NC STATE UNIVERSITY

Knowledge Graph-Based Multi-Agent System for Fertilizer Optimization

Nahed Abu Zaid, Jaya Shruti Chintalapati, Mihir Shah

CSC 722: Advanced Topics in Machine Learning

Professor: Ranga R. Vatsavai

Overview

- KG-MASFO: Modular AI using KGs, GATv2, and multi-agent reasoning
- Specialized agents process soil, crop, and weather subgraphs
- Meta-agent fuses outputs into tailored fertilizer suggestions
- Symbolic module adds interpretable agronomic explanations
- Validated on *Isanti City*, MN — outperforms baselines

Outline

- Motivation
- Problem Statement
- Our Proposed Approach: KG-MASFO
- Experiments and Results
- Strengths & limitations
- Conclusion & Future Work

Motivation

Motivation

Traditional Methods

- Static, rule-based decisions
- Ignore dynamic soil and weather changes
- Cause runoff, yield loss, and ecological harm

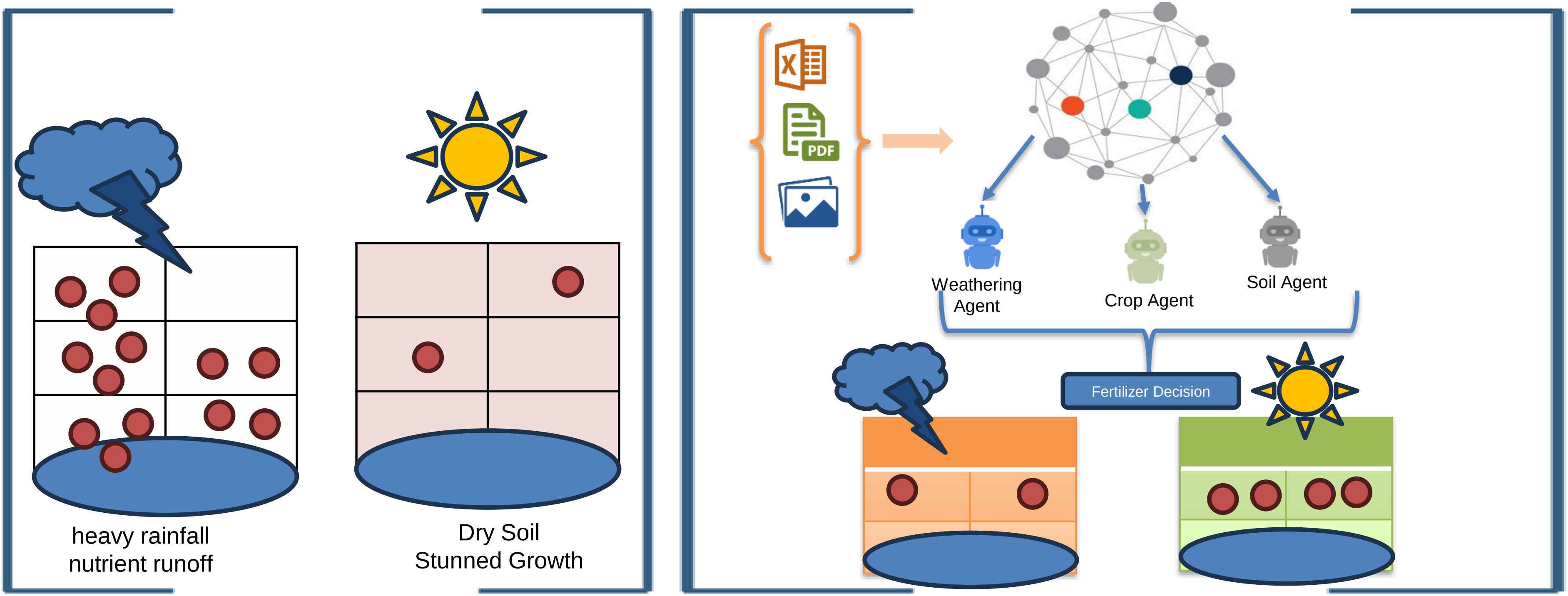
AI System Gaps

- Use outdated, historical data
- No integration across soil, crop, and weather
- Lack transparency and reduce user trust

