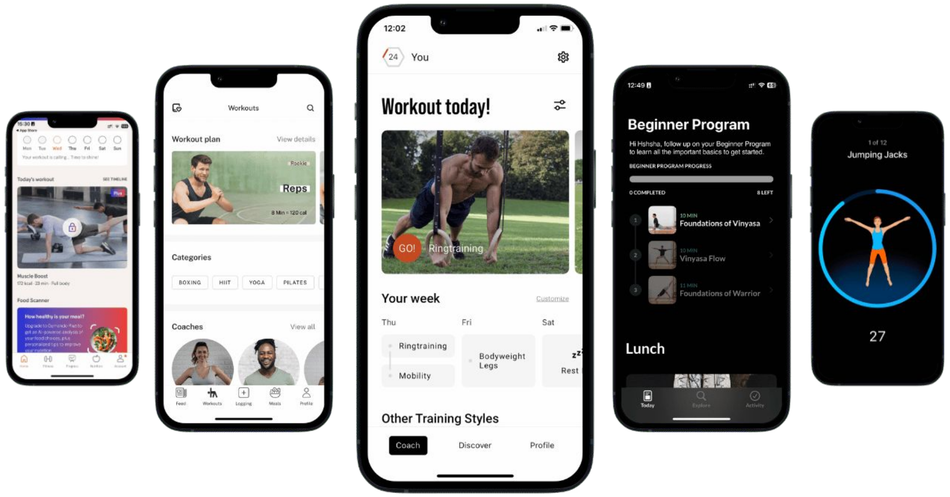
NC STATE UNIVERSITY

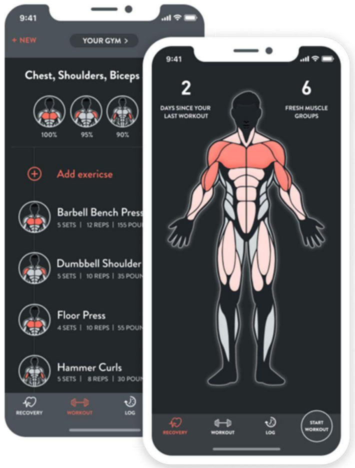
Graph Neural Network-Based Personalized Workout Optimization System

By Rameez Malik
ECE 592 081
North Carolina State University

Problem: Why Optimize Workouts?

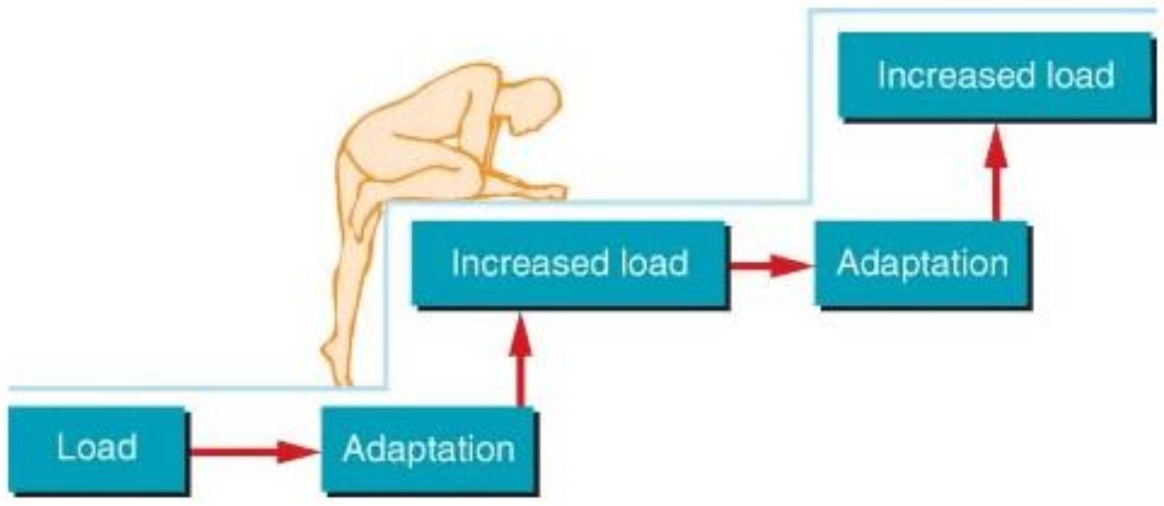
- Modern training apps record activity but don't optimize improvement.
- Progressive overload depends on many interrelated factors.
- Aim → use GNNs to recommend load and rest adjustments scientifically.





