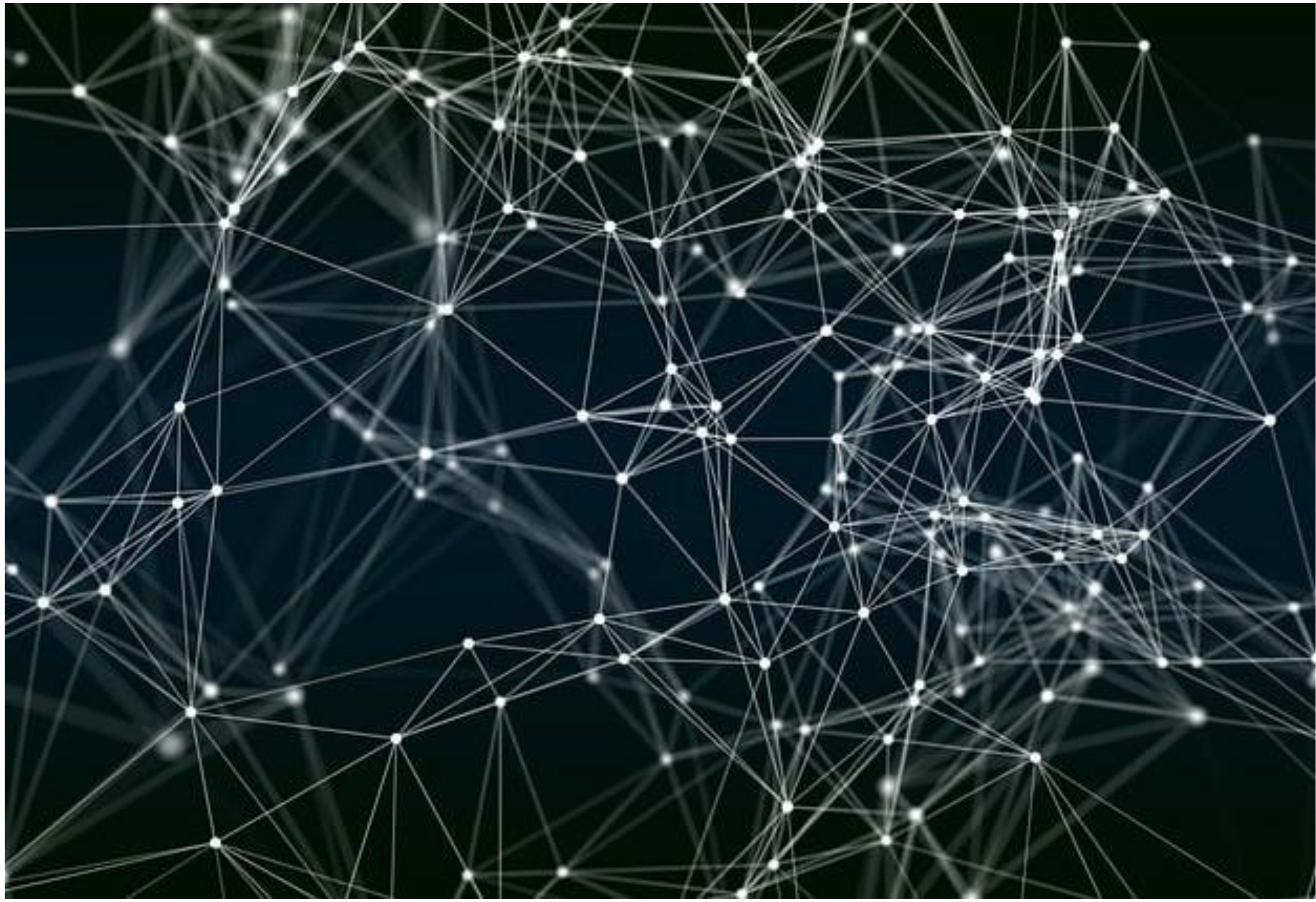
Graph Learning for Efficient and Explainable Operation of Particle Accelerators

Chris Tennant

T. Larrieu, J. Li, D. Moser, S. Wang

DOE PI AI/ML Exchange Meeting | December 5, 2024



Jefferson Lab UNIVERSITY of VIRGINIA

U.S. DEPARTMENT OF ENERGY Office of Science JSA

Outline

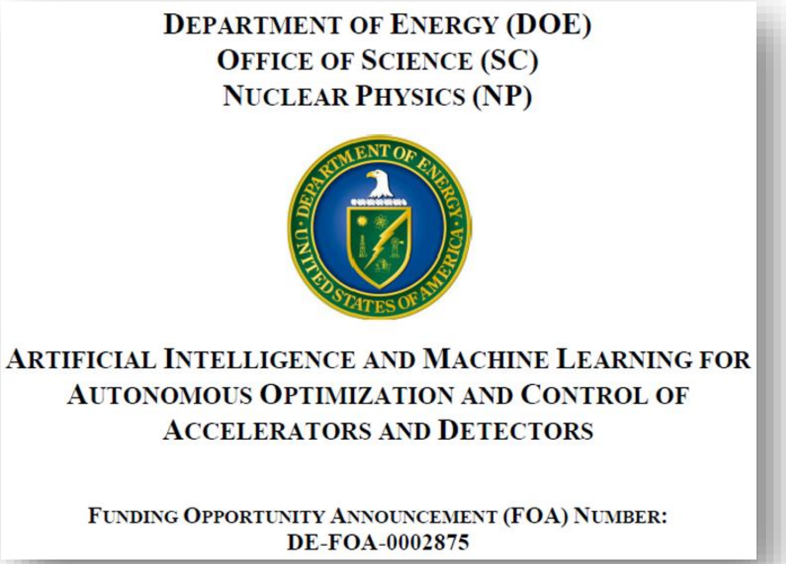
- Introduction
- Motivation
- Method
- Applications
  - ✓ *Embeddings*
  - ✓ *Beam tuning trajectories*
  - ✓ *Explainability framework*
- Project Summary

“Graph Learning for… Operation of Particle Accelerators”

“The fundamental idea is to apply deep learning over graph representations of the CEBAF injector beamline in order to extract information-rich, low-dimensional embeddings. With access to embeddings that capture the complex relationships of a many-dimensional beamline over time, several unique applications will be developed…”

leverage deep learning on graph representations of a beamline for improving the efficiency of beam tuning tasks

- this project represents the intersection of:
  - ✓ graph analysis
  - ✓ deep learning
  - ✓ self-supervised learning
  - ✓ dimensionality reduction
  - ✓ visualization
  - ✓ explainable AI


DEPARTMENT OF ENERGY (DOE)
OFFICE OF SCIENCE (SC)
NUCLEAR PHYSICS (NP)

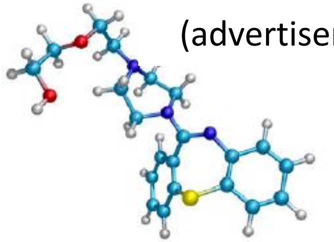
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR AUTONOMOUS OPTIMIZATION AND CONTROL OF ACCELERATORS AND DETECTORS

FUNDING OPPORTUNITY ANNOUNCEMENT (FOA) NUMBER: DE-FOA-0002875

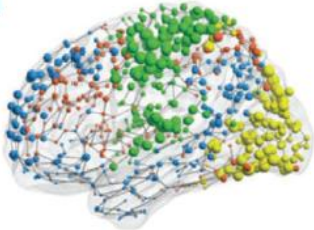
Introduction: What is a Graph?

