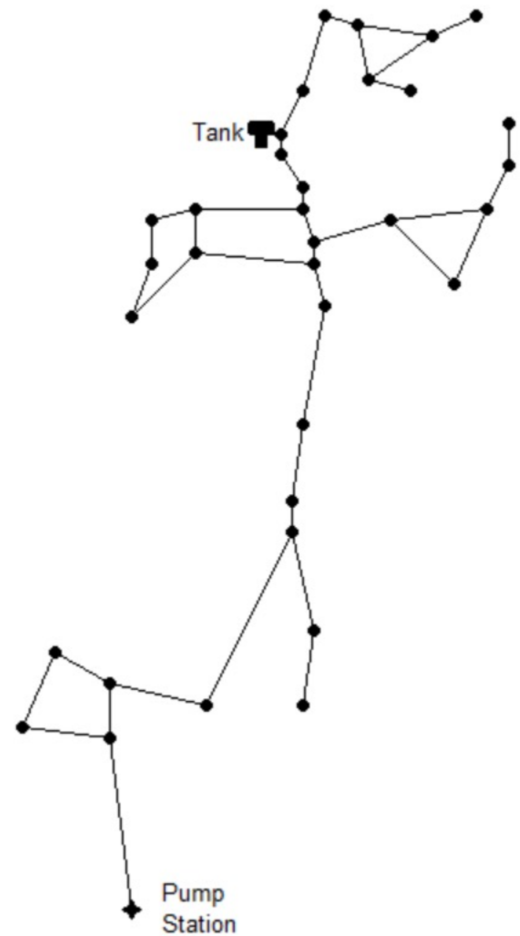


Matrix Reordering for Graph Learning

Water Distribution Network

Heatmap:



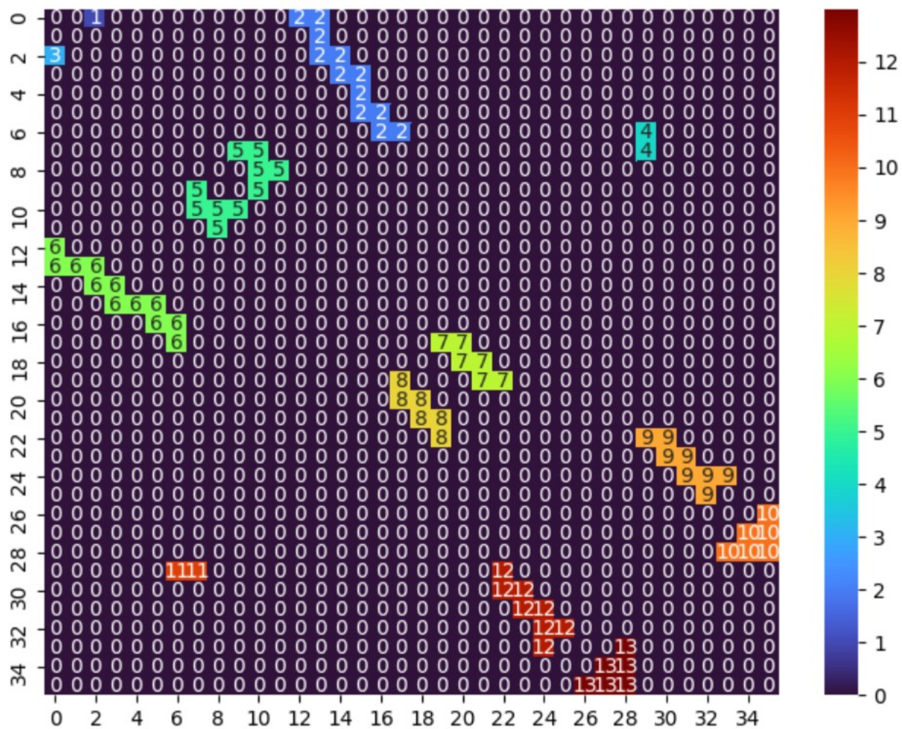


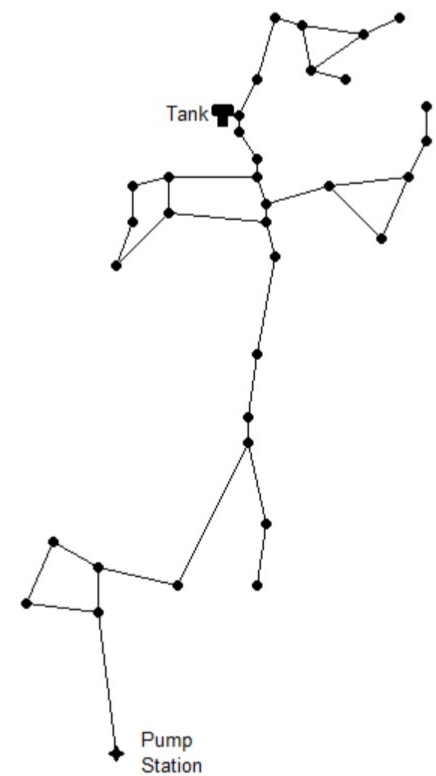
Matrix Reordering for Graph Learning

Matrix Reordered Water Distribution Network

Traveling Salesman:

runtime of
~2 secs.



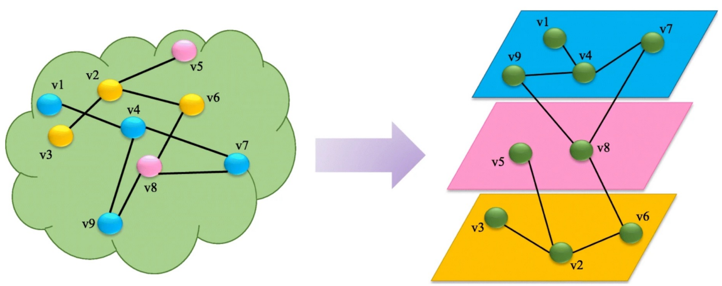


Transition to Networked Decomposition and Incremental Learning of Multi-Domain Graphs

