Physics-Informed Machine Learning for SWAT-Based Nutrient Prediction

Mohamed Fawzy Mahmoud[1], Mazdak Arabi[1]

International SWAT Conference in Colorado, USA
24 October 2025

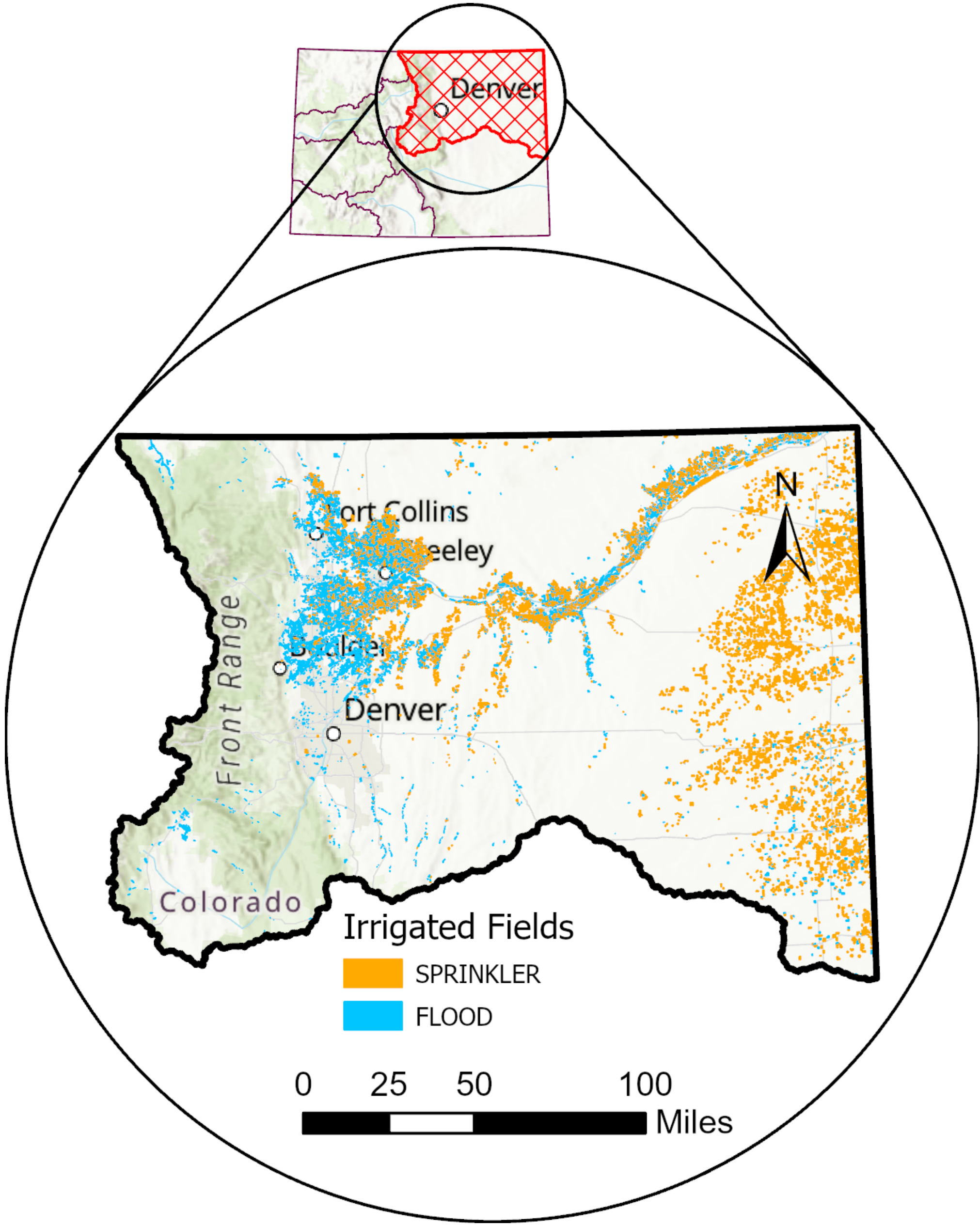
[1]Department of Civil and Environmental Engineering, Colorado State University

Introduction

Background & Motivation

The Power and Challenge of Field -Scale SWAT Modeling

- Field-scale SWAT 2012 for water quality assessments
- Improved auto-irrigation subroutine (Chen et al., 2017; Jobin, 2018)
- One field is represented as a single HRU SWAT model
- ~70K irrigated crop fields across Colorado river basins
- **Goal:** Robust modeling for web-based decision support
- **Challenge:** Computational scalability



The Core Problem

The Scalability Bottleneck

- SWAT is process-based but is computationally slow for large scale computations at the state or regional levels.
- Millions of model simulation are needed to represent thousands of fields and several management scenarios over multi-decade simulation periods.
- Near real-time web applications need fast results.
- Current workflow limits practical deployment.

Exploring Potential Solutions

Finding the Right Path to a Solution

Option 1: Standard Machine Learning (ML)

✓ Fast predictions

✗ "Black box" - physically implausible results

✗ No guarantee of mass balance

Option 2: Knowledge-Guided ML (KGML)

✓ Speed of ML + Scientific integrity

✓ Enforces physical rules during training

✓ "Glass box" - fast, accurate, trustworthy

The KGML-SWAT Emulator

A Hybrid Approach for the Best of Both Worlds

Physics-Informed Loss Functions

Penalize violations of core physical principles.
Enforces process relationships.