- graphs are a general language for describing and analyzing entities with relations/interactions

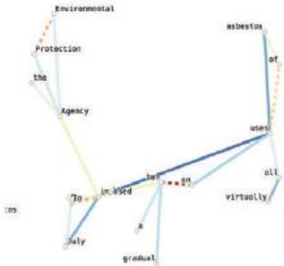
Social Networks
(advertisement)



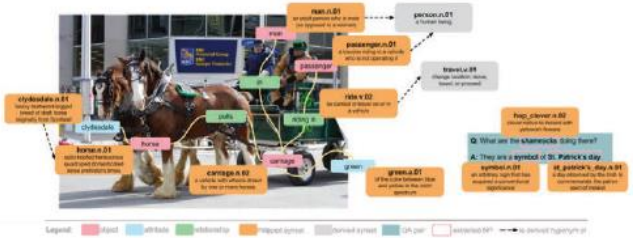
Molecules
(chemistry)



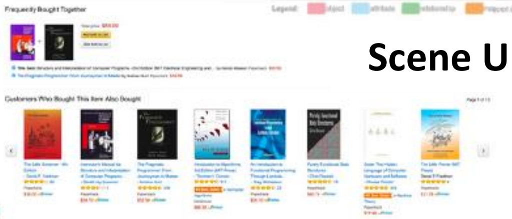
Brain Connectivity
(neuroscience)



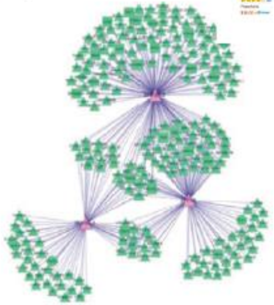
Word Relationships
(NLP)



Scene Understanding

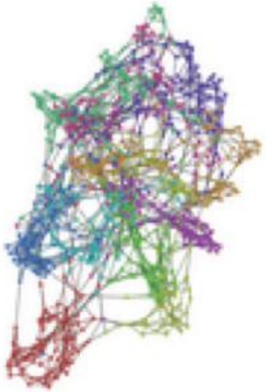


Recommender Systems

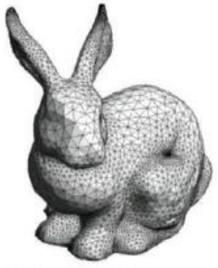


Gene Regulatory Network

=

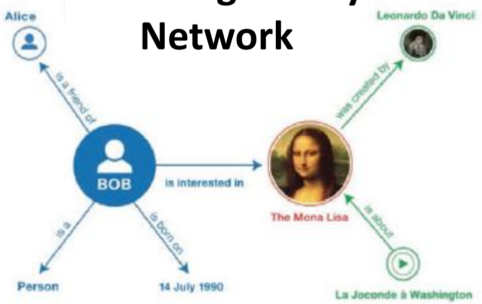


GRAPH

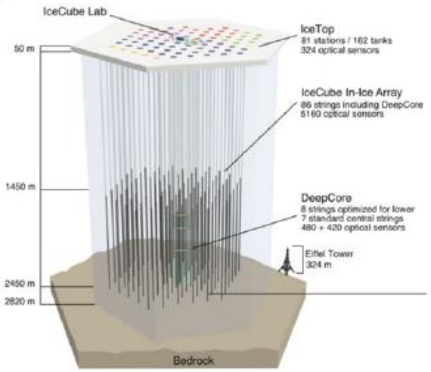


3D Meshes
(computer graphics)

Transportation Networks



Knowledge Graphs

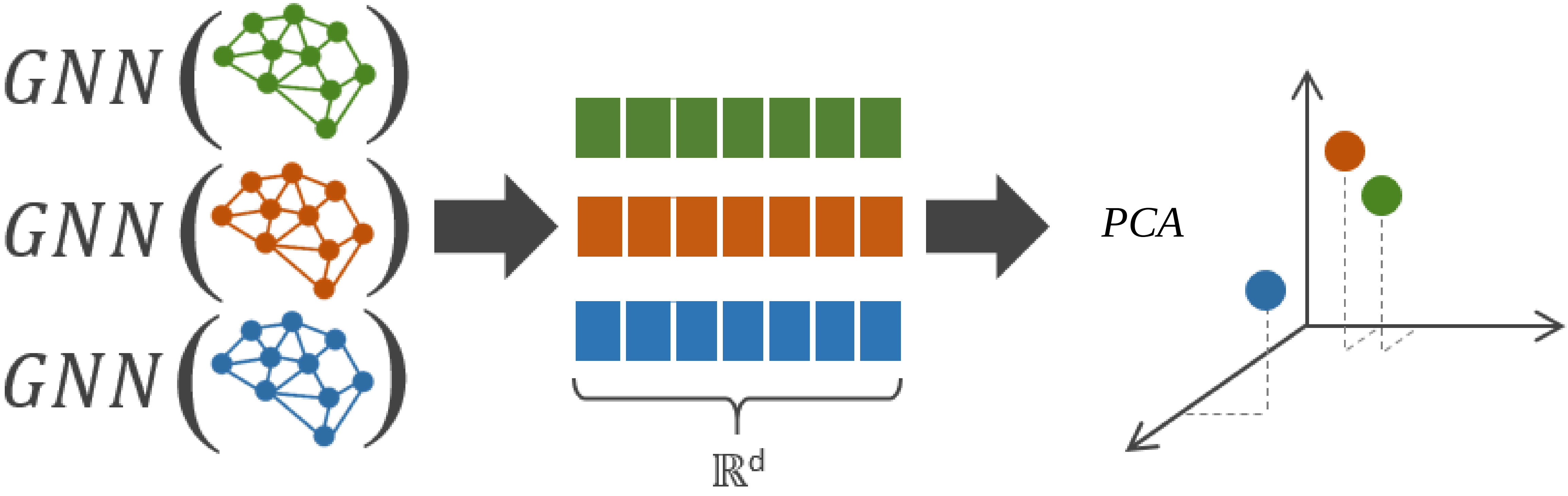


Neutrino Detection
(HEP)

(courtesy X. Bresson)

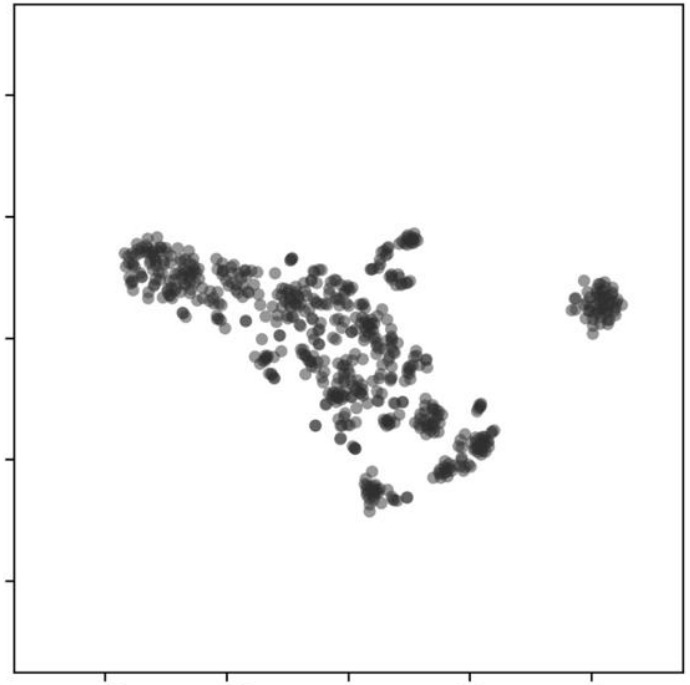
Introduction: Whole-Graph Embedding

- represent the state of a beamline at a specific date and time as a graph
- embed graph into 2- or 3-dimensions using a graph neural network (GNN)
    - ✓a GNN provides a framework for defining a deep neural network on arbitrary graph data



