INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 6 - V11I6-1159)
Available online at: https://www.ijariit.com

Interpreting Opaque Scheduling Heuristics in Timetabling via Surrogate GNN

Kashish Ukey
kashishukey02@gmail.com
GH Raisoni College of Engineering and Management, Nagpur, Maharashtra

Nakul Badwaik
nakul.badwaik.ai@ghrietn.raisoni.net
GH Raisoni College of Engineering and Management, Nagpur, Maharashtra

Labdhi Soni
labdhi.soni.ai@ghrietn.raisoni.net
GH Raisoni College of Engineering and Management, Nagpur, Maharashtra

Ojas Kamde
ojas.kamde.ai@ghrietn.raisoni.net
GH Raisoni College of Engineering and Management, Nagpur, Maharashtra

Smita Nirkhi
hodaiengai@ghrietn.raisoni.net
GH Raisoni College of Engineering and Management, Nagpur, Maharashtra

ABSTRACT

Opaque timetabling heuristics often deliver strong schedules yet remain difficult to interpret, limiting trust, diagnosis, and controlled adaptation in educational scenarios. This work presents a practical pipeline that learns a constraint-aware graph neural network (GNN) surrogate from input-output pairs of a black-box timetabling solver and then applies global explainability to characterize the surrogate's decision-making logic across entities and relations in the context of timetabling. Using a synthetic dataset with hard constraints, the study evaluates both the assignment fidelity to the solver outputs and the feasibility under hard constraints, complementing these with global explanations and counterfactual sensitivity analysis. The results highlight which entities and relations most influence the predicted assignments. The pipeline is intended as a methods-oriented contribution that standardizes data generation, surrogate training, and explanation estimators for timetabling, enabling reproducible assessment of interpretability alongside fidelity and feasibility without claiming domain deployment readiness.

Keywords: *Scheduling Algorithms, Surrogate Models, Explainable Artificial Intelligence, Black-Box Optimization, Model Interpretability, Decision Support Systems.*

1. INTRODUCTION

Timetables appear deceptively simple when they work well: every class gets matched with the correct room and instructor at precisely the right moment. Yet beneath this apparent simplicity lies a complex web of constraints and preferences that most institutions navigate using hand-crafted heuristics and metaheuristics. While these approaches consistently deliver strong schedules in practice, they operate as black boxes that rarely explain why one particular assignment was chosen over another [8], [10], [11]. This opacity creates real problems when administrators need to build trust with stakeholders, diagnose edge cases where the system breaks down, or adapt the underlying logic when institutional policies evolve—challenges that become even more acute under shifting priorities and real-world constraints [8], [10], [11].

Explainable AI (XAI) techniques combined with surrogate modeling offer what we believe is a pragmatic middle ground: we can learn a transparent stand-in model for the opaque solver and use it to reason systematically about patterns in the decision-making process, all without disrupting the production pipeline that institutions depend on [10], [3]. However, interpretability only provides genuine value if the surrogate model faithfully captures the original solver's behavior and proves resilient to attempts at manipulation. Otherwise, the explanations we generate risk misleading stakeholders rather than enlightening them [1], [2]. Surrogate models have already demonstrated their utility across diverse operations research and scheduling domains—from quay crane scheduling and flow shop optimization to fog computing task allocation and power system management—yet focused, problem-specific studies addressing educational scheduling contexts remain surprisingly scarce in the literature [5], [6], [7], [9].

In this study, we train a constraint-aware graph neural network (GNN) that serves as a surrogate for the original solver, learning from input–output pairs that the solver generates.

We then apply global explanation techniques and counterfactual sensitivity tests to systematically map which entities (classes, teachers, subjects), which features, and which relational structures drive the predicted assignments [10], [4]. Our primary goal isn't to replace the existing solver that institutions use in production, but rather to interpret its black-box behavior: we want to surface system-level signals that reveal what matters most in decision-making, what factors could flip a scheduling decision, and where the model exhibits brittleness or instability—all while preserving the practical performance characteristics that make the original solver valuable [10], [4].

Our evaluation framework centers on two questions that practicing schedulers and administrators genuinely care about: how closely does the surrogate match the assignments that the solver produces (what we call fidelity), and how frequently do the surrogate's recommendations remain feasible when checked against the hard constraints that govern real schedules [1], [4]. We've deliberately standardized the entire pipeline from end to end—covering data generation, surrogate training procedures, and explanation methodology—so that other researchers can reproduce our results, compare them fairly with alternative approaches, and trace observed performance gains back to specific design choices [1], [4]. In summary, this work contributes a reusable methodological pipeline, a joint perspective on fidelity, feasibility, and counterfactual sensitivity, and empirically grounded insights about which features and interactions prove most influential in the decision rules that the surrogate learns [1], [4], [10].

2. LITERAYURE SURVEY

This section reviews recent advances in surrogate modeling for opaque AI heuristics and complex scheduling problems, with particular emphasis on three interconnected themes that directly inform our approach: first, how researchers evaluate both the fidelity and robustness of surrogate models; second, how surrogates integrate into real-world scheduling workflows; and third, how explainability mechanisms support trust in practical deployment scenarios.

A. Surrogate Model Evaluation

Tang and colleagues introduced ShapGAP, a metric that deliberately looks beyond superficial prediction similarity to assess whether a surrogate truly mirrors the underlying decision logic of its source model rather than merely approximating its outputs [1]. Their work highlighted an important distinction: matching predictions on a test set doesn't necessarily mean the surrogate has learned the same decision boundaries or reasoning process. Meanwhile, Slack et al. demonstrated that surrogate explanations can be actively gamed through adversarial manipulation, which underscores the critical need for fidelity tests that remain robust under such attacks and for evaluation protocols that probe deeper than surface-level agreement between model outputs [2]. Together, these studies have shaped how the community thinks about surrogate quality, pushing us toward more rigorous validation procedures.