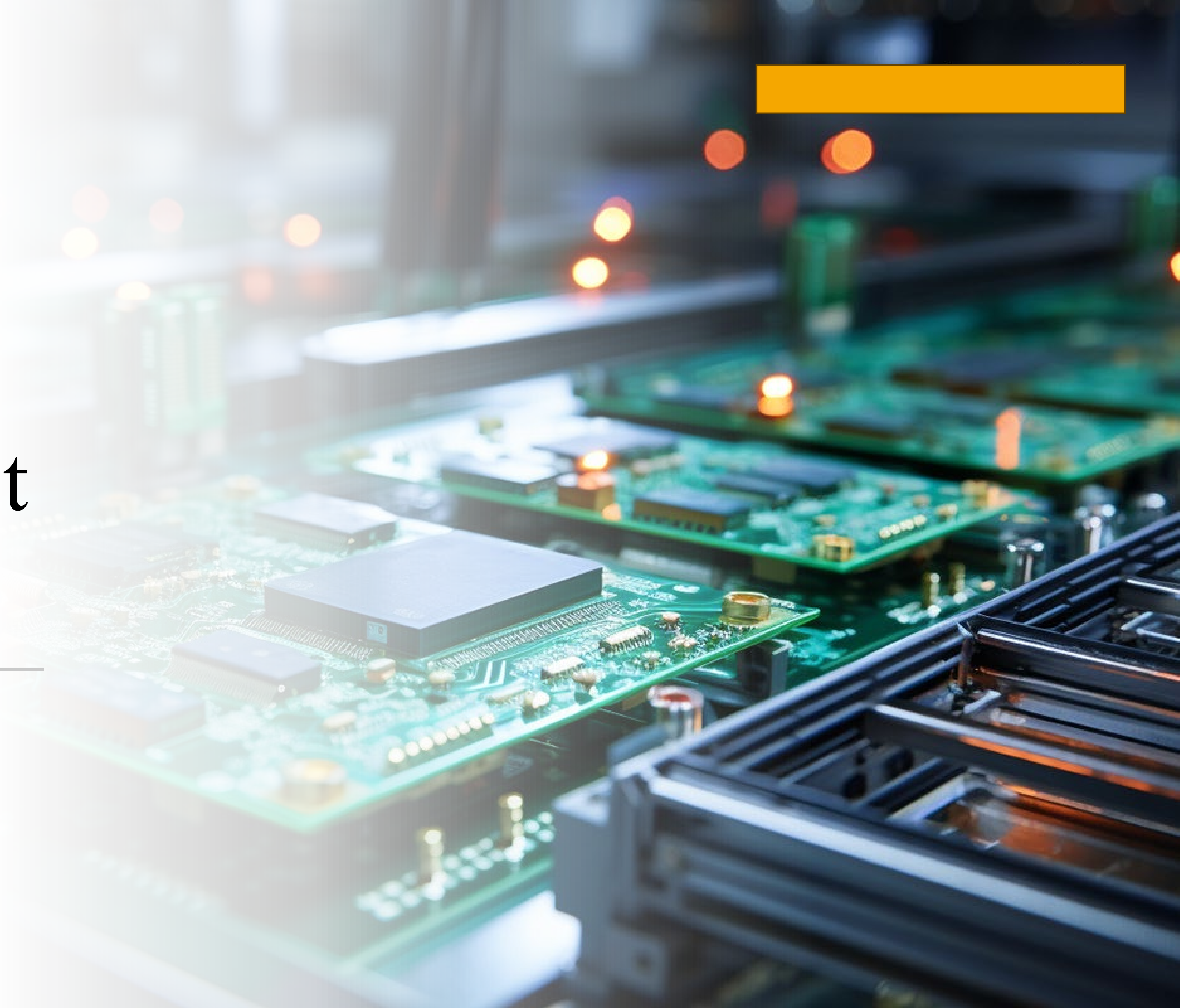
Unified Multi-Scenario Architecture

One model handles all management practices.
Learns tillage & irrigation effects directly.

Three-Phase Training Strategy

Gradual physics constraint introduction.
Prevents training instability.

Model Development & Training



The Foundation

Training on a Rich SWAT Simulation Dataset

1.56M+

Field -Year Records

~21,000

Agricultural fields in the South Platte River Basin

18

Years of data (2003-2020)

4

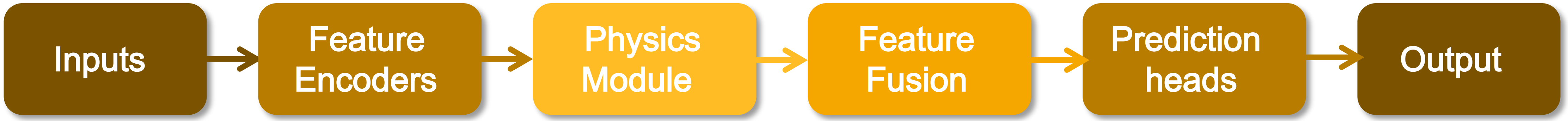
Scenario management combinations (Tillage/irrigation)

Validation Strategy

Spatial Cross-Validation, holding out entire fields to ensure the model generalizes to new locations

