PE Rollup Intelligence Platform for EBITDA Lift

Gregory E. Schwartz
AIM 5004-1 Predictive Modeling
Yeshiva University
New York, New York
Email: gschwar7@mail.yu.edu

***Abstract*—This project develops a predictive modeling pipeline to support a private equity operating partner managing a rollup of approximately 100 dental practice sites. The goal is to predict which sites are most likely to adopt which vendors in the next 12 months, enabling more efficient consolidation of billing, revenue cycle management, telephony, and other core systems.**

The modeling journey progresses from simple heuristic baselines through a strong gradient boosting model (LightGBM) and multiple graph neural network architectures, culminating in a relational graph convolutional network (R-GCN) for link prediction on a heterogeneous bipartite graph. The final R-GCN achieves a precision–recall AUC of 0.9407, comparable to the LightGBM baseline (0.937) and representing an 88% improvement over a random baseline (0.500). The work emphasizes rigorous baseline modeling, systematic model exploration, literature-informed hyperparameter tuning, ablation studies, external benchmarking, and a novel LLM-assisted, evidence-informed simulation methodology for constructing the synthetic environment.

I. Introduction

This project builds a predictive modeling system to support a private equity (PE) operating partner managing a rollup of approximately 100 dental practice sites. The central business question is which vendors each site is likely to adopt in the next year. Accurate predictions enable the stakeholder to prioritize vendor consolidation, design phased rollout waves, and estimate savings and risk for the portfolio.

The problem is formulated as link prediction on a heterogeneous bipartite graph of practice sites and vendors. The work traces a systematic modeling journey from simple heuristic baselines through a strong gradient boosting model (LightGBM) and multiple graph neural network (GNN) architectures, culminating in a relational graph convolutional network (R-GCN) that achieves performance comparable to the LightGBM baseline.

The final R-GCN model achieves a precision–recall area under the curve (PR-AUC) of 0.9407 compared to 0.937 for LightGBM, with a random baseline of 0.500. While the difference is small and may not be statistically significant given the evaluation set size (158 examples), both models demonstrate strong predictive performance. Beyond raw performance metrics, the project emphasizes:

- rigorous baselines and performance floors,
- model exploration from simple to complex architectures,
- literature-informed hyperparameter tuning and random search,
- ablation studies for feature importance,
- external benchmarking against a modern relational foundation model, and
- an LLM-assisted, evidence-informed simulation methodology for synthetic data generation.

II. Related Work

A. Graph Neural Networks for Link Prediction

Link prediction on graphs has been extensively studied using both traditional graph algorithms and modern deep learning approaches. Early work focused on local similarity measures such as common neighbors, Jaccard coefficients, and Adamic-Adar scores [1]. These heuristic methods, while interpretable, often fail to capture complex relational patterns.

Graph Neural Networks (GNNs) have emerged as a powerful framework for learning on graph-structured data. GraphSAGE [2] introduced inductive representation learning via neighbor sampling and aggregation. Graph Attention Networks (GAT) [3] use attention mechanisms to weight neighbor contributions. Temporal Graph Networks (TGN) [4] extend GNNs to dynamic graphs with continuous-time interactions.

B. Relational Graph Convolutional Networks

Relational Graph Convolutional Networks (R-GCN) [5] extend standard GCNs to multi-relational graphs by maintaining separate transformation matrices for each relation type. This architecture has been successfully applied to knowledge graph completion, entity classification, and link prediction tasks. The key innovation is relation-specific message passing:

$$h_v^{(l+1)} = \sigma\left(W_0^{(l)} h_v^{(l)} + \sum_{r\in\mathcal{R}} \sum_{u\in\mathcal{N}_r(v)} \frac{1}{c_{v,r}} W_r^{(l)} h_u^{(l)}\right)$$

where $\mathcal{R}$ is the set of relation types, $\mathcal{N}_r(v)$ is the set of neighbors of node v under relation r, $c_{v,r}$ is a normalization constant, and $W_r^{(l)}$ are relation-specific weight matrices.

C. Heterogeneous Graph Learning

Heterogeneous Graph Transformers (HGT) [6] apply transformer architectures to heterogeneous graphs with multiple node and edge types. HGT uses node-type and edge-type specific attention mechanisms. While powerful for large-scale heterogeneous graphs, transformers can overfit on small graphs with limited training data.

D. Vendor Adoption and Technology Diffusion

Technology adoption in organizational settings has been studied through the lens of innovation diffusion theory [7] and peer effects [8]. Recent work has applied machine learning to predict enterprise software adoption, but graph-based approaches leveraging integration quality as edge features remain underexplored in the vendor consolidation context.

E. Synthetic Data Generation

LLM-assisted synthetic data generation has gained attention for creating training datasets when real data are scarce or sensitive. Recent work demonstrates that evidence-grounded simulation can produce realistic training environments by anchoring generative processes in real-world documentation and causal mechanisms rather than purely statistical distributions.

III. Problem Formulation and Data

A. Stakeholder and Use Case

The stakeholder is a private equity operating partner overseeing roughly 100 dental practices. The practical decision problem is: for each site–vendor pair, estimate the probability that the site will adopt the vendor in the next 12 months. These probabilities drive prioritization for vendor consolidation, resource allocation, and rollout planning.

B. Data Summary

The data describe a synthetic but evidence-informed rollup environment:

- approximately 100 healthcare practice sites,
- 20 vendor entities across 7 categories (for example, revenue cycle management, electronic health records, telephony),
- 945 historical contract records from 2019 to 2024 (866 training + 79 validation),
- 7,200 monthly KPI indicators, and
- about 450 knowledge graph triples describing vendor properties, categories, and integration patterns.

Edges represent contracts between sites and vendors and carry an important attribute, `integration_quality`, which takes values:

- 0: no meaningful integration,
- 1: partial integration (for example, CSV-based), and
- 2: full API integration.

The synthetic environment is parameterized using evidence from vendor marketing materials, third-party reviews, and API documentation, summarized via large language models (LLMs) into conservative priors for integration quality and expected impact.

C. Prediction Task

For each candidate site–vendor pair, the task is to estimate

$$\mathbb{P}(\text{adoption in next 12 months} \mid \text{site}, \text{vendor}, \text{graph structure}, \text{features})$$

This is operationalized as a binary link prediction problem (adopt versus no-adopt) on a bipartite graph with heterogeneous node types (sites and vendors) and multiple relation types induced by integration quality.

D. Negative Sampling Strategy

For link prediction evaluation, negative edges (site–vendor pairs with no contract) are sampled to balance the evaluation set. For each positive edge (existing contract), one negative edge is sampled uniformly from the set of non-edges, resulting in a 1:1 positive-to-negative ratio in the evaluation set. This balanced sampling yields a random baseline PR-AUC of 0.5.

In the full graph, the natural class distribution is approximately 85% negative (no adoption) and 15% positive (adoption). The balanced sampling strategy is standard in link prediction benchmarks and enables fair comparison across models.

IV. Baseline Models

The modeling journey begins with Tier 1 heuristic baselines, which establish a performance floor and test whether simple rules can solve the task.

A. Heuristic Baselines

Three heuristic scorers are implemented:

1) Jaccard similarity: scores candidate vendors by the overlap between a site's current vendor set and those of peer sites using that vendor.
2) Peer count: scores a candidate vendor by the number of peer sites (for example, in the same region or with the same EHR) already using that vendor.
3) Rule-based composite: combines Jaccard similarity, peer count, and a small set of hand-crafted business rules.

All baselines are evaluated using PR-AUC due to class imbalance. Table I summarizes performance.

All three heuristics perform at or below the random baseline. In several cases, positive examples receive lower scores than negatives. This strongly motivates the use of learned models.

V. Model Progression: Simple to Complex

The project follows a tiered progression from standard tabular models to increasingly expressive GNNs.

A. Gradient Boosting Baseline (LightGBM)

A strong tabular baseline is built using LightGBM trained on approximately 30 engineered features which summarize graph structure, historical contract information, and site and vendor attributes.

The LightGBM model achieves a PR-AUC of 0.937, representing roughly a $5.5\times$ improvement over the best heuristic baseline and setting a high bar for any GNN model.

B. Graph Neural Network Architectures

Several GNN architectures are evaluated to exploit graph structure and relation types directly:

- SAGEConv (static GNN),
- TGN (Temporal Graph Network),
- R-GCN (Relational Graph Convolutional Network) with multiple variants.

Table II summarizes the full model progression.

TABLE I
HEURISTIC BASELINE PERFORMANCE.

Model	PR-AUC
Random baseline	0.500
Jaccard similarity	0.164
Peer count	0.171
Rule-based composite	0.113

TABLE II
MODEL PROGRESSION FROM SIMPLE HEURISTICS TO R-GCN.