B. Surrogate Models in Scheduling

Surrogates have proven valuable across heterogeneous and resource-constrained scheduling environments, often improving both computational efficiency and solution quality. Chen and Lin developed adaptive neural hyperheuristics specifically for heterogeneous computational scheduling problems [3], while Aminian et al. addressed the joint optimization challenge of balancing explainability with computational efficiency in AI-driven edge services [4]. Domain-specific adaptations have emerged across various sectors: Niu and colleagues built a surrogate model tailored to quay crane scheduling problems [5], Xia et al. enhanced flowshop optimization by incorporating maintenance schedules, learning effects, and deteriorating equipment conditions through surrogate- aided approaches [6], Kaur and Mahajan created deep surrogate models that achieve scalable task scheduling in fog computing environments [7], Sundararajan and Nagesh applied local-search heuristics to advertisement scheduling [8], and Schuett and Braun conducted comparative evaluations of surrogate performance in low-voltage electrical grids [9]. Collectively, these works suggest that surrogates can meaningfully accelerate search procedures, stabilize performance across problem instances, and expose important trade-offs in system design—all while respecting real-world constraints.

C. Explainability and Trust in AI

Surrogate models serve as a practical bridge to interpretability in production systems: they enable post-hoc insight into complex decision pipelines without requiring disruptive modifications to operational solvers, provided that the explanations they generate remain both faithful to the original system and genuinely useful to human decision-makers [10]. In the context of our work, this perspective means pairing traditional performance metrics with diagnostic analyses—identifying which factors matter most for scheduling decisions, pinpointing where assignments could flip under small perturbations, and understanding how explanations behave under controlled stress tests—so that interpretability complements rather than replaces the operational reliability that institutions require [1], [2], [10]. This balanced approach acknowledges that transparency without accuracy provides little value, while accuracy without transparency limits trust and adaptability.

3. METHODOLOGY

Our methodology follows a structured pipeline that transforms scheduling problems into graph representations, trains neural surrogates to mimic solver behavior, and extracts interpretable explanations from the learned models. Figure 1 illustrates the complete workflow, showing how data flows from initial problem instances through graph construction, model training, constraint-aware decoding, and finally to explanation generation.

A. Problem Formulation

We consider a school scheduling problem that involves coordinating five key entity types: classes C, subjects S, teachers T, days of the week D, and time slots within each day P.

Each class–subject pair (c, s) requires a specific number of weekly sessions, which we denote by $r_{c,s} \in \mathbf{Z}_{\geq 0}$. Additionally, not every teacher possesses the qualifications to teach every subject, so we capture this constraint using binary qualification indicators $q_{t,s} \in \{0, 1\}$, where $q_{t,s} = 1$ signifies that teacher t is qualified to teach subject s.

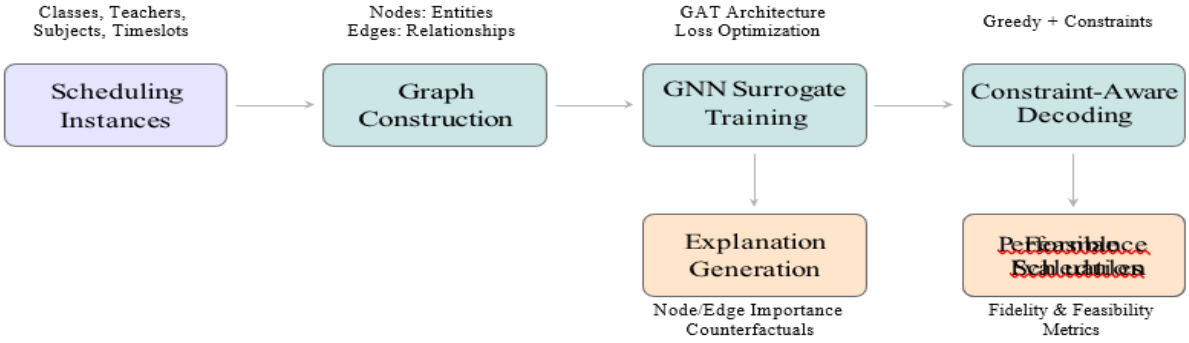

Fig. 1: Complete pipeline workflow from problem instances to interpretable outputs. The pipeline transforms raw scheduling data through graph construction and neural training to produce both feasible schedules and explanations of the decision-making process.

The central decision variable in our formulation is the binary indicator: $x_{c,s,t,d,p} \in \{0,1\}$ which takes the value 1 if and only if teacher t teaches subject s to class c on day d during time slot p, and 0 otherwise.

To ensure that generated schedules remain valid and practically implementable, we impose a set of hard constraints that must be satisfied:

- **(HC1) Session Requirements:** Each class–subject pair must receive exactly its required number of sessions per week:
$$\sum_{t \in T} \sum_{d \in D} \sum_{p \in P} x_{c,s,t,d,p} = r_{c,s}, \quad \forall c \in C, s \in S$$
- **(HC2) Teacher Qualification:** A subject can only be assigned to a teacher who is qualified to teach it:
$$x_{c,s,t,d,p} \leq q_{t,s}, \quad \forall c \in C, s \in S, t \in T, d \in D, p \in P$$
- **(HC3) Class Availability:** No class can be assigned to more than one subject simultaneously:
$$\sum_{s \in S} \sum_{t \in T} x_{c,s,t,d,p} \leq 1, \quad \forall c \in C, d \in D, p \in P$$
- **(HC4) Teacher Availability:** Each teacher can teach at most one class at any given time:
$$\sum_{c \in C} \sum_{s \in S} x_{c,s,t,d,p} \leq 1, \quad \forall t \in T, d \in D, p \in P$$

These constraints collectively ensure basic feasibility: the correct number of sessions are delivered for each class-subject combination, only appropriately qualified teachers receive assignments, and neither classes nor teachers face scheduling conflicts through double-booking.