Model Architecture

A Model That Thinks Like a Hydrologist



Model Architecture

A Model That Thinks Like a Hydrologist

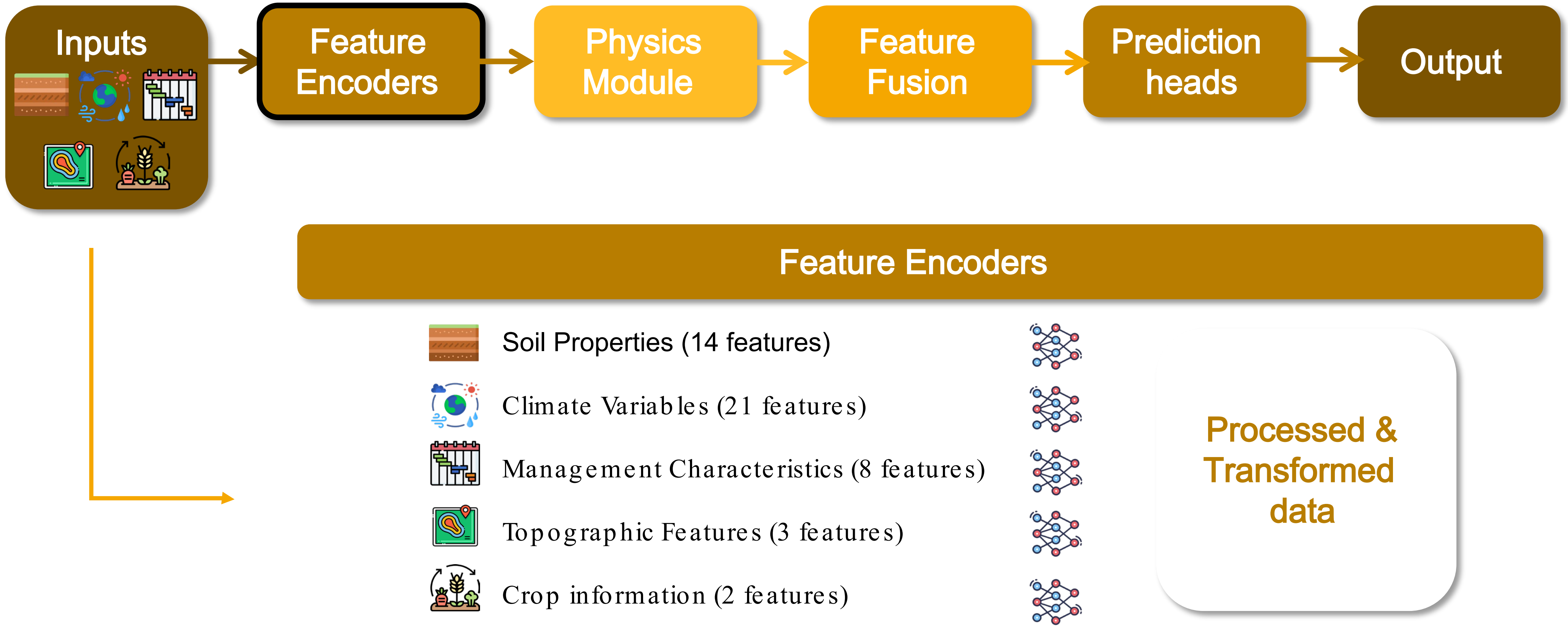
Inputs → Feature Encoders → Physics Module → Feature Fusion → Prediction heads → Output

Inputs

- Soil Properties (14 features)
- Climate Variables (21 features)
- Management Characteristics (8 features)
- Topographic Features (3 features)
- Crop information (2 features)

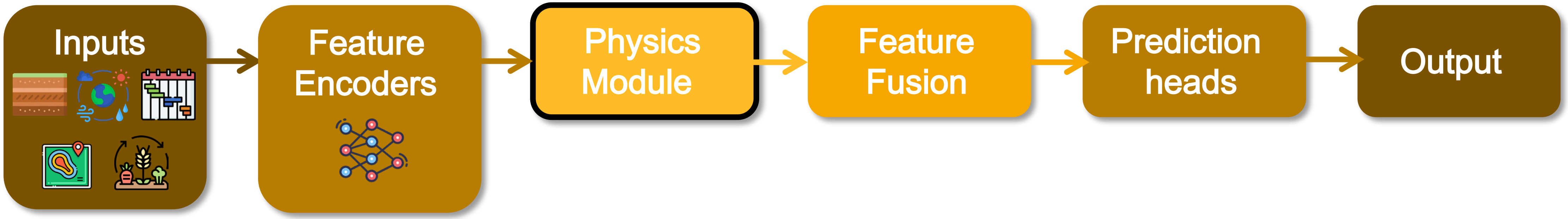
Model Architecture

A Model That Thinks Like a Hydrologist



Model Architecture

A Model That Thinks Like a Hydrologist



Physics Module

Runoff

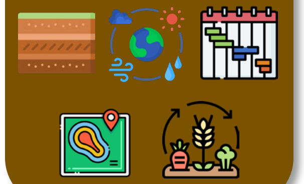
Erosion

Nutrient Cycling

Model Architecture

A Model That Thinks Like a Hydrologist

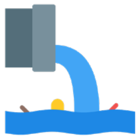
Inputs



→ Feature Encoders

→ **Physics Module** → Feature Fusion → Prediction heads → Output

Physics Module



Runoff

Precipitation

$$Runoff_{rain} = \frac{(P - I_a)^2}{P + 0.8 * S}$$

P is precipitation. Initial abstraction (Ia) & Retention parameter (S) are calculated based in the CN

Irrigation

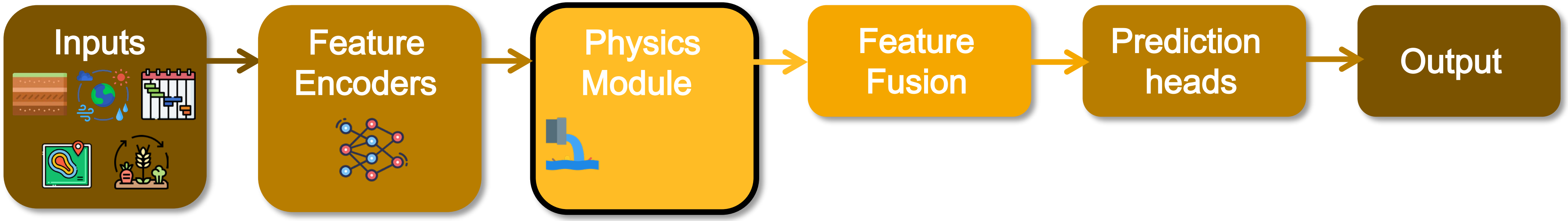
$$IrrigationNeed = CropWaterReq - PrecipGrowingSeason$$

$$Runoff_{irr} = IrrigationNeed\ * IrrigationRatio$$

Irrigation ratio is based on irrigation type (Flood/Sprinkler) and efficiency (Management data)