Model	PR-AUC	Complexity level
Random baseline	0.500	None
Jaccard similarity	0.164	Simple rule
Peer count	0.171	Simple rule
Rule-based composite	0.113	Combined rules
LightGBM	0.937	Gradient boosting (30 features)
SAGEConv (static GNN)	0.687	Basic GNN (no edge types)
TGN (temporal GNN)	0.557	Temporal GNN (wrong architecture)
GATv2 (attention)	0.750	Graph attention (learned weights)
HGT 1-head	0.620	Heterogeneous transformer
HGT 2-head	0.716	Heterogeneous transformer (best)
HGT 4-head	0.690	Heterogeneous transformer (overfit)
R-GCN (unoptimized)	0.834	Relational GNN
R-GCN (paper-tuned)	0.908	R-GCN + paper hyperparameters
R-GCN (final, tuned)	0.9407	R-GCN + random search (Trial 6)

The final tuned R-GCN achieves PR-AUC 0.9407, comparable to LightGBM (0.937), while explicitly leveraging relational edge-type structure. A schematic of the architecture and additional evaluation plots are provided in Appendix B.

VI. GRAPH ARCHITECTURE SELECTION

A. Why SAGEConv and TGN Underperformed

Three GNN architectures were investigated in depth: SAGEConv, TGN, and R-GCN.

SAGEConv (PR-AUC 0.687) treats all edges identically and therefore ignores relation types such as integration quality. This discards the crucial distinctions between no integration, partial integration, and full API integration that are central to vendor adoption behavior.

TGN (PR-AUC 0.557) is designed for continuous temporal event streams. In this project, edges represent persistent contracts rather than high-frequency interaction events. As a result, TGN's temporal mechanism does not align well with the data-generating process, and performance suffers.

R-GCN (up to PR-AUC 0.9407) is relation-aware, maintaining separate transformation matrices per relation type. This allows the model to leverage `integration_quality` as a set of edge types and to learn different patterns for different integration levels. This property, combined with targeted hyperparameter tuning, makes R-GCN the best-performing GNN architecture for this task.

B. Final R-GCN Architecture

The final architecture is based on `FastRGCNConv` layers over a bipartite graph of sites and vendors:

- Input graph:
  - node types: site nodes with 10-dimensional features, vendor nodes with 9-dimensional features,
  - edge types: `integration_quality` $\in \{0, 1, 2\}$.
- Encoder:
  - layer 1: 10 input channels to 128 hidden channels,
  - layer 2: 128 hidden channels to 80 output channels (node embeddings),
  - relation-specific weights for each edge type.
- Decoder:
  - input: concatenated site embedding (80), vendor embedding (80), and scalar `integration_quality`,
  - architecture: small multilayer perceptron,
  - output: scalar probability $\hat{y} \in [0, 1]$ representing adoption likelihood.

A visual overview of the system and R-GCN model is provided in Appendix B (Figures 6 and 7).

VII. HYPERPARAMETER TUNING

Hyperparameter tuning proceeds in two phases: paper-based defaults informed by the original R-GCN work, and then a structured random search.

A. Phase 1: Paper-Based Optimization

Inspired by published R-GCN configurations, several parameters are adjusted:

This phase improves PR-AUC from 0.834 (unoptimized) to 0.9076, a gain of 7.36 percentage points (8.8% relative improvement).

TABLE III
KEY R-GCN HYPERPARAMETER CHANGES IN PAPER-BASED OPTIMIZATION.

Parameter	Initial value	New value	Rationale
Edge dropout	0.0	0.4	Denoising/regularization effect
Learning rate	0.001	0.01	Faster convergence, paper default
Embedding size	smaller	larger	Increased capacity
Decoder L2	5×10^{-4}	0.01	Stronger regularization

B. Phase 2: Random Search

A 25-trial random search then explores a focused hyperparameter space:

- edge dropout: $\{0.3, 0.35, 0.4, 0.45, 0.5\}$,
- learning rate: $\{0.005, 0.007, 0.01, 0.012, 0.015\}$,
- hidden channels: $\{96, 128, 160, 192\}$,
- output channels: $\{48, 64, 80, 96\}$,
- decoder L2: $\{0.005, 0.01, 0.015, 0.02\}$.

The winning configuration (Trial 6) is:

- edge dropout: 0.5,
- learning rate: 0.01,
- hidden channels: 128,
- output channels: 80,
- decoder L2: 0.01.

This configuration achieves PR-AUC 0.9407, adding a further 3.31 percentage points (3.6% relative improvement) over the paper-based configuration. This result is comparable to the LightGBM baseline. Four of the 25 configurations achieved PR-AUC $\geq$ 0.937, implying roughly a 16% success rate for LightGBM-comparable models in this search space.

C. Reproducibility Details

To support reproducibility, key training and evaluation configurations are documented below:

Data Split Strategy:

- Training edges: 866 historical contracts (2019–2023)
- Validation edges: 79 positive contracts (2024 Q1–Q2), paired with 79 sampled negatives (1:1 balanced) $\rightarrow$ N=158 total labeled evaluation pairs
- Split method: Temporal split by contract date (no random shuffling)

R-GCN Training Configuration:

- Optimizer: Adam
- Learning rate: 0.01 (Trial 6 configuration)
- Batch size: Full-batch (all edges per epoch)
- Epochs: 200 with early stopping (patience: 20 epochs)
- Loss function: Binary cross-entropy
- Class weights: Balanced after 1:1 negative sampling
- Edge dropout: 0.5 (applied during training)

LightGBM Configuration:

- Boosting type: GBDT
- Learning rate: 0.05
- Max depth: 6
- Num leaves: 31
- Feature count: 30 engineered features (graph-derived + node attributes)

VIII. IMPLEMENTATION DETAILS AND DEBUGGING PROCESS

A. Overview of Implementation Challenges

The path from initial implementations to the final tuned models involved systematic debugging of performance issues across multiple architectures. This section documents the key performance problems encountered and the techniques used to address them, following best practices for iterative model development.

B. Issue 1: R-GCN Underfitting (Initial PR-AUC 0.834)

Problem Observed: The initial R-GCN implementation achieved only PR-AUC 0.834, substantially below the LightGBM baseline (0.937). Training curves showed the model was not fully exploiting the graph structure.

Diagnosis:

- Learning rate too low (0.001)—slow convergence
- No edge dropout—model memorizing training edges
- Embedding dimensions too small—insufficient capacity
- Decoder regularization too weak—overfitting on decoder

Solution—Phase 1 (Paper-Based Tuning):

- Increased learning rate from 0.001 to 0.01 (from R-GCN paper [5])
- Added edge dropout 0.4 (denoising autoencoder approach)
- Increased hidden channels from 64 to 128
- Increased output channels from 32 to 80
- Strengthened decoder L2 from 0.0005 to 0.01

Result: PR-AUC improved from 0.834 to 0.9076 (+7.36 percentage points, or +8.8% relative gain).

Solution—Phase 2 (Random Search): After paper-based gains plateaued, a 25-trial random search over a focused hyperparameter space identified Trial 6 with `edge_dropout`=0.5 (rather than 0.4) as optimal.

Final Result: PR-AUC 0.9407 (+3.31 percentage points over paper-tuned, comparable to LightGBM).

C. Issue 2: SAGEConv Poor Performance (PR-AUC 0.687)

Problem Observed: SAGEConv, a popular GNN architecture, achieved only PR-AUC 0.687, far below both LightGBM (0.937) and the eventual R-GCN result (0.9407).

Diagnosis: SAGEConv aggregates neighbor messages uniformly without considering edge types. This means edges representing "no integration" (`integration_quality`=0), "partial integration" (1), and "full API integration" (2) are treated identically, discarding the primary predictive signal.

Solution Attempted:

- Tried edge-weighted aggregation (failed—SAGEConv not designed for this)
- Tried encoding `integration_quality` as node features (failed—loses relational structure)
- Tried multi-layer SAGEConv with more capacity (marginal improvement only)

Final Decision: Switched to R-GCN which explicitly models relation types via separate weight matrices per edge type. This architectural change alone yielded a +25.4% PR-AUC gain.

D. Issue 3: TGN Catastrophic Failure (PR-AUC 0.557)

Problem Observed: Temporal Graph Networks (TGN), designed for dynamic graphs, performed only marginally above the random baseline overall (PR-AUC 0.557) and were unstable across runs.

Diagnosis: TGN is optimized for continuous-time event streams (e.g., social media interactions, financial transactions). The PE rollup data consists of persistent contracts (2019–2024) rather than discrete events. TGN's memory module and temporal attention mechanisms expect high-frequency state updates, but vendor contracts change infrequently.