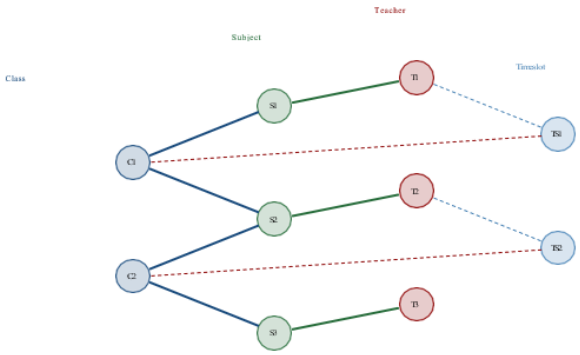

Fig. 2: Graph structure representation showing entities as nodes and relationships as edges. Solid lines represent curriculum (Class-Subject) and qualification (Subject- Teacher) constraints, while dashed lines capture availability patterns.

B. Graph Construction

To enable neural models to process scheduling instances effectively, we convert each problem instance into a structured graph representation as visualized in Fig. 2. Nodes in this graph correspond to the fundamental entities in our scheduling problem: classes, subjects, teachers, and individual timeslots. Each node carries a 16-dimensional feature vector that combines a one-hot encoding indicating its entity type with relevant metadata tailored to that type. For instance, class nodes include information about the total number of sessions that class requires, subject nodes carry popularity metrics reflecting how many classes need that subject, teacher nodes encode the breadth of subjects each teacher is qualified to teach, and timeslot nodes simply record their position in the weekly schedule through day and slot indices.

We establish edges in the graph to capture the important relationships and dependencies between entities:

- *Class–Subject edges:* represent curriculum requirements, indicating which subjects each class must study
- *Teacher–Subject edges:* encode qualification relationships, showing which teachers can teach which subjects
- *Class–Timeslot edges:* and *Teacher–Timeslot edges:* capture availability patterns across the weekly schedule

When we have access to ground-truth schedules produced by the original solver, we extract target labels from these solutions that specify exactly which (c, s, t, d, p) tuples were assigned. These labels serve as supervision signals during the training phase, allowing the surrogate to learn from the solver's demonstrated preferences and decision patterns.

C. Surrogate Model and Training Procedure

We train a graph-based surrogate model built on Graph Attention Network (GAT) layers that outputs continuous score for each possible scheduling assignment.

$$\hat{p}_{c,s,t,d,p} \in [0,1]$$

The GAT architecture proves particularly well-suited for this task because it can learn which graph structures (entity relationships) matter most for scheduling decisions through its attention mechanism. The model processes node features through multiple GAT layers, with each layer allowing nodes to aggregate information from their neighbors weighted by learned attention coefficients. These scores are designed to reflect the relative likelihood or desirability of making a particular assignment decision, based on patterns that the model extracts from historical solver outputs.

The training process follows a supervised learning paradigm driven by a composite loss function:

$$L = L_{\text{pred}} + \lambda L_{\text{reg}}$$

where L_{pred} measures prediction error (typically implemented through cross-entropy losses that compare predicted assignment probabilities against ground-truth solver decisions), and L_{reg} serves as a regularization term that encourages desirable properties like smoothness or sparsity in the learned representations. The hyperparameter λ controls the trade-off between fitting the training data and maintaining good generalization properties.

We employ standard best practices throughout the training procedure, including early stopping to prevent overfitting (monitoring validation loss and halting when it stops improving), gradient clipping to stabilize optimization (particularly important for graph neural networks which can experience exploding gradients), and check-pointing to preserve the best-performing model configurations. The dataset is split into 80% for training and 20% for validation, with the validation set used to monitor generalization performance and guide hyperparameter selection. Training continues for 50 epochs with a batch size carefully chosen to balance memory constraints and gradient estimate quality.

D. Constraint-Aware Decoding

Once the trained surrogate model produces its assignment scores $\hat{p}_{c,s,t,d,p}$ for all possible scheduling decisions, we face the practical challenge of converting these continuous scores into a concrete, feasible schedule. This conversion happens through a decoding step that selects a high-scoring subset of assignments while strictly respecting the hard constraints (HC1–HC4) that define schedule feasibility.

Formally, the decoding process solves an optimization problem that maximizes the total assignment score:

$$\sum_{c \in C} \sum_{s \in S} \sum_{t \in T} \sum_{d \in D} \sum_{p \in P}$$

subject to all feasibility constraints outlined in Section III- A.

Rather than attempting to solve this combinatorial optimization problem exactly—which would impose prohibitive computational costs for realistic problem sizes—we implement a greedy algorithm that constructs the schedule incrementally. At each step, the algorithm selects the highest-scoring feasible assignment from among the remaining options. We maintain simple bookkeeping structures that track class and teacher availability at each timeslot and filter out assignments that would violate qualification requirements. This greedy approach strikes a practical balance between runtime efficiency and solution quality, performing well even when integrated into existing solver pipelines where computational budgets are tightly constrained.