Model Architecture

A Model That Thinks Like a Hydrologist



Physics Module

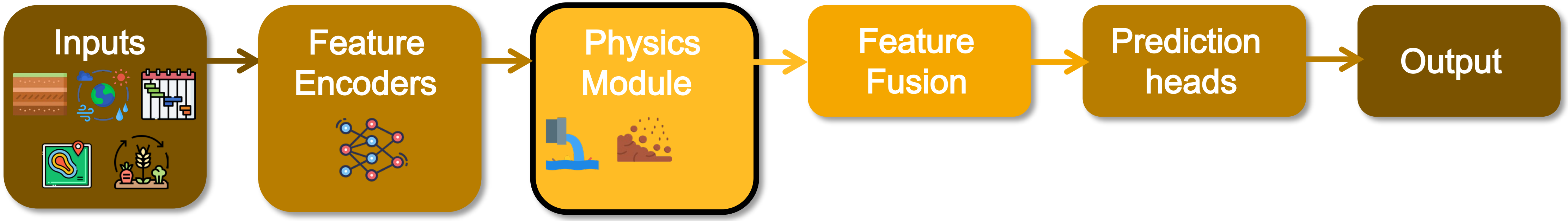
$$Sediment\ Yield = R * K * LS * C * P$$

Erosion

- R is the runoff erosivity factor (calculated from runoff volume and peak flow)
- K is the soil erodibility factor (learned from soil texture, organic matter, and structure)
- LS is the topographic factor (learned from slope steepness, as slope length is constant at 50m)
- C is the cover and management factor (learned from crop type, tillage system, and residue management)
- P is the support practice factor (constant at 1.0 in this dataset)

Model Architecture

A Model That Thinks Like a Hydrologist



Physics Module

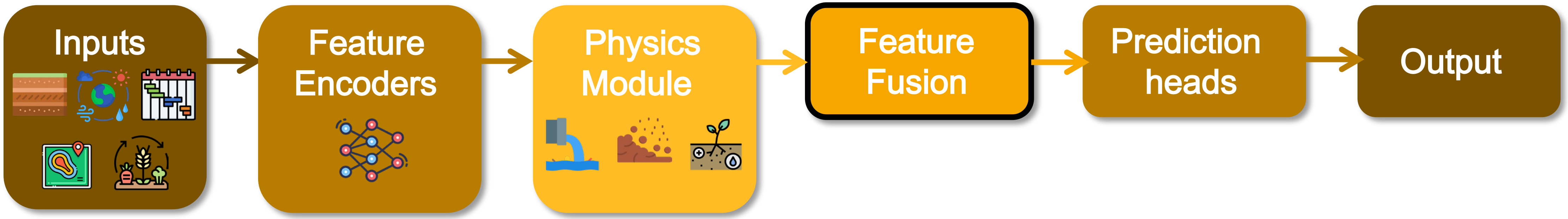
$$Available_N = Fertilizer_N + Residue_N + Mineralization_N$$

Nutrient Cycling

- Mineralization rates are temperature and moisture dependent, with higher rates under warm, moist conditions.
- The model learns that tillage enhances mineralization through increased soil aeration and residue incorporation.
- Maximum mineralization is constrained to 3% of soil organic nitrogen per year based on established literature values.

Model Architecture

A Model That Thinks Like a Hydrologist



Feature Fusion

Physics-Informed Loss Functions

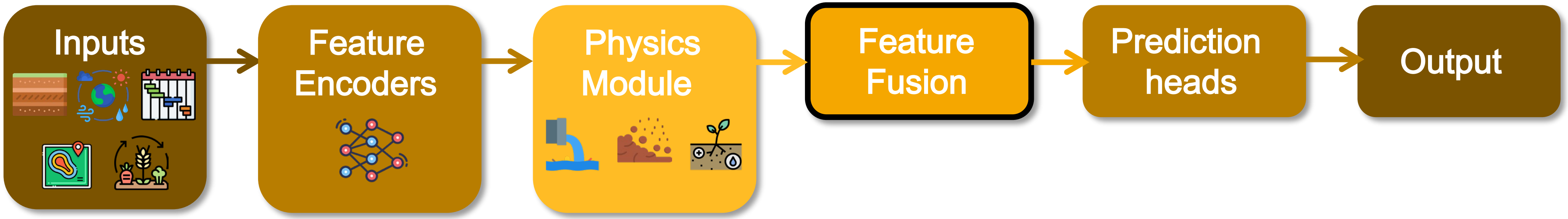
$$L_{total} = L_{prediction} + \lambda_{mass} * L_{mass} + \lambda_{process} * L_{process} + \lambda_{bounds} * L_{bounds}$$

Prediction Loss

Mean Square Error in predictions

Model Architecture

A Model That Thinks Like a Hydrologist



Feature Fusion

Physics-Informed Loss Functions

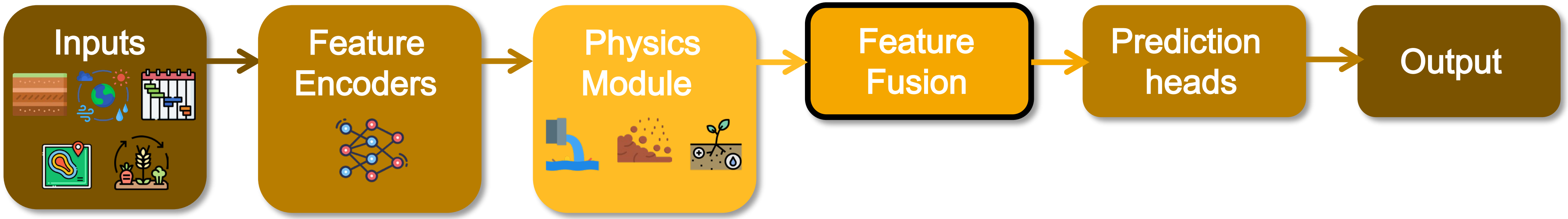
$$L_{total} = L_{prediction} + \lambda_{mass} * L_{mass} + \lambda_{process} * L_{process} + \lambda_{bounds} * L_{bounds}$$

Mass balance Constraints

$$Max_{TN} = Fertilizer_N + Residue_N + 0.03 * Soil_{OrgN}$$

Model Architecture

A Model That Thinks Like a Hydrologist



Feature Fusion

Physics-Informed Loss Functions

$$L_{total} = L_{prediction} + \lambda_{mass} * L_{mass} + \lambda_{process} * L_{process} + \lambda_{bounds} * L_{bounds}$$