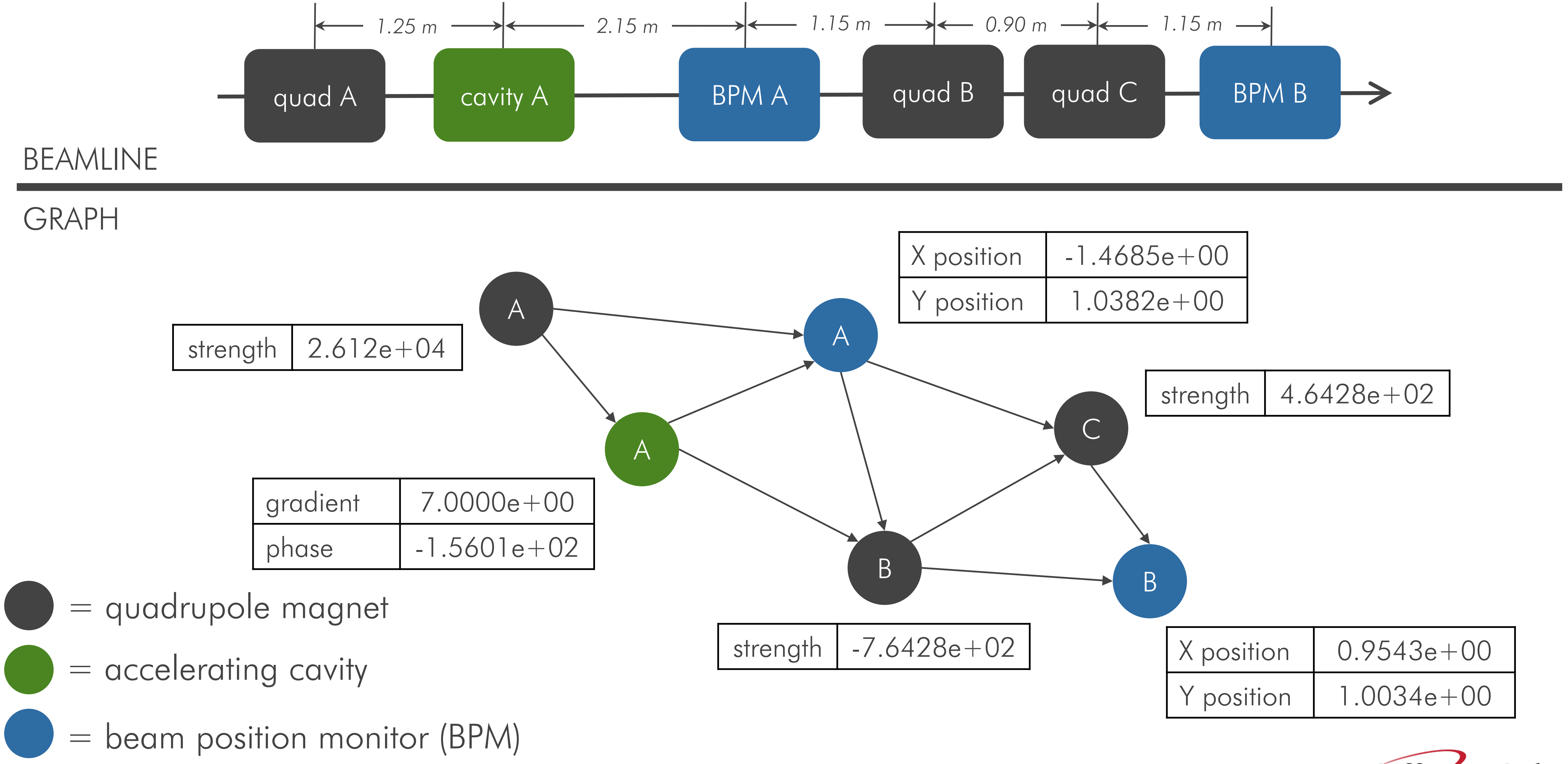
Motivation: Application for Beam Operations

- visualize the underlying distribution/pattern of beamline states
- use cluster analysis to identify regions of parameter space
- make changes to machine and observe trajectory in latent space



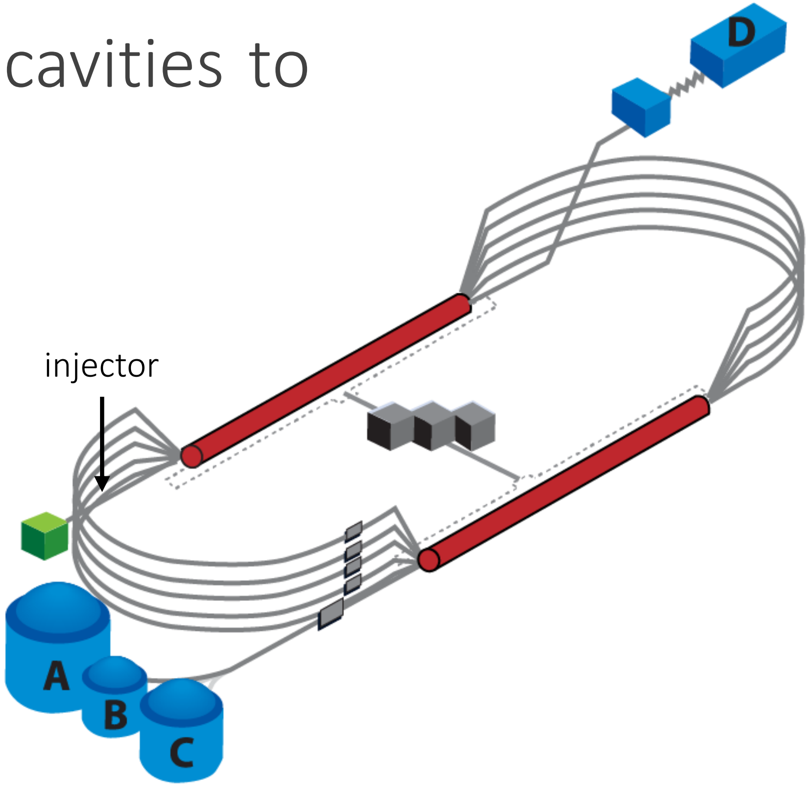
- provides a data-driven framework with visual feedback to more quickly converge to known optimal beamline configurations

Method: Beamline-to-Graph



Method: Beamline-to-Graph

- CEBAF is a CW recirculating linac utilizing 418 SRF cavities to accelerate electrons up to 12 GeV through 5-passes
- consider a 95 m portion of the injector
- software developed so that:
  - ✓ given a date and time stamp, a beamline is defined from the CEBAF Element Database
  - ✓ node features are populated by process variables stored in the operational archiver
  - ✓ each graph: 12 node types, 207 nodes, 393 node features, and 528 edges
- have generated over 100K graphs from the last few years of operational data for model training and analysis



`info.dat`
defines the node types

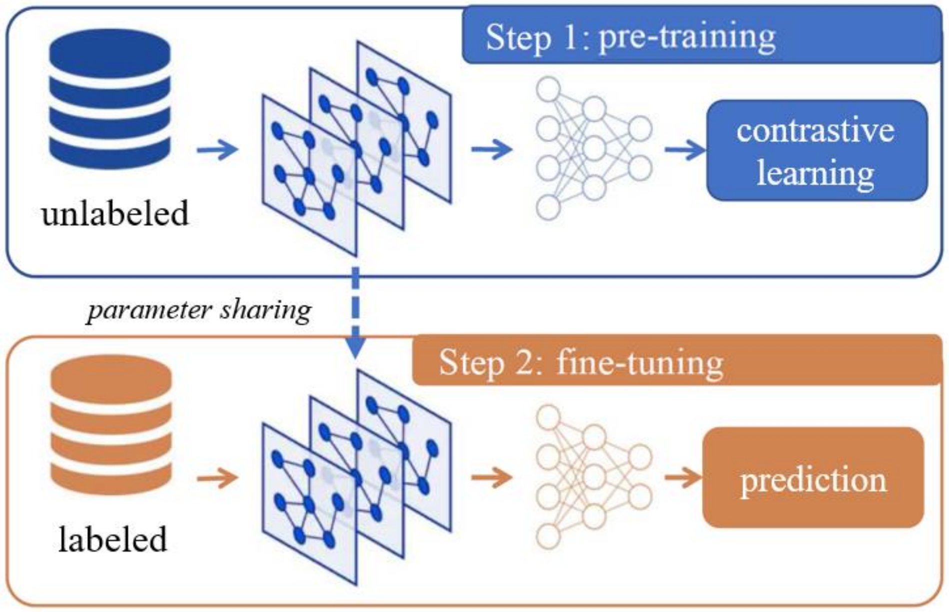
`meta.dat`
lists the number of each node type in the graph