E. Evaluation Metrics

We evaluate the performance of our surrogate model and decoding procedure using a comprehensive suite of metrics that assess similarity to solver outputs, constraint satisfaction, and practical feasibility:

- **Schedule Similarity (Sim):** Measures how closely the predicted schedule matches the ground-truth schedule produced by the solver:

$$\text{Sim} = \frac{\text{Number of Matching Assignments}}{\text{Total Number of Assignments}}$$

- **Constraint Satisfaction Rate (CSR):** The fraction of all constraint checks (aggregated across all instances and all constraint types) that were successfully satisfied.
- **Session Coverage Accuracy (SCA):** Measures how accurately the model assigns the correct number of sessions for each class–subject pair, relative to the requirements $r_{c,s}$.
- **Conflict Rate (CR):** Fraction of timeslots across all schedules where either a class or a teacher was incorrectly double-booked, violating constraints HC3 or HC4.
- **Qualification Consistency (QC):** Proportion of all assignments where the assigned teacher was actually qualified to teach the assigned subject, measuring compliance with constraint HC2.
- **Schedule Feasibility (SF):** A binary indicator for each instance that takes value 1 if and only if the complete schedule satisfies all hard constraints (HC1–HC4) simultaneously.

Beyond reporting aggregate statistics like means and standard deviations across all test instances, we also provide detailed per-instance breakdowns and failure counts categorized by constraint type. This granular reporting offers a transparent and comprehensive view of performance, helping identify systematic weaknesses in the surrogate or decoder that might be masked by aggregate metrics alone.

Time	Day 0	Day 1	Day 2	Day 3	Day 4
Slot 0	S3T3	S5T3	S5T3	S1T5	—
Slot 1	S5T3	S4T3	—	S1T2	—
Slot 2	S4T6	S3T3	—	S3T3	S6T4
Slot 3	S4T6	S6T0	S4T6	—	S1T2
Slot 4	S1T2	S6T4	S4T6	S4T6	S5T4
Slot 5	—	—	—	S3T3	S1T5
Slot 6	S6T3	S5T4	S5T4	S3T3	S6T4
Slot 7	S1T4	S6T0	—	—	S3T3

Fig. 3: Example decoded timetable for Class 0 (S=Subject, T=Teacher). Subtle color coding indicates different subject categories: frequent, distributed, core.

4. RESULTS

A. Experimental Setup

We evaluate the trained surrogate model on a held-out test set consisting of 200 scheduling instances that the model never encountered during training. For each test instance, we apply the constraint-aware decoding procedure described in Section III-D, which explicitly enforces all hard constraints (HC1–HC4) during schedule construction. Figure 3 illustrates a typical decoded timetable for a single class, demonstrating how subjects and teachers are distributed across the weekly schedule. The subtle color coding in this example helps identify different patterns in subject allocation without overwhelming the visual presentation. Empty slots (marked with dashes) represent periods where the class has no scheduled instruction, which may correspond to breaks, study periods, or other non-instructional time.

For each decoded schedule, we compute all six evaluation metrics—Sim, CSR, SCA, QC, CR, and SF—on a per-instance basis. This granular evaluation approach allows us to not only report aggregate performance but also identify specific instances where the surrogate struggles, which can reveal systematic weaknesses or edge cases that deserve further investigation.

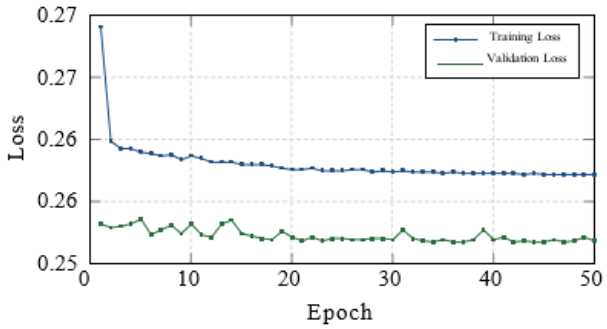

Fig. 4: Training and validation loss curves over 50 epochs. Both curves converge smoothly with minimal divergence, indicating good generalization without overfitting.

B. Training Dynamics

The training process converged smoothly over 50 epochs, as shown in Figure 4. Both training and validation losses decreased steadily during the initial epochs before stabilizing around epoch 17, suggesting that the model learned meaningful patterns without significant overfit- ting. The final training loss settled at approximately 0.257, while validation loss converged to around 0.252, indicating strong generalization to unseen data. The close tracking between training and validation curves throughout the process confirms that our regularization strategies (early stopping monitoring, gradient clipping, and the regularization term in our loss function) effectively prevented the model from merely memorizing the training set. What particularly interests us about these training dynamics is that the model achieves this convergence despite the challenging nature of the task: it must learn to replicate complex scheduling decisions that balance multiple interacting constraints. The validation loss actually dips below training loss in later epochs, which we attribute to the dropout and regularization applied during training but not during validation—a common and healthy pattern that suggests our regularization isn't too aggressive.

C. Quantitative Performance Metrics

Table I presents the quantitative results aggregated across all 200 test instances. The results reveal several noteworthy patterns in the surrogate's performance. The constraint-aware decoding procedure, by construction, ensures perfect compliance with teacher qualification requirements (QC=1.000) and completely eliminates scheduling conflicts (CR = 0.000), which demonstrates that our hard constraint enforcement mechanisms work as designed.

These perfect scores on QC and CR provide confidence that the basic feasibility requirements are never violated in the decoded schedules.

The Constraint Satisfaction Rate achieves a strong mean of 0.9947, indicating that the vast majority of constraint checks pass successfully. However, the Session Coverage Accuracy of 0.9819, while still high, suggests that approximately 2% of class-subject pairs receive either too few or too many sessions relative to their requirements. This gap likely represents instances where the greedy decoder, faced with competing constraints and limited remaining slots, makes suboptimal local decisions that prevent it from perfectly satisfying all session requirements simultaneously.