Process Relationship Constraints

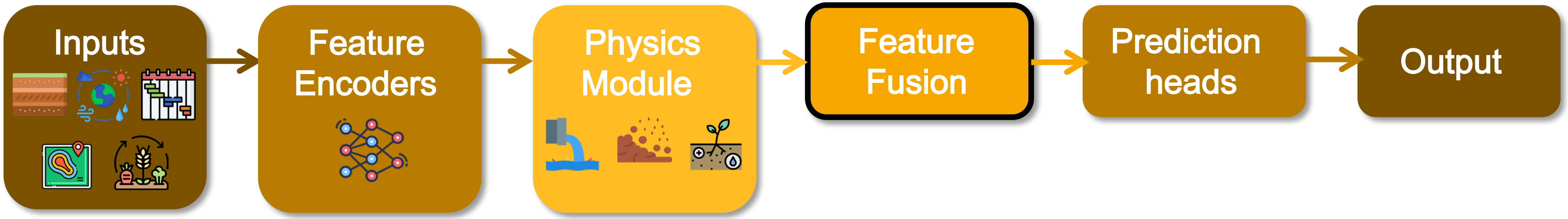
Concentration (TN/Runoff) within realistic range

Correlation (TN, Erosion) ≥ 0.9

Lower Erosion in reduced tillage, lower runoff in sprinkler irrigation

Model Architecture

A Model That Thinks Like a Hydrologist



Feature Fusion

Physics-Informed Loss Functions

$$L_{total} = L_{prediction} + \lambda_{mass} * L_{mass} + \lambda_{process} * L_{process} + \lambda_{bounds} * L_{bounds}$$

Physical Bounds Constraints

Total load per hectare is within realistic range

Model Architecture

A Model That Thinks Like a Hydrologist

Inputs → Feature Encoders → Physics Module → Feature Fusion → Prediction heads → Output

Feature Fusion

Three-Phase Training

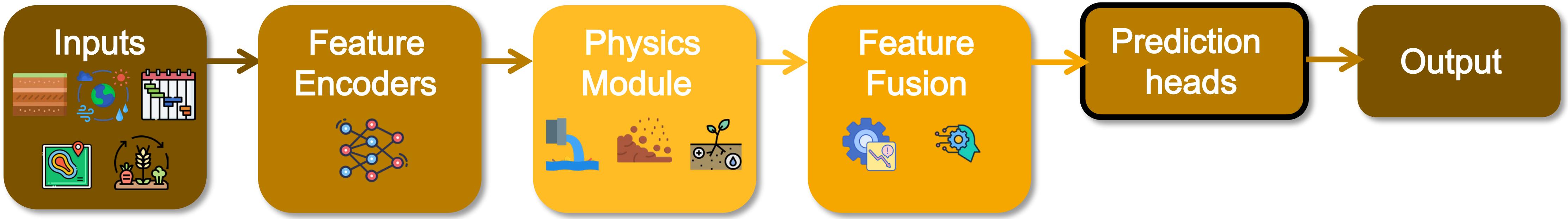
Phase 1
Pure Machine Learning (Epochs 1-20)

Phase 2
Gradual Physics Integration (Epochs 21-50)

Phase 3
Full Physics Constraints (Epochs 51-150)