`node.dat`
[*node ID, node name, node type, node attributes*]

`link.dat`
[*start node ID, end node ID, edge type, edge weight*]

Method: GNN Model Training

1. pre-training the model on a large set of *unlabeled* data using a technique called self-supervised learning (SSL)
   - ✓ tries to learn as much as possible from the data alone
   - ✓ take advantage of years of operational data stored in the archiver without having to do the expensive task of hand labeling the data
   - ✓ a special class of loss function, known as contrastive loss, maximizes agreement between latent representations from similar graph pairs while minimizing agreement from unlike pairs

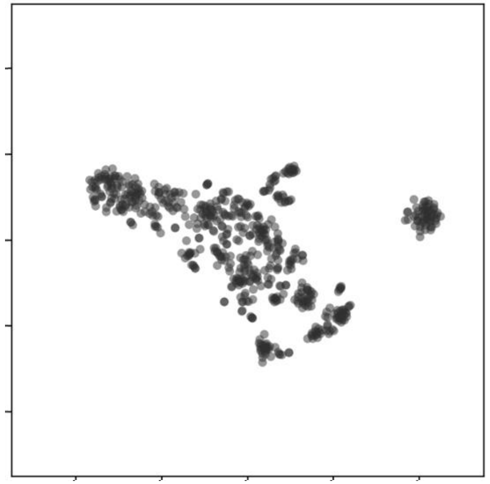


2. fine-tune the model on the downstream task of classifying good and bad setups using a smaller *labeled* dataset
3. a dimensionality reduction to visualize the results in 2D

this same basic workflow is being leveraged to produce state-of-the-art results in natural language processing and vision tasks

Applications

1. Represent accelerator beamline as a graph
2. Pre-train a GNN in a self-supervised way



enable data exploration
(identifying trends and patterns)

identify good and bad regions of parameter space

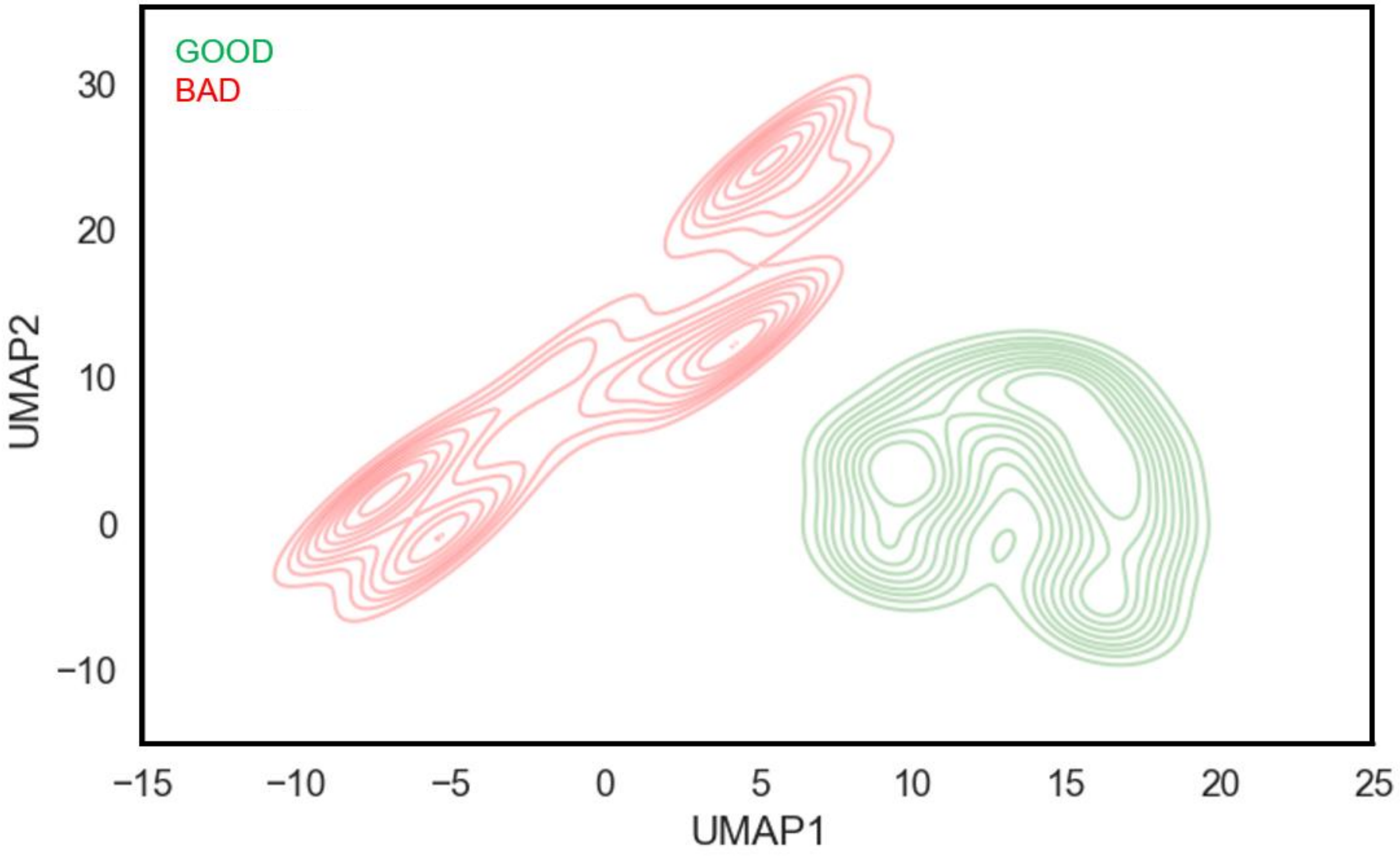
On-line Tool: provides near real-time feedback on effect of tuning changes