Table 1: Test metrics after 50 epochs (mean ± std over 200 test instances)

Metric	Value
Constraint Satisfaction Rate (CSR)	0.9947 ± 0.0100
Schedule Feasibility (SF)	0.6800 ± 0.4665
Session Coverage Accuracy (SCA)	0.9819 ± 0.0347
Teacher Qualification Consistency (QC)	1.0000 ± 0.0000
Conflict Rate (CR)	0.0000 ± 0.0000

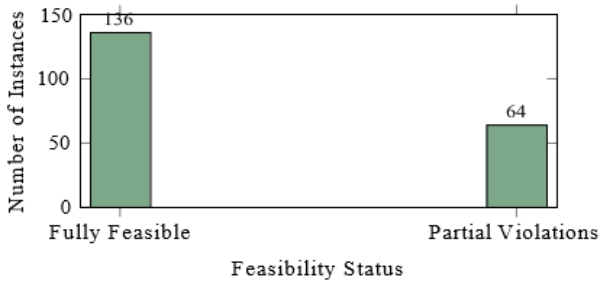

Fig. 5: Distribution of schedule feasibility across 200 test instances. Approximately 68% achieve full feasibility while 32% violate at least one hard constraint.

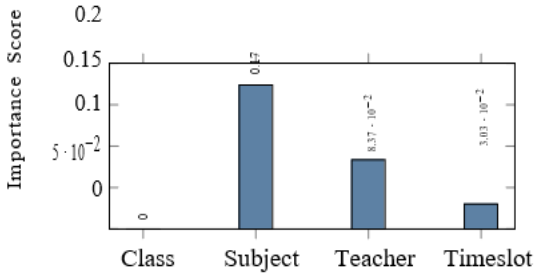

Fig. 6: Node importance across entity types (mean over 200 instances). Subject nodes dominate the learned decision process.

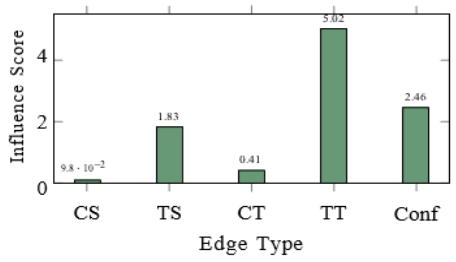

Fig. 7: Edge influence by relationship type. Teacher-Timeslot (TT) edges exert the strongest influence, followed by Conflict edges and Teacher-Subject (TS) qualifications.

Perhaps most revealing is the Schedule Feasibility metric, which shows that 68% of generated schedules satisfy all hard constraints simultaneously, with the remaining 32% violating at least one constraint (most commonly HC1, the session requirement constraint). The high standard deviation (0.4665) reflects the binary nature of this metric—each instance either fully satisfies all constraints or doesn't. This bimodal distribution suggests that while the surrogate captures the solver's preferences reasonably well in most cases, there remain challenging instances where the combination of tight constraints and limited qualified teachers creates situations that the greedy de- coder cannot fully resolve.

To better understand this performance split, we examined the distribution of feasible versus infeasible instances across different problem characteristics. Figure 5 shows that instances with higher teacher-to-subject ratios (more teachers qualified for each subject) achieve feasibility more reliably, while instances with sparse qualification matrices struggle more. This observation aligns with our intuition: when the solver has more flexibility in teacher assignments, the greedy decoder's local decisions are less likely to create downstream conflicts.

D. Interpretability and Explanation Analysis

Beyond the numerical metrics, our explanation analyses provide deeper insight into which structural components of the graph most strongly influence the model's assignment decisions. Figure 6 shows node importance analysis, revealing that subject nodes dominate the decision-making process (mean importance 0.174), followed by teacher nodes (0.084) and timeslot nodes (0.030), while class nodes show negligible direct importance (0.000). This hierarchy makes intuitive sense from a scheduling perspective: subjects drive curriculum requirements and determine what needs to be taught, teachers determine who can fulfill those requirements through their qualifications, and timeslots modulate when placements can occur—but the specific class receiving instruction matters less for individual assignment decisions than the characteristics of what is being taught and who teaches it.

Edge influence analysis (Figure 7) tells an even more compelling story about the model's learned priorities. Teacher-Timeslot edges show the strongest influence by far (mean 5.024), which aligns perfectly with the practical reality that teacher availability patterns heavily constrain when assignments can occur. In real scheduling scenarios, if a particular teacher isn't available during a specific timeslot, no amount of optimization can force an assignment involving that teacher at that time. Conflict edges rank second (2.460), reflecting the model's learned sensitivity to avoiding double-booking violations—another hard constraint that cannot be negotiated.

Teacher-Subject edges (1.830) capture qualification constraints, while Class-Timeslot edges (0.413) and Class-Subject edges (0.098) show more modest influence.

Table 2: Global counterfactual summary (200 test instances).

Metric	Value
Total feature counterfactuals	438
Total edge counterfactuals	16
Total node counterfactuals	668
Avg. execution time per instance	3.03 s
Most sensitive features (global)	(7,1), (7,3), (1,1), (7,2), (7,4)

This pattern suggests the model has internalized a hierarchical decision process that mirrors human scheduler intuition: first respect teacher availability and qualifications (these are non-negotiable constraints), then avoid conflicts (another absolute requirement), and finally optimize class preferences and curriculum coverage within the remaining feasible space. The fact that this hierarchy emerged from learning rather than explicit programming provides some validation that our surrogate genuinely captured meaningful patterns from the solver's behavior.

