GRAIL: A Benchmark for GRaph ActIve Learning on Longitudinal Sensing Data

Anonymous authors
Paper under double-blind review

Abstract

Modern embedded and wearable sensing systems continuously collect high-dimensional data for health and behavior monitoring, yet obtaining high-quality labels remains costly and resource-intensive. Graph-based Active Learning (Graph-AL) offers an efficient solution by exploiting network relationships among sensors or users to prioritize the most informative data for labeling, reducing energy use and user burden. However, existing Graph-AL evaluations rely on one-shot experiments over static citation-style graphs and overlook the temporal variability, repeated-querying dynamics, and per-user sampling burden that characterize real-world longitudinal sensing deployments. To bridge this gap, we introduce GRAIL, a benchmarking framework that evaluates Graph-AL strategies on longitudinal sensing data, where node features and labels evolve daily on a fixed underlying social graph. GRAIL is architected to also handle evolving graph topologies, but our two contributed real-world datasets (SNAPSHOT and Friends-and-Family) exercise the longitudinal-features, static-topology setting due to the scarcity of public datasets with both rich sensing features and evolving social ties. GRAIL introduces metrics to assess sustained effectiveness, sampling diversity, and per-user query burden, allowing systematic comparison of AL methods under realistic conditions. Experiments across 11 AL strategies on real-world data reveal clear trade-offs between predictive performance and sensing cost, identify hybrid strategies (AGE, GraphPartFar) as Pareto-optimal across two markedly different topologies, and quantify the conditions under which classical Graph-AL methods fail to outperform random sampling.

1 Introduction

The increasing availability of sensor-rich, real-world datasets from mobile and wearable devices has enabled longitudinal modeling of user behavior, health, and well-being. These environments produce high-dimensional, time-varying data streams, often embedded in complex social or sensor networks, where traditional machine learning methods struggle to balance predictive performance with the cost of continuous data labeling. A similar challenge applies to longitudinal networked environments, such as social networks, sensor networks, or Internet of Things (IoT) applications, where the continuous operation and data demands on individual nodes (e.g., users, sensors, or devices) can lead to resource depletion, sensor battery drain, or data overload, impacting overall system performance and reliability. Active Learning (AL) offers a compelling solution by selectively querying the most informative instances for labeling.

In network/graph-based settings, graph active learning(Graph-AL) can further address this challenge by leveraging the structure of user networks to selectively query only a subset of users or sensors for data collection. This approach typically integrates two main components: AL strategies and graph-embedding techniques, balancing performance metrics through an optimization function such as:

$$x^* = \arg\min_x \mathcal{L}(x) \tag{1}$$

$$\mathcal{L} = f(U, C, N) \tag{2}$$

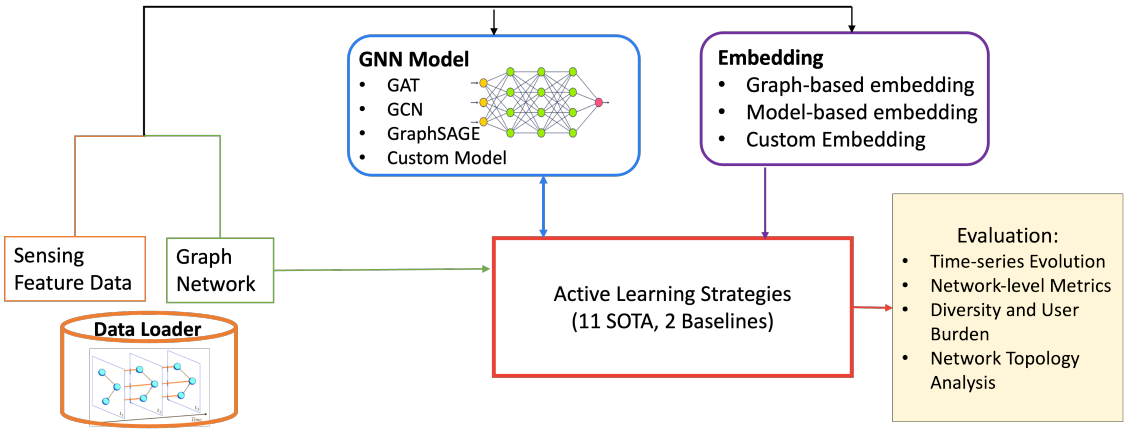

Figure 1: GRAIL Framework for Graph Active Learning on Longitudinal Sensing Data

where x is the sample to query and $\mathcal{L}$ is a loss function based on a combination of metrics: uncertainty U, clustering C, and network metrics N. Uncertainty is often measured using entropy, mutual information, or prediction confidence, guiding the model to query points with the highest ambiguity (Nair & Wen, 2020; Cai et al., 2017; Moore et al., 2011). Clustering metrics leverage the graph structure to identify representative nodes, using techniques such as clustering coefficient, K-Medoids clustering, and modularity clustering (Nair & Wen, 2020; Wu et al., 2019; Ma et al., 2022; Bilgic et al., 2010). Network metrics like degree centrality and pagerank centrality prioritize influential nodes, optimizing model performance even with limited labeled data (Nair & Wen, 2020; Cai et al., 2017). However, in longitudinal studies or real-time monitoring scenarios, the application of graph-based AL poses unique and largely unaddressed challenges.

Despite their contributions, existing evaluations of these approaches exhibit significant limitations. First, they focus on static snapshots and assume fixed feature and label distributions, i.e. node attributes and labels remain unchanged throughout the learning process. Popular Graph-AL studies overwhelmingly rely on static benchmark datasets such as Cora, Citeseer, PubMed, Coauthor-CS, Coauthor-Physics, Amazon-Photo, Amazon-Computers, and ogbn-arxiv(Nair & Wen, 2020; Cai et al., 2017; Moore et al., 2011). These datasets, while valuable for benchmarking node classification tasks, assume fixed graph topologies and node attributes where neither node features nor labels evolve over time. As a result, evaluations are typically one-shot and fail to capture the complexities of temporal learning, re-querying behavior, or real-time generalization.

This static evaluation paradigm poses a major limitation for applying AL in real-world longitudinal sensing settings such as health monitoring, IoT systems, or mobile behavior prediction, domains where data streams arrive continuously and node features and labels evolve over time, even when the underlying network topology changes only slowly relative to the sensing cadence. Moreover, most existing work prioritizes prediction accuracy and overlooks user-centric constraints such as the cost of repeatedly querying individuals, user dropout risk, or the uneven sampling that may arise due to graph topology biases (e.g., high-centrality nodes being queried disproportionately).

In contrast, real-life in-the-wild sensing data, collected from mobile phones, wearables, or social interactions, introduces several practical challenges: missing data, irregular sampling, noisy labels, heterogeneous feature distributions, and privacy constraints. These factors compound the difficulty of evaluating and deploying graph AL strategies effectively. Although several recent works develop graph-AL methods for specific dynamic application domains – including patient-specific seizure prediction (Li et al., 2022), structural-uncertainty-based AL (Wu et al., 2021), and active learning under preemptive query generation (Regol et al., 2020), with a recent survey of dynamic-data Graph-AL provided in (Zhang et al., 2022) – these works each address a single application or theoretical aspect, and no standard benchmark currently exists to compare AL strategies head-to-head under longitudinal, noisy, and user-burden-sensitive conditions, leaving a critical gap in the literature.

To address these limitations, we introduce GRAIL, a systematic benchmarking framework for graph-based active learning on longitudinal sensing data, i.e. stream-like settings where node features and labels evolve from one timestep to the next on a fixed underlying social graph. While GRAIL's framework supports evolving graph topologies, the two real-world datasets we contribute exercise only the longitudinal-features, static-topology setting; this reflects the current state of available public sensing datasets, which rarely include both rich multi-modal features and a longitudinal record of evolving social ties. We view the longitudinal-features, static-topology regime as a realistic and underexplored first step, and discuss the implications of this scope choice in Section 8.