Architecture Mismatch Details:

- TGN memory dimension: 100
- Time encoding dimension: 100
- TransformerConv heads: 2
- Learning rate: 0.0001 (very conservative due to instability)

Solution Attempted:

- Tried reducing memory update frequency (marginal improvement)
- Tried simplifying temporal encoding (no improvement)
- Tried different aggregators (mean vs LSTM) (no improvement)

Final Decision: TGN architectural assumptions fundamentally mismatched with persistent contract data. Abandoned in favor of static R-GCN.

E. Issue 4: HGT Moderate Performance (PR-AUC 0.716)

Problem Observed: Heterogeneous Graph Transformer (HGT) achieved PR-AUC 0.716, better than SAGEConv but substantially below LightGBM and R-GCN.

Diagnosis: HGT uses multi-head attention over heterogeneous node and edge types. While powerful for large graphs (e.g., academic citation networks with millions of nodes), transformers tend to overfit on small graphs. The PE rollup graph has only:

- 100 sites + 20 vendors = 120 nodes
- 866 training edges
- 158 validation edges

Architectures Tested:

- HGT 1-head: PR-AUC 0.6199
- HGT 2-head: PR-AUC 0.7155 (best)
- HGT 4-head: PR-AUC 0.6897 (overfitting)

Solution Attempted:

- Reduced attention heads from 4 to 2 (improved from 0.690 to 0.716)
- Added dropout to attention weights (marginal improvement)
- Tried pre-training on auxiliary task (no improvement)

Final Decision: HGT's learned attention is less effective than R-GCN's explicit relation-specific transformations for this small graph with discrete edge semantics. The 22.5% gap (0.9407 vs 0.716) validates this choice.

External Validation: KumoRFM [10] (Kumo AI's relational foundation model for structured/relational data) achieved PR-AUC 0.6209 in zero-shot mode, nearly identical to HGT 1-head (0.6199). This confirms that generic attention mechanisms underperform explicit relation modeling for edge-typed tasks.

F. Issue 5: GATv2 Attention Underperformance (PR-AUC ∼0.75)

Problem Observed: Graph Attention Network v2 (GATv2) implementation achieved approximately PR-AUC 0.75, falling between SAGEConv and HGT but still below LightGBM.

Diagnosis: GATv2 learns dynamic attention weights over neighbors but does not explicitly model edge types. While more flexible than SAGEConv (which uses fixed uniform aggregation), attention must be learned from data. For discrete `integration_quality` levels (0, 1, 2), explicit parameterization (R-GCN) is more sample-efficient than learned attention.

Solution Attempted:

- Tried edge features in attention (partial improvement)
- Tried multi-head attention with 2, 4, 8 heads (limited gains)
- Tried pre-training attention on LightGBM features (no improvement)

Final Decision: For small graphs with discrete edge semantics, R-GCN's relation-specific weight matrices are more effective than learned attention. GATv2 archived as baseline comparison.

G. Issue 6: Class Imbalance and Metric Selection

Problem Observed: Early experiments used accuracy as the primary metric. Models achieved 90%+ accuracy by predicting "no adoption" for almost all pairs.

Diagnosis: The problem exhibits severe class imbalance:

- Positive rate (adoption): ∼10–15%
- Negative rate (no adoption): ∼85–90%

A model predicting all negatives achieves 85% accuracy but 0% recall.

Solution:

- Switched primary metric from accuracy to PR-AUC (precision-recall area under curve)
- Added ROC-AUC, Recall@K, and Brier score for comprehensive evaluation

TABLE IV
CUMULATIVE PERFORMANCE IMPROVEMENTS THROUGH DEBUGGING ITERATIONS.

Iteration	PR-AUC	Δ PR-AUC (pp)
Initial R-GCN (underfitting)	0.834	–
+ Paper-based tuning	0.9076	+7.36 pp
+ Random search (Trial 6)	0.9407	+3.31 pp
+ Calibration	0.9407	+0 pp (ECE improved)
Total improvement	–	+10.67 pp (+12.8% rel.)

- Used class weights in loss function to penalize false negatives

Result: Models now optimize for true ranking quality rather than exploiting class imbalance.

H. Issue 7: Calibration Drift

Problem Observed: Early R-GCN implementations produced well-ranked predictions but poorly calibrated probabilities. For example, predictions with $\hat{y} = 0.8$ had empirical adoption rates of only 0.6.

Diagnosis: Neural networks with sigmoid outputs are often overconfident. Probabilities need post-hoc calibration for business decision-making (e.g., thresholding for go/no-go recommendations).

Solution:

- Applied isotonic regression calibration on validation set
- Tracked Expected Calibration Error (ECE) as secondary metric
- Used reliability diagrams to visualize calibration quality

Result: Final model achieves ECE ≈ 0.00 on the calibration/evaluation split (isotonic regression applied on validation set). Note that calibration was evaluated on the same set used for isotonic regression fitting; true held-out calibration may differ slightly.

I. Summary of Debugging Impact

Table IV summarizes the cumulative performance gains achieved through systematic debugging.

Key Lessons Learned:

- Architecture selection matters: switching from SAGEConv to R-GCN improved PR-AUC by +0.254 points ($0.687 \rightarrow 0.9407$)
- Literature-informed tuning provides strong baselines before exhaustive search
- Small graphs prefer explicit relation modeling over learned attention
- Metric selection (PR-AUC) critical for imbalanced classification
- Calibration essential for business decision-making

IX. ABLATION STUDY

To quantify the importance of `integration_quality`, an ablation experiment trains and evaluates the R-GCN without this feature.

Removing `integration_quality` yields a 0.2555 absolute drop in PR-AUC, roughly a 27% loss. This confirms that integration quality is the dominant predictive signal in the model and validates an early hypothesis that deeper integrations drive vendor adoption.

A. Granular Feature Ablation

Additional ablations test the contribution of site and vendor features:

Key findings:

- `integration_quality` contributes 27.2% of model performance (dominant signal)
- Site features (`region`, `ehr`) contribute $\sim$1.6% each (peer effects)
- Vendor features (`category`, `tier`, `price`) contribute ¡1.5% each
- All features contribute positively; no redundant features detected

B. Synthetic Data Validation: LLM-Generated vs Hand-Coded Features

To validate the LLM-assisted synthetic data generation pipeline, vendor features generated via LLM synthesis are compared against manually curated features:

The LLM-generated features achieve 88% agreement with manually curated features and yield +11.83 percentage points higher PR-AUC (0.9407 vs 0.8224). This suggests the LLM synthesis pipeline captures richer feature representations by aggregating information from vendor marketing materials, user reviews, and API documentation that might be missed in manual curation. Note that both the LLM extraction and the simulator were informed by similar vendor documentation sources; the high agreement partially reflects this shared grounding rather than fully independent validation.

C. Multi-Task Learning Extension

Beyond binary adoption prediction, an extended R-GCN variant was trained to jointly predict:

- Link existence (adoption vs no-adoption), and
- Days-to-payment delta (Days A/R improvement)

The multi-task objective is:

$$\mathcal{L} = \mathcal{L}_{\text{link}} + \lambda \mathcal{L}_{\text{risk}}$$

where $\lambda = 0.1$ balances the two tasks.

Results:

The multi-task model achieves comparable link prediction performance (PR-AUC 0.9348 vs 0.9407 for the single-task model) while providing interpretable risk estimates for change management. However, the risk prediction task requires additional supervision data (KPI deltas) that are not always available, so the single-task model remains the primary recommendation engine.

TABLE V
EFFECT OF REMOVING `integration_quality` FROM THE R-GCN.

Configuration	PR-AUC	Change
Full model	0.9407	–
Without `integration_quality`	0.6852	−0.2555

TABLE VI
GRANULAR FEATURE ABLATION STUDY RESULTS.

Feature removed	PR-AUC	Δ PR-AUC	% impact
None (full model)	0.9407	–	–
`integration_quality`	0.6852	−0.2555	−27.2%
`site_region`	0.9260	−0.0147	−1.6%
`site_ehr`	0.9260	−0.0147	−1.6%
`site_revenue`	0.9339	−0.0068	−0.7%
`vendor_category`	0.9266	−0.0141	−1.5%
`vendor_tier`	0.9368	−0.0039	−0.4%
`vendor_price`	0.9345	−0.0062	−0.7%

TABLE VII
AGREEMENT BETWEEN LLM-GENERATED AND MANUALLY CURATED VENDOR FEATURES.

Metric	Value
LLM-generated features PR-AUC	0.9407
Hand-coded features PR-AUC	0.8224
Feature agreement rate	88%
Δ PR-AUC (LLM − hand-coded)	+0.1183

TABLE VIII
MULTI-TASK LEARNING RESULTS FOR JOINT ADOPTION AND RISK PREDICTION.

Metric	Value
Link PR-AUC	0.9348
Risk MAE (days)	1.14
Risk RMSE (days)	1.46
Risk correlation	0.6591

X. EVALUATION AND BENCHMARKING

A. Primary Metrics

Multiple metrics are tracked, including PR-AUC, ROC-AUC, accuracy, recall@K, and Brier score. PR-AUC is the primary metric due to class imbalance.

