Seeing the System: A World Model for Scheduling

Zhenghao Gong z3gong@ucsd.edu
Bingyan Liu bil010@ucsd.edu
Stephanie Wang yuw168@ucsd.edu
Weijie Zhang wez042@ucsd.edu
Mentor: Phi Nguyen pnguyen@sdge.com
Mentor: Fatemeh Aarabi FAarabi@sdge.com
Mentor: Mike Hilden MHilden@sdge.com

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

01 INTRODUCTION

WebSite

Shift from Piecewise Prediction to Graph Modeling to capture systemic interactions and hidden links in scheduling:

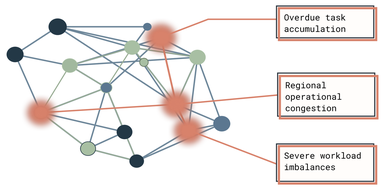
- **Operational Links:** Dependencies and priorities.
- **Human Factors:** Team collaboration and workload.
- **Proximity:** Spatial and temporal patterns.

The limits of short term rule satisfaction

Executing tasks in isolation.

Current industrial scheduling pipelines emphasize feasibility and rule based optimization.

Hidden cascading effects.

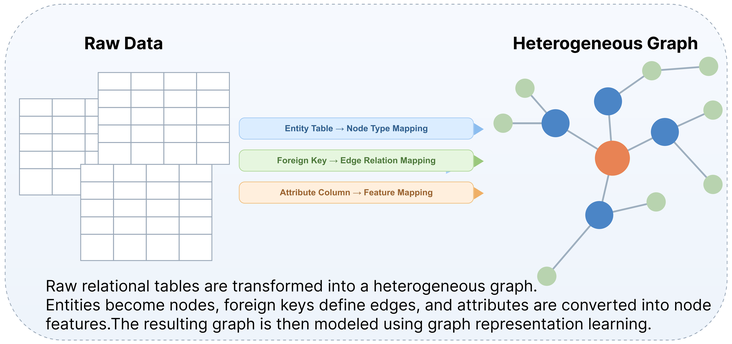


02 METHOD

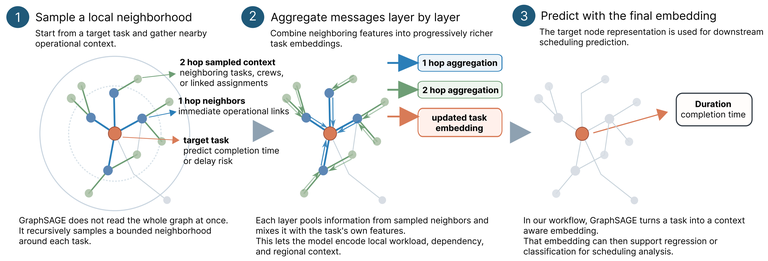
System Scale: Analyzed ~10M historical activity records to capture real-world scheduling dynamics.

Heterogeneous Architecture:

- **Nodes (7 types):** Tasks, Assignments, Engineers, Districts, Departments, Task Status, and Task Types.
- **Edges (20 types):** Encode operational links (Task-Assignment), organizational context (Assignment-Engineer), and regional metadata.

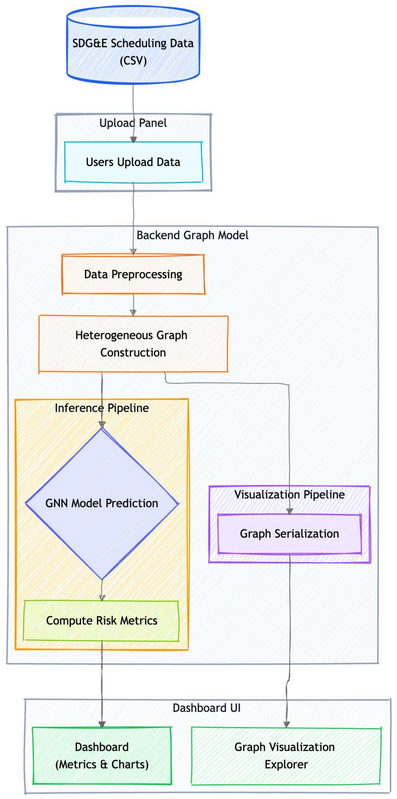


GraphSAGE Learning:



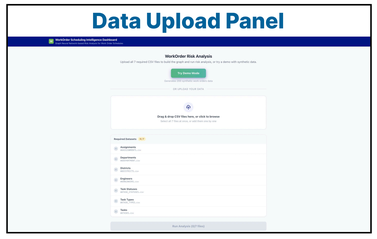
03 APPLICATION

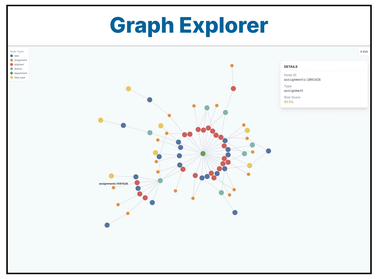
Application Workflow



- **Data Ingestion:** Validate input tables, standardize schema, and generate scheduling features.
- **Graph Construction:** Convert relational data into a heterogeneous graph connecting tasks, engineers, districts, and assignments.
- **Model Inference:** Train a Graph Neural Network (GNN) to capture relational dependencies and predict task completion time.
- **Risk Analysis:** Generate assignment-level predictions and compute aggregated operational metrics.
- **Visualization:** Present model predictions through an interactive dashboard and graph explorer.

Application Demo

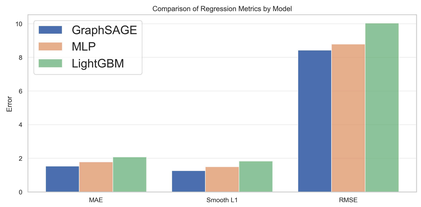




- **Data Upload Panel**: Upload required datasets and initiate the scheduling analysis workflow.
- **Dashboard**: Display key scheduling metrics (completion time, workload imbalance, district-level performance, etc).
- **Graph Explorer**: Visualize node relationships and predicted risk in the scheduling network.

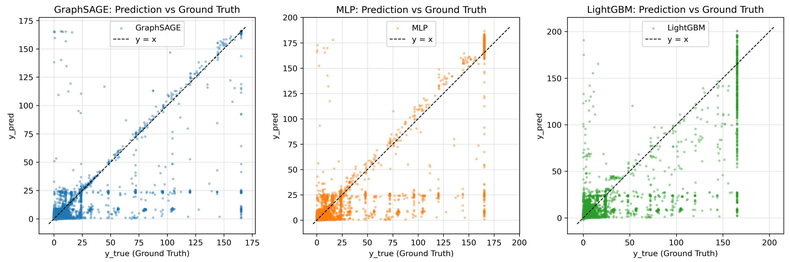
04 RESULT

We compare **GraphSAGE**, **MLP**, and **LightGBM** for task completion time prediction.
GraphSAGE leverages **relational information** from the **heterogeneous graph**, while MLP and LightGBM rely on tabular features.
GraphSAGE achieves the best performance across all metrics.



Scheduling tasks are **relational and interdependent**.
GraphSAGE aggregates neighbor information in the graph,
leading to **more accurate predictions**.

Piecewise models: partition space → leaf average → repeated predictions → horizontal bands
GraphSAGE: continuous embedding + relational aggregation → smoother predictions



05 CONCLUSION

Key Takeaway:

- This work shows that modeling utility scheduling as a **heterogeneous graph of interconnected entities** capture operational dependencies overlooked by traditional tabular approaches and improves predictive performance.
- The proposed pipeline demonstrates how graph learning can be applied to analyze large-scale operational scheduling systems.

Limitation:

- Our current graph representation does not fully capture dynamic scheduling changes, sparse engineer collaborations, or external factors such as weather and emergency priorities.

Future Work:

- Future work can extend the framework with richer contextual data and advanced graph models to support dynamic schedule evaluation and future decision-support systems.

References:
[1] Hamilton et al., Inductive Representation Learning on Large Graphs, NeurIPS 2017
[2] Hu et al., Heterogeneous Graph Transformer, WWW 2020
[3] Schlichtkrull et al., Modeling Relational Data with Graph Convolutional Networks, ESWC 2018...