To complement the framework, we also contribute two real-world longitudinal graph datasets, Friends and Family (FnF)(Aharony et al., 2011) and SNAPSHOT(Khalid & Sano, 2023), that include rich multi-modal sensing features, longitudinal ground truth labels, and explicit social network graphs. While publicly accessible, these datasets were not previously formatted for graph active learning. We perform substantial preprocessing including dynamic label alignment, graph-based feature extraction, and time-series formatting, to make them suitable for AL evaluation in real-life settings. This enables the community to go beyond academic benchmarks and test algorithms in realistic, noisy, user-facing environments, where concerns like query fatigue, temporal generalization, and per-user sampling-rate equity are crucial for real-world impact.

The benchmark introduces novel metrics, such as the Cumulative Performance Index (CPI), which evaluates the model's sustained effectiveness over time, and sampling diversity metrics that track the distribution of queries across users, helping to identify patterns of over-exertion on high-centrality nodes. By examining each strategy's robustness across various node types, from newly queried nodes to long-held-out test sets, the framework provides a holistic view of AL strategy performance in graph-based, health monitoring or IoT applications.

The benchmarking framework evaluates 11 AL strategies along three key dimensions:

- Performance Metrics: Including accuracy, Area Under the Curve (AUC), and CPI to quantify the predictive capabilities of each strategy over time.
- Diversity and Burden Metrics: Measuring the degree of query diversity and potential user fatigue alongside time-gap analyses that reveal how frequently individual users are queried.
- Network-Level Analysis: Examining the influence of network topology on strategy effectiveness, particularly focusing on centrality and exertion patterns across users.

Findings from these experiments reveal critical insights into the trade-offs AL strategies must manage between prediction performance and user burden. They also highlight key limitations in existing approaches, inspiring future development of AL methods that are sustainable, efficient, and mindful of the user experience in longitudinal health and well-being monitoring systems. By providing a structured, reproducible evaluation framework, this benchmark serves as a foundational resource for advancing AL strategies across multiple domains and datasets, supporting the development of adaptable, real-time applications. The code and processed datasets are available in an anonymous OSF repository(Anonymous, 2025).

2 GRAIL: Graph-Based Active Learning Framework for Longitudinal Sensing Data

This section presents a high-level overview of the GRAIL framework, outlining its components and design motivations to guide conceptual understanding and act as a user guide for its usage. GRAIL's core is composed of five flexible components: Data Loader, GNN Models, Embedding module, AL Strategies, and multi-dimensional Evaluation module. The framework is shown in Figure. 1.

- **Data Loader** The data loader provides options to load processed data, including features and graphs, for multiple datasets with various labels and graph types (e.g., call, SMS, friend networks). Examples include SNAPSHOT and FnF (with labels such as mood or sleep).
- **Model Options**: GRAIL supports various Graph Neural Network (GNN) architectures, including GAT (Graph Attention Network)(Veličković et al., 2018), SAGE GNN (GraphSAGE)(Hamilton

et al., 2017), GCN (Graph Convolutional Network)(Kipf & Welling, 2017), and allows for Custom Models. This adaptability accommodates different data and task requirements, allowing users to tailor the framework to their needs.

- **Active Learning (AL) Strategy Options**: GRAIL includes 11 AL strategies listed in Table 1 for selectively querying nodes which can be broadly categorized into four main groups: (1) Uncertainty-based sampling techniques(Settles & Craven, 2008; Nair & Wen, 2020); (2) Graph-structure-based sampling methods, including Degree-Based and PageRank-Based sampling (Nair & Wen, 2020; Cai et al., 2017); (3) Embedding-space clustering methods such as dentisy-based, coreset, FeatProp(Cai et al., 2017; Sener & Savarese, 2017; Wu et al., 2019) and (4) Hybrid strategies that combine graph structure, input features, and model outputs, such as GraphPart, GraphPartFar, and Active Graph Embedding (AGE) (Ma et al., 2022; Cai et al., 2017). Users can also implement Custom Strategies to evaluate novel methods suited to specific contexts.

Table 1: Summary of Active Learning Strategies. Families: *Base* = baseline; *Unc* = uncertainty-only (no graph); *Graph* = graph-structure-only; *Embed* = embedding-space clustering (graph + features); *Hybrid* = partitioning + hybrid (graph + features + model).

