Graph Machine Learning for 2024

Agenda

1. GML Overview
2. Looking Forward: Where is this Going? (Hint: GenAI)
3. Using GML to Enrich a Grounding Knowledge Graph
4. Demo - Semantic Search with Recommendations

Zach Blumenfeld
Product Specialist DS/ML

GML Overview

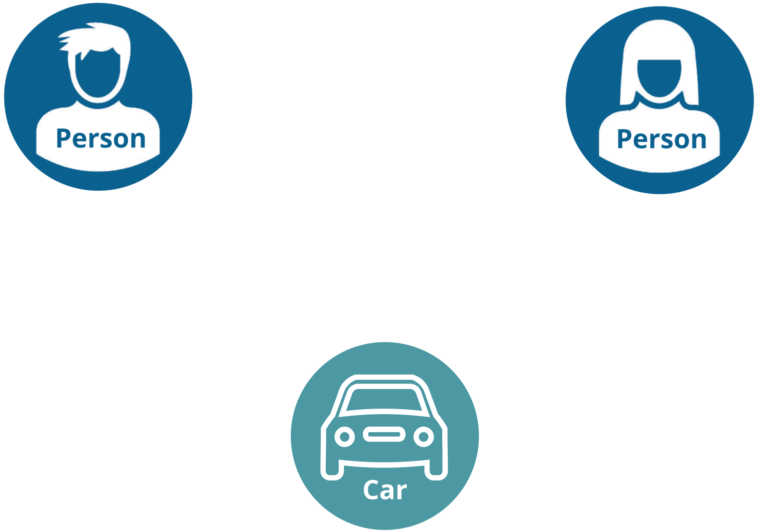
What is a Graph?