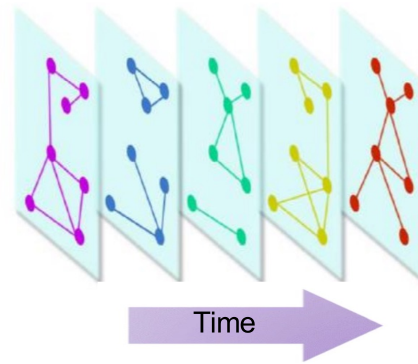
Transition to Networked Decomposition

Attribute-Driven Decomposition of System Graphs

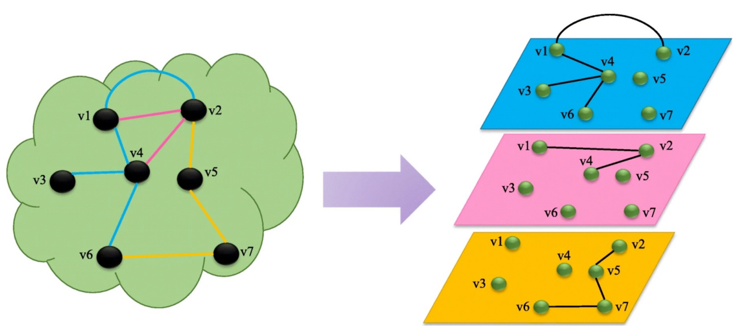
Component Characteristics



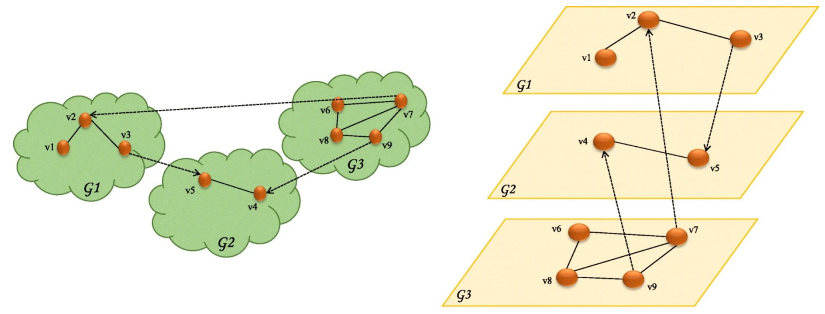
Temporal Characteristics



Connection Characteristics

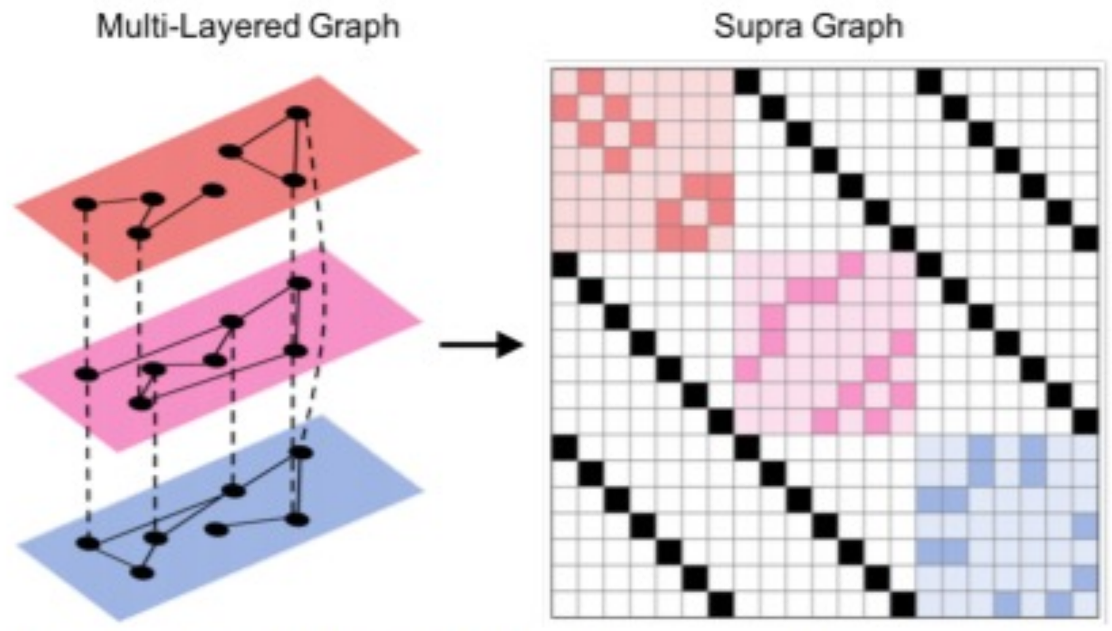


Spatial Characteristics



Transition to Networked Decomposition

Supra Graph Framework: Support for multi-layer / multi-domain graphs, graph zones, viewpoints, etc.