Strategy	**Family**	**Description**
No Active Learning	Base	Trained on first L days only; never updated.
Random Sampling	Base	Uniform-random node selection.
Entropy (Settles & Craven, 2008)	Unc	Highest predictive entropy.
Least Confidence (Nair & Wen, 2020)	Unc	Lowest top-class probability.
Max Margin (Nair & Wen, 2020)	Unc	Smallest gap between top-2 probabilities.
Degree (Nair & Wen, 2020)	Graph	Highest degree centrality.
PageRank (Cai et al., 2017)	Graph	Highest PageRank.
Density (Cai et al., 2017)	Embed	Clusters in embedding space; selects nodes near cluster centers.
Coreset (Sener & Savarese, 2017)	Embed	K-center clustering for embedding-space coverage.
FeatProp (Wu et al., 2019)	Embed	Propagates features; selects nodes minimizing classification-error bound.
GraphPart (Ma et al., 2022)	Hybrid	Modularity partition + K-medoids representatives.
GraphPartFar (Ma et al., 2022)	Hybrid	GraphPart + far-from-previously-selected penalty.
AGE (Cai et al., 2017)	Hybrid	Clustering + centrality + predictive uncertainty.

- **Embedding Options**: For embedding computation, GRAIL offers two distinct approaches:
  - Model-based: Utilizing the trained GNN model to generate embeddings, leveraging learned representations from the trained model for more integrated, task-specific embeddings.
  - Direct GNN: Applying a separate GNN model to pass data and graph structure through its layers, obtaining embeddings independent of the trained model, thus allowing flexibility in embedding generation separate from specific model tasks or training phases.
- **Evaluation Module**: The GRAIL framework features a comprehensive evaluation pipeline, assessing AL strategies along three critical dimensions: Model Performance, Sampling Analysis, and Network Analysis.
  - **Model Performance (Section 3)**: This component includes metrics such as Accuracy, F1 Score, AUC, and Cumulative Performance Index (CPI), which assess the predictive strength of the model across dynamic, real-world conditions. In addition to the custom metrics like CPI, users can select from up to seven metrics: accuracy, micro F1, macro F1, recall, precision, Area Under the ROC Curve (AUC-ROC), and Area Under the Precision-Recall Curve (AUC-PR).
  - **Sampling Analysis (Section 4)**: GRAIL evaluates the diversity and distribution of queries with metrics like Diversity Metrics (measuring range and query balance), Coverage Ratio (extent of node sampling), Time Distribution (frequency of queries over time), and Burden Metrics (assessing user query fatigue).
  - **Network Analysis (Section 4)**: To assess how AL strategies perform across node types, GRAIL includes Node Types analysis (e.g., high-centrality vs. peripheral nodes), Time-Gap