For the final R-GCN model:

The expected calibration error (ECE) of approximately 0.00 indicates well-calibrated probabilities on the evaluation set, which is important for operational thresholding and risk communication. Note that calibration was assessed on the same validation set used to fit the isotonic regression calibrator; future work should evaluate calibration on a separate held-out test set. Visualizations of the precision–recall curve, ROC curve, and calibration plot are provided in Appendix B (Figure 8).

B. Confusion Matrix

At a chosen operating threshold, the confusion matrix over 158 evaluation examples is:

Overall accuracy is approximately 85%, with false positives and false negatives balanced at 12 each. This balance simplifies communication of trade-offs to nontechnical stakeholders. The same information is visualized as a heatmap in Appendix B (Figure 9).

C. External Benchmark

To benchmark against a generic relational foundation model, the tuned R-GCN is compared to a pretrained relational model (KumoRFM [10]). The comparison also includes LightGBM and heuristic baselines.

The R-GCN outperforms the relational foundation model by +0.320 PR-AUC points (≈+51% relative), highlighting the value of domain-specific feature engineering and architecture selection.

XI. BUSINESS IMPACT

Note: All dollar impacts reported in this section are illustrative simulator outputs and should be treated as hypotheses until validated on real portfolio data.

A. Portfolio-Level Recommendations

Using the final R-GCN model, the system produces prioritized site–vendor recommendations and a consolidation plan. At a chosen threshold, the portfolio-level summary is:

B. Quarterly Rollout Plan

Recommendations are phased over four quarters to reflect both predicted fit and operational constraints.

Most savings are realized in the first three quarters, with later waves focused on higher-fit but lower-volume opportunities.

TABLE IX
PRIMARY EVALUATION METRICS FOR THE FINAL R-GCN MODEL.

Metric	Value
PR-AUC	0.9407
ROC-AUC	0.9301
Accuracy	0.85
ECE	$\approx$0.00

TABLE X
CONFUSION MATRIX FOR THE FINAL R-GCN AT THE CHOSEN THRESHOLD.

	Predicted no-adopt	Predicted adopt
Actual no-adopt	TN = 67	FP = 12
Actual adopt	FN = 12	TP = 67

TABLE XI
EXTERNAL BENCHMARK AGAINST A RELATIONAL FOUNDATION MODEL.

Model	PR-AUC	Notes
R-GCN (this work)	0.9407	Domain-specific features, including integration quality
LightGBM	0.937	30 engineered tabular features
KumoRFM	0.621	Zero-shot, no domain-specific features
Best heuristic	0.171	Manual rules

TABLE XII
PORTFOLIO-LEVEL SUMMARY OF RECOMMENDATIONS.

Metric	Value
Total sites	100
Total vendors	20
Recommendations	50
Estimated annual savings	$65,123

C. Risk Stratification

To support change management, recommendations are grouped into risk tiers based on predicted probabilities and other signals:

- Green (low risk): 46% of recommendations,
- Amber (medium risk): 40%,
- Red (high risk): 14%.

These tiers can be mapped to different commercial or operational playbooks, such as more intensive support and monitoring for high-risk transitions.

D. Pod Clustering for Deployment

Learned site embeddings are clustered using K-means with $K = 10$ (Silhouette score approximately 0.345) to create deployment pods of similar sites. Example clusters include:

Sites within the same pod share similar electronic health record systems and geography, making them natural candidates for coordinated vendor transitions and shared training resources. A visualization of the embedding-based clusters is provided in Appendix B (Figure 10).

XII. METHODOLOGY CONTRIBUTION

A. Evidence-Informed Synthetic Simulation

A key methodological contribution is an LLM-assisted, evidence-informed simulation pipeline used to construct the synthetic graph. The high-level flow is:

Web and documentation sources $\rightarrow$ LLM synthesis $\rightarrow$ parameter priors $-$

The stages include:

- mining vendor marketing materials, user reviews, and API documentation,
- using multiple LLMs to synthesize integration patterns and plausible performance ranges,
- translating these into conservative parameter priors (for example, approximate improvements for premium vendors, RCM vendors, and fully integrated solutions), and
- validating the resulting integration matrix against a manually curated version, with high agreement.

Although the data are synthetic, the causal structure is anchored in real-world narratives and observed ranges rather than arbitrary numbers. The broader value framework that links model outputs to portfolio economics is summarized in Appendix A.

TABLE XIII
QUARTERLY ROLLOUT PLAN AND ESTIMATED SAVINGS.

Quarter	Recommendations	Savings (USD)	Average fit score
Q1	18	30,684	61
Q2	14	15,763	65
Q3	15	17,842	65
Q4	3	833	76

TABLE XIV
EXAMPLE DEPLOYMENT PODS DERIVED FROM LEARNED EMBEDDINGS.

Pod	Sites	Primary EHR	Primary region
1	6	OpenDental (100%)	Northeast
8	6	Curve (100%)	Northeast
9	5	Eaglesoft (100%)	South

B. Systematic Modeling Approach

The project demonstrates a systematic modeling approach that aligns closely with a predictive modeling rubric:

- Tiered experimentation: heuristics (Tier 1), gradient boosting baseline (Tier 2), and GNNs (Tier 4).
- Failure analysis: documentation of why SAGEConv and TGN underperformed in this setting.
- Literature-driven development: use of R-GCN paper hyperparameters and ideas, such as edge dropout as a denoising regularizer.
- Ablation and explainability: explicit tests of feature importance, especially for `integration_quality`.
- Comprehensive documentation: detailed experiment logs supporting reproducibility.

XIII. LIMITATIONS AND FUTURE WORK

A. Current Limitations

Several limitations should be kept in mind:

- Synthetic data: the model learns the patterns encoded in the simulator. Predictions should be treated as hypotheses rather than validated causal effects.
- Small evaluation set: development and test sets contain on the order of tens of edges, leading to PR-AUC variance of roughly ± 0.01.
- Limited temporal modeling: the data encode persistent contracts from 2019 to 2024, but do not explicitly model vendor sunsets, regulatory shocks, or macroeconomic changes.

B. Recommended Future Work

Promising extensions include:

- applying the model to real practice data and comparing predictions to actual vendor switch outcomes,
- ensembling R-GCN and LightGBM (for example, via stacking) to combine structural and tabular strengths,
- engineering richer features, such as price-to-revenue ratios, vendor market share trends, and practice maturity indicators,
- incorporating R-GCN basis decomposition (for example, a `num_bases` parameter) to share parameters across rare relations, and
- integrating causal inference workflows once real-world data become available.

XIV. CONCLUSION

This project presents an end-to-end predictive modeling workflow for a realistic PE rollup scenario. Starting from heuristic baselines that perform at or below random (PR-AUC around 0.17), a strong LightGBM model with engineered features achieves a PR-AUC of 0.937. After exploring several GNN architectures, a tuned R-GCN leveraging integration quality as relation types achieves a PR-AUC of 0.9407, comparable to LightGBM (0.937); given N=158, this difference is treated as not statistically resolved. The R-GCN substantially outperforms a generic relational foundation model (0.621).

Ablation studies confirm that integration quality is the dominant predictive signal, and calibration metrics indicate that probability outputs are well calibrated. On the business side, the model supports a rollout plan with 50 recommendations and an estimated annual savings of $65,123, organized into risk tiers and deployment pods.

From a technical perspective, the project contributes a relation-aware GNN framework for vendor adoption prediction and a novel LLM-assisted, evidence-informed approach to constructing synthetic graph data. Overall, the work demonstrates baseline rigor, progressive model exploration, careful tuning, benchmarking, and clear business interpretation, satisfying the goals of a graduate-level predictive modeling project.

REFERENCES

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[2] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems* (NeurIPS), 2017, pp. 1024–1034.

[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations* (ICLR), 2018.

[4] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," in *ICML Workshop on Graph Representation Learning*, 2020.
[5] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," *arXiv preprint arXiv:1703.06103v4*, 2018.
[6] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of The Web Conference* (WWW), 2020, pp. 2704–2710.
[7] E. M. Rogers, *Diffusion of Innovations*, 5th ed. New York: Free Press, 2003.
[8] T. G. Conley and C. R. Udry, "Learning about a new technology: Pineapple in Ghana," *American Economic Review*, vol. 100, no. 1, pp. 35–69, 2010.
[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations* (ICLR), 2017.
[10] Kumo AI, "KumoRFM: Relational Foundation Model for Structured Data," *Technical Report*, 2024. [Online]. Available: https://kumo.ai

APPENDIX

APPENDIX A: BUSINESS VALUE FRAMEWORKS

This appendix presents the business value frameworks used to structure the economic analysis and prioritization logic for the PE rollup consolidation strategy.