Monitor Stability: plot the beamline configuration in latent space over time

Jefferson Lab

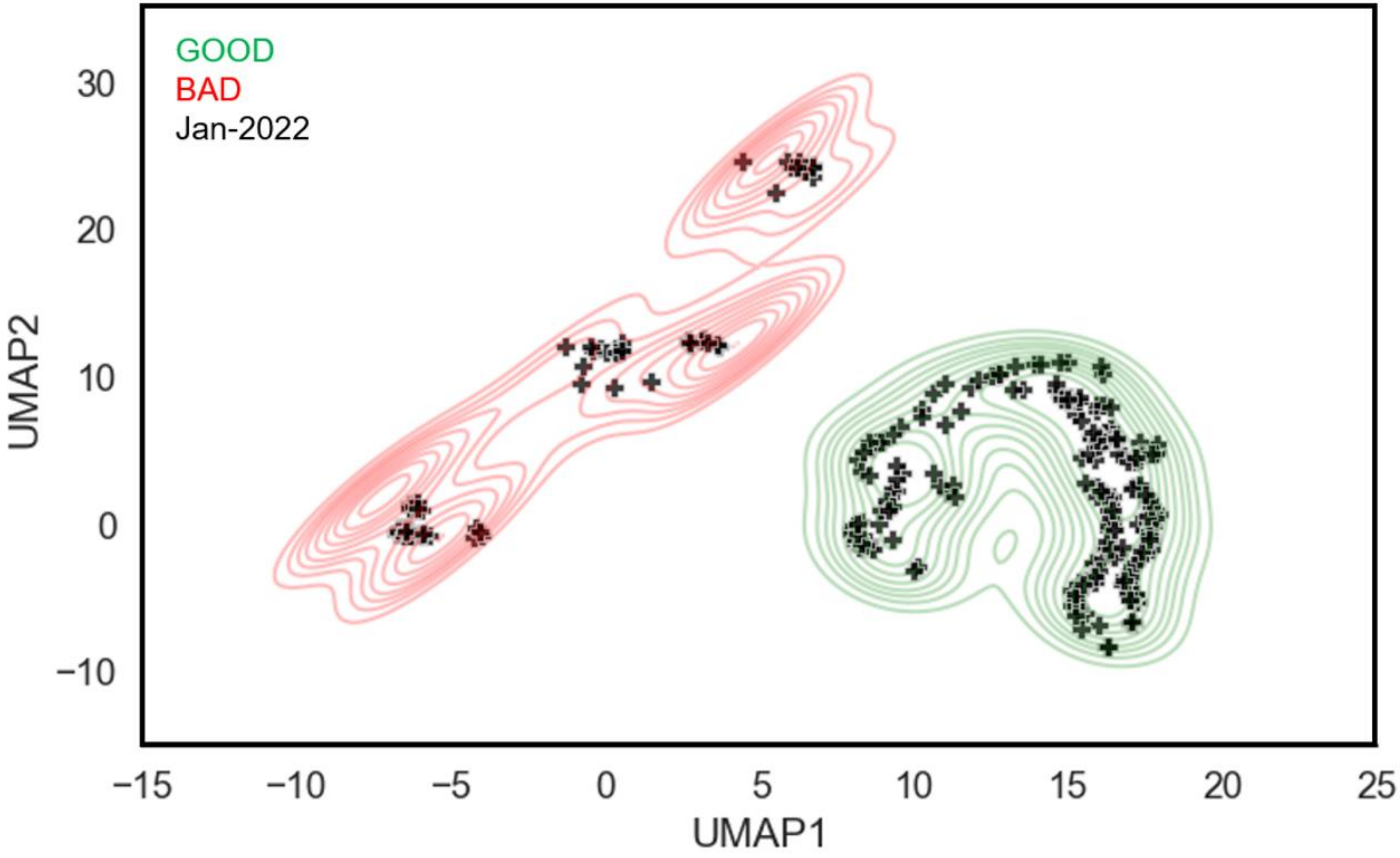
Identifying Regions of Parameter Space

- a GNN encoder pre-trained on the 2,129 unlabeled graphs from 2021
- fine-tune model on the smaller set of labeled (354 good, 254 bad) graphs



Identifying Regions of Parameter Space

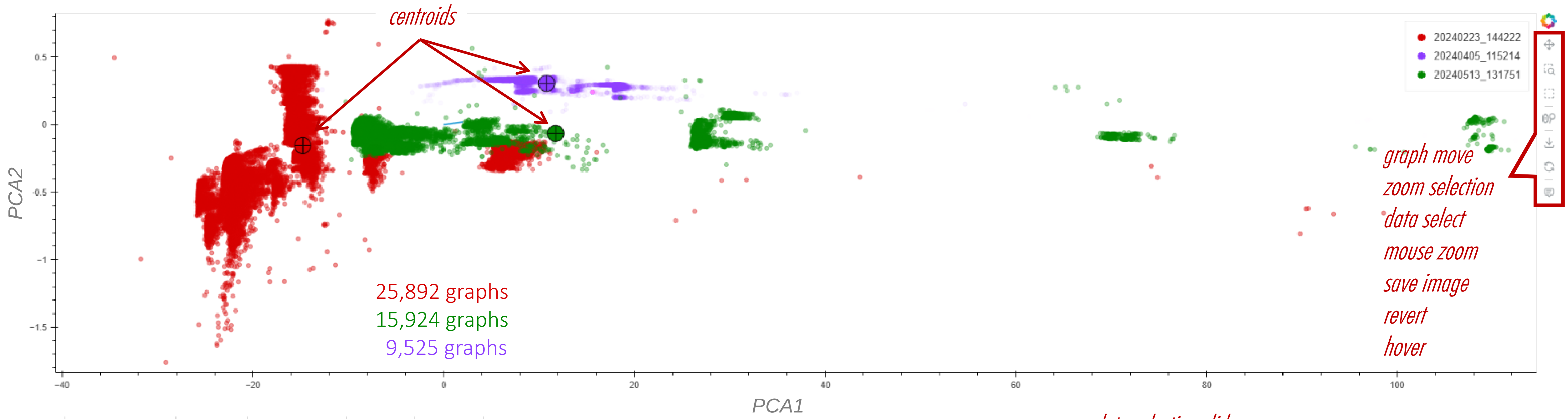
- a set of 353 *unlabeled* graphs representing operations in January 2022 were input to the model and their resulting latent representations plotted



S. Wang, et al., Front. Big Data, Sec. Data Mining and Management, vol 7 (2024).

Jefferson Lab

Embeddings: Interactive Visualization



index	Datetime	IBC0L02	Label	UMAP1	UMAP2
0	2022-07-01 00:00:00	0.085934	20240223_144222	7.108283	-0.154056
1	2022-07-01 00:02:00	-0.364039	20240223_144222	6.51527	-0.212205
2	2022-07-01 00:04:00	0.717826	20240223_144222	7.578921	-0.139675
3	2022-07-01 00:08:00	0.675061	20240223_144222	6.627912	-0.180567
4	2022-07-01 00:18:00	0.775135	20240223_144222	6.536095	-0.215873
5	2022-07-01 00:22:00	0.698479	20240223_144222	7.639106	-0.150539
6	2022-07-01 00:24:00	0.715541	20240223_144222	7.264797	-0.206922
7	2022-07-01 00:26:00	0.54385	20240223_144222	6.65573	-0.169164
8	2022-07-01 00:28:00	0.282189	20240223_144222	6.353577	-0.165169
9	2022-07-01 00:30:00	0.563995	20240223_144222	6.383954	-0.141519

First Prev 1 2 3 4 5 Next Last

sortable table of selected data

Date: 04 Jan 2022 14:48:00 .. 31 Aug 2022 13:55:00 — *date selection slider*

Current: -1.27 .. 4.17 — *beam current selection slider*

Labels — *data selector*

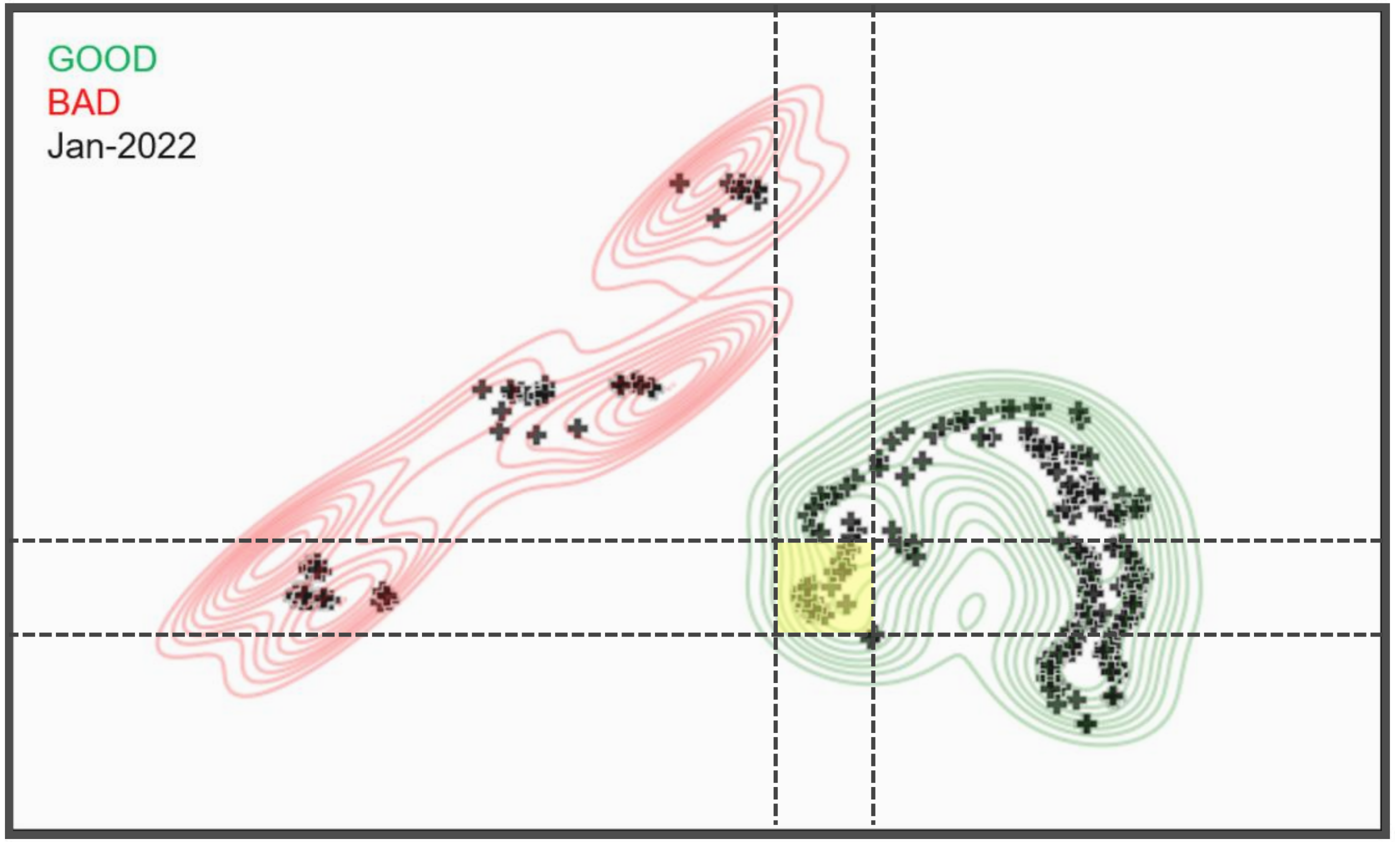
20240223_144222
20240405_115214
20240513_131751
20240729_114431_BeamStudy