Simply put, a graph consists of nodes connected by relationships

Graph databases, like Neo4j, structure and store data as graphs

Property Graph Components

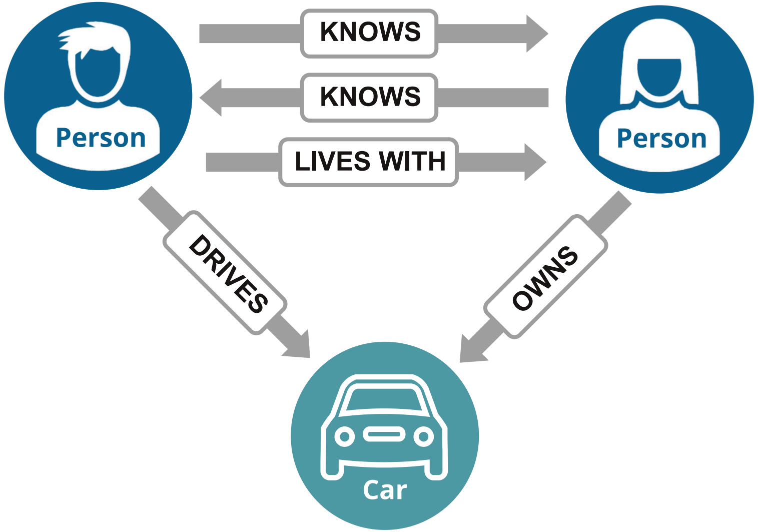
Nodes represent entities in the graph



Property Graph Components

Nodes represent entities in the graph

Relationships represent associations or interactions between nodes

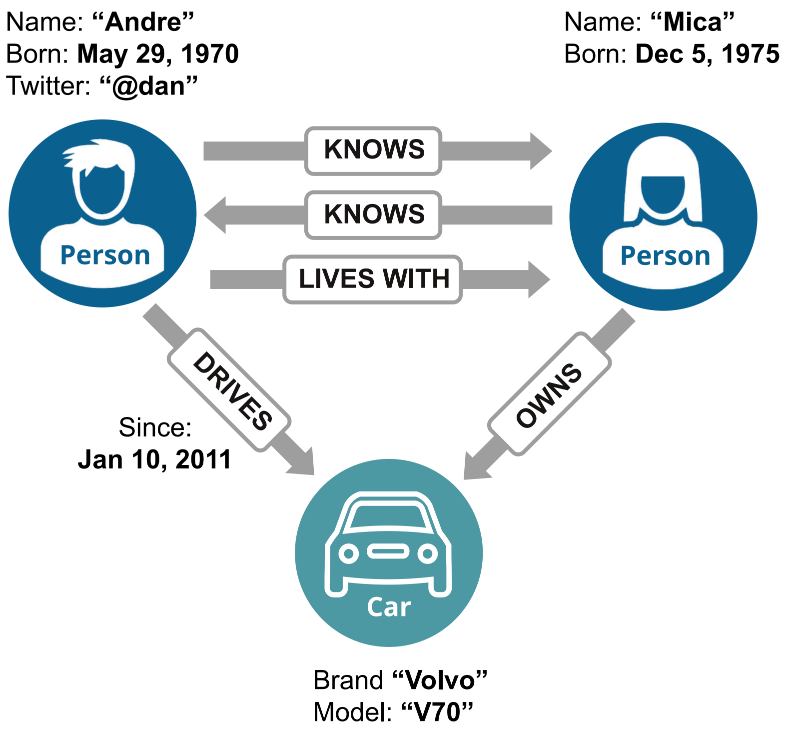


Property Graph Components

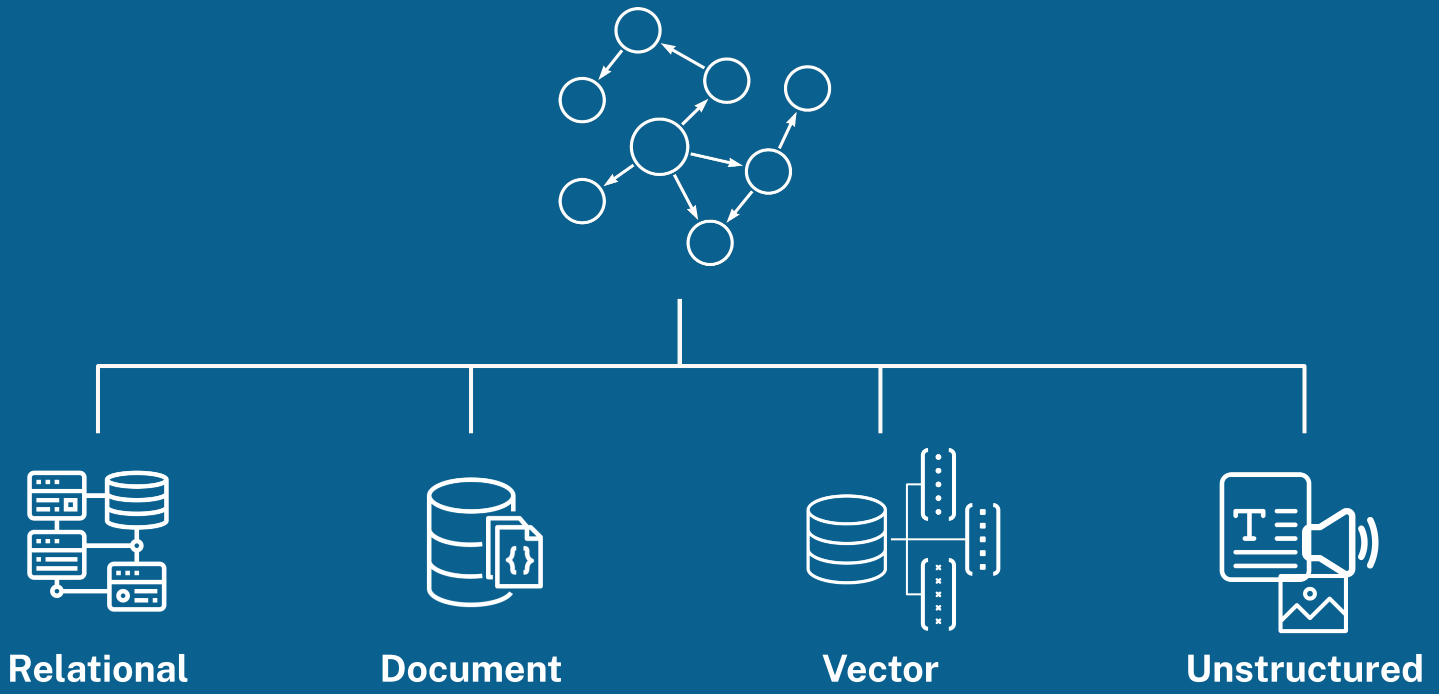
Nodes represent entities in the graph

Relationships represent associations or interactions between nodes

Properties represent attributes of nodes or relationships



Graph is the Superset of All Data Structures for AI

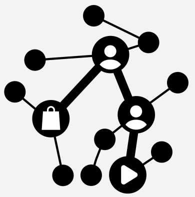


What is Graph Machine Learning (GML)?

The application of machine learning to graphs, specifically for predictive and prescriptive tasks

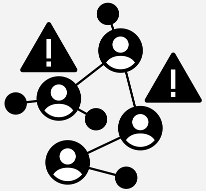
Many Applications

Recommendation



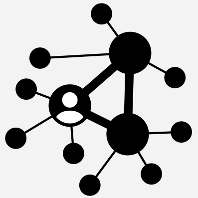
What should I recommend to a customer next?

Fraud Detection



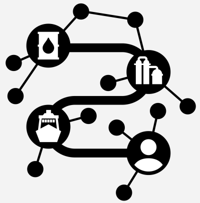
How do I identify fraud rings and suspect activity?

Customer 360



Can I resolve information and relationships across fragmented customer records?

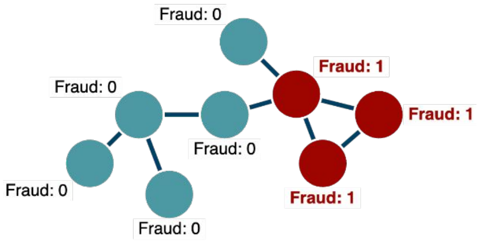
Supply Chain



What is the most optimal route through a supply chain network?