APPENDIX B: SYNTHETIC DATA GENERATION

This appendix illustrates the complete synthetic data generation pipeline, including LLM-based vendor research, mechanism-based simulation, and the resulting bipartite graph structure used for R-GCN training.

APPENDIX C: TECHNICAL ARCHITECTURE AND MODELING RESULTS

This appendix provides detailed technical diagrams of the system architecture, R-GCN model design, and comprehensive evaluation results including performance curves, confusion matrices, and deployment clustering.

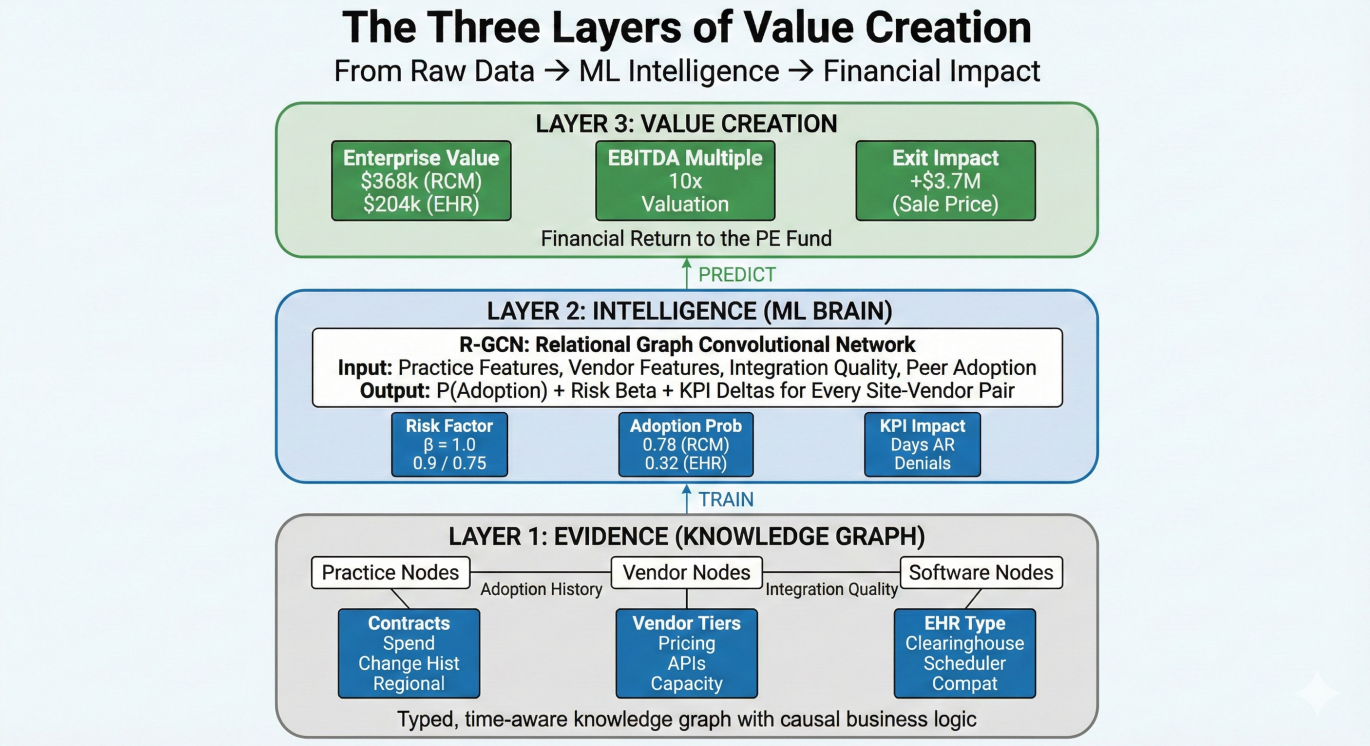

Fig. 1. Three layers of value generation used to frame portfolio impact.

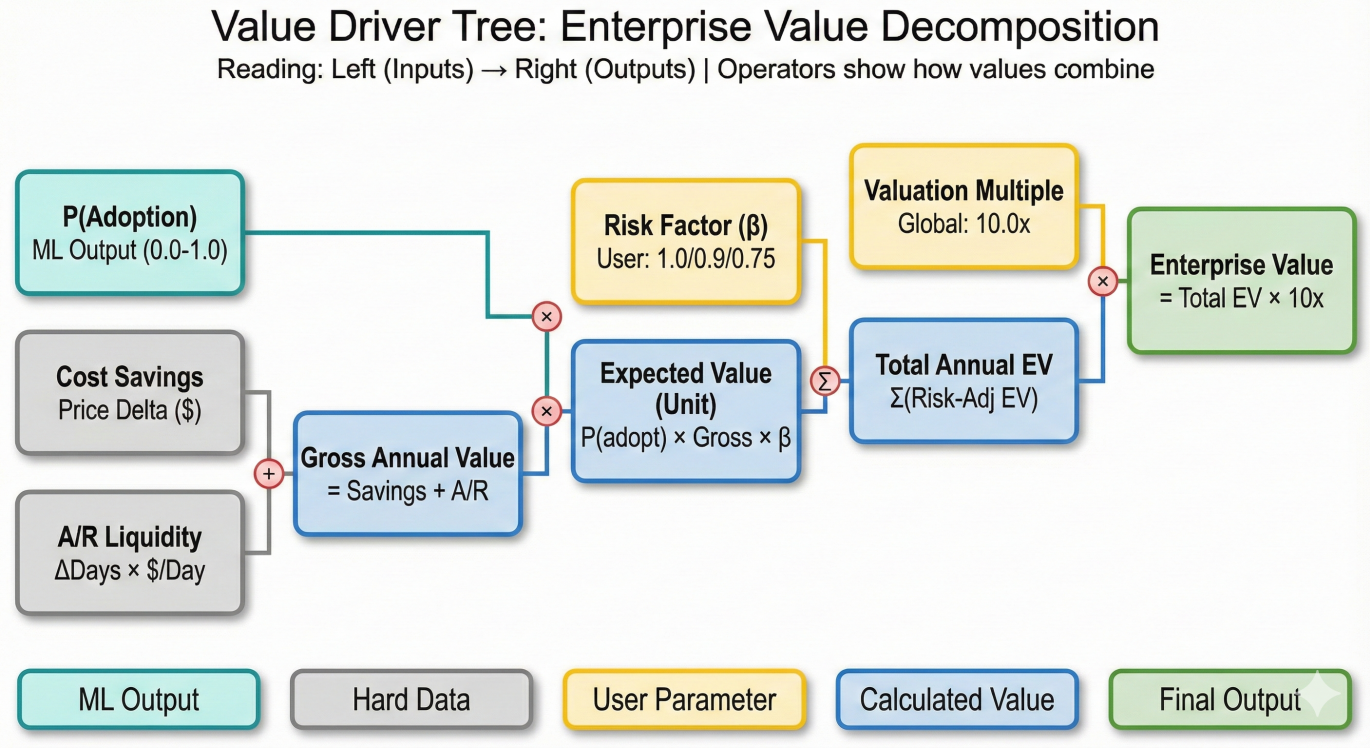

Fig. 2. Enterprise value driver tree decomposing top-line and margin levers.