Related Work & Limitations

Domain	Example Works	Limitation
AI in Agriculture	John Deere (2023), Talaviya et al. (2020), Adewusi (2024)	No real-time environmental adaptation
ML in Fertilization	Patel et al. (2021), Sharma et al. (2023)	Static models, no domain coordination
Knowledge Graphs	Yan et al. (2025), Gelal et al. (2024), Garcia et al. (2024)	No real-time reasoning or GNN integration
Multi-Agent Systems	Mahajan et al. (2024), SmythOS (2024)	Agents act in silos, lack unified logic

Motivating Example



A. Traditional Fertilizer system

B. KG-powered Multi agent system

Goal

To develop a **scalable**, **explainable**, and **real-time AI system** that leverages knowledge graphs and multi-agent reasoning to optimize fertilizer recommendations based on dynamic soil, crop, and weather data.

Problem Formulation

Problem Statement

Given a set D of heterogeneous agricultural datasets—including soil, crop, fertilizer, and weather data—the objective is to **construct a unified knowledge graph (KG)** that semantically integrates these sources and to **enable fertilizer decision-making** via a **modular multi-agent system**.

The system must satisfy the following constraints:

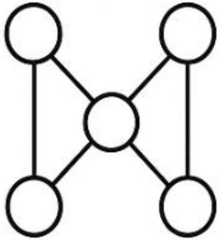
(i) Handle noisy and incomplete data;
(ii) Preserve agent-level modularity;
(iii) Ensure interpretability through rule-based explanations.

The KG-MASFO Workflow

KG-MASFO Methodology

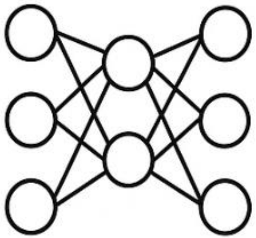
Goal: Build a modular, interpretable, real-time system for fertilizer optimization

Phase1:
Knowledge graph Construction



→

Phase2:
Graph Neural Network Learning



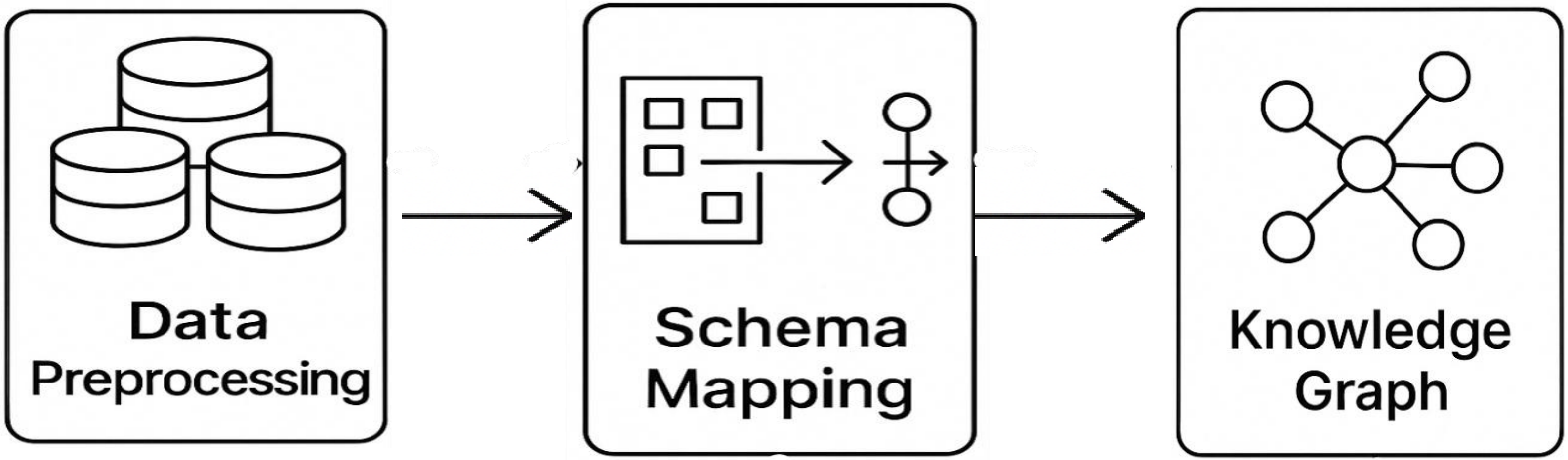
→

Phase3:
Agent System Reasoning

- Single Agent
- Multi Agents
- Multi Agents + Post-hoc Symbolic Exlan.