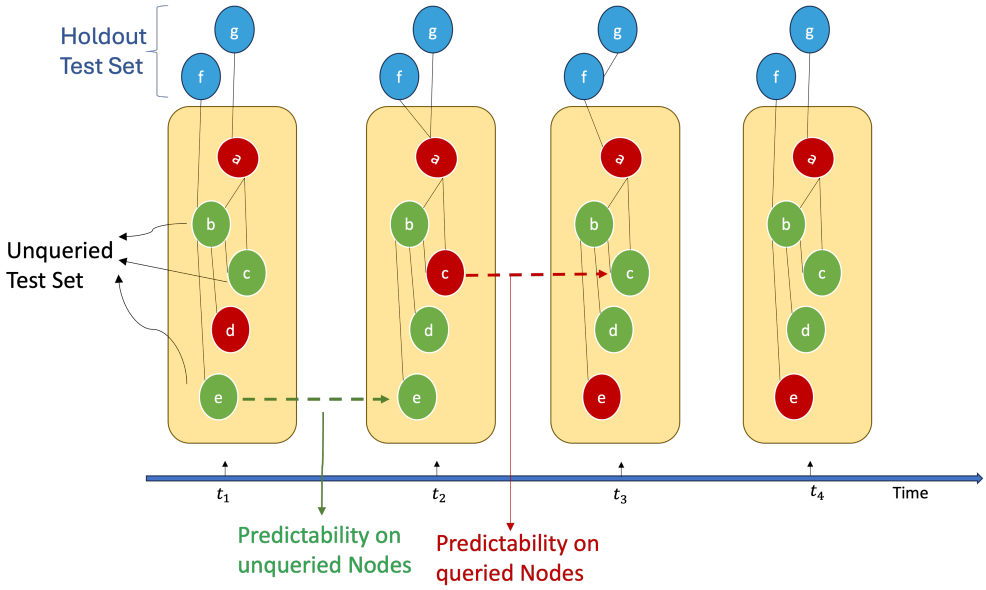

Figure 2: Evaluation Strategy overview of how nodes are categorized and divided over time. Timepoints represent instances where the network state is assessed and nodes are queried. Blue nodes are part of the holdout set that remains unseen by the model. The yellow box indicates the query pool. Red nodes represent queried nodes, while green nodes are unqueried.

Analysis (intervals between repeated queries on nodes), Centrality Impact (influence of high-centrality nodes on sampling), and Graph Properties (such as density and clustering, affecting AL effectiveness).

3 Network-level Performance Evaluation for Longitudinal Graph Sensing

This section outlines the strategy and metrics to evaluate Graph AL strategies at a network-level and distinguishing between different types of nodes.

3.1 Categorizing Nodes for Evaluation

Given the model's reliance on spatial (graph) and temporal (time-series) components, an effective evaluation strategy is necessary to differentiate the model's predictive performance across various node categories. This ensures a comprehensive understanding of how the model generalizes beyond its training set, particularly when predicting outcomes for unqueried or unseen nodes. The experiment is designed to answer questions such as: How well does the model perform on unqueried nodes when k nodes have been queried from the query pool? The node categories for each query timepoint (e.g., each day) in the time series are explained below and in Figure 2:

- **Train Nodes**: These nodes are used for training the model on each day of the time series. Metrics are computed based on the model's predictions for these nodes, providing insight into how well the model is fitting the training data.
- **Unqueried Nodes**: These nodes are excluded from the training process on each specific day. The model's predictions on these unqueried nodes help assess how well it generalizes to unseen data.
- **Same-Day Unqueried Nodes**: These are the nodes that are not used for training on the same day but are included for predictions on subsequent days. This category provides insight into the model's performance on unqueried nodes but is later predicted in the next step.

- **Holdout Test Set**: This set consists of nodes that are never used for training or querying throughout the experiment. Metrics from the holdout set provide a benchmark for the model's ability to predict on an entirely unseen data, showcasing its overall generalization capability and robustness.

Dividing nodes into these four categories is crucial for stream AL as it provides a comprehensive view of the model's performance under varied data exposure. Distinguishing between queried and unqueried nodes helps evaluate the effectiveness of AL strategies where only a subset of nodes is queried. This analysis reveals how well the model can leverage limited labeled data to make accurate predictions on unseen nodes.

Assessing same-day unqueried nodes highlights the model's immediate predictive capabilities and adaptability to changing data. Evaluating the holdout test set ensures that the model's performance generalizes to entirely unseen data.

3.2 Performance Metrics

The following metrics are generated during each bootstrap iteration and across the three node categories:We evaluated model performance in each bootstrap iteration and across the three node categories using the following metrics: Accuracy, AUC ROC, AUC Precision-Recall, Precision, and Recall. We further propose a new metric called the CPI.

- **AUC**: a metric used to represent the cumulative performance of a model over multiple time points in AL experiments. As the AL cycle progresses, queries are made at each timepoint, and model performance (e.g., F1 score, accuracy, or any other relevant metric that takes values between 0 and 1) is evaluated on the unqueried nodes or future timepoints. The AUC is computed by summing the performance values across all timepoints, capturing the overall model effectiveness over time.

  Formally, the AUC can be represented as:

$$\text{AUC} = \sum_{t=1}^{T} \frac{\text{Performance}(t) + \text{Performance}(t+1)}{2} \times \Delta t \tag{3}$$

  where T is the total number of timepoints, and Δt is the interval between timepoints (e.g., 1 hour or 1 day). This trapezoidal approximation integrates the performance curve over the specified timepoints.