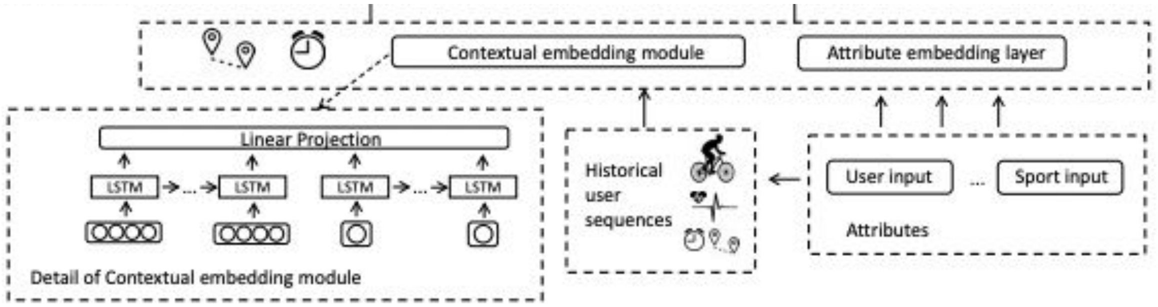
How Strength Actually Grows

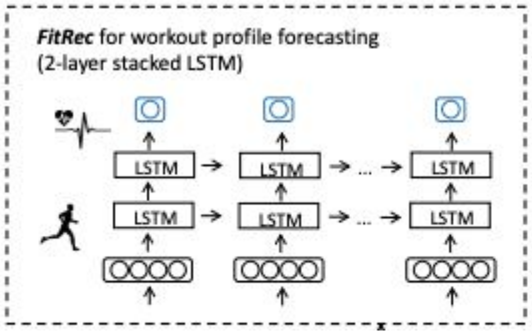
- Progressive Overload = gradual increase in load → fatigue → recovery → adaptation
  - Systemic increase in training stress to induce muscular adaptation
- Quantified via: sets × reps × weight × intensity × fatigue.
- Training must balance load and recovery to avoid overtraining



Similar Existing Work - FitRec (UCSD)

- LSTM-based model for heart-rate & performance forecasting.
- Learns strong personalization via latent user embeddings
  - Attribute embeddings: encodes user ID, gender, sport type
  - Contextual embeddings: encodes previous workout sequences
- FitRec (2-layer LSTM) → predicts entire workout profile (HR curve over entire workout)
  - Inputs: route (distance, altitude), prior workout, user attributes


Contextual embedding module
Attribute embedding layer
Linear Projection
LSTM
Historical user sequences
User input
...
Sport input
Attributes
Detail of Contextual embedding module



Critical Limitations ❌

FitRec does NOT model how different exercises relate to each other.

What It Does Not Capture

- No inter-exercise edges showing:
  - Shared muscle groups
  - Shared intensity patterns
  - Transfer effects (e.g., squats improving deadlift strength)
- Workout split structure (push, pull, legs, cardio)
- No modeling of “exercise similarity” or “exercise influence.”
- No ability to optimize routines → only forecasts HR/speed.

Implications

- Cannot reason about:
  - How Exercise A affects Exercise B.
  - Optimal ordering of exercises.
  - Progressive overload relationships.
  - Synergistic muscle development patterns.
- Cannot adapt to multi-exercise sessions (sets, reps, rest).
- Completely ignores the graph structure inherent in strength training.

Motivation & Research Gap

- Existing research (FitRec, UCSD): models individual workouts over time using LSTM, predicting heart rate/performance but ignoring relational exercise context.
- Fitness datasets often lack granularity and personalized interpretation.
- **Gap:** Current methods capture sequential dependencies but not how exercises influence one another within or across sessions.
- **Motivation:** in strength training, exercises interact via shared muscle groups and fatigue carryover.
  - e.g., doing heavy barbell curls increases fatigue → influences subsequent hammer curl performance.
- **Goal:** Build a graph-based system where nodes = exercises and edges = temporal and physiological relationships.