Phase 1: Knowledge Graph Construction

Goal: is to construct a unified knowledge graph by preprocessing and mapping this data into semantically meaningful triples



(Soil, hasPH, pH_Value)

- 3,792 nodes
- 7,708 edges

Phase 2 - GNN Learning: Fertilizer Prediction via Graph Attention (Example)

Goal: To **train a GNN model** that takes soil, crop, and weather information from a **Knowledge Graph** and predicts the **fertilizer amount (kg/ha)**.

Input Tables

Feature Matrix X

Soil Table

ID	pH	Nitrogen (ppm)	Moisture (%)
Soil_1	6.5	5	30

Crop Table

ID	Crop Type	Growth Stage (%)	Nutrient Demand
Corn_1	Corn	25	High

Weather Table

ID	Temp (°C)	Rainfall (mm)	Humidity (%)
Weather_ 1	20	5	60

Target Value → Fertilizer needed = 40 kg/ha

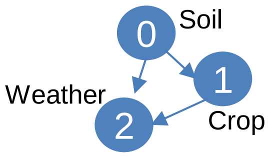
Step A: Graph Tensor Construction

Normalized X:

```
X = [pH   Nitrogen   Moisture
  [   0.0,   1.0,      0.0],    # Soil
  [1.0,       0.0,      - ],    # Crop
  [0.789,     1.0,      1.0]    # Weather
]
```

Graph Edges E

```
E = [
  [0, 0, 1],  # From
  [1, 2, 2]   # To
]
```



$$\mathrm{MinMax}(z_i) = \frac{z_i - \min(z)}{\max(z) - \min(z)}$$

y_normalized = (40 - 20) / (50 - 20) = 0.67

Min = 20, Max = 50

Step B: GNN Model Definition

$$\boxed{f = GATv(X,E)}$$

$$L = \begin{cases} 0.5(\hat{y} - y)^2 & \text{if } |\hat{y} - y| < 1 \\ |\hat{y} - y| - 0.5 & \text{otherwise} \end{cases}$$

Step C: Training Loop with Early Stopping

- Predict fertilizer amount y^
- Compute loss
- Update model
- Stop if no improvement after P epochs

Step D: Inference

- Reload best checkpoint
- Output prediction:

$$\hat{y} = 0.67 \Rightarrow \text{Unnormalize} = 0.67 \times (50 - 20) + 20$$

$$\boxed{40.1\ \mathrm{kg/ha}}$$

Phase 3 - A GNN Training (Single-Agent)

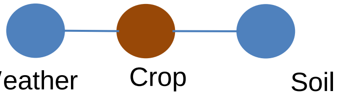
Goal: training **a GATv2-based neural network** on a **typed subgraph**, where: (i) Each subgraph is **domain-specific** (Soil, Crop, or Weather). (ii) A single agent (e.g., "Soil Agent") learns from its subgraph. (iii) The goal is to **predict a continuous target value**, like "Nitrogen %".

Input Node Features

Predict the fertilizer needed for a crop using features from:

Node	Feature Vector
Soil	[pH=6.5, N=5]
Crop	[Growth=29%]
Weather	[Rainfall=5mm]

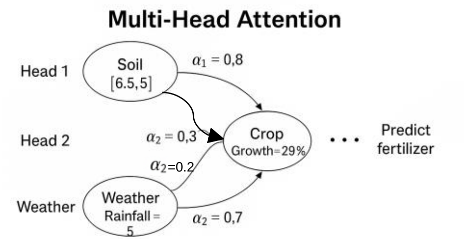
Multi-Head Attention



Crop is the target node (center), and it will attend to **Soil** and **Weather**.