Overlay Hooks — *custom functions*

avg_point

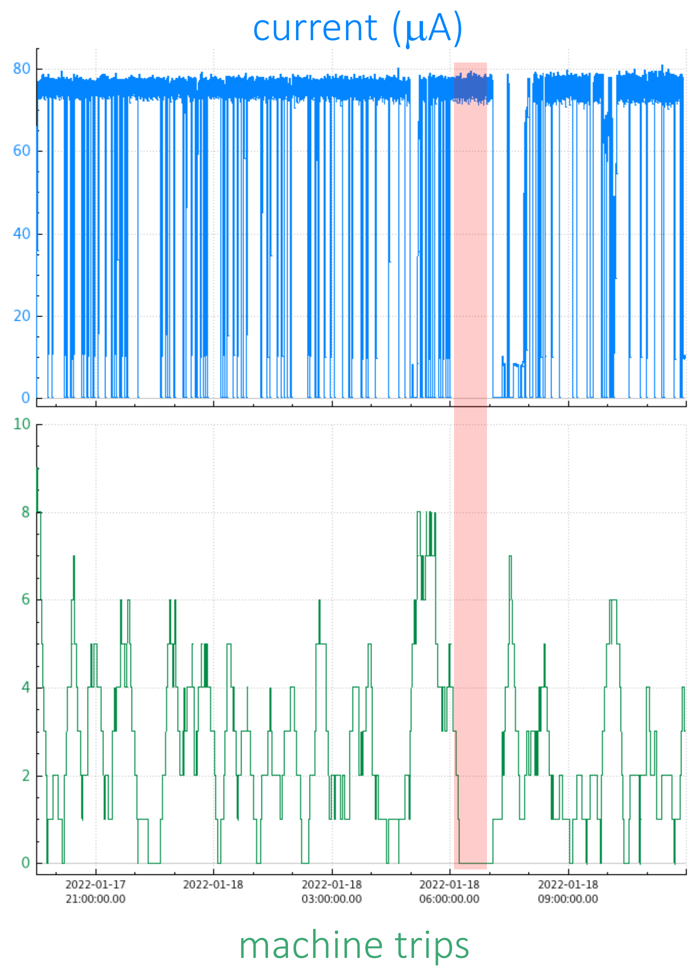
Applications: Reproducibility and Stability

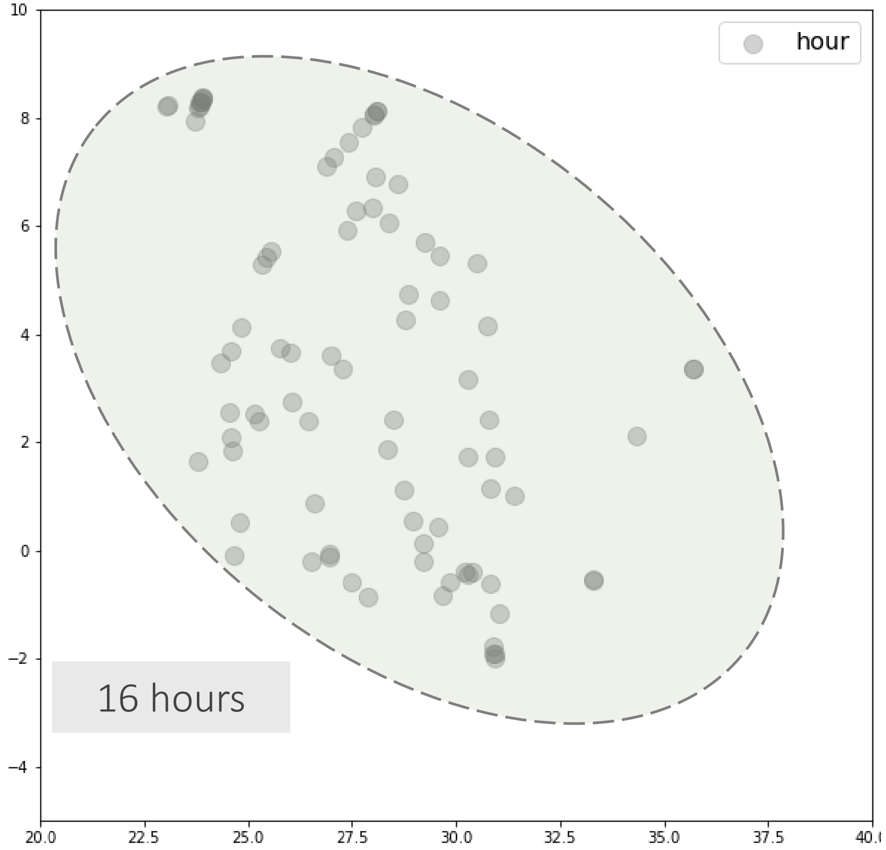
- embeddings provide a way to measure the *reproducibility* of the machine in a quantifiable way → use an appropriate distance metric in the latent space
- embeddings provide a way to monitor system *stability*
  - ✓ it's trivial to track one PV over time, but this work provides a means to track a 393-dimensional space over time
  - ✓ consider a tool where a user defines a bounding box of stable/good running → in the background, the beamline configuration is queried every minute, a graph generated, embedded in the latent space, and an alert is sent to operators anytime the system crosses the boundary