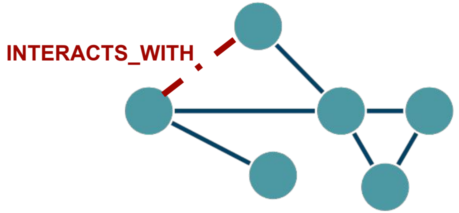
Supervised GML Tasks

Train on labeled graphs to make predictions



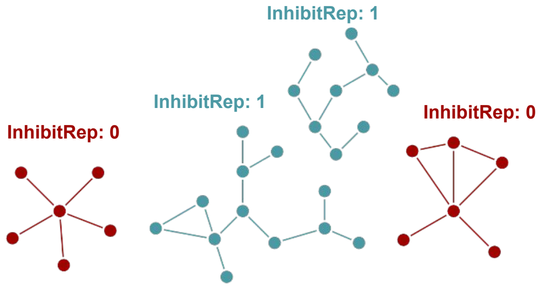
Node Property Prediction

Predict a discrete or continuous node property, called **node classification** and **node regression** respectively.



Link Prediction

Predict if a relationship should exist between two nodes. Often a binary classification task, but can sometimes include more link types or continuous properties.

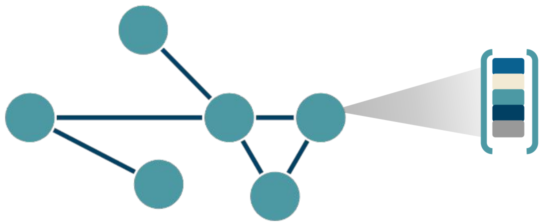


Graph Property Prediction

Predict a discrete or continuous property of a graph or subgraph.

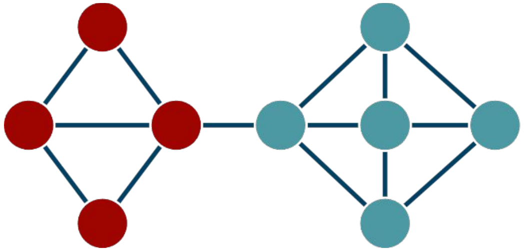
Unsupervised GML Tasks

Training on unlabeled graphs to learn patterns



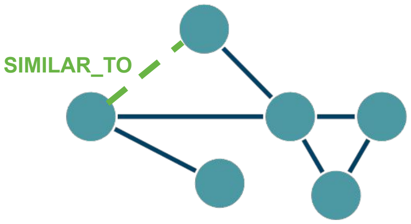
Representation Learning

Automatically generate features based on graph structure for downstream ML and EDA.



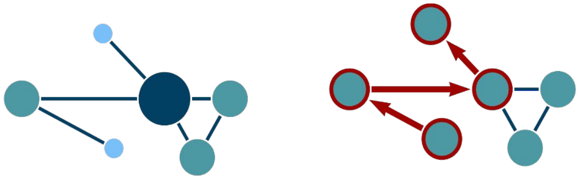
Clustering / Community Detection

Identify groups of nodes that have higher connectivity between each other than the rest of the graph.



Similarity

Find and measure similar pairs of nodes in the graph. Use for recommendation, entity resolution, and more.



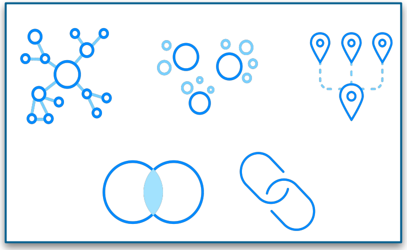
Centrality & Pathfinding

Find important and influential entities in the graph. Identify and evaluate more efficient paths and trees.

How to Accomplish Graph Machine Learning?

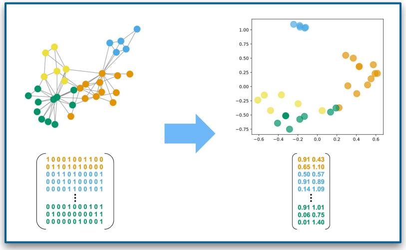
Classic Graph Algorithms

Results from algorithms like pagerank for centrality, Louvain for community detection, or node similarity.



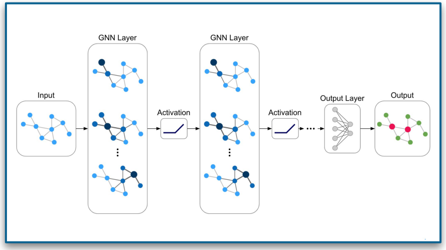
Non-GNN Graph Embeddings

Low-dim vector representations of nodes s.t. similarity between vectors approximates similarity between nodes
(can also be for triples, paths, or sub-graphs)



Graph Neural Networks (GNNs)

End-to-end solution for the ML task. Compression happens in hidden layers and is learned during model training



Graph Algorithms

- Use independently for unsupervised community detection, similarity, centrality, or pathfinding.
- Use as features for conventional downstream models, such as linear and logistic regressions, random forests, or neural networks to perform GML tasks.