Node	Feature Vector
Soil	[6.5, 5]
Crop	[29]
Weather	[5]

Multi-Head Attention (K = 2):



Soil [pH, N] → Crop
Weather [Rainfall] → Crop

Aggregation Formula (*):

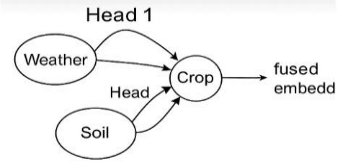
$$h^{(l+1)}_{\text{Crop}} = \Big\|_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k)} \cdot \mathbf{W}^{(k)} \cdot h_j^{(l)} \right)$$

Where:

- $\alpha_{ij}^{(k)}$ is the attention weight from neighbor j
- $\mathbf{W}^{(k)}$ is a learnable transformation matrix
- $\|$ means concatenating both heads' output
- σ is an activation function like ELU

Final Crop Representation:

After multi-head attention, the crop node receives a fused embedding that integrates soil and weather features, **each weighted by head-specific importance.**



Optimization

Training Setup
•Optimizer: **AdamW**
•Loss: **Smooth L1 (Huber)**
•Activation: **ELU**

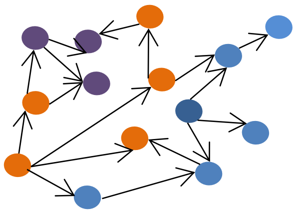
Final Crop Embedding → Used to Predict Fertilizer (kg/ha)

Phase 3 – B Multi-Agent Fusion

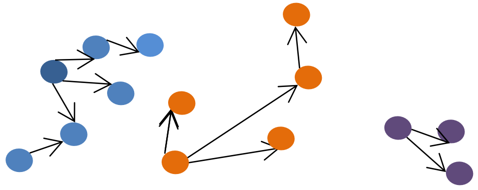
Goal: To **combine** the predictions **from multiple specialized GNNs (agents)** into a **single final fertilizer prediction**. Each agent sees one type of data: soil, crop, or weather.

Step A: Setup

Divide the entire **Knowledge Graph (KG)** into **3 subgraphs**:



Each subgraph is processed **separately** by a GNN agent.

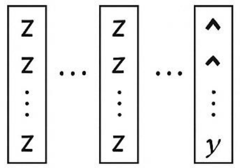


Step B: Fusion Layer

Concatenate all embeddings from the 3 agents into one long vector (Fusion Layer)

$$Z = [Z_{\text{soil}} \,\|\, Z_{\text{crop}} \,\|\, Z_{\text{weather}}]$$

Step C: Feed into MLP(*)



$$\hat{y} = f_{\text{fuse}}(Z) \quad \textbf{(…..**)}$$

Step D: Training with Joint Optimization

Update parameters in all GATv2 agents: $f_{\text{soil}},\ f_{\text{crop}},\ f_{\text{weather}}$

Update parameters in the fusion MLP

$$f_{\text{fuse}}$$

Comparison using SmoothL1Loss:

$$\mathcal{L}_{\text{cur}} = \text{SmoothL1Loss}(\hat{y}, y)$$

Step E: Apply Early Stopping

Stop if loss does not improve for P epochs

Step F: Final Inference

- Reload the best saved model
- Use it to predict new fertilizer amounts given new soil/crop/weather data.

NC STATE UNIVERSITY

Phase 4 - Symbolic Post-Hoc Explanation

Goal: To generate an interpretable symbolic rule that approximates the behavior of the trained GNN model, allowing users to understand why a specific fertilizer prediction was made based on input features like soil pH and crop progress.

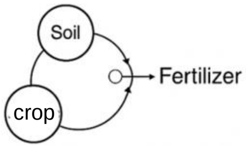
Step A: Rule Induction

Given a trained predictive model, we aim to infer a symbolic rule that explains its behavior.

Feature	Value
pH	5.0
Progress (%)	10

Step B: Rule Exploration

We explore symbolic expressions (rules) to approximate the model's predictions using graph-based features (e.g., Soil, Crop).



Step C: Rule Selection

We evaluate each rule using a combined score of prediction loss and rule complexity. Only rules satisfying

$$\alpha < \chi^*$$

are selected.

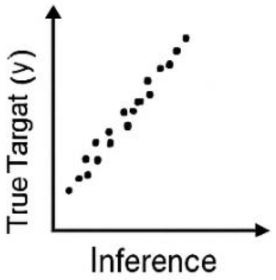
Step D: Filtering & Final Rule

Implausible rules are filtered. The final interpretable rule might look like:

Fertilizer=2.1×pH+1.3×Progress

Step E: Validation

We assess the final rule's quality by comparing predicted and true values. Ideally, predictions align well with ground truth.