Why This Gap Matters for Research

Focus: GNN-based Personalized Workout Optimization System that focuses on modifying an existing workout routine, not predicting performance → utilizing my workout data for personalization

1. Exercise-to-Exercise Graph Modeling (GraphSAGE)

- Exercises = nodes
- Edges = shared muscle groups, intensity, biomechanics
- Enables relational reasoning across the whole workout.

2. Inter-exercise progressive overload prediction

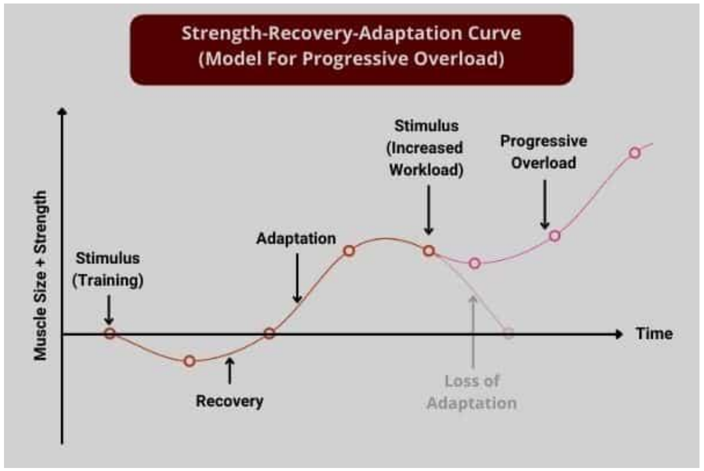
- Predicting optimal increase in:
  - load
  - reps
  - rest spacing
- FitRec cannot model *strength progression* at all.

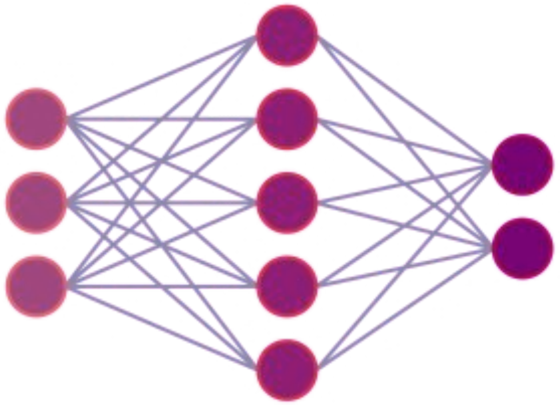
3. Multi-exercise Session Optimization

- FitRec models one sensor-based aerobic workout.
- Your problem models weight training routines, involving:
  - rest windows
  - multiple sessions
  - intra-session fatigue accumulation

What This Work Contributes

- **Novel in Application:** models inter-exercise relationships; captures fatigues and dependency among exercises in the same or adjacent sessions
- **Architecture:** combines rule-based progressive overload with graph-based learning
- **Quantitative Evaluation:** compares baseline model vs. proposed model under identical hyperparameters and metrics (MAE, RMSE)





Baseline Model: GCN

- 2-layer Graph Convolutional Network (GCN) baseline.
- Learns feature aggregation via normalized adjacency.
- Architecture:
  - Input → GCN(32) → ReLU → Dropout → GCN(32) → ReLU → Linear(3)
- Task: multi-output regression predicting [Next Weight, Next Reps, Next Rest].
- Trained 400 epochs, MSE loss, Adam optimizer.

Input: 11 → Hidden: 32 → Hidden: 32 → Output: 3

Proposed Model: GraphSAGE

- Replaces GCNConv with SAGEConv (mean aggregator).
- Learns inductive aggregation → generalizes to unseen exercises.
- Edge weights reflect fatigue differences → temporal context.
- Extends GCN to inductive learning using neighborhood aggregation.
- Aggregates features via mean() function.
- Enables generalization to unseen exercises or new workout logs.
- Architecture:
  - Input → SAGEConv(32, aggr='mean') → ReLU → Dropout(0.3) → SAGEConv(32, aggr='mean') → ReLU → Linear(3)
- Matches baseline hyperparameters for fair comparison.