Pathfinding & Search

- A* Shortest Path
- All Pairs Shortest Path
- Breadth & Depth First Search
- Delta-Stepping Single-Source
- Dijkstra Single-Source
- Dijkstra Source-Target
- Minimum Spanning Tree & K-Spanning Tree
- Random Walk
- Yen's K Shortest Path
- Minimum Directed Steiner Tree
- Topological Sort
- Longest Path

Community Detection

- Conductance Metric
- K-1 Coloring
- K-Means Clustering
- Label Propagation
- Leiden Algorithm
- Local Clustering Coefficient
- Louvain Algorithm
- Max K-Cut
- Modularity Optimization
- Speaker Listener Label Propagation
- Strongly Connected Components
- Triangle Count
- Weakly Connected Components

Centrality

- ArticleRank
- Betweenness Centrality & Approx.
- Closeness Centrality
- Degree Centrality
- Eigenvector Centrality
- Harmonic Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (CELF)
- PageRank
- Personalized PageRank

Topological Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- K-Nearest Neighbors (KNN)
- Node Similarity
- Filtered KNN & Node Similarity
- Cosine & Pearson Similarity Functions
- Euclidean Distance Similarity Function
- Euclidean Similarity Function
- Jaccard & Overlap Similarity Functions

Node Embeddings

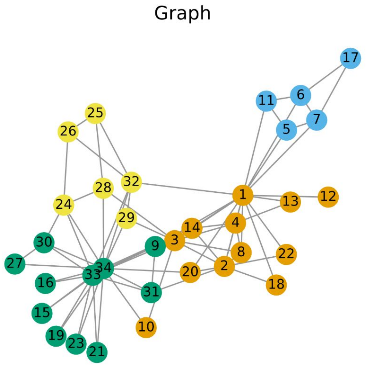
Most Common Type of Graph Embedding

What?

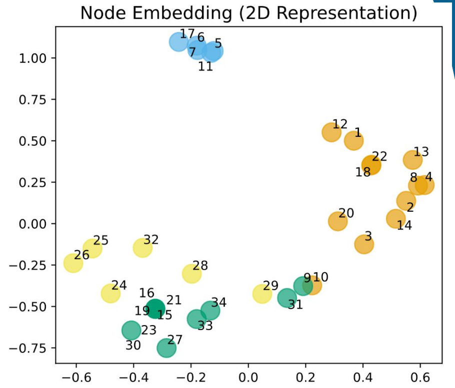
- Low-dimensional representations of nodes
- Similarity between vectors approximate similarity between nodes in the graph

Why?

- Save time & reduce work for generating ML features - Generating custom features manually can be time consume and imprecise, embeddings help automate and scale the process.
- Increase downstream ML performance



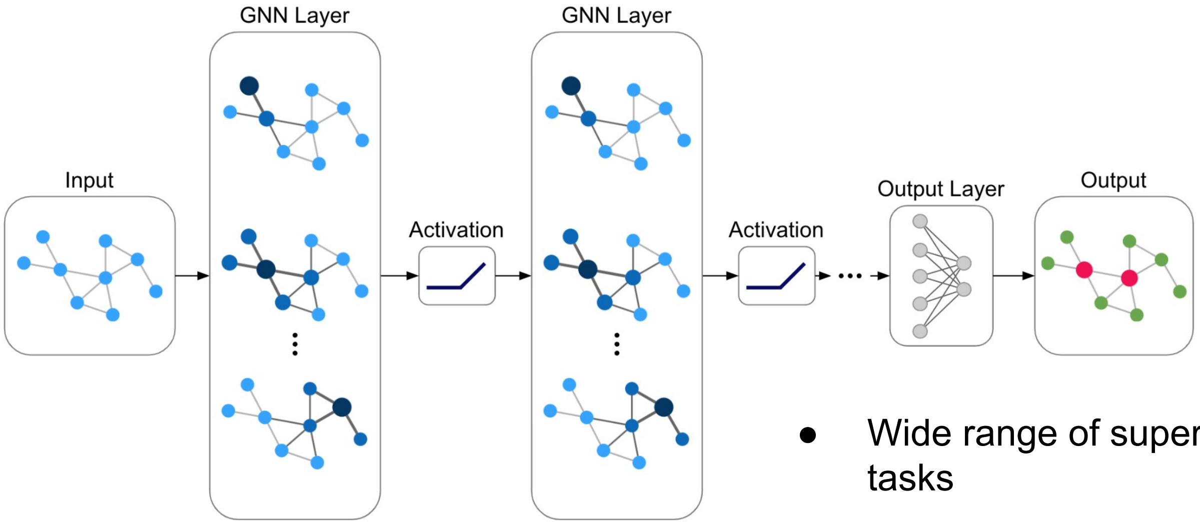
$$\begin{pmatrix} 1\,0\,0\,0\,1\,0\,0\,1\,1\,0\,0 \\ 0\,1\,1\,0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,1\,1\,0\,1\,0\,0\,0\,0\,1 \\ 0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,1\,0\,0\,1\,0\,1 \\ \vdots \\ 0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1 \end{pmatrix}$$



$$\begin{pmatrix} 0.91 & 0.43 \\ 0.65 & 1.10 \\ 0.50 & 0.57 \\ 0.91 & 0.89 \\ 0.14 & 1.09 \\ \vdots & \\ 0.91 & 1.01 \\ 0.06 & 0.75 \\ 0.01 & 1.40 \end{pmatrix}$$

Graph Neural Networks (GNN)

Take Graph as Input → **Transform Into Intermediate Embeddings** → **Feed into Final Layer for a Prediction Task**



- Wide range of supervised and unsupervised tasks
- Common examples: GCN, GraphSAGE, GAT