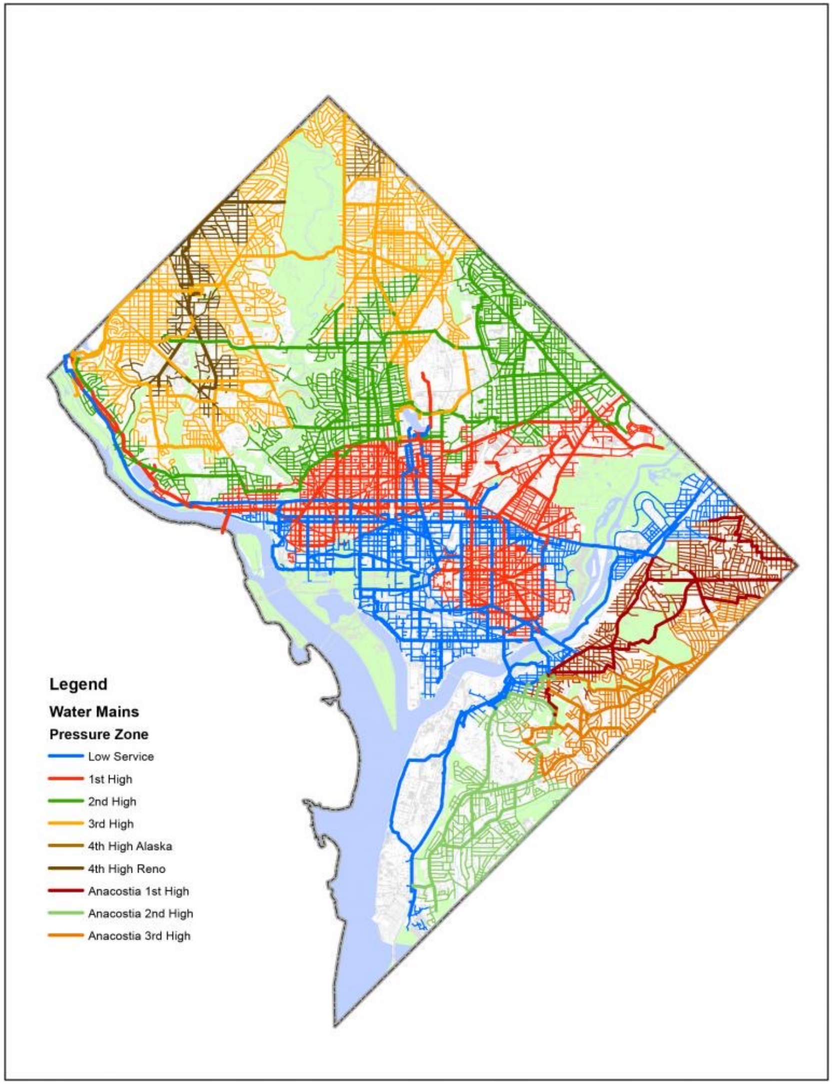


Shanthamallu et al., 2019

Transition to Networked Decomposition

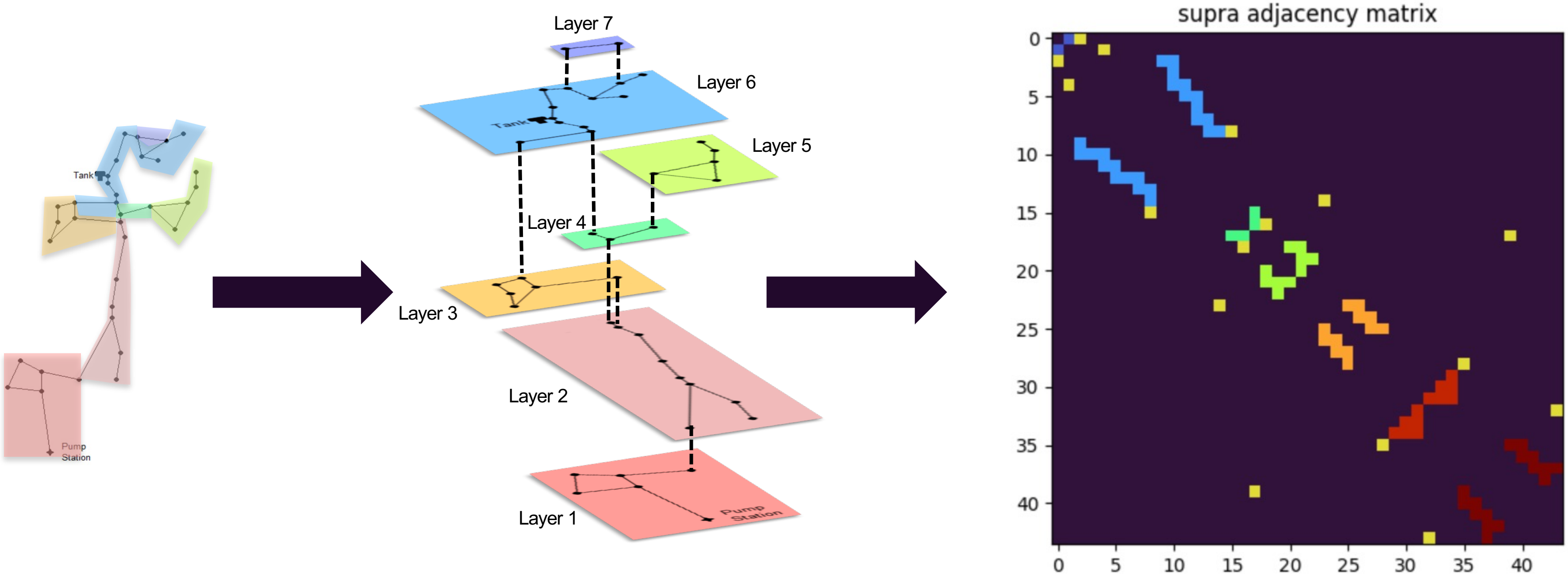
Example: Washington DC Water Network

- Washington DC's drinking water is distributed by elevation levels.
- Distribution network is divided into "pressure zones".



Transition to Networked Decomposition

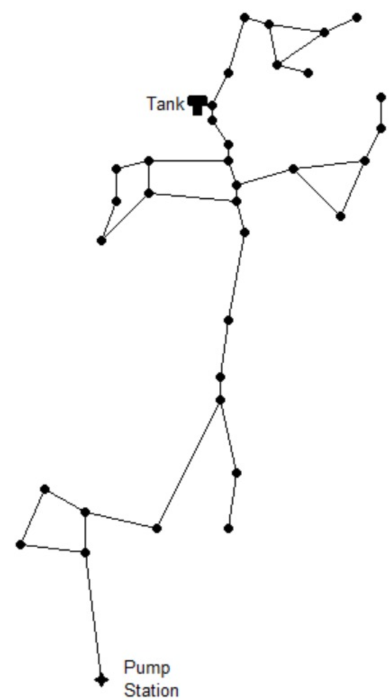
Water Network Decomposition into Graph Layers