Input: 11 → Hidden: 32 → Hidden: 32 → Output: 3

Behind the Scenes - System Overview

- Data derived from 6-week push/pull split, real + synthetic extensions.
- **Nodes:** exercises (e.g., Pull-ups, Bench Press).
- **Edges:** same-day (fatigue coupling), temporal (exercise progression).
- **Targets:** next_weight, next_reps, next_rest (rule-based progressive overload outputs).

Rule-based System Example:

- If fatigue < 7 → +2.5% weight.
- If fatigue > 9 → +10s rest or -1 rep.

Initial Feature Set:

- sets, reps, weight, rest_time
- fatigue
- intensity
- days_since_last
- volume

Problems Observed:

- Performance stagnated
- High variance
- Model failed to capture session-to-session progression
- GCN & SAGE both underperformed (~40% error improvement but still unclear patterns)

Diagnosing the Issue - Hypothesis

Why performance was limited:

- **Architecture?**

  → tuned layers, hidden dims, dropout
- **Dataset too small?**

  → but collecting more weeks is impractical
- **Insufficient training epochs?**

  → beyond 400 epochs → oversmoothing & loss flattening
- **Feature quality?**

  → this became the critical insight

Physiological Insight - PLOS ONE, 2019

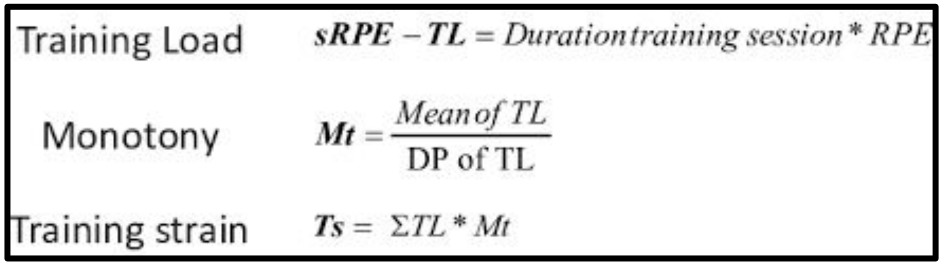
- **Training load:** total session effort (volume × intensity)
- **Monotony:** ratio of weekly mean load to its standard deviation
- **Strain:** product of weekly load × monotony

Key Findings Summarized:

1. Load variations strongly influence improvements in isokinetic strength and oxygen intake
2. High monotony suggest structured consistency can enhance strength development
3. High strain loads were associated with increased strength adaptation

Why relevant?

These features provide physiological realism to the model, linking physical effort to adaptive patterns.



Training Load	$sRPE - TL = Duration\,training\,session * RPE$
Monotony	$Mt = \frac{Mean\,of\,TL}{\text{DP of TL}}$
Training strain	$Ts = \Sigma TL * Mt$