GNNs Have Their Pros and Cons

Pros

- Can automatically learns important signals in the graph
- Most recent GNNs are inductive (train a model and predict on new graph data)
- Potential to handle deep complex graph structure
- End-to-end solution for supervised learning

Cons

- Relatively complicated. Can still be difficult to construct, tune, and avoid overfitting. Requires high degree of technical expertise.
- Can be difficult to scale. High time & space complexity. Usually requires accelerated hardware - like GPU.
- Limited depth. Usually “shallow” to prevent over-smoothing and/or reaching the diameter of the graph
- Low interpretability/explainability

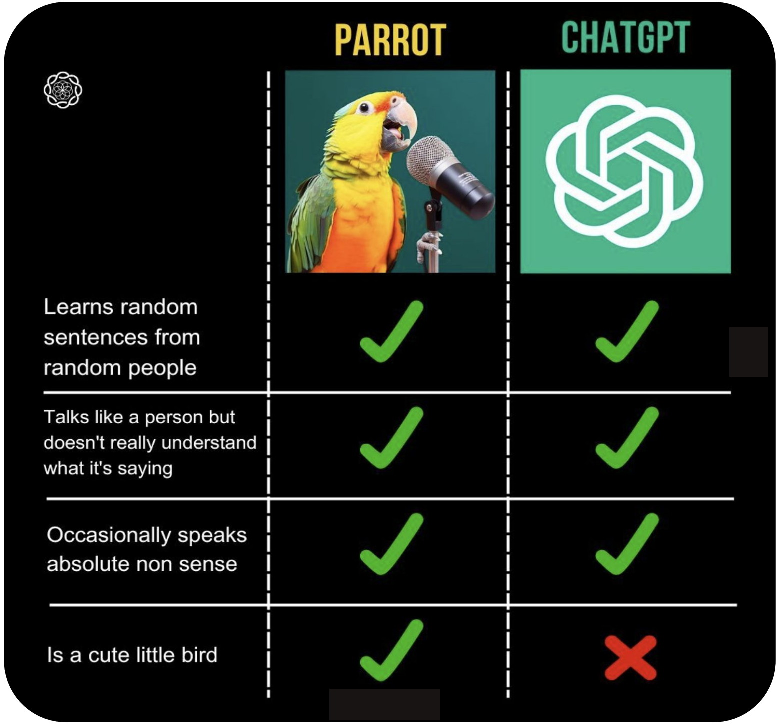
Looking Forward - Where is this Going?

Generative AI is Taking Storm!

1. Automate data retrieval tasks
2. Improve customer service experiences
3. Expedite reading, understanding, & summarizing
4. Generate content & code

POWERING GENERATIVE AI APPS

What GenAI Can't Do!



Lack of enterprise domain knowledge

Limited input sizes for fine tuning

Inability to verify answers

Sensitive to prompt phrasing & injection

Hallucinates

Ethical & data bias concerns

neo4j

Generative AI, LLMs and Knowledge Graphs

Knowledge
Facts
Context

KGs

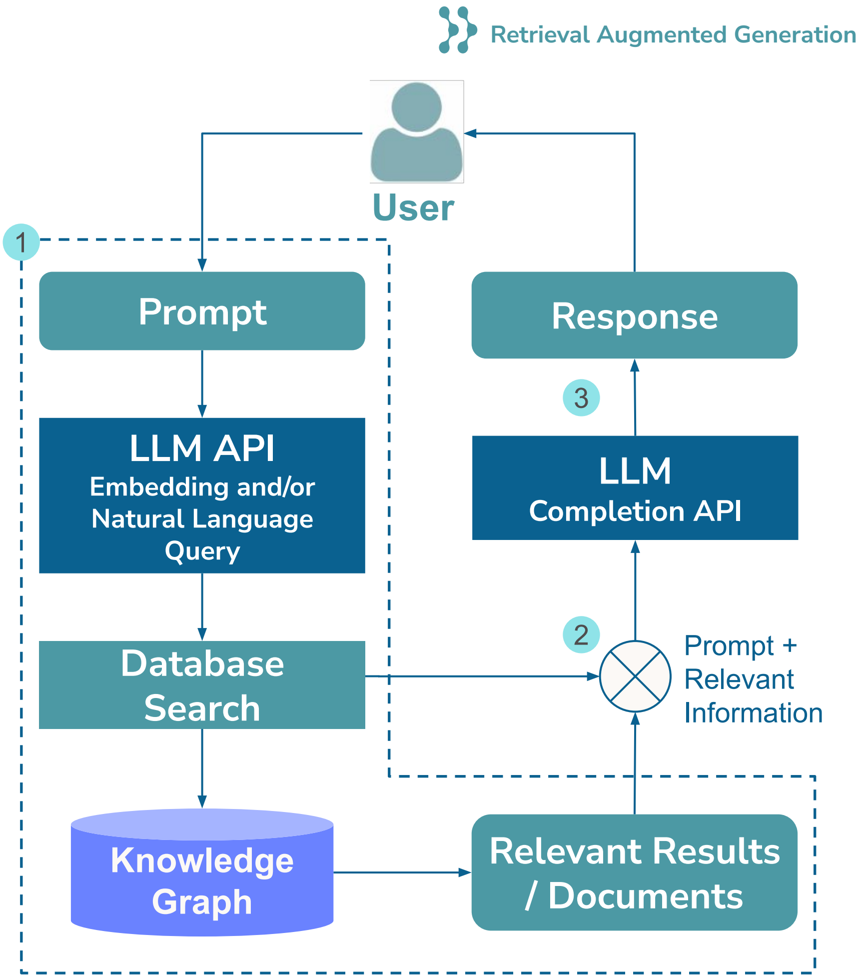
LLMs

Language
Statistics
Creativity

BETTER TOGETHER

neo4j

RAG enables… Natural language search on factual information retrieved from a database



Where is GML Valuable?