Model Architecture

A Model That Thinks Like a Hydrologist

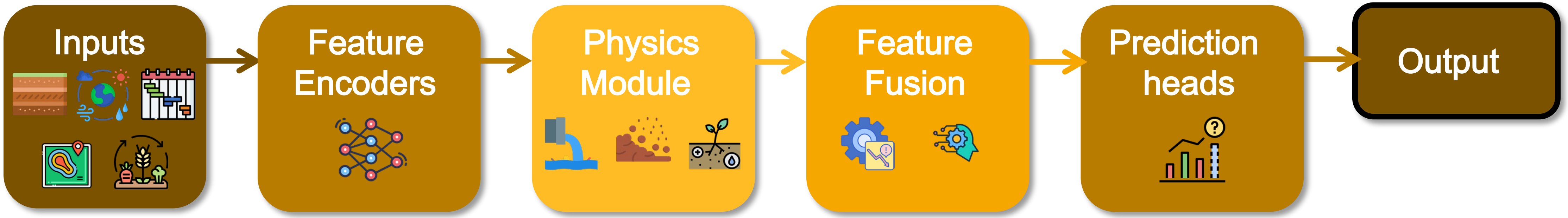


Prediction heads

Total Nitrogen Prediction
Total Phosphorus Prediction

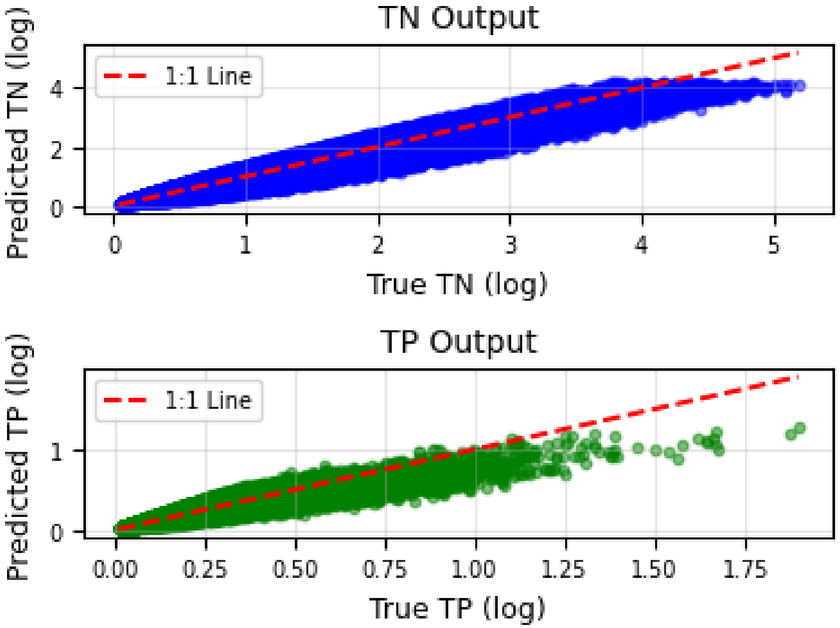
Model Architecture

A Model That Thinks Like a Hydrologist



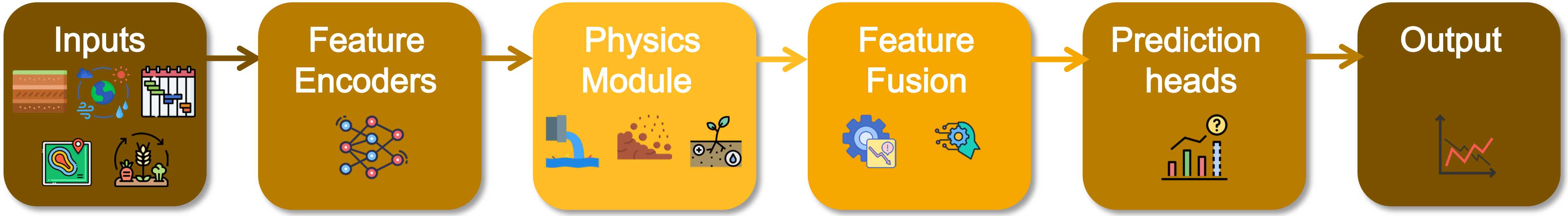
Output

Annual TN and TP load per field



Model Architecture

A Model That Thinks Like a Hydrologist

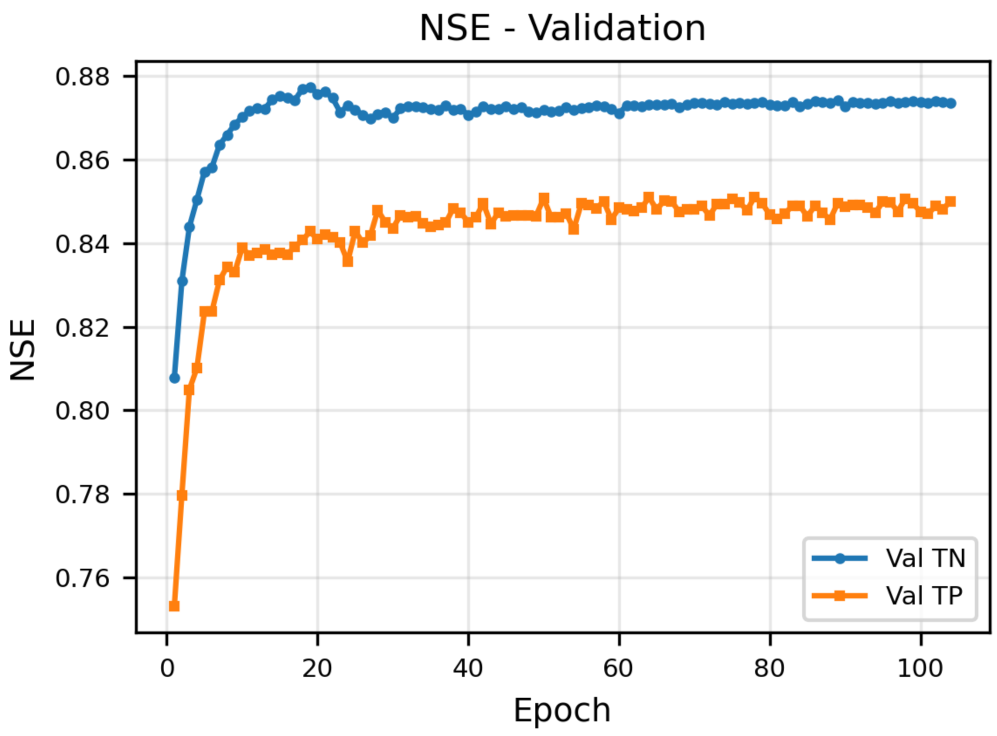


Model Performance

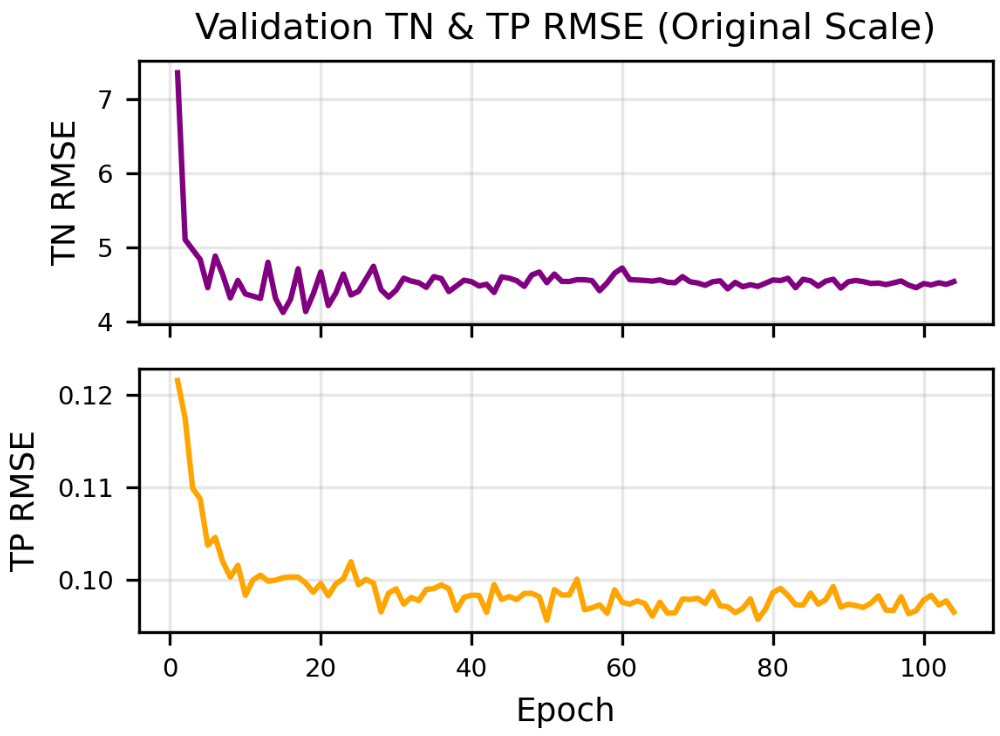
Performance at a Glance

Watching the Model Learn

Model validation shows high accuracy, with Nash-Sutcliffe Efficiency values stabilizing near 0.88 for TN and 0.85 for TP