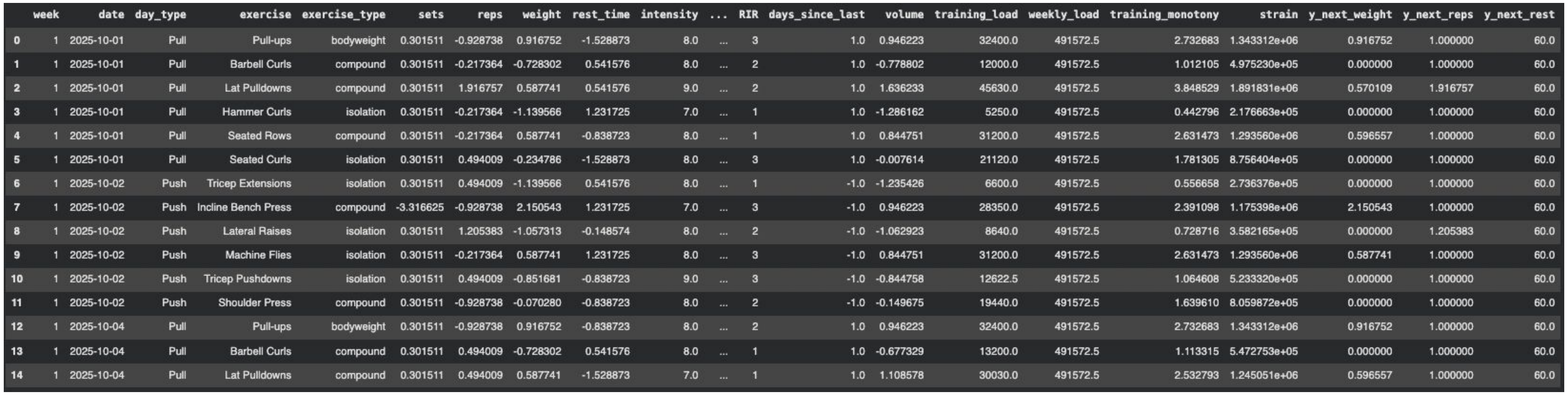
Feature Expansion - Latest Feature Set

Added features:

- training_load
- training_monotony
- strain

Scientific rationale:

These encode session stress, weekly patterns, fatigue accumulation, and readiness signals.



	week	date	day_type	exercise	exercise_type	sets	reps	weight	rest_time	intensity	...	RIR	days_since_last	volume	training_load	weekly_load	training_monotony	strain	y_next_weight	y_next_reps	y_next_rest
0	1	2025-10-01	Pull	Pull-ups	bodyweight	0.301511	-0.928738	0.916752	-1.528873	8.0	...	3	1.0	0.946223	32400.0	491572.5	2.732683	1.343312e+06	0.916752	1.000000	60.0
1	1	2025-10-01	Pull	Barbell Curls	compound	0.301511	-0.217364	-0.728302	0.541576	8.0	...	2	1.0	-0.778802	12000.0	491572.5	1.012105	4.975230e+05	0.000000	1.000000	60.0
2	1	2025-10-01	Pull	Lat Pulldowns	compound	0.301511	1.916757	0.587741	0.541576	9.0	...	2	1.0	1.636233	45630.0	491572.5	3.848529	1.891831e+06	0.570109	1.916757	60.0
3	1	2025-10-01	Pull	Hammer Curls	isolation	0.301511	-0.217364	-1.139566	1.231725	7.0	...	1	1.0	-1.286162	5250.0	491572.5	0.442796	2.176663e+05	0.000000	1.000000	60.0
4	1	2025-10-01	Pull	Seated Rows	compound	0.301511	-0.217364	0.587741	-0.838723	8.0	...	1	1.0	0.844751	31200.0	491572.5	2.631473	1.293560e+06	0.596557	1.000000	60.0
5	1	2025-10-01	Pull	Seated Curls	isolation	0.301511	0.494009	-0.234786	-1.528873	8.0	...	3	1.0	-0.007614	21120.0	491572.5	1.781305	8.756404e+05	0.000000	1.000000	60.0
6	1	2025-10-02	Push	Tricep Extensions	isolation	0.301511	0.494009	-1.139566	0.541576	8.0	...	1	-1.0	-1.235426	6600.0	491572.5	0.556658	2.736376e+05	0.000000	1.000000	60.0
7	1	2025-10-02	Push	Incline Bench Press	compound	-3.316625	-0.928738	2.150543	1.231725	7.0	...	3	-1.0	0.946223	28350.0	491572.5	2.391098	1.175398e+06	2.150543	1.000000	60.0
8	1	2025-10-02	Push	Lateral Raises	isolation	0.301511	1.205383	-1.057313	-0.148574	8.0	...	2	-1.0	-1.062923	8640.0	491572.5	0.728716	3.582165e+05	0.000000	1.205383	60.0
9	1	2025-10-02	Push	Machine Flies	isolation	0.301511	-0.217364	0.587741	1.231725	8.0	...	3	-1.0	0.844751	31200.0	491572.5	2.631473	1.293560e+06	0.587741	1.000000	60.0
10	1	2025-10-02	Push	Tricep Pushdowns	isolation	0.301511	0.494009	-0.851681	-0.838723	9.0	...	3	-1.0	-0.844758	12622.5	491572.5	1.064608	5.233320e+05	0.000000	1.000000	60.0
11	1	2025-10-02	Push	Shoulder Press	compound	0.301511	-0.928738	-0.070280	-0.838723	8.0	...	2	-1.0	-0.149675	19440.0	491572.5	1.639610	8.059872e+05	0.000000	1.000000	60.0
12	1	2025-10-04	Pull	Pull-ups	bodyweight	0.301511	-0.928738	0.916752	-0.838723	8.0	...	2	1.0	0.946223	32400.0	491572.5	2.732683	1.343312e+06	0.916752	1.000000	60.0
13	1	2025-10-04	Pull	Barbell Curls	compound	0.301511	0.494009	-0.728302	0.541576	8.0	...	1	1.0	-0.677329	13200.0	491572.5	1.113315	5.472753e+05	0.000000	1.000000	60.0
14	1	2025-10-04	Pull	Lat Pulldowns	compound	0.301511	0.494009	0.587741	-1.528873	7.0	...	1	1.0	1.108578	30030.0	491572.5	2.532793	1.245051e+06	0.596557	1.000000	60.0