In Model Enhancement

Use GML in Language Models & Foundation Modeling

Knowledge Graph Enrichment

Use GML to Enrich Knowledge Graphs for Improved RAG

Graph Machine Learning in GenAI Models

Large graph model research underway in academia

GreaseLM & **DRAGON**

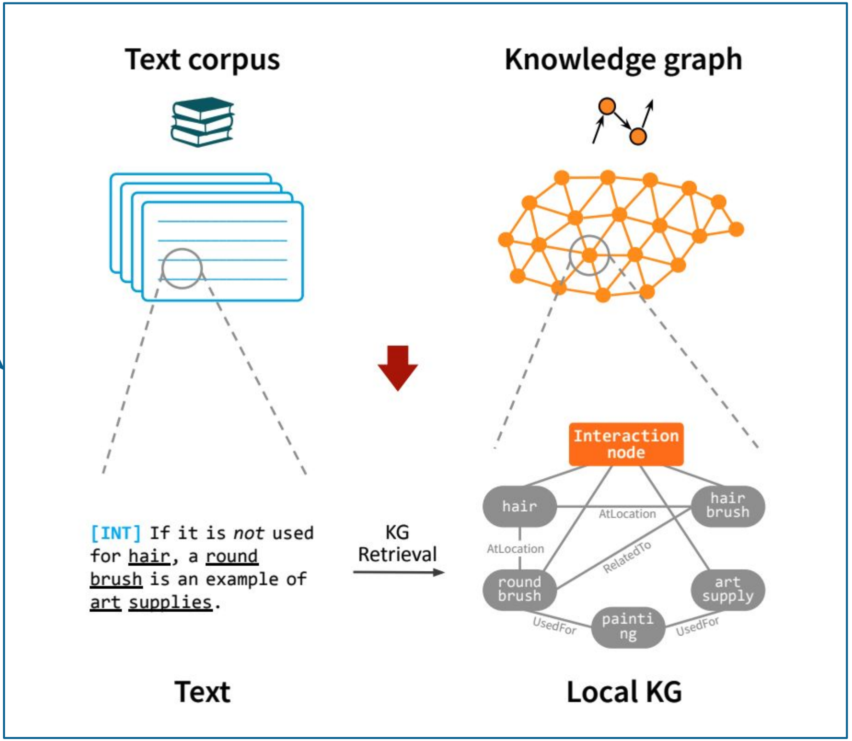
High-Level Methodology:

1. **Retrieve:** Retrieve entities from KG based on prompt using Entity Linking
2. **Fuse:** Leverage model that fuses LM and GNN together to generate a response based on prompt and subgraph

Graph Machine Learning in GenAI Models

GreaseLM (**G**raph **Reas**oning **E**nhanced **L**anguage **M**odels For Question Answering)