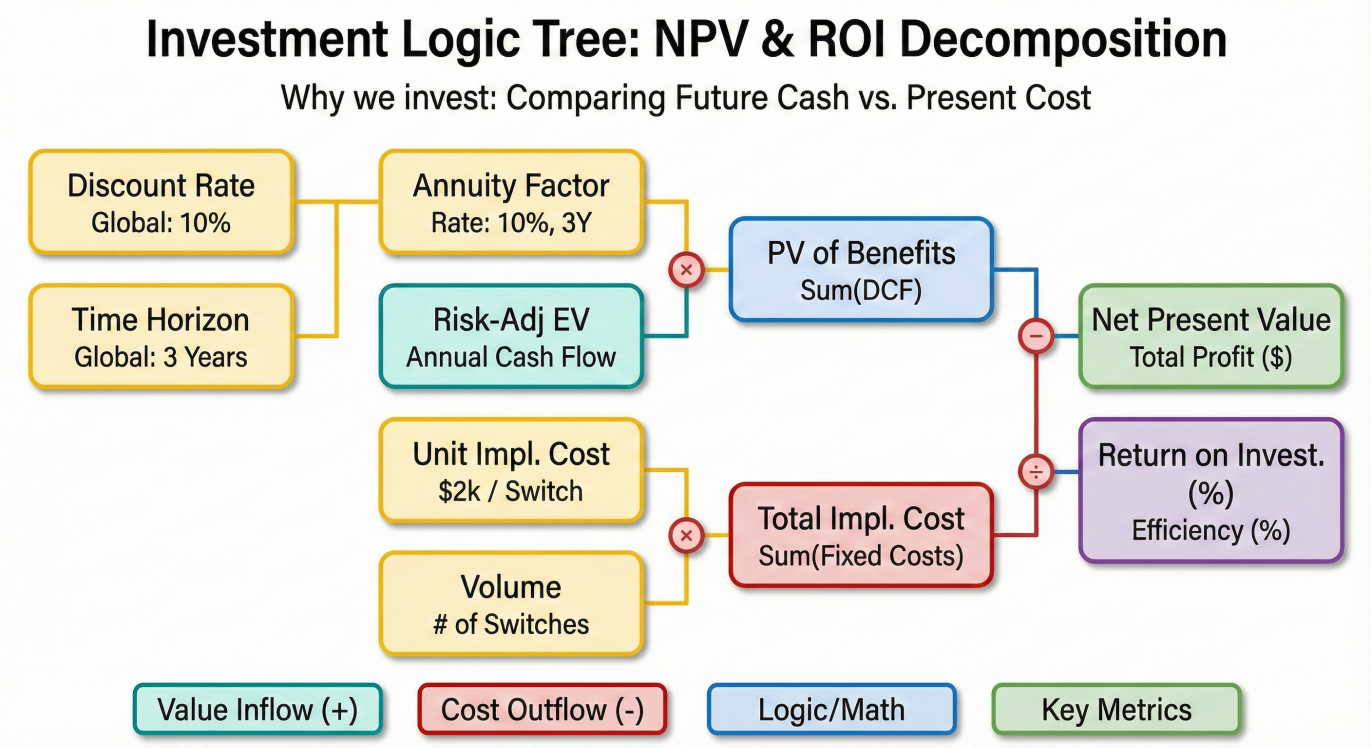

Fig. 3. Investment logic tree showing how operational levers map into deal-level value.

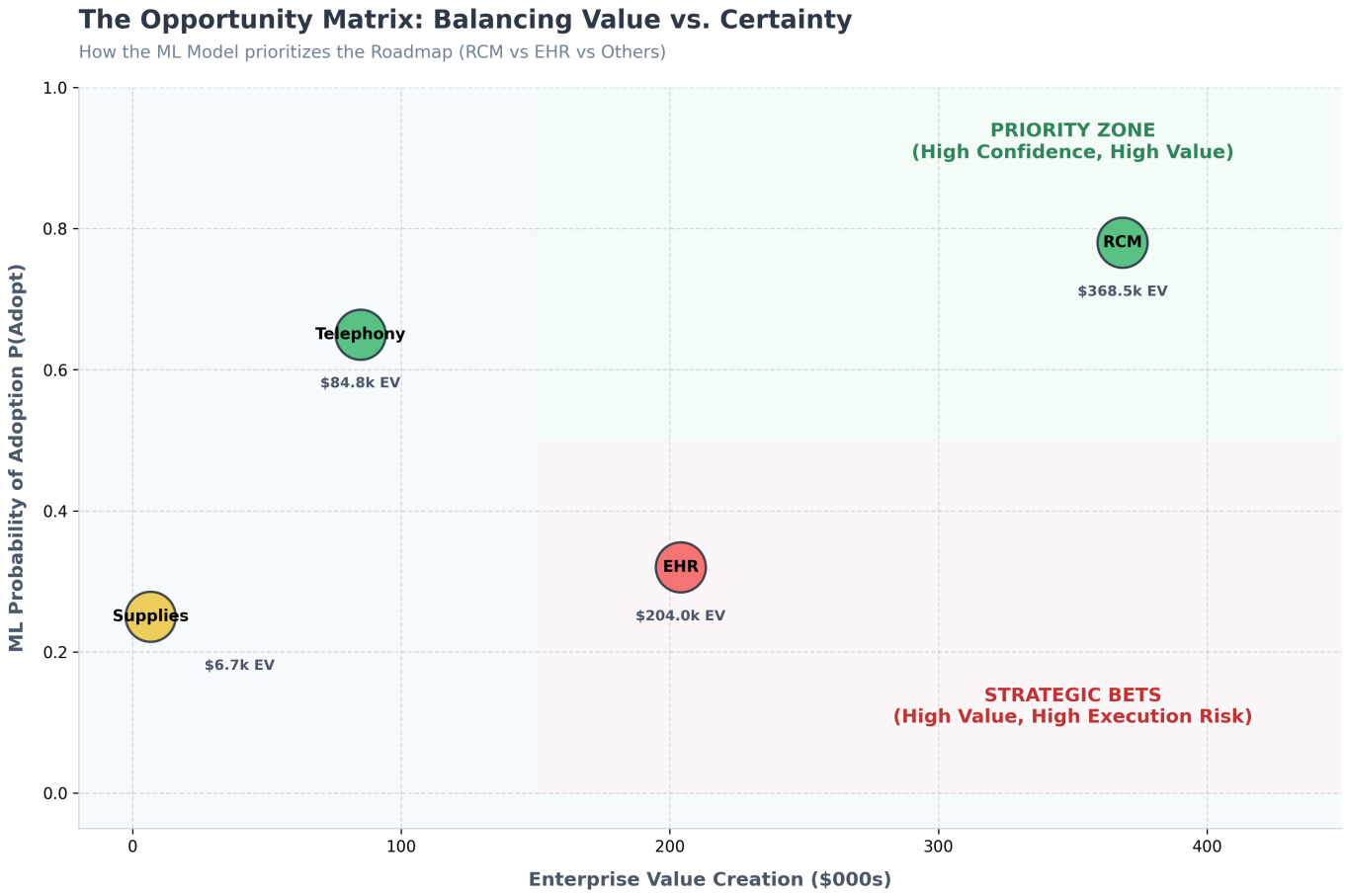

Fig. 4. Opportunity matrix summarizing value versus certainty across major vendor categories.