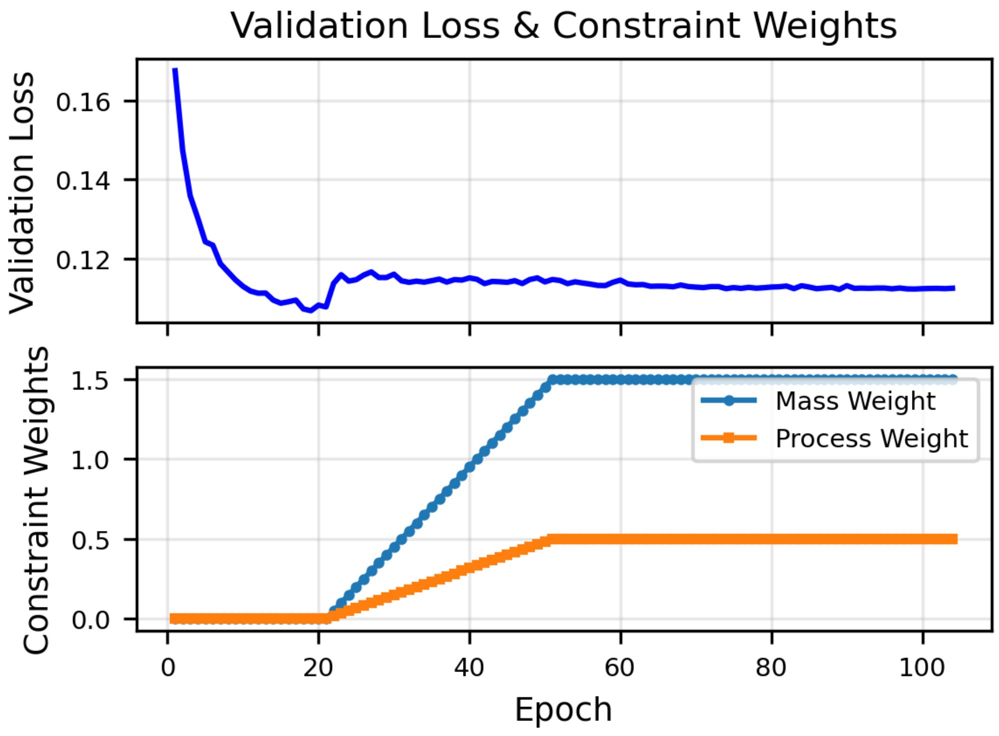
Model error (RMSE) rapidly minimizes and stabilizes by epoch 50, reaching ~4.5 for TN and ~0.1 for TP.



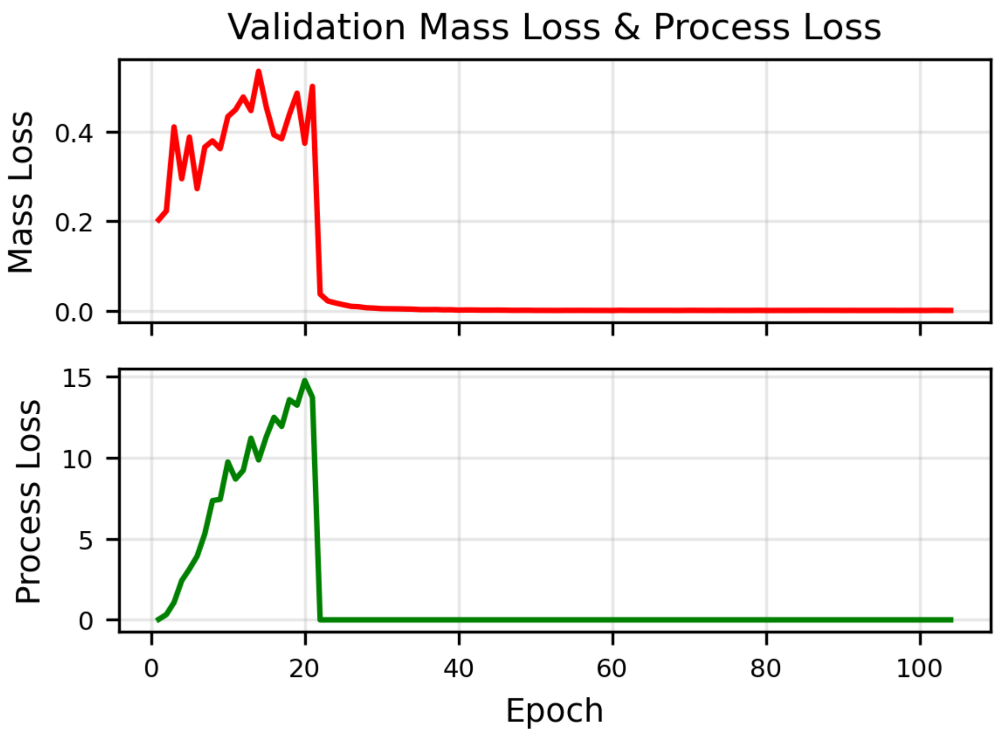
Performance at a Glance

Watching the Model Learn

The three-phase training, pure ML loss increase by epoch 20, ramping Mass Weight to 1.5 and Process Weight to 0.5 between epochs 20-50.