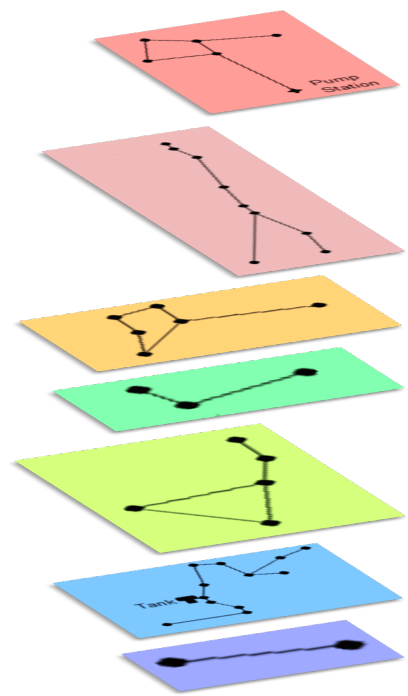
Transition to Networked Decomposition

Incremental Learning of Network / Graph Zones

Original Network



Decomposed Network



GPU Runtime

~91.9 s

~81.6 s

~91.1 s

~90.6 s

~92.3 s

~98.2 s

~8.2 s

Total GPU Runtime = 2935.4 s

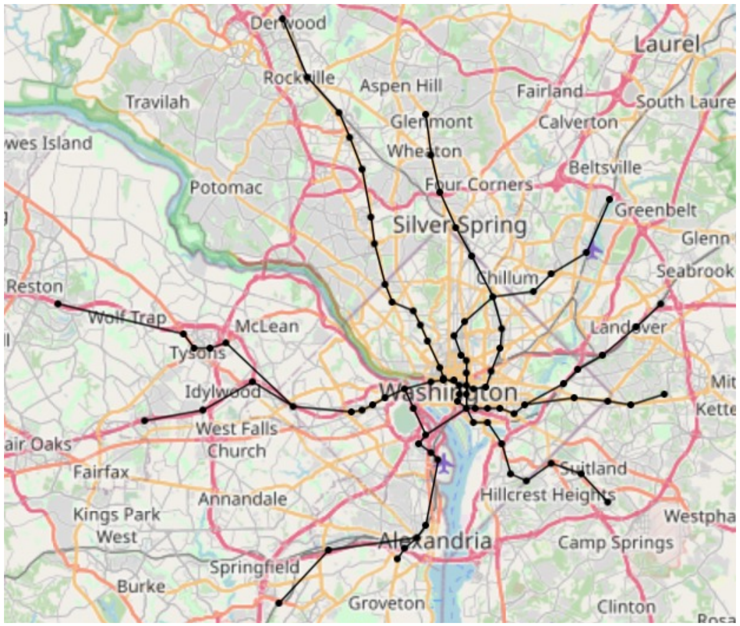
Total GPU Runtime = 553.9 s

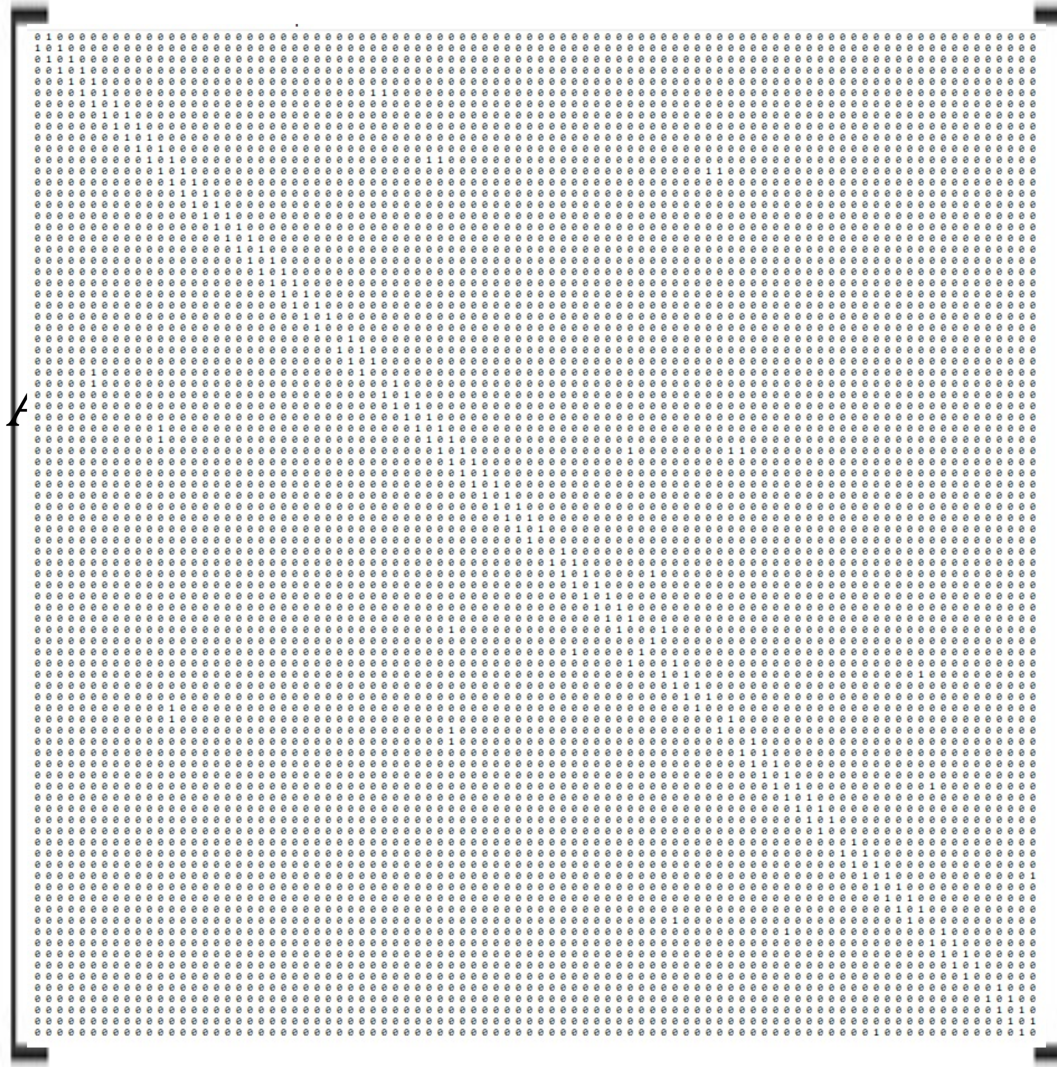
Accelerated
Learning

Transition to Networked Decomposition

Washington DC Metro System Network

Topology:

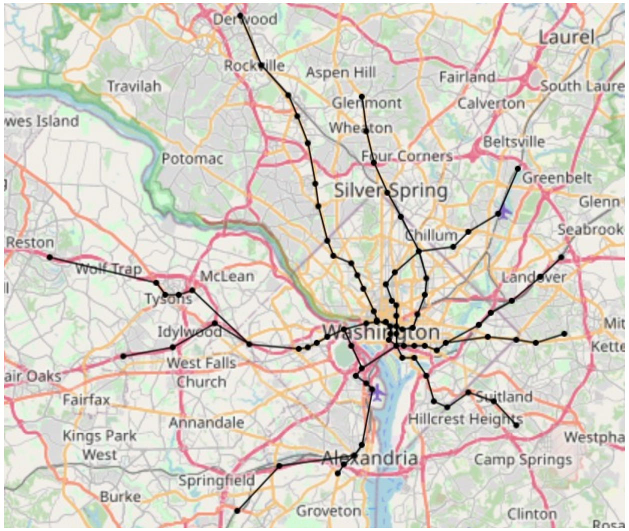




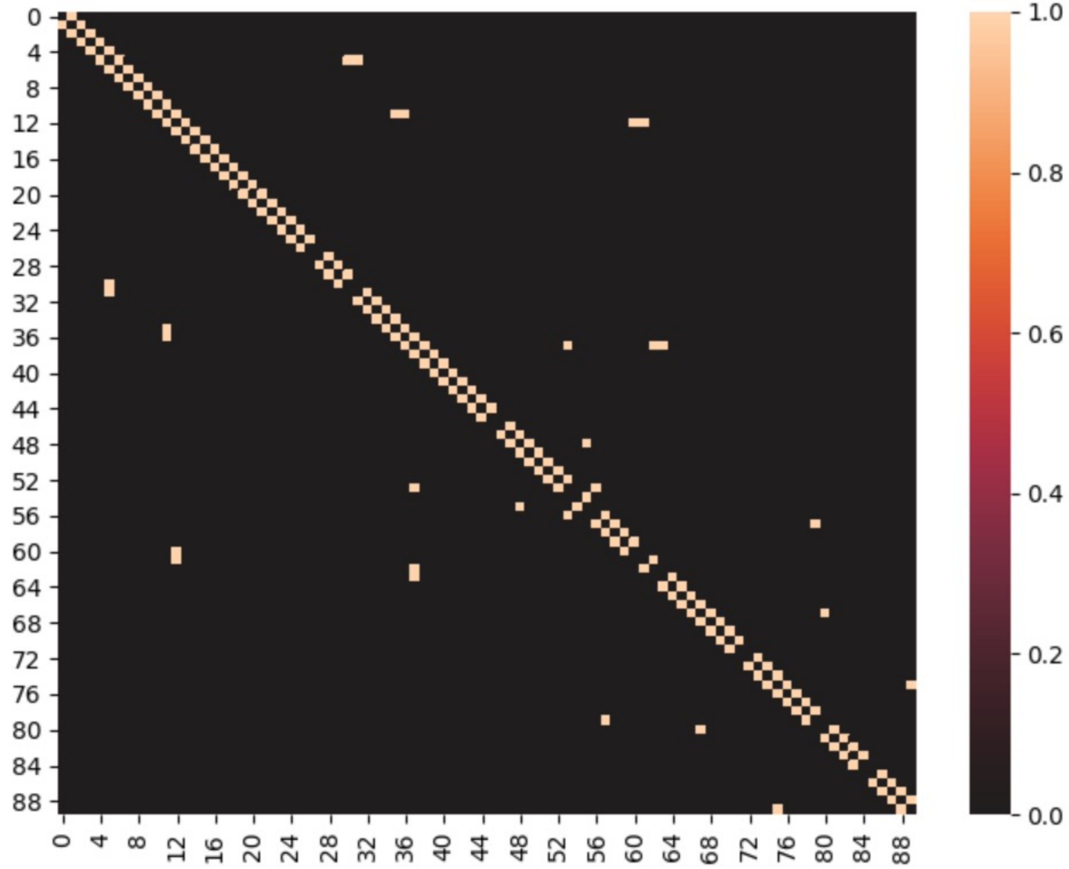
SYSTEMS ENGINEERING RESEARCH CENTER

Transition to Networked Decomposition

Washington DC Metro System Network

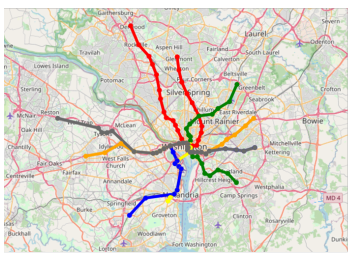


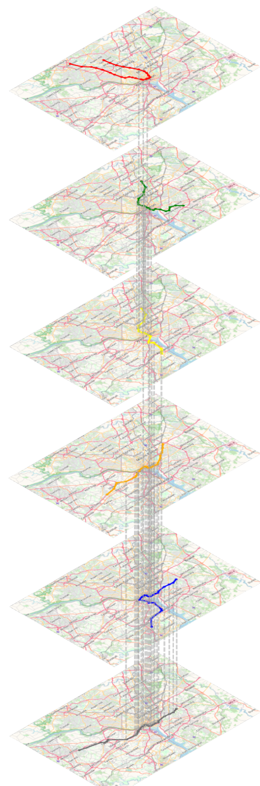
Heatmap:

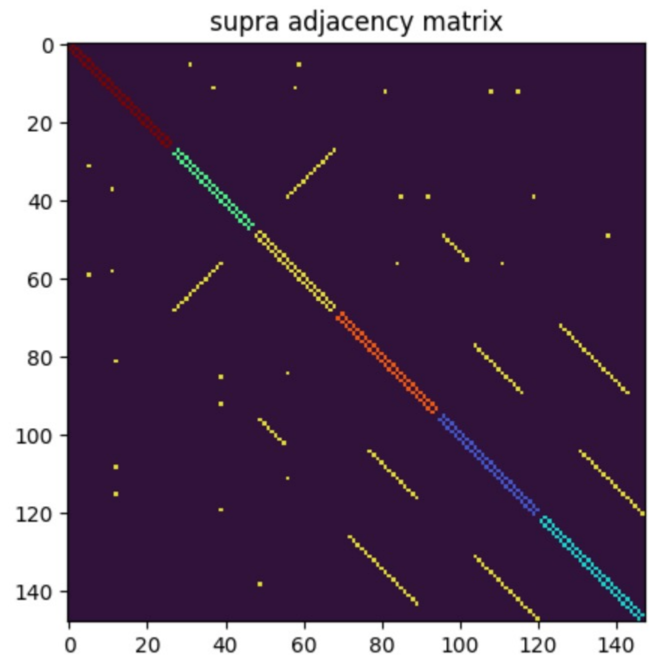


Transition to Networked Decomposition

Washington DC Metro System Network



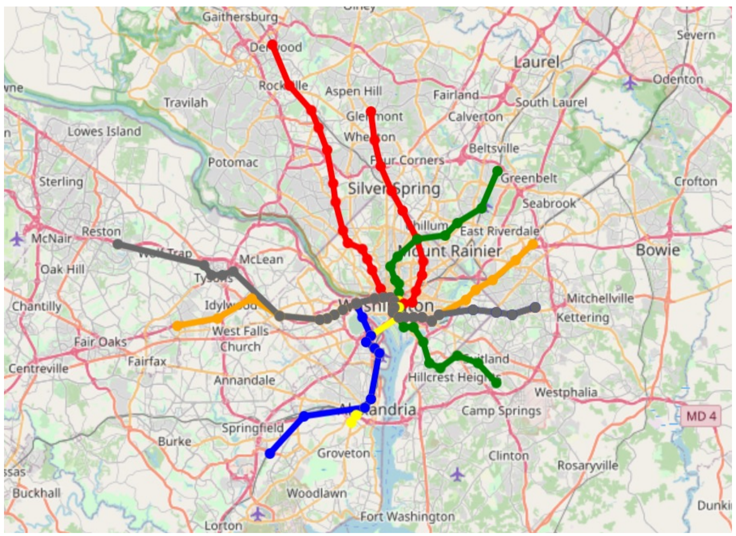




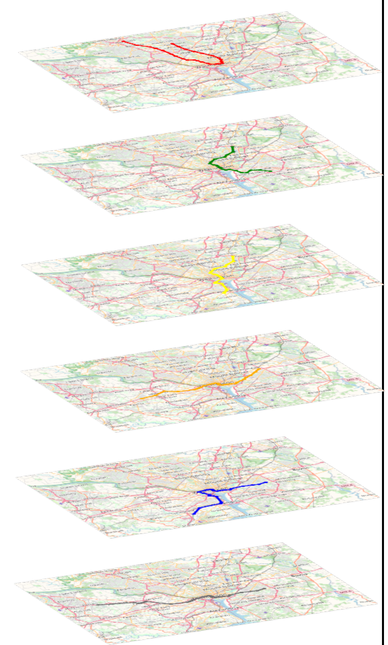
Transition to Networked Decomposition

Accelerated Learning of Network / Graph Zones

Original Network



Total GPU Runtime = 25582.10 s

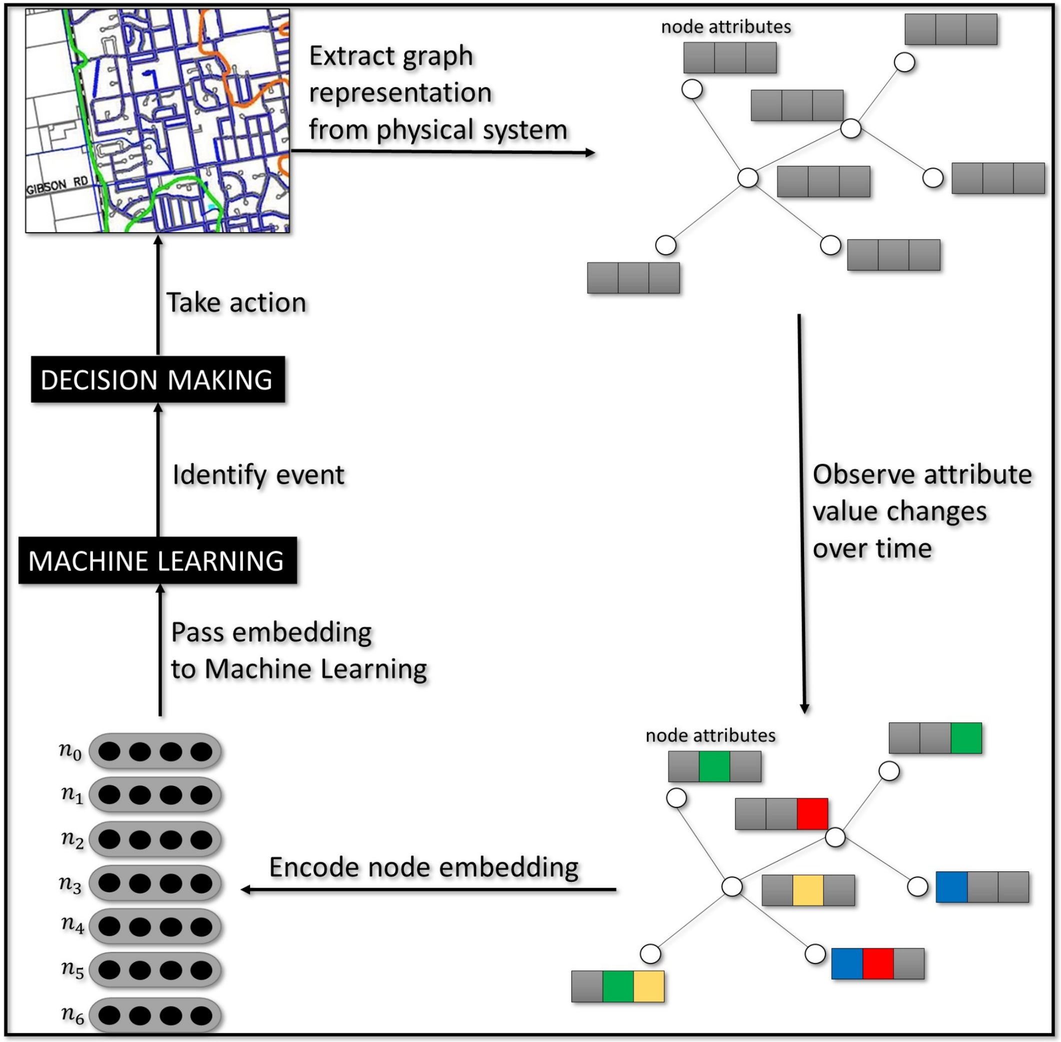
Decomposed Network	**GPU Runtime**
	~115.03 s ~97.26 s ~98.33 s ~102.95 s ~102.86 s ~101.12 s

Total GPU Runtime = 617.55 s

Next Big Step

Semantic Model and Machine Learning Liaison

- To date we have achieved considerable success in understanding semantic modeling and ML.
- Need to show their collaboration being applied to a domain-specific networked structure. infrastructure.



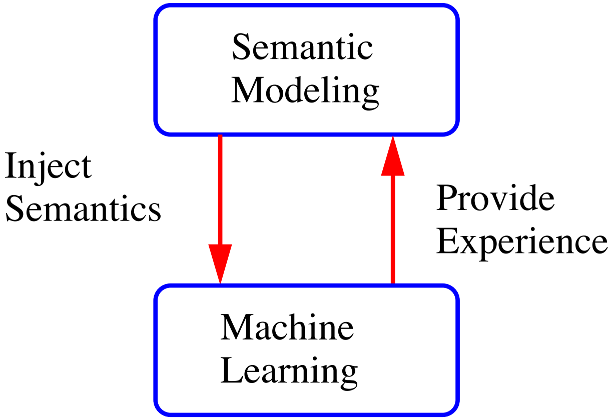
Summary of Results/Future Work

Year 1: Teaching Machines to Understand Graphs

- Teaching machines to understand small graphs having static graph topologies.
- Auto-encoder design (guarantees on system graph representation).
- Formulae for design of neural network architectures for specific types of graph.
- Explore opportunities for composition of neural network architectures.
- Identification of events via time-series anomaly detection.
- Basic mechanisms for semantic / machine learning interaction.
- Integration of simulation and machine learning.

Year 2: Go Deep, Dynamic, Broad, Hybrid

- Deep graph neural networks / dynamic graph topologies.
- Reasoning with events, space and time.
- Inject semantics into machine learning models.
- Applications.



Year 3: Create Digital Twin Experience

- AI/ML architectures for digital twin experience.
- Applications.

Thank You

Questions?

Contact Information

Mark Austin: austin@umd.edu
Maria Coelho: mecoelho@terpmail.umd.edu
Mark Blackburn: mblackbu@stevens.edu

References

- **Austin M.A., Delgoshaei P., Coelho M. and Heidarinejad M.** , Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach (Special Collection on Engineering Smarter Cities with Smart City Digital Twins), Journal of Management in Engineering, ASCE, Volume 36, Issue 4, July, 2020.

- **Coelho M., Austin M.A., Mishra S., and Blackburn M.R.** , Teaching Machines to Understand Urban Networks: A Graph Autoencoder Approach, International Journal on Advances in Networks and Services, Vol 13, No 3&4, December, 2020, pp. 70-81

- **Coelho M., Austin M.A. and Blackburn M.R.** , Semantic Behavior Modeling and Event-Driven Reasoning for Urban System of Systems, International Journal on Advances in Intelligent Systems, Vol. 10, No 3 and 4, December 2017, pp. 365-382.

- **Coelho M., and Austin M.A.** , Teaching Machines to Understand Urban Networks, The Fifteenth International Conference on Systems (ICONS 2020), Lisbon, Portugal, February 23-27, 2020, pp. 37-42.

- **Coelho M., Austin M.A. and Blackburn M.** , Distributed System Behavior Modeling of Urban Systems with Ontologies, Rules and Many-to-Many Association Relationships, The 12th International Conference on Systems (ICONS 2017), Venice, Italy, April 23-27, 2017, pp. 10-15.