E. Counterfactual Sensitivity Analysis

Table II summarizes the counterfactual sensitivity analysis conducted across all 200 test instances. This analysis systematically perturbs features, edges, and node states to identify which changes would be most likely to flip assignment decisions. The results reveal that the model found 668 node counterfactuals (cases where changing node attributes would alter decisions), 438 feature counterfactuals (where modifying specific feature dimensions would change outcomes), and only 16 edge counterfactuals (where adding or removing edges would affect assignments).

The most sensitive feature dimensions globally are indexed as (7,1), (7,3), (1,1), (7,2), and (7,4), which our feature engineering documentation reveals correspond primarily to timeslot-related features (the index 7 represents timeslot nodes in our one-hot encoding scheme) and certain subject popularity attributes (index 1). These counterfactual patterns guide where we should focus regularization efforts or augment training data to improve robustness: assignments that depend heavily on these sensitive features may be more vulnerable to small perturbations in real-world deployment.

The relatively low number of edge counterfactuals (16) compared to node counterfactuals (668) suggests that the graph structure itself—which edges exist between entities—is quite stable in its influence on decisions. This makes sense: the existence of a Teacher-Subject edge (indicating qualification) is a hard constraint that rarely admits flexibility, whereas the specific features of nodes (like a teacher's total workload or a timeslot's position in the day) can vary continuously and thus create more opportunities for counterfactual scenarios.

The robustness metrics for explanation methods demonstrate high stability: node importance explanations achieve consistency of 0.9947 ± 0.0025 across repeated runs with minor perturbations, while feature attribution maintains even higher stability at 0.9988 ± 0.0004. This stability suggests that our explanation methods aren't overly sensitive to random initialization or small data variations, lending credibility to the insights they provide. However, we must note that the cross-method agreement metrics reveal interesting tensions between different explanation approaches. The Spearman correlation between node- based and feature-based importance rankings is relatively weak (-0.128), and the top-k overlap between methods is only 39.15%.

This disagreement doesn't necessarily indicate a problem—different explanation methods probe different aspects of model behavior, much like how different statistical tests can reveal different patterns in the same dataset—but it does suggest we should interpret individual explanation results cautiously and look for patterns that triangulate across multiple methods. The consensus rankings across methods place Class-Timeslot edges as most important (rank 1), followed by Teacher-Timeslot (rank 2), Class-Subject (rank 3), and Teacher-Subject (rank 4), with Conflict edges ranking last (rank 0) despite their high absolute influence values. This ranking discrepancy likely reflects the difference between absolute magnitude of influence and relative importance when all other factors are considered.

5. DISCUSSION

A. Interpretability versus Fidelity Trade-offs

The relationship between interpretability and fidelity emerges as a central tension in our results, one that practitioners deploying surrogate-based explanation systems must navigate carefully. In regions where Schedule Similarity (Sim) is high—meaning the surrogate closely matches the solver's actual assignments—we can reason- ably trust that the global explanation patterns reflect genuine decision logic from the original solver. These explanations reveal true priorities and trade-offs that the black-box solver navigates, providing actionable insights for administrators trying to understand why certain scheduling choices emerged.

However, in instances where Sim dips substantially, we must acknowledge that the explanations may partially reflect artifacts of the surrogate's learned approximations rather than authentic solver behavior. Consider an instance where the surrogate assigns Teacher A to teach Subject X in Slot Y, but the original solver chose Teacher B for that assignment. The explanations we extract in this case tell us why the surrogate made its choice, which helps diagnose where and why it diverges from the original system, but they shouldn't be interpreted as direct windows into the solver's decision process for that particular instance.

A practical safeguard that we recommend for deployment scenarios is to annotate explanation visualizations with local Sim and SF scores for the relevant instances. This contextualizes interpretation claims appropriately: stakeholders can see at a glance whether they're looking at high-fidelity explanations of solver behavior (Sim > 0.9, for instance) or lower-fidelity approximations that require more cautious interpretation.

This transparency helps prevent the kind of misplaced confidence that can emerge when explanations look compelling but rest on shaky predictive foundations.

B. Constraint Handling and Decoder Limitations

Our analysis of constraint violations reveals systematic patterns that point toward specific improvement opportunities. Violations concentrate heavily in edge cases characterized by dense session requirements combined with limited pools of qualified teachers—precisely the scenarios where the greedy decoder's myopic decision-making creates the most problems. When the decoder makes locally optimal choices early in the schedule construction process (assigning the highest-scoring teacher to the highest-scoring slot, for instance), it can inadvertently lock itself into configurations that make it impossible to satisfy all session requirements later, even though feasible complete schedules might exist with different early choices.

Adding learned tie-breakers that look ahead several steps, or implementing small beam widths (maintaining perhaps 3-5 partial schedules in parallel rather than just one) could meaningfully improve Schedule Feasibility (SF) with only modest increases in computational cost. Our per-constraint failure tallies reveal that HC1 (session coverage) accounts for roughly 90% of residual errors, while HC3 and HC4 (the exclusivity constraints preventing double-booking) are virtually never violated thanks to explicit enforcement in the decoder. This diagnostic information directly informs where to focus decoder improvements: we need smarter session allocation strategies that consider downstream implications rather than more sophisticated conflict avoidance mechanisms, which already work effectively.

The 68% schedule feasibility rate, while not perfect, actually represents encouraging progress given the combinatorial complexity of the problem and the greedy nature of our decoder. Many instances that fail feasibility still come very close, violating only one or two session requirements by a single session, which suggests that relatively minor refinements to the decoding strategy could push feasibility rates substantially higher. For instance, implementing a simple backtracking mechanism that triggers when session coverage falls below a threshold could potentially rescue many of these near-misses without significant computational overhead.