As physics constraints are applied at epoch 20, the model was initially violating physics, but it rapidly corrects after activating the constraints

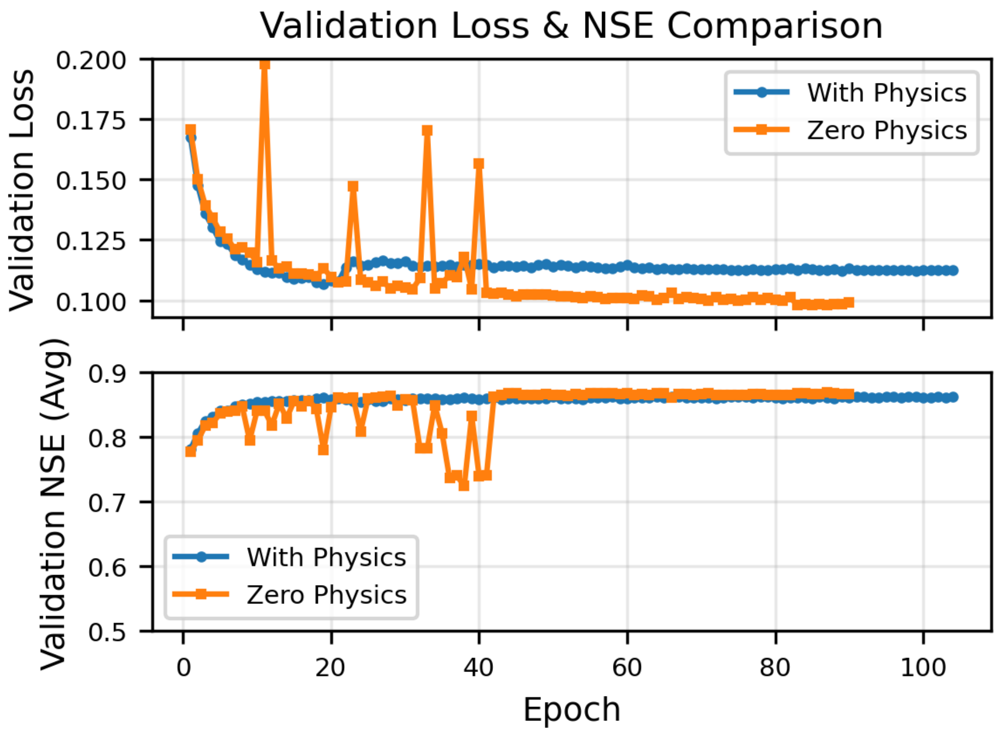


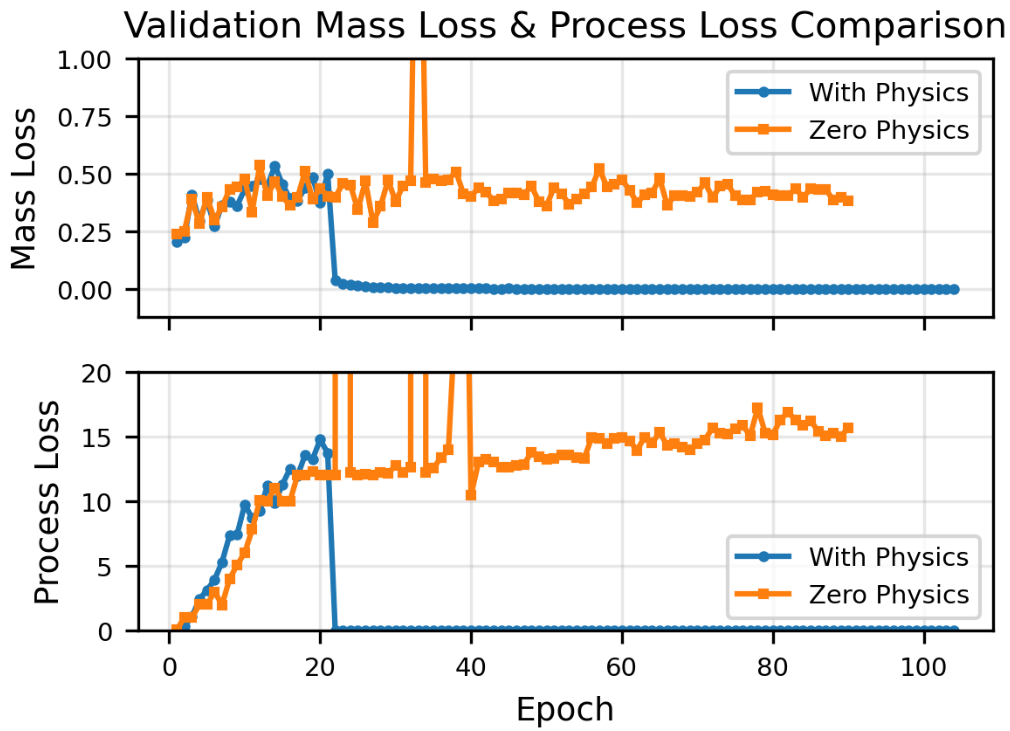
Performance at a Glance

Watching the Model Learn

Both models achieve a high avg. NSE of ~0.9, with the zero physics model has slightly lower validation loss

The "Zero Physics" model consistently violates physics (Mass Loss ~0.4, Process Loss > 15), while the "With Physics" model learns to force both physics losses to 0.





Breaking the Bottleneck: A Time & Reliability Comparison

From Days to Minutes: The End User Prospective

	Setup	Execution	Post-Process	Final Outcome
SWAT 1000 fields x4 scenarios	Collect data (web/user), create input files **1-2 Hours**	Run SWAT field-by-field for 20 years **20 Hours**	Aggregate results from all 4000 outputs **0.5 Hours**	Prone to errors from complex input files, requiring manual debugging. **Total Time: ~1 Day**
KGML 1000 fields x4 scenarios	Prepare a single input data table **5 Minutes**	Make 80,000 predictions (1000x4x20) **10 Seconds**	Output is already aggregated **0 Seconds**	Robust & stable; eliminates input file errors **Total Time: ~5 Minutes**

Conclusions

Key Takeaways

- KGML emulator solves the SWAT bottleneck, enabling rapid predictions for large-scale, web-based decision support.
- By incorporating physical laws, the model produces scientifically reliable and trustworthy results.
- This "glass box" method makes AI a robust and defensible tool for agricultural and environmental modeling.

Thank you

Mohamed Fawzy Mahmoud

M.Fawzy.Mahmoud@colostate.edu