- **Coelho M., Austin M.A., and Blackburn M.R.,** The Data-Ontology-Rule Footing: A Building Block for Knowledge-Based Development and Event-Driven Execution of Multi-Domain Systems, 2018 Conference on Systems Engineering Research, Charlottsville, VA, May 8-9, 2018. Also see: Chapter 21, Systems Engineering in Context , Springer, 2019.

- **Coelho M., Austin M.A. and Blackburn M.R.,** ``Semantic Behavior Modeling and Event-Driven Reasoning for Urban System of Systems,'' International Journal on Advances in Intelligent Systems, Vol. 10, No 3 and 4, December 2017, pp. 365-382.

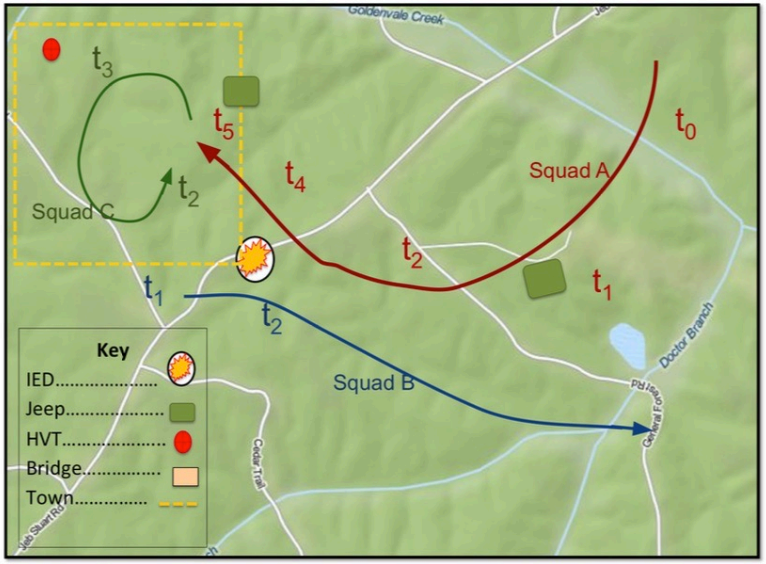
Extra Slides

Contact Information

Mark Austin: austin@umd.edu
Maria Coelho: mecoelho@terpmail.umd.edu
Mark Blackburn: mblackbu@stevens.edu

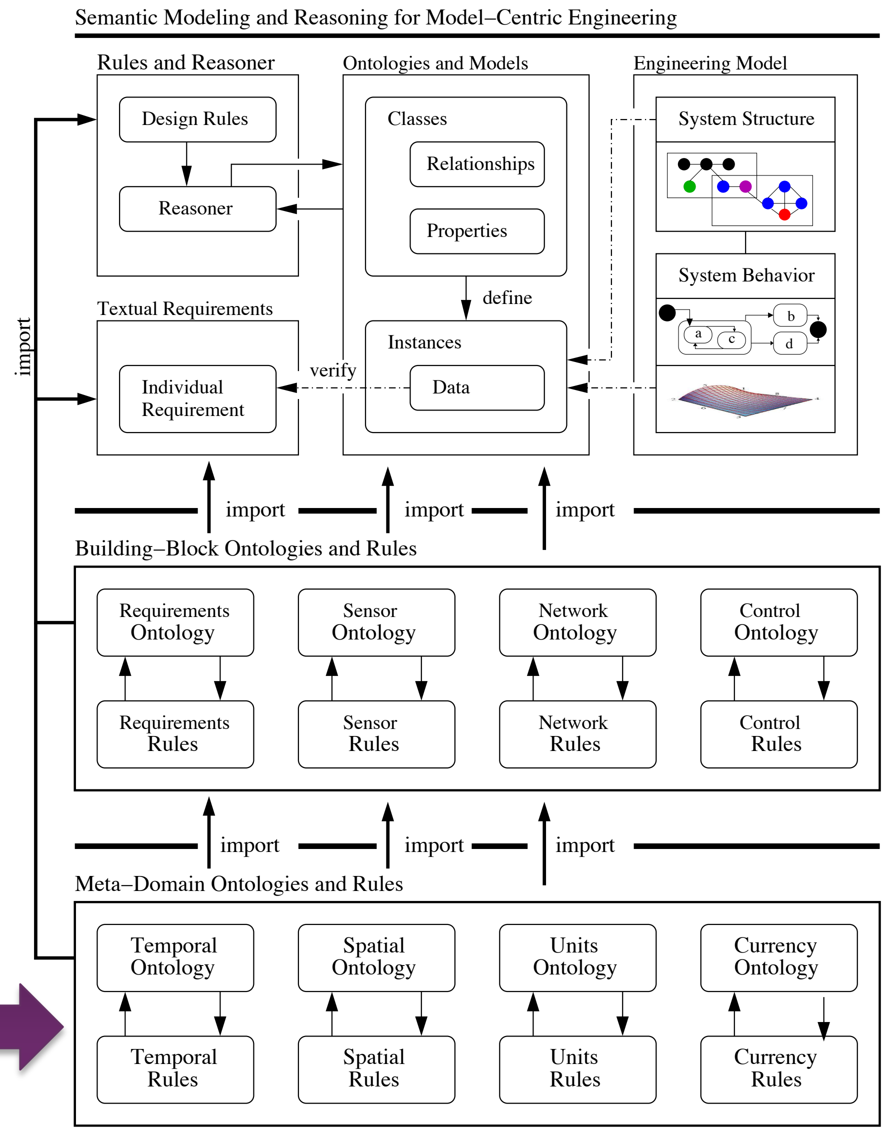
Semantic Modeling for Model-Centric Engineering