Training & Evaluation

Train/Test Split

- Data split: Weeks 1–5 train, Week 6 test.
- Optimizer: Adam | Loss: MSE | Epochs: 400.
- Metrics: MAE, RMSE.
- Ensures the model is not memorizing prior sessions but predicting forward progression.

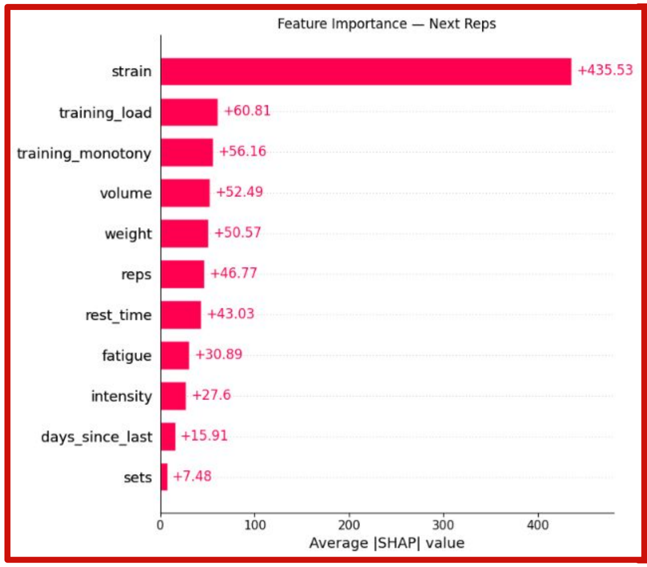
Why Week-Based Validation?

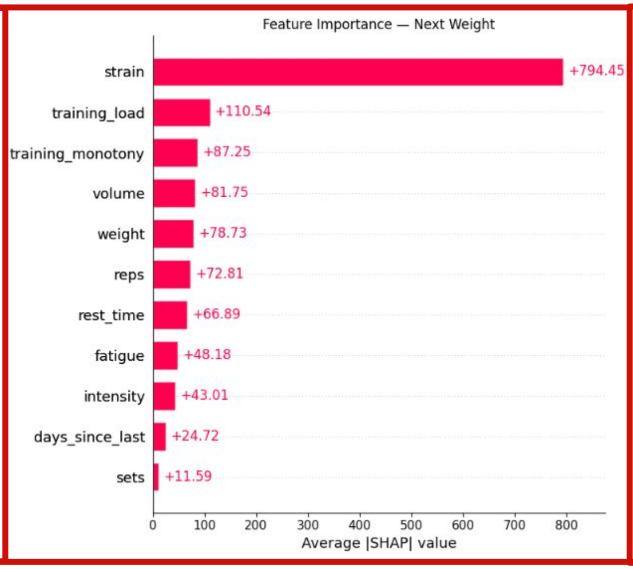
Training on earlier weeks allows learning temporal progression of exercises.