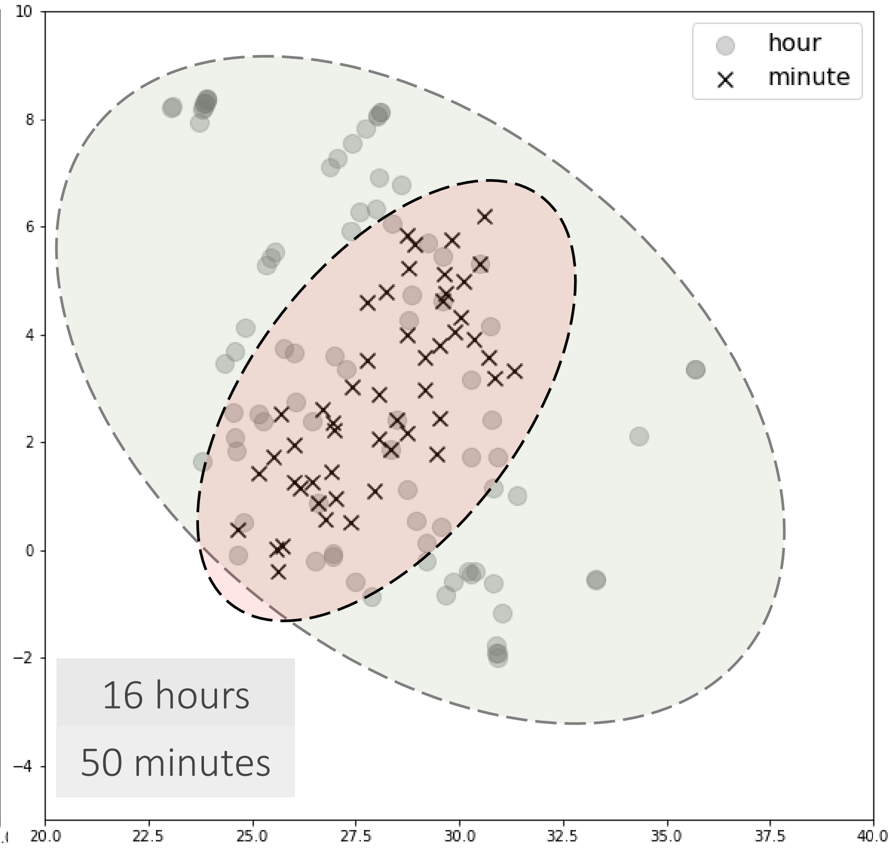


Applications: Monitoring Stability

- identify a period of stable running and observe associated embeddings







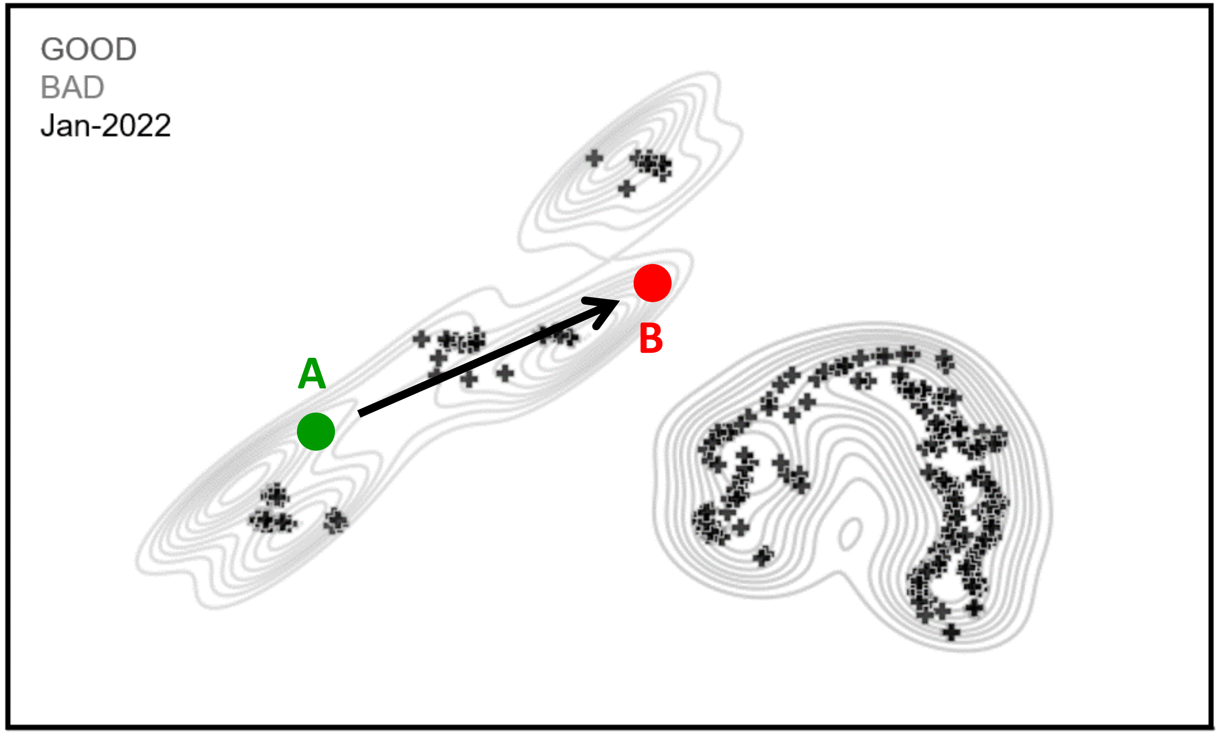
Explainability

- **GOAL:** what setting node is most important (i.e. best *explains*) in moving from point A to B in latent space?
- the downstream impact of changing an element depends on the:

✓ element type
  - *a quadrupole affects the beam differently than a corrector than an RF phase*

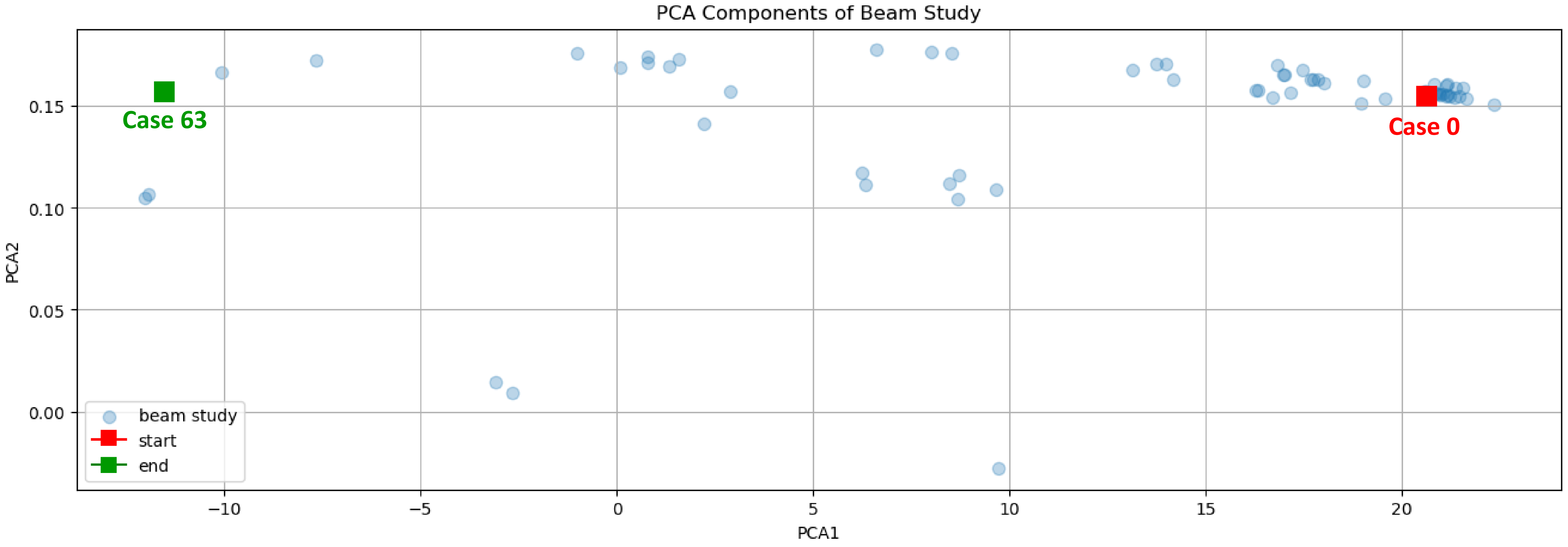
✓ element location
  - *a change further upstream provides a bigger lever arm, but it also depends on beam properties at location of change*

✓ magnitude of the change
  - *all other things being equal, a larger magnitude will have a bigger affect, however, even if we compare two of the same type of element, location matters*