Simple Military Exercise



Decision Making / Exercise Actions

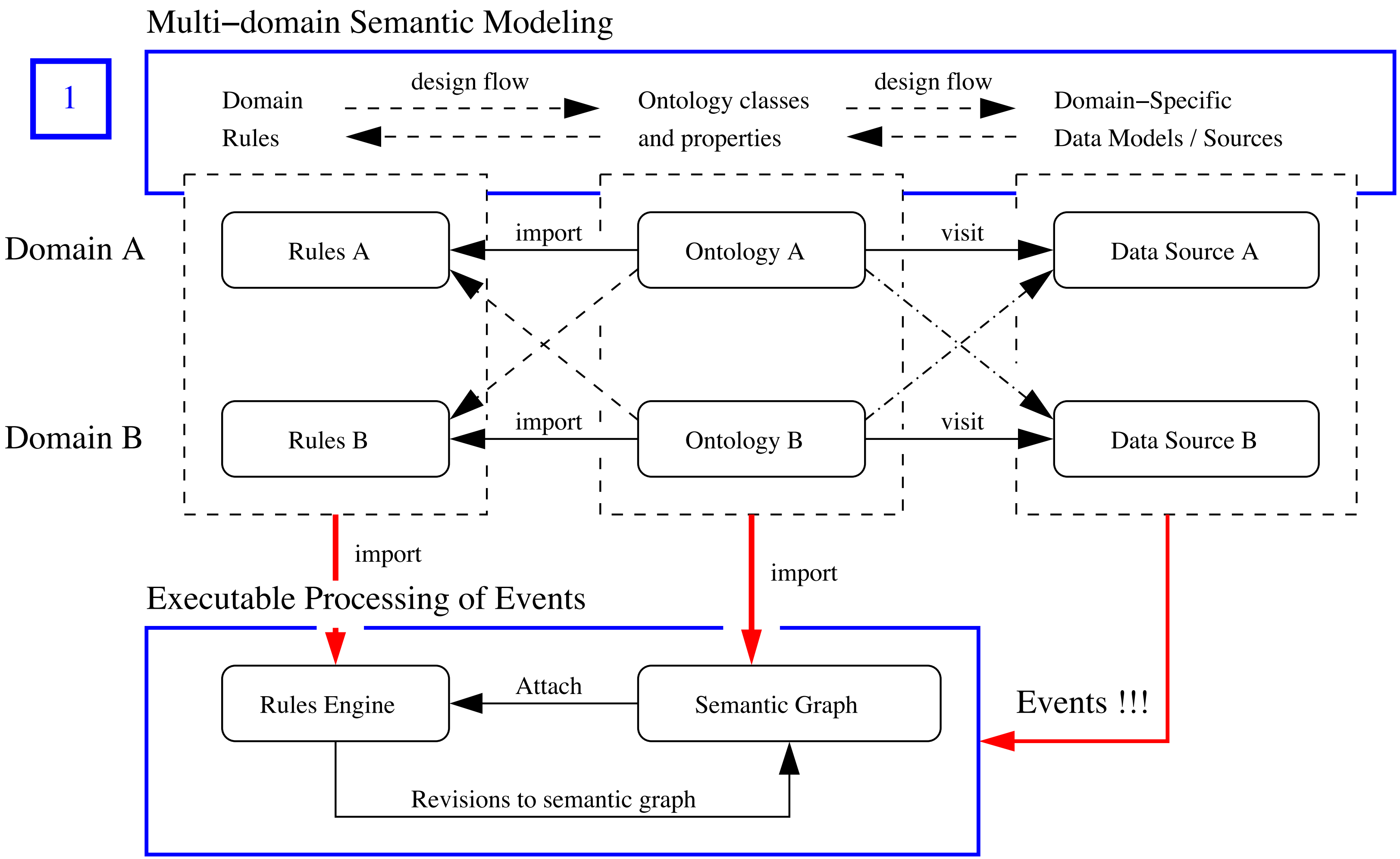
Military exercise actions need to occur at the right time and in the right place.



Source: Regli W., et al.

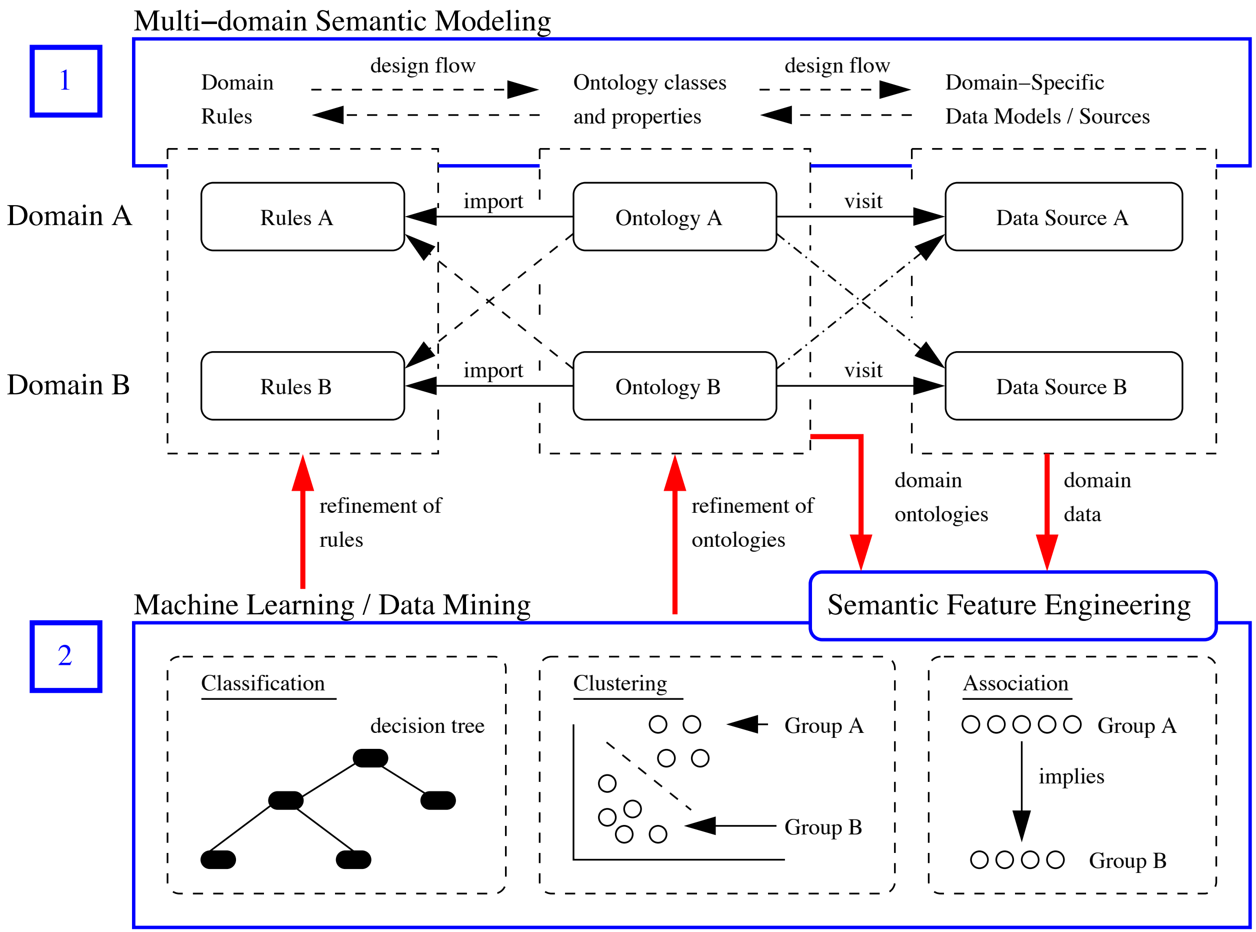
Multi-Domain Semantic Modeling

Data-Ontology-Rule Footing (Work at UMD / NIST / SERC in 2017).



Combined Semantics + Data Mining

Work at UMD / Building Energy Group at NIST / NCI, 2018-2019



Research Question: How can semantic modeling + machine learning / data mining work together as a team?