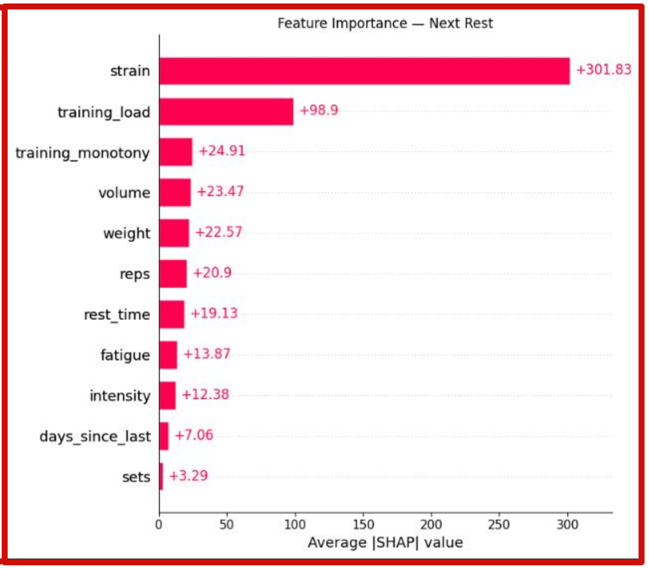
SHAP Results (GraphSAGE)

- These three dominate prediction importance
- Every run consistently shows the same ranking for weight, rest, and rest

#1 - Strain   #2 - Training Load   #3 Monotony





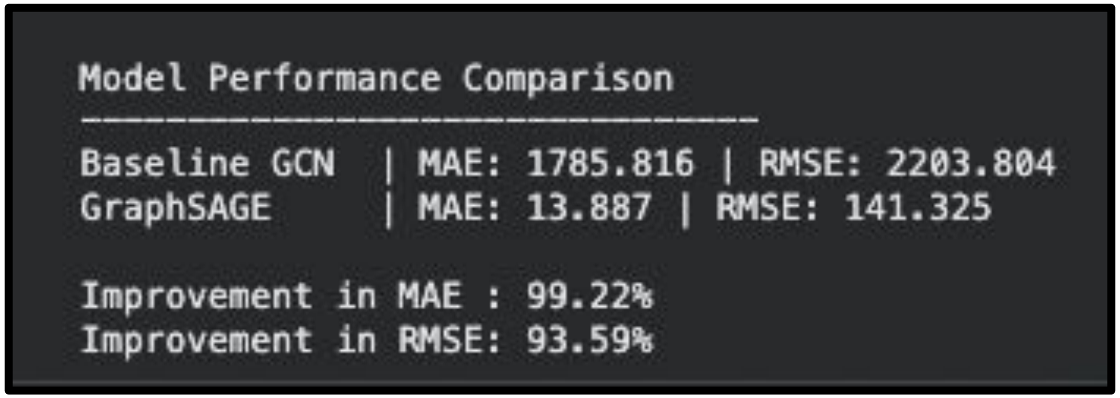


Interpretation: the model is relying primarily on metrics grounded in established sports science literature