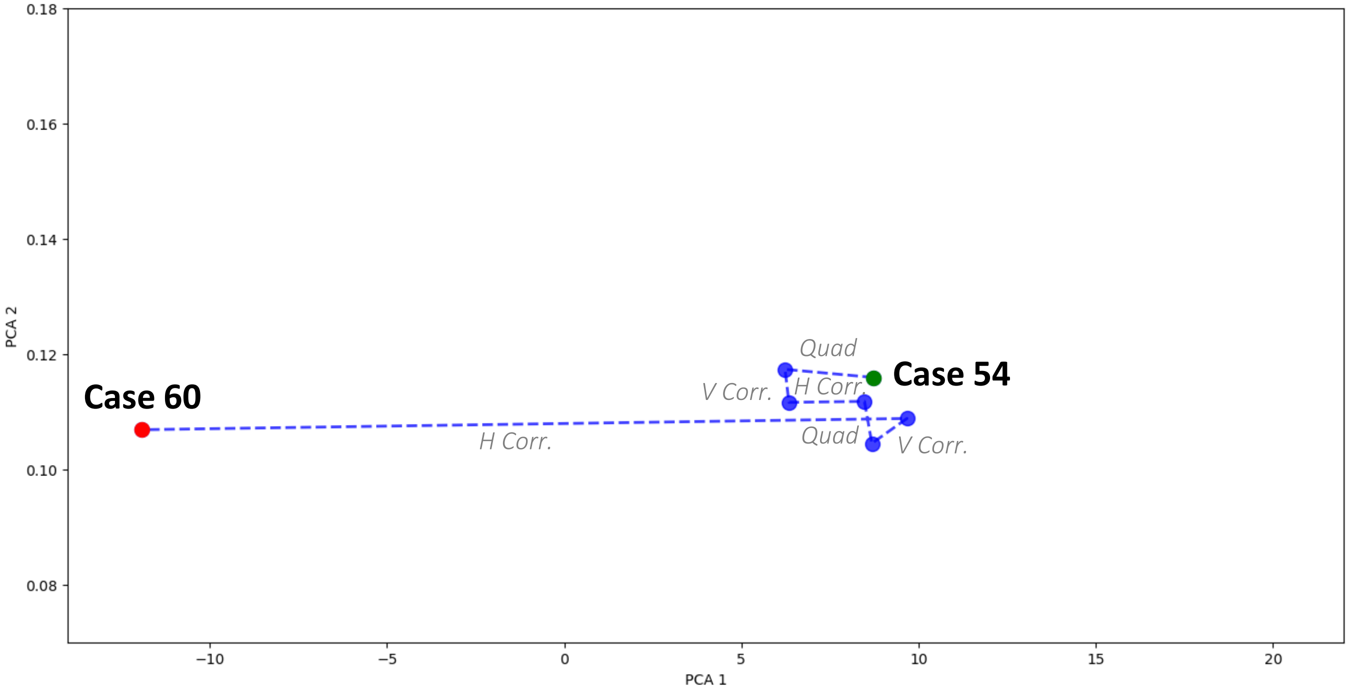
Controlled Beam Test

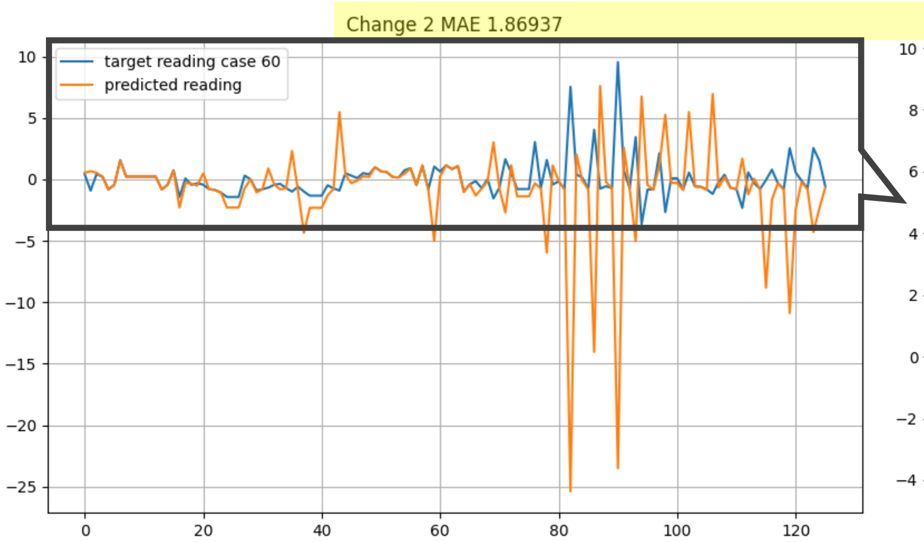
- migrate from one injector setup to another in a methodical, incremental way
- use graph embeddings to visualize trajectory through latent space (off-line)
- start with an injector configuration from Feb. 8, 2022 and make 63 small, incremental changes (starting downstream and working upstream) to reach a configuration from Aug. 30, 2022

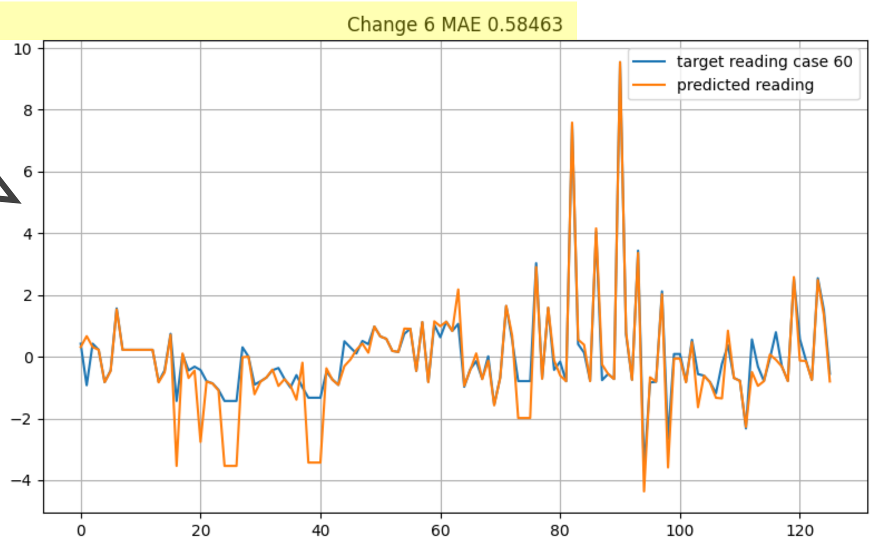


Explainability: Case 54 → Case 60

- consider case 54 vs case 60
  - ✓ 6 settings were changed
- for each setting that changed
  - ✓ predict the reading nodes
  - ✓ compute the MAE with reading nodes from case 60
  - ✓ lower MAE → more important







Summary

- the objective is to leverage deep learning on graph representations of a beamline in order to create a tool for improving the efficiency of beam tuning tasks
- encouraging results in FY24
  - ✓have software to generate a beamline as a graph
  - ✓have software to train a GNN in a self-supervised way
  - ✓demonstrated ability to visualize beamline changes as trajectories in latent space
  - ✓converging on a framework to explain changes in embedding space
- this represents a general framework applicable to any <u>arbitrary beamline</u>

Project Summary: Major Deliverables and Schedule

Project	Deliverable	Date
Graph Learning for Efficient and Explainable Operation of Particle Accelerators	Finalize explainability workflow	01/2025
	Re-train models for use in Spring 2025 CEBAF operation	03/2025
	Design and execute beam study tests for tuning	06/2025
	Document and publish work	09/2025

Project Summary: Annual Budget

	FY 2024	Total
a) Funds allocated	$500,000	$500,000
b) Actual costs to date	$206,993	$206,993
c) Uncosted commitments	$211,917	$211,917
d) Uncommitted funds (d=a-b-c)	$81,090	$81,090

Thank you.

This work is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under Contract No. DE-AC05-06OR23177.