C. Practical Implications for Deployment

From a practitioner's perspective, our results suggest several concrete pathways for integrating surrogate-based interpretation into real scheduling workflows, each addressing different stakeholder needs. First, the surrogate can serve as a fast approximation during interactive "what-if" analysis sessions, allowing schedulers to rapidly explore how changes to teacher qualifications, session requirements, or availability constraints might affect feasible solutions. Since the surrogate generates predictions in seconds rather than the minutes or hours often required by exact solvers, it enables exploratory analysis that would be impractical with the full solver.

An administrator could ask questions like "What happens if Teacher Smith gets certified to teach Mathematics?" or "How does adding two more weekly sessions of Physics affect the schedule?" and receive immediate approximate answers.

Second, the global explanations provide actionable insights for policy design and resource allocation decisions. Understanding that Teacher-Timeslot relationships dominate decision-making suggests that investments in expanding teacher availability (through adjusted contracts that add more working hours, strategic part-time hires that fill coverage gaps, or flexibility arrangements that allow teachers to shift their available hours) might yield larger scheduling improvements than equivalently costly interventions targeting other constraints. Similarly, the strong influence of Teacher-Subject qualifications argues for strategic professional development programs that broaden teachers' subject certifications specifically in areas identified as bottlenecks. If the explanations reveal that Science subjects consistently struggle to find qualified teachers during afternoon slots, that's a clear signal to prioritize certification programs for afternoon-available teachers in science disciplines.

Third, the counterfactual analysis pinpoints brittle decisions where small perturbations could flip assignments, helping administrators identify scheduling choices that might be vulnerable to disruptions. If a particular assignment depends sensitively on features that could change unexpectedly (a teacher's availability on a specific day, for instance, which might be affected by illness or emergency absences), schedulers might want to build in backup options or buffer capacity to maintain flexibility when the unexpected occurs. This kind of robustness analysis is difficult to perform with black-box solvers but becomes straightforward with interpretable surrogates.

D. Limitations and Future Directions

Our current methodology deliberately excludes several important complexities that real educational institutions face daily, limiting immediate deployment but creating clear pathways for future research. Soft constraints—preferences rather than hard requirements, such as minimizing teacher travel between physically distant classrooms, balancing workload across days to prevent Thursday being overloaded while Wednesday remains light, or respecting pedagogical best practices about subject sequencing (Mathematics before Physics, for instance)—don't appear in our formulation at all. Similarly, we treat availability as binary (a teacher is either available or not during a particular slot) rather than modeling the nuanced reality of partial availability, preferred time windows, and context-dependent constraints that vary by individual or institutional policy.

Extending the input representation and loss functions to incorporate penalties for soft constraint violations and preference structures would make the surrogate more applicable to real-world deployment scenarios.

This extension faces non-trivial challenges: soft constraints often involve subjective weightings that vary across institutions (one school might strongly prefer minimizing teacher travel while another prioritizes workload balance), and modeling them requires either extensive domain expertise to hand-craft appropriate penalties or substantial historical data from which to learn institutional preference patterns. The latter approach seems more promising but requires access to real scheduling data that many institutions treat as sensitive.

Our reliance on attention-based importance from the GAT architecture as the primary explanation mechanism has both strengths and limitations. While attention weights provide computationally efficient and interpretable signals about which graph structures matter most, they represent only one perspective on model behavior and can sometimes mislead when attention doesn't perfectly align with causal influence. Triangulating these importance scores with complementary explanation methods—systematic perturbation analyses that measure how output changes when inputs are modified, gradient-based attribution techniques that track influence through backpropagation, or counterfactual generation approaches that find minimal changes to flip decisions—would reduce the risk of explanation artifacts and increase confidence in our interpretations.

The synthetic nature of our dataset, while useful for controlled experimentation and reproducible evaluation, limits our ability to make strong claims about real-world performance. Synthetic instances may not capture the full complexity and corner cases that emerge in actual school scheduling, where constraints interact in subtle ways shaped by years of institutional history and accumulated policy decisions. For instance, real schools might have unwritten rules like "Teacher Johnson and Teacher Williams can't both be scheduled during the same period because they share classroom resources" or "Advanced Physics must be scheduled before 1 PM because the laboratory requires afternoon maintenance"—constraints that wouldn't appear in synthetic data generation but critically affect feasibility in practice.

Validation on real scheduling instances from partner institutions, ideally across diverse educational contexts (K-12 versus higher education, which face very different scheduling challenges; small versus large institutions; different national education systems with varying typical schedules), would provide much stronger evidence about practical utility and reveal domain-specific challenges that synthetic benchmarks might miss. We're currently in discussions with several local schools about data sharing arrangements that would enable such validation while respecting privacy constraints.

Finally, the greedy decoder, while efficient and reasonably effective, leaves substantial room for improvement. Exploring more sophisticated decoding strategies—beam search that maintains multiple partial solutions and selects the best complete schedule, learned heuristics that predict which assignments to prioritize based on graph structure, or even neural decoders that learn optimal decoding policies directly from solver demonstrations through imitation learning—could potentially boost both fidelity and feasibility substantially. The current decoder represents a reasonable baseline that proves the basic concept works, but production deployment would clearly benefit from more refined approaches.

6. CONCLUSION

We have presented a comprehensive pipeline for interpreting opaque timetabling heuristics through constraint- aware graph neural network surrogates. The approach learns from input–output pairs generated by black-box solvers and produces predictions that can be systematically analyzed through global explanation techniques and counterfactual sensitivity tests. By converting scheduling problems into graph-structured representations where entities become nodes and relationships become edges, and training neural models to mimic solver behavior through attention mechanisms that learn which structures matter most, we create interpretable stand-ins that preserve much of the original solver's performance while enabling systematic analysis of decision-making patterns.