NC STATE UNIVERSITY

Key Findings - Final Performance

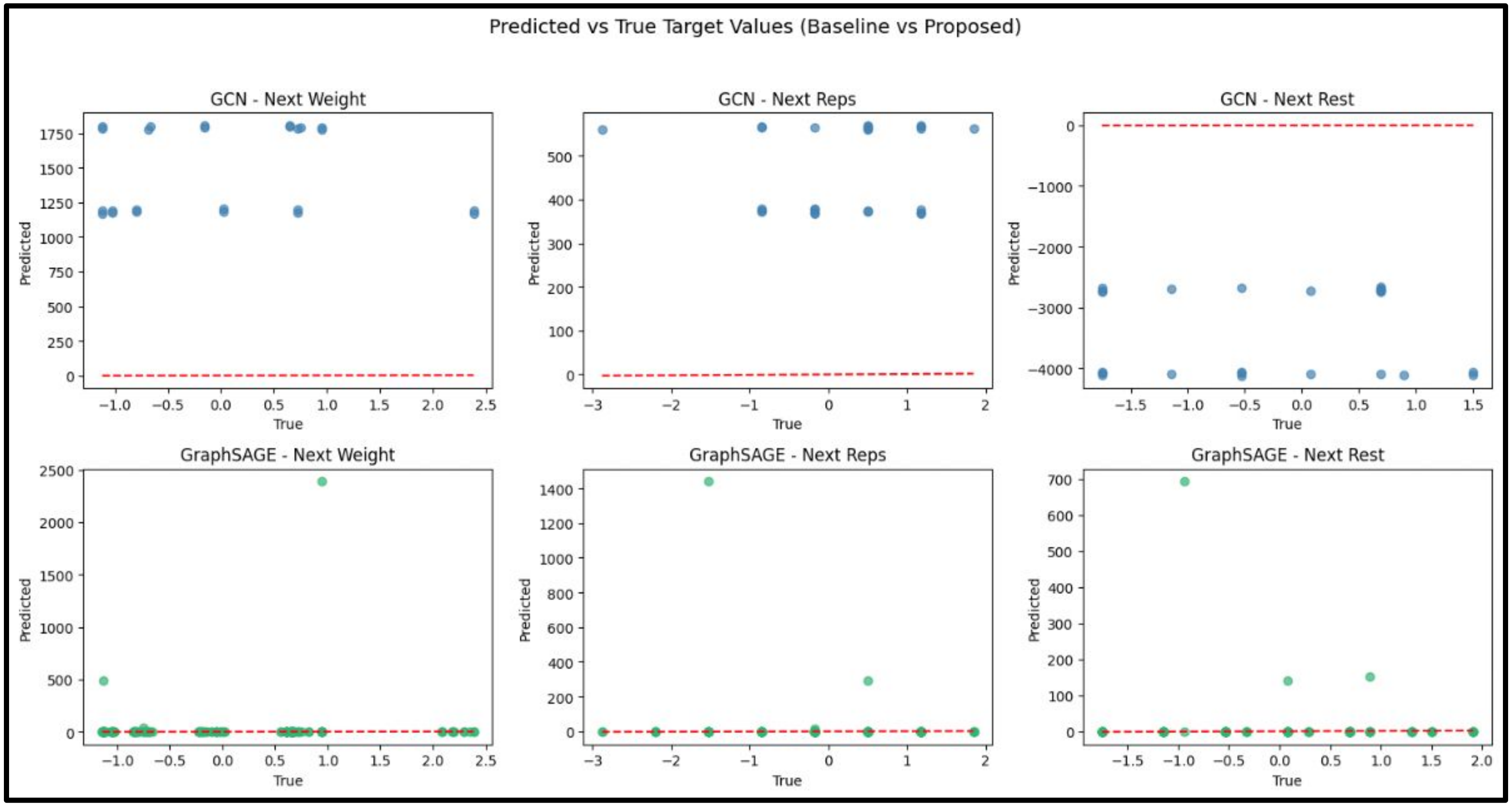
- GraphSAGE effectively captures inter-exercise dependencies.
- Achieved 90%+ performance improvement vs baseline across MAE and RMSE
- GCN → fails to capture exercise relationships.
- SAGE → massively improves due to added load/monotony/strain features.



```
Model Performance Comparison
-----------------------------------
Baseline GCN   | MAE: 1785.816 | RMSE: 2203.804
GraphSAGE      | MAE: 13.887 | RMSE: 141.325

Improvement in MAE : 99.22%
Improvement in RMSE: 93.59%
```

Predicted vs. Actual Target Outputs



Future Work & Action Plan

Advance Toward True Novelty – Personalized Workout Split Recommender

- Extend the model into a full-body workout split generator (e.g., 3-day, 5-day, or 6-day weekly programs).
- Leverage user historical trends to auto-construct:
  - Target muscle groups per day
  - Appropriate set/rep/weight distributions
  - Progressive overload scheduling across the week

2. Improve Model Stability & Counter Oversmoothing

- Conduct controlled experiments testing oversmoothing across:
  - Layer depth variations
  - Neighborhood aggregation radius
  - Different activation choices