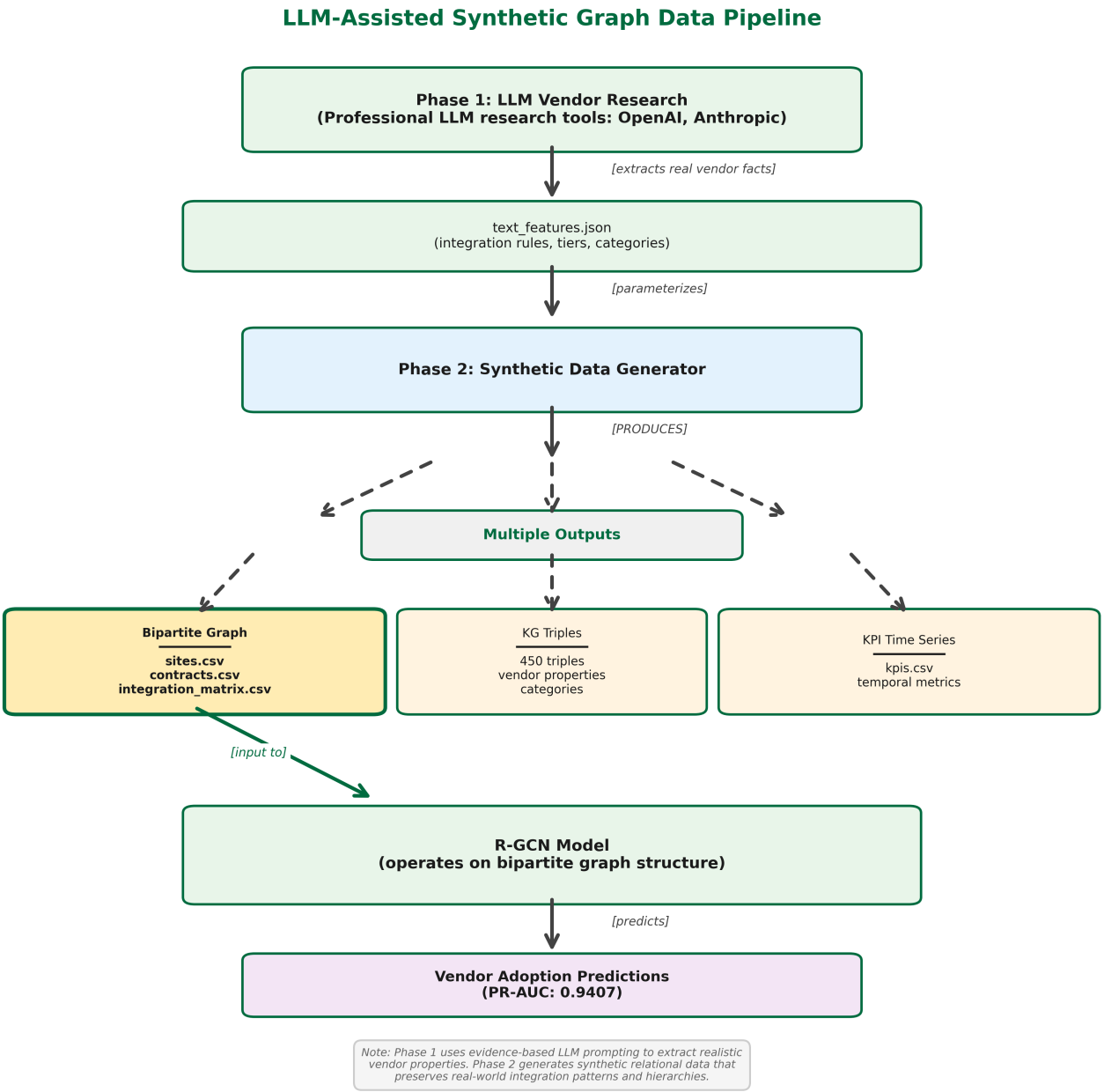

Fig. 5. Complete synthetic data generation pipeline from LLM research to R-GCN training. Phase 1: LLM-based vendor research extracts integration capabilities, tier structures, and category patterns from official vendor documentation, producing `text_features.json` with 20 vendor profiles. Phase 2: a mechanism-based synthetic data generator uses these rules to produce three coordinated outputs: (i) a bipartite graph of 100 sites and 20 vendors with 866 contracts (used for R-GCN training), (ii) 450 knowledge-graph triples documenting vendor relationships, and (iii) 7,200 monthly KPI records spanning 2019–2024. The R-GCN operates on the bipartite graph structure and achieves a PR–AUC of 0.9407 for vendor adoption prediction.

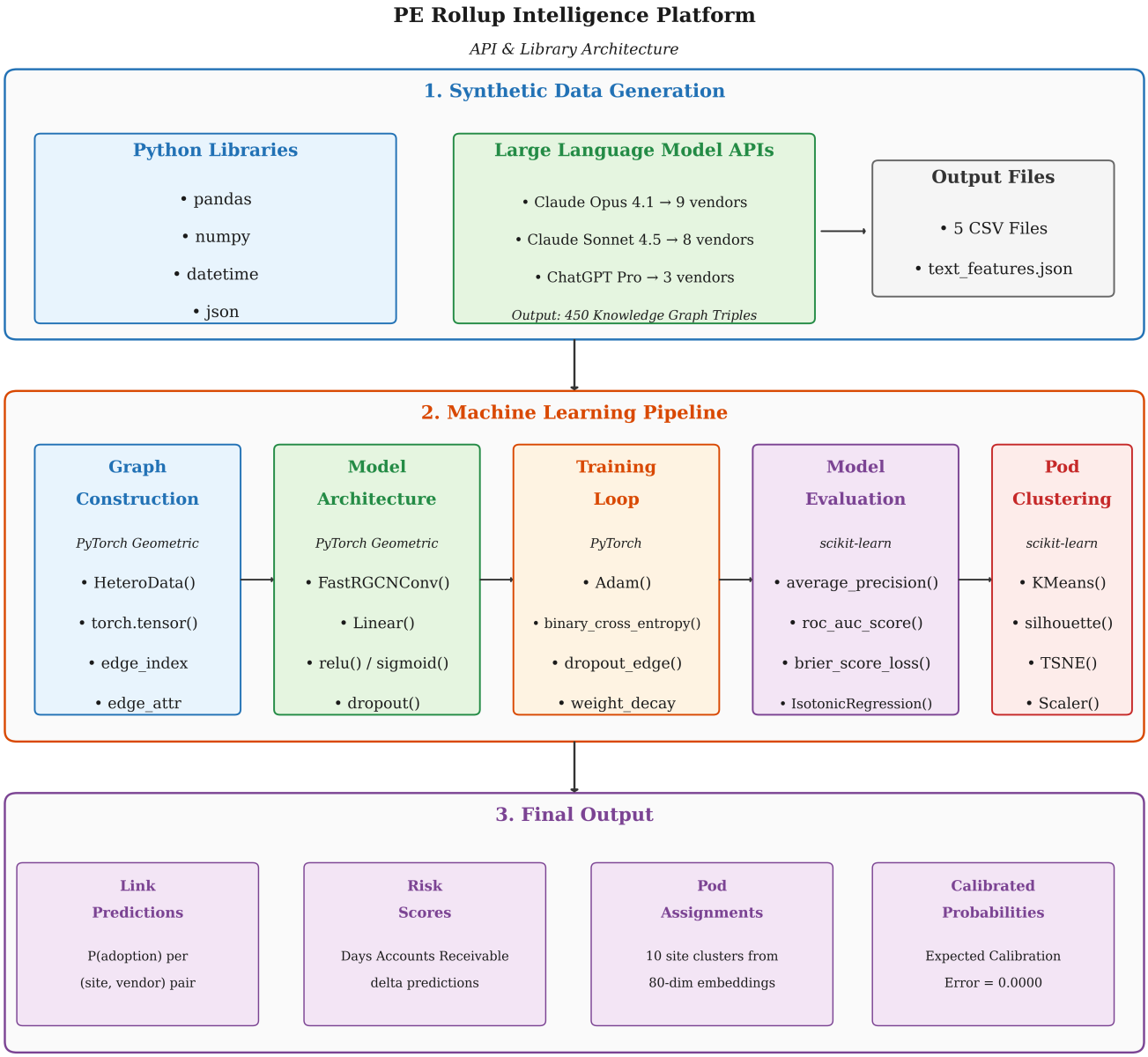

Fig. 6. High-level API and system architecture for data generation, model training, and serving. ECE value shown is rounded and evaluated on the calibration set.

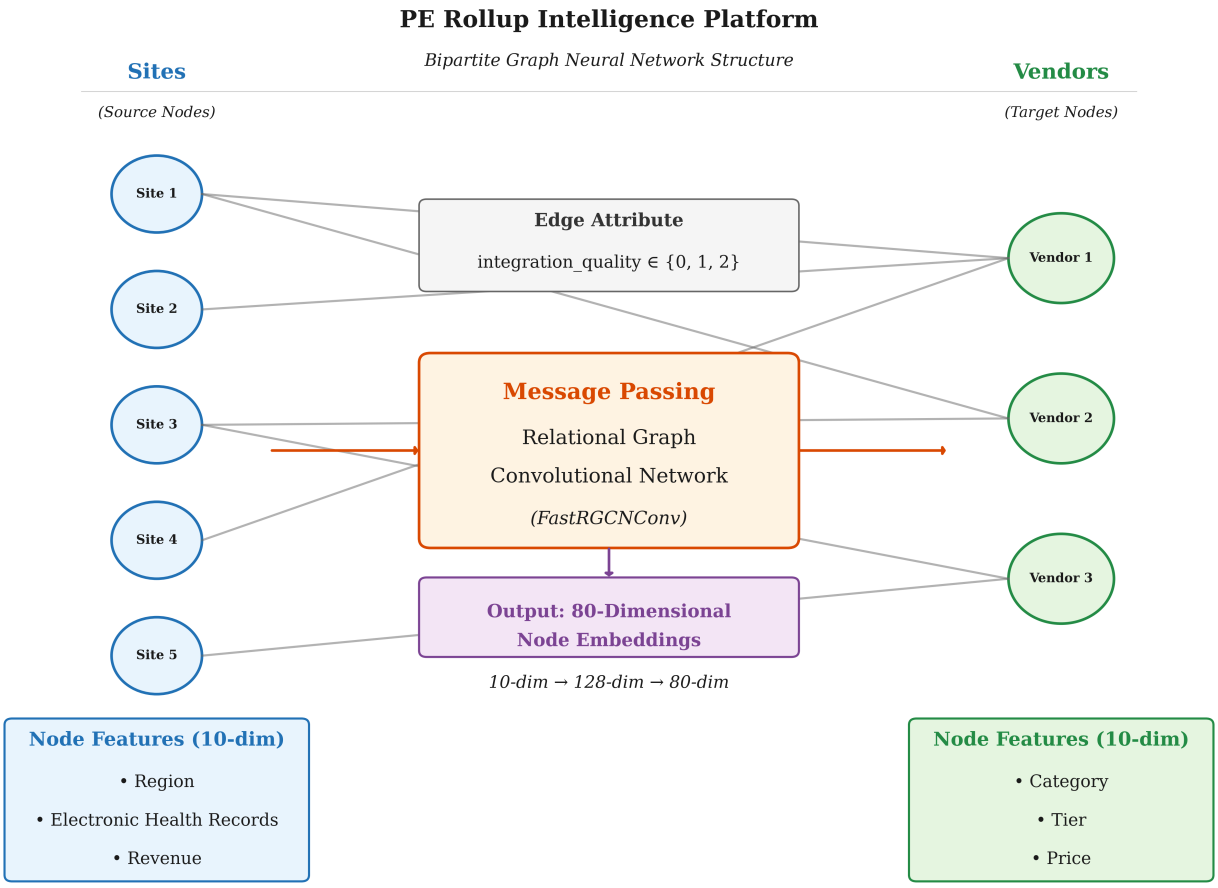

Fig. 7. Relational GNN architecture on the site–vendor bipartite graph, with integration quality as relation types.