1. Retrieve (Entity LInking)
2. Fuse (Cross-Modal Layers)

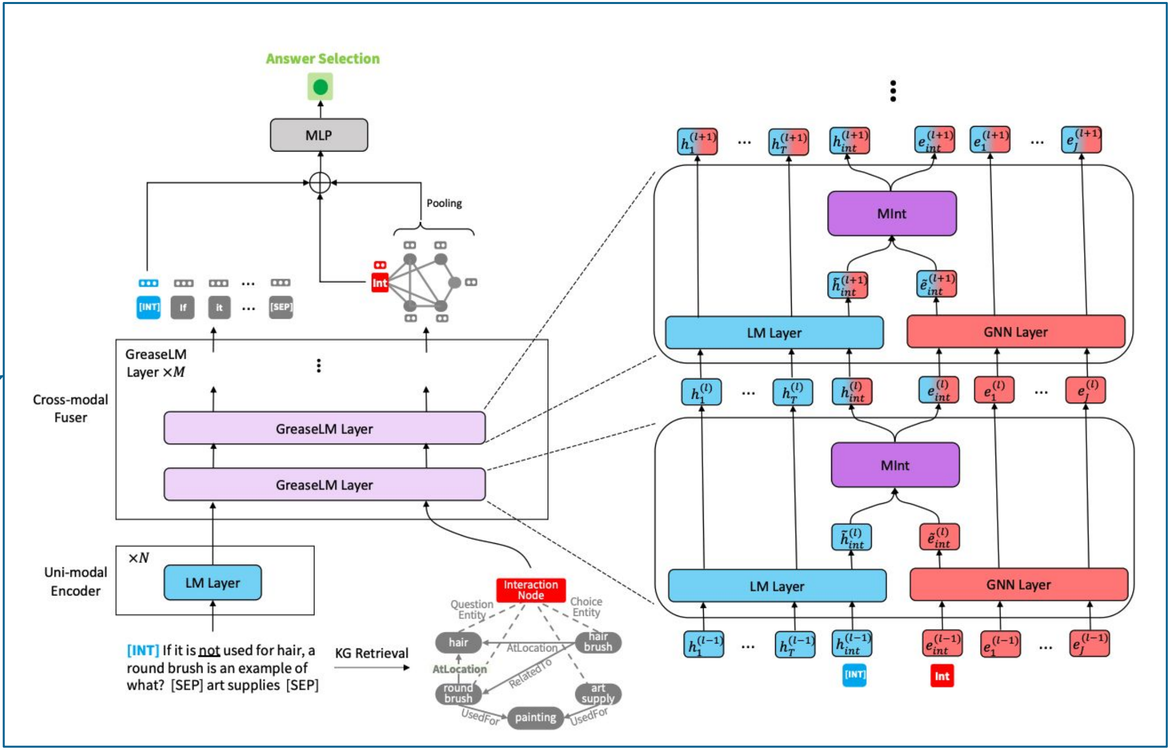


Graph Machine Learning in GenAI Models

GreaseLM (**G**raph **Reas**oning **E**nhanced **L**anguage **M**odels For Question Answering)

1. Retrieve (Entity LInking)
2. Fuse (Cross-Modal Layers)

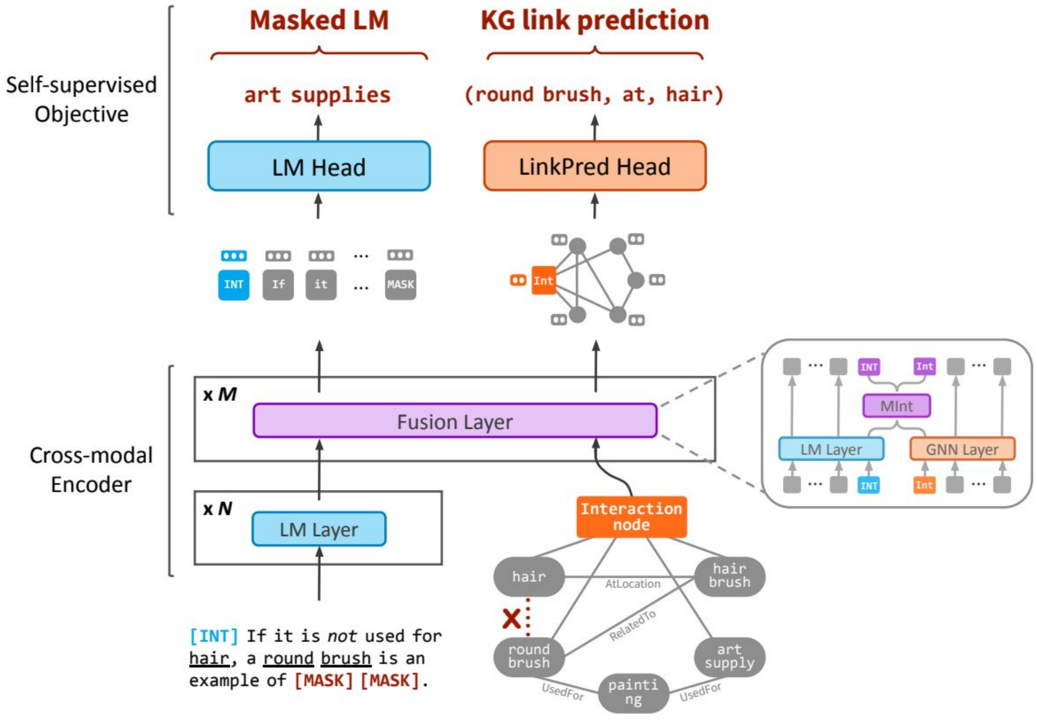
*Use Pre-Trained LM and GNN



Graph Machine Learning in GenAI Models

DRAGON (**D**eep Bidi**r**ectional L**a**n**g**uage-Kn**o**wledge Graph Pretrai**n**ing)

Similar to GreaseLM but as its own foundation model with pre-training / self-supervised objective function



Graph Machine Learning in GenAI Models

Quick Note on Upcoming Work: Integrating Semi-Structured Knowledge into Language Models

- Splitting prompts into sub-portions for different structured and unstructured tasks
- Use different retrieval mechanisms (graph, text, etc.) for each subportion as appropriate

Still In Development. Research Ongoing

Graph Machine Learning in GenAI Models

Overall:

- **Future potential** - Research indicates Graph Models in RAG paradigm can improve performance in some instances
- **Still very new** - Nascent academic research at this point
- **Not the most accessible** for application at the moment

Where is GML Valuable?