Experimental Setup

Location: Isanti City, Minnesota

Data Sources:

- USDA Web Soil Survey: https://websoilsurvey.nrcs.usda.gov/app/WebSoilSurvey.aspx
- Visual Crossing Weather: https://www.visualcrossing.com/weather-query-builder/
- EDI Fertilizer Dataset: https://portal.edirepository.org/nis/simpleSearch
- NASS Crop Data : https://www.nass.usda.gov/Statistics_by_State/Minnesota/

Baselines:

- GNN GAT
- Single-Agent GAT
- Multi-agent GAT

Evaluation metrics

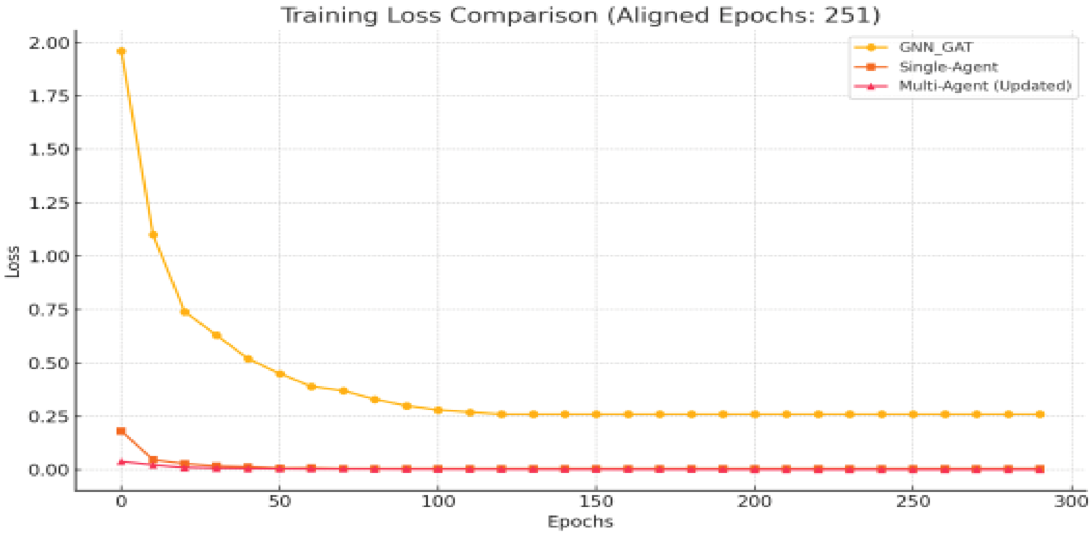
- Mean Squared Error (MSE)
- R-Squared

Experimental Results

Model Variant	MSE ↓	R^2 Score ↑
GNN_GAT	0.0074	0.9025
Single-Agent GAT	0.0058	0.9515
KG-MASFO	**0.0020**	**0.9739**

Performance comparison across model phases. The proposed **multi-agent KG-MASFO** framework achieves the **lowest MSE** and **highest R^2** score, demonstrating superior prediction accuracy and generalization.

Training Curves



- GNN GAT: Slow convergence, higher loss
- Single-Agent: Faster convergence
- **KG-MASFO: Fast + Stable convergence**

Strength and Limitations

Strengths

- Modular Learning: GATv2 agents specialize in soil, crop, and weather data.
- Context-Aware Decisions: Real-time sensor and weather input drive optimization.
- Scalable & Lightweight: Suitable for edge deployment with efficient agents.
- Transparent Reasoning: Symbolic rules generate clear, interpretable outputs.

Limitations

- Generic Outputs: Predicts dosage (kg/ha), not specific fertilizers (e.g., Urea, DAP).
- Single Objective: Ignores cost, emissions, or economic trade-offs.
- No Interaction Modeling: Lacks nutrient synergy/antagonism (e.g., N vs. K).
- Not Time-Aware: No temporal encoding or dynamic edge tracking.

Future Work

- Fertilizer Type Classification: Predict specific types (e.g., Urea, DAP) in addition to dosage.
- Multi-Objective Optimization: Balance yield, cost, and environmental impact.
- Interaction Modeling: Capture nutrient synergy/antagonism (e.g., N ↔ K effects).
- Temporal GNNs: Use models like T-GAT for season-aware recommendations.
- User Feedback Loop: Adapt predictions based on farmer input to improve trust.

Thank you!