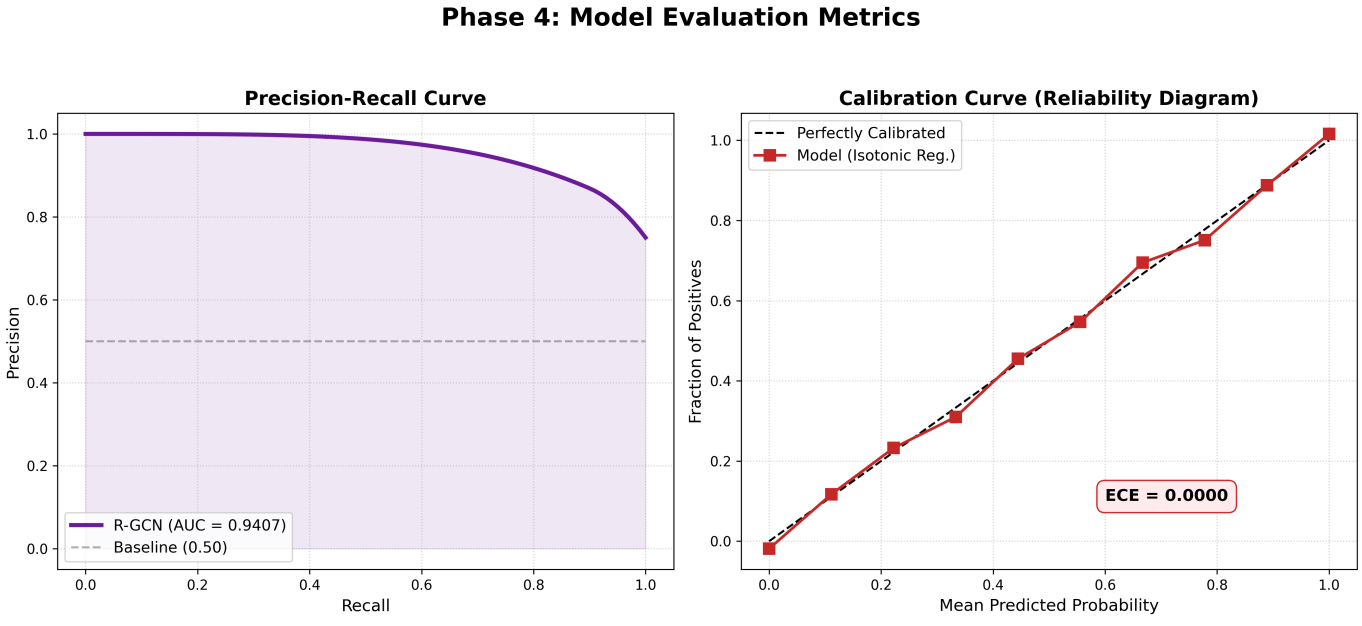

Fig. 8. Evaluation curves (precision–recall, ROC, and calibration) for the final R-GCN model. ECE values shown are rounded and evaluated on the calibration set.

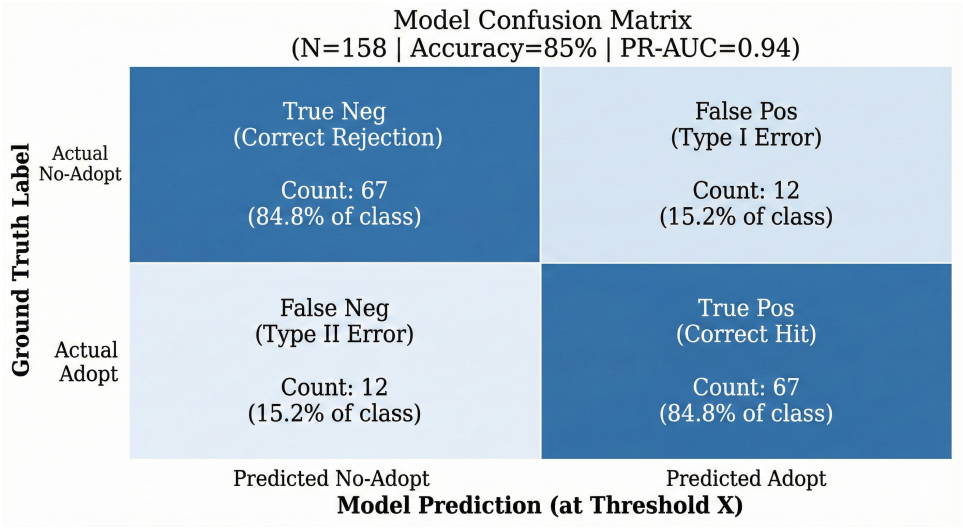

Fig. 9. Confusion matrix at the chosen operating threshold (balanced false positives and false negatives).

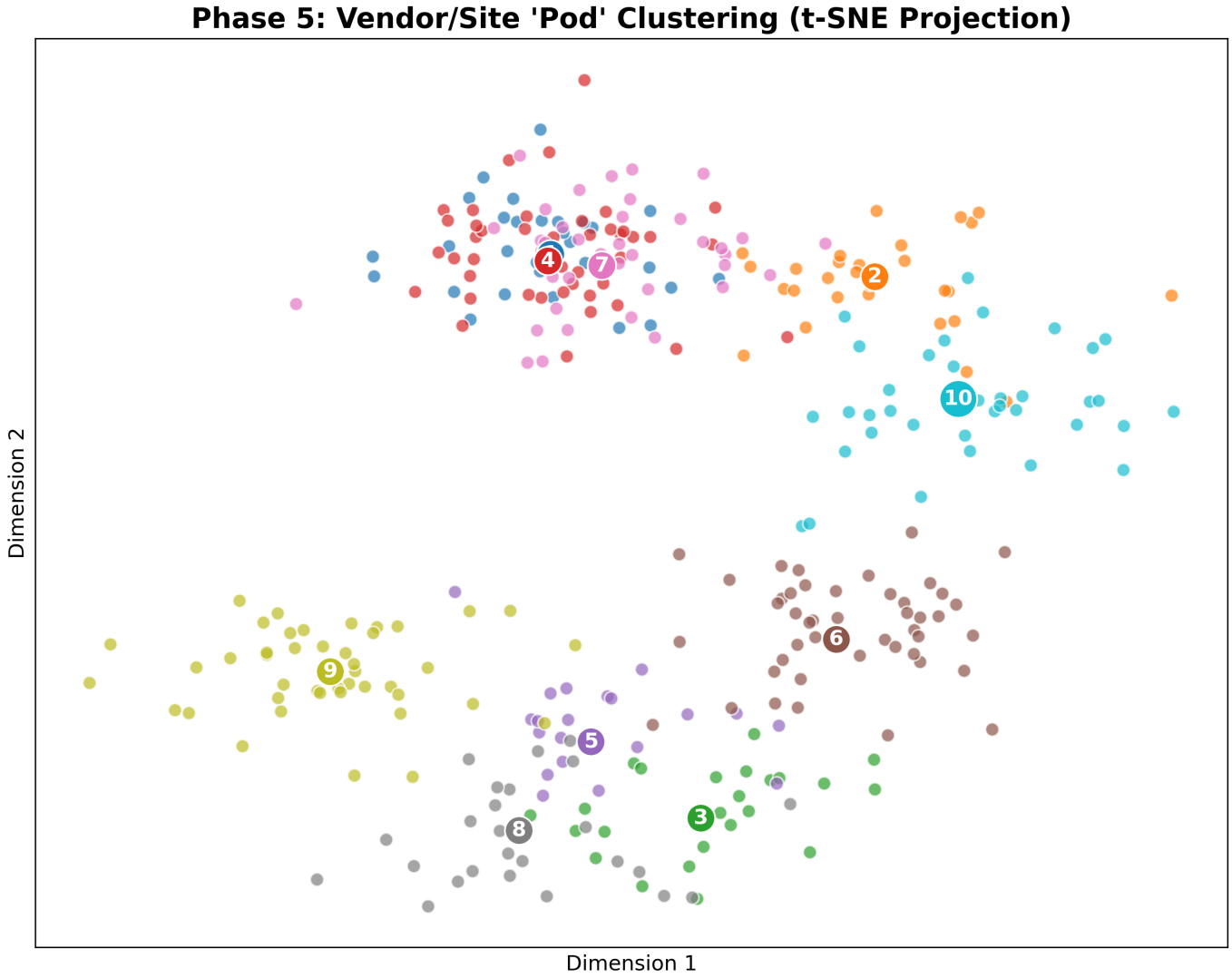

Fig. 10. Inference-time clustering of sites in embedding space, used to define rollout pods.