In Model Enhancement

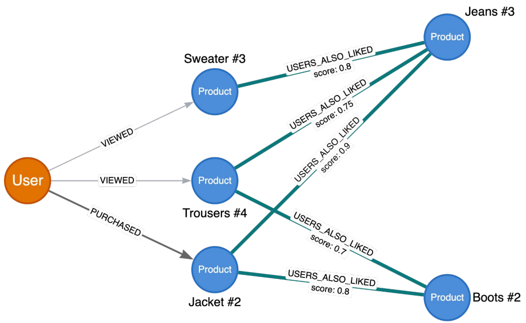
Use GML in Language Models & Foundation Modeling

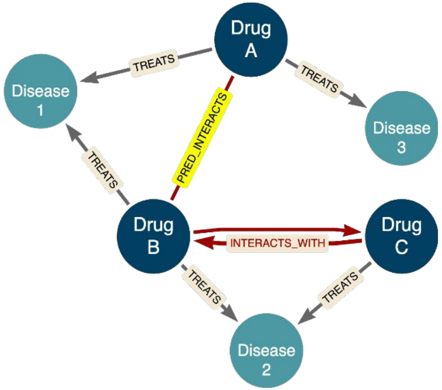
Knowledge Graph Enrichment

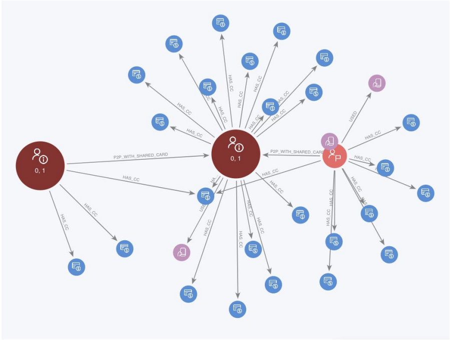
Use GML to Enrich Knowledge Graphs for improved RAG

GML in Knowledge Graph Enrichment

Enhance the Graph to enable LLMs to not only use facts, but also make intelligence inferences on your enterprise data







Similarity for Recommendation and Entity Resolution

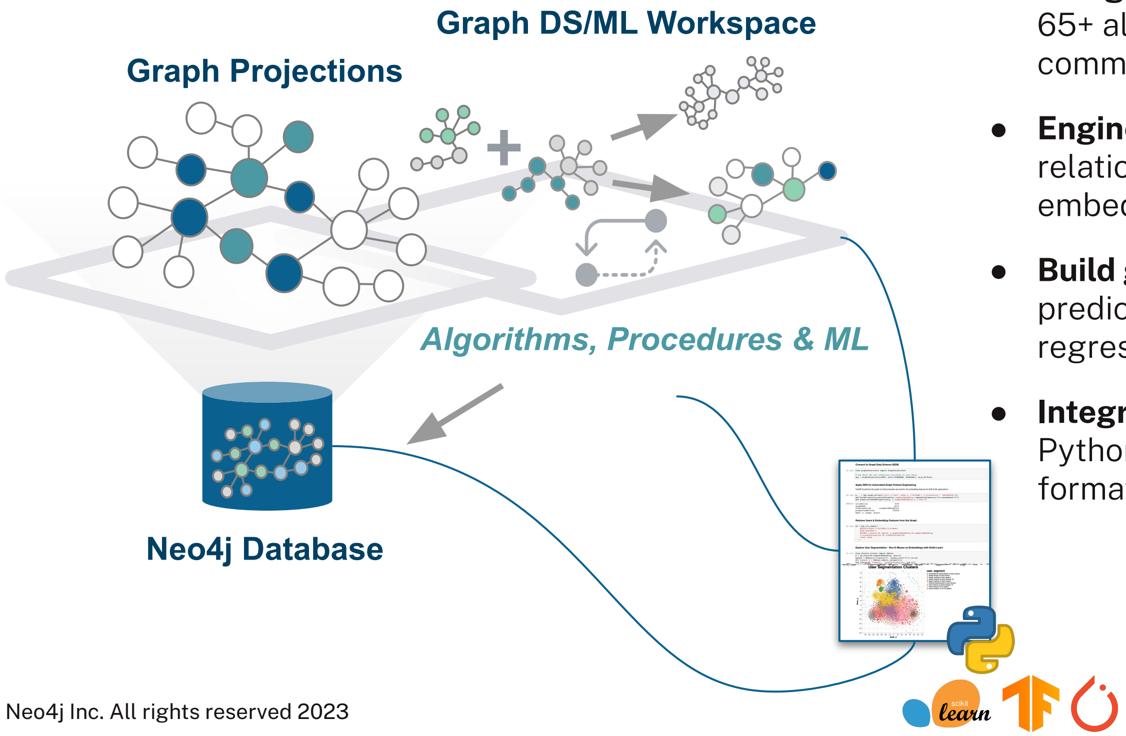
Link Prediction to Discover Missing Relationships

Node Prediction, Community Detection & Centrality Scoring For Tagging and Identifying Entities

Demo with Neo4j

Neo4j Graph Data Science

highly optimized, massively parallel, scalable



- **Run graph algorithms to generate insights:** 65+ algorithms across centrality, path finding, community detection, similarity, and more
- **Engineer graph features for ML:** Leverage relationship information with algorithms & node embeddings
- **Build graph native ML pipelines:** Link prediction, node classification & property regression
- **Integrate with external ML frameworks:** Python client, blazing fast import & export, formatting for dataframes and tensors

Demo

neo4j-product-examples/ml-genai/tree/main/retail-hm

Thank you!