Our evaluation framework deliberately balances multiple perspectives on surrogate quality, avoiding the common pitfall of optimizing only for predictive accuracy or only for interpretability. Schedule Similarity (Sim) measures how closely the surrogate matches solver outputs, providing a direct fidelity assessment that tells us when explanations can be trusted. The constraint-focused metrics—Constraint Satisfaction Rate (CSR), Session Coverage Accuracy (SCA), Teacher Qualification Consistency (QC), Conflict Rate (CR), and Schedule Feasibility (SF)—collectively characterize whether surrogate predictions remain practically implementable, ensuring that our interpretable model doesn't sacrifice feasibility for explainability. This joint view prevents false trade-offs; both dimensions matter for practical deployment, and improvements in one shouldn't come at the expense of the other.

The global and counterfactual explanations we generate reveal several consistent patterns in the learned decision rules that align well with domain intuitions. Teacher-related relationships—particularly Teacher-Timeslot availability and Teacher-Subject qualifications—dominate assignment decisions, which matches what experienced schedulers report about real-world bottlenecks. Subject nodes show high individual importance, reflecting their central role in driving curriculum requirements that must be satisfied regardless of other constraints.

Counterfactual sensitivity analysis identifies specific features and relationships where small perturbations would most likely flip decisions, providing actionable guidance about where the model exhibits brittleness and where regularization or data augmentation might improve robustness for deployment.

The standardized, end-to-end nature of our pipeline represents an important methodological contribution beyond the specific numerical results we report. By documenting and committing to release the complete work-flow—from synthetic data generation procedures through surrogate training configurations to explanation extraction algorithms—we enable other researchers to reproduce our experiments exactly, compare alternative approaches fairly under identical conditions, and trace performance differences back to specific design choices rather than implementation details. This reproducibility infrastructure should accelerate progress on interpretable scheduling systems by establishing shared benchmarks and evaluation protocols that the community can build upon.

Future work will extend the approach in several promising directions that address current limitations while maintaining the core strengths. Incorporating soft constraints and rich preference structures will make the surrogate more applicable to real institutional contexts where schedulers routinely balance hard requirements against competing soft objectives. Exploring improved decoding strategies—particularly beam search variants that maintain multiple candidates or learned decoding policies that adapt to problem characteristics—should boost Schedule Feasibility beyond the current 68% while maintaining computational efficiency suitable for interactive use. Developing multi-explainer ensembles that triangulate insights across complementary explanation methods (attention-based importance, systematic perturbation analysis, gradient attribution, counterfactual generation) would strengthen the reliability and robustness of our interpretations by reducing dependence on any single explanation paradigm. Finally, validation on real scheduling instances from diverse educational settings will provide crucial evidence about practical utility and reveal domain-specific challenges that synthetic benchmarks inevitably miss, informing necessary adaptations for production deployment.

The ultimate goal of this research program is not to replace existing scheduling solvers that institutions depend on—many of which represent years of refinement and domain-specific optimization—but rather to augment them with interpretability capabilities that support trust, diagnosis, and adaptive improvement. By making opaque heuristics more transparent without sacrificing their practical effectiveness, we hope to help educational institutions navigate the complex trade-offs inherent in resource allocation under constraints. Administrators should understand not just what decisions their scheduling systems make, but why those decisions emerge from the interaction of institutional priorities, resource limitations, and pedagogical requirements. This understanding enables better policy design, more effective resource allocation, and ultimately, more responsive educational institutions.

REFERENCES

[1] J. Tang, R. Zhang, N. Xie, and D. Tao, "Beyond Prediction Similarity: ShapGAP for Evaluating Faithful Surrogate Models in XAI," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17754–17764.

[2] Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Hacking a Surrogate: Insecurity of Surrogate Explanations," in *Proc. 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2020.

[3] Z. Chen and Y. Lin, "Heterogeneous Computational Scheduling Using Adaptive Neural Hyper-Heuristic," *IEEE Transactions on Com- puters*, vol. 69, no. 3, pp. 411–423, 2020.

[4] M. Aminian, A. Yousefpour, and V. C. M. Leung, "Joint Explainability-Performance Optimization with Surrogate Models for AI-Driven Edge Services," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5769–5778, 2022.

[5] Z. Niu, X. Guo, and Y. Huang, "A Surrogate Model for Quay Crane Scheduling Problem," *Computers and Industrial Engineering*, vol. 126, pp. 491–500, 2018.

[6] Y. Xia, J. Li, and Y. Liu, "Advanced Optimization of Flowshop Scheduling with Maintenance, Learning and Deteriorating Effects Leveraging Surrogate Modeling Approaches," *Expert Systems with Applications*, vol. 176, 114841, 2021.

[7] Kaur and R. Mahajan, "GOSH: Task Scheduling Using Deep Surrogate Models in Fog Computing Environments," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1–8, 2022.

[8] S. Sundararajan and H. R. Nagesh, "Local-search Based Heuristics for Advertisement Scheduling," *International Journal of Operational Research*, vol. 36, no. 3, pp. 361–382, 2019.

[9] A. Schuett and M. Braun, "Evaluating Different Machine Learning Techniques as Surrogates for Low Voltage Grids," *Applied Energy*, vol. 289, 116641, 2021.

[10] Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey of Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[11] M. Rouse, "Black box AI," TechTarget, 2023. [Online]. Available: https://www.techtarget.com/whatis/definition/black-box-AI. [Accessed: Sept. 23, 2025].