- **Cumulative Performance Index (CPI)**: a novel metric designed to capture sustained model effectiveness over time in a dynamic AL setting. Unlike standard metrics that assess static performance, CPI quantifies a model's performance trajectory across multiple timepoints.

  We define the CPI by normalizing the AUC. The CPI is obtained by dividing the AUC by T, yielding a value between 0 and 1:

$$\text{CPI} = \frac{\text{AUC}}{T} \tag{4}$$

  The maximum value of CPI is 1, which is when the model provides 100% performance at all time points. This CPI metric provides a standardized way to compare the effectiveness of AL strategies across various experimental setups, regardless of the number of timepoints.

4 Diversity & Sampling Burden Metrics for Longitudinal Graph Sensing

In longitudinal graph-based environments (e.g., social networks, sensor networks, IoT), AL can impose significant query burden on specific nodes/sensors. Even with static graph structures, node features (e.g., sensor data) can change from one timestep to the next. We introduce and evaluate a set of metrics designed to assess AL strategies on two key aspects: sampling diversity and user burden. These metrics allow us to quantify how well an AL strategy distributes queries across nodes/sensors, avoids over-exertion, and maintains proportional representation of the network.

- **Sampling Entropy:**It measures the diversity in node selection by quantifying how evenly nodes are sampled over the course of the AL process. Higher entropy values indicate a more uniform distribution, meaning that the AL strategy is querying a broad set of nodes rather than focusing on a few. Entropy is computed as:

$$H = -\sum_{i=1}^{N} p_i \log(p_i) \tag{5}$$

  where p_i is the probability of sampling node/sensor i, and N is the total number of nodes/sensors.

- **Coverage Ratio:** It assesses the proportion of nodes/sensors that are sampled at least once over the evaluation period. It ensures that the AL strategy is not only sampling a few key nodes but is capturing a broader section of the graph. The coverage ratio is given by:

$$C = \frac{\text{Number of unique nodes sampled}}{\text{Total number of nodes (N)}} \tag{6}$$

- **Time-Gap Analysis** evaluates how frequently the same nodes are being sampled across consecutive or nearby days. This metric helps identify over-exertion, where certain nodes are queried too frequently, leading to fatigue (for users) or battery depletion (for sensors).
  - **Average Time Gap**: This metric calculates the mean number of days between consecutive samples for each node:

$$\text{Average Time Gap} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|T_i|} \sum_{t=1}^{|T_i|-1} (T_{i,t+1} - T_{i,t}) \tag{7}$$

    where T_i is the set of days when node i is sampled, and N is the total number of nodes.
  - **Within-Gap Percentage**: This metric computes the percentage of nodes sampled within a specified gap (e.g., 1 or 2 days):

$$\text{Within-Gap Percentage} = \frac{\sum_{i=1}^{N} \mathbb{1}(\text{gap} < k)}{N} \tag{8}$$

    where k is the specified time gap threshold and $\mathbb{1}$ is an indicator function.
  - **Over-Exertion Score**: The **over-exertion score** quantifies how often nodes are sampled more frequently than a defined time-gap threshold. High over-exertion scores indicate that certain nodes are being repeatedly sampled within short intervals:

$$E = \frac{\text{Number of nodes sampled within the gap threshold}}{\text{Total number of nodes sampled}} \tag{9}$$

4.1 Centrality Metrics

In graph-based AL strategies, nodes have varying levels of importance based on their position in the graph. Centrality metrics help us understand which nodes are more critical in the network and how their centrality correlates with their sampling frequency. Key centrality metrics included in GRAIL are Degree Centrality, Betweenness Centrality, Closeness Centrality, Eigenvector Centrality, Harmonic, Load, PageRank, and Clustering coefficientNewman (2010). Analyzing centrality metrics alongside sampling and time-gap statistics can help determine whether certain high-centrality nodes are being disproportionately favored by the AL strategy.

5 Datasets

We use two longitudinal, dynamic sensor social network datasets: SNAPSHOT (Khalid & Sano, 2023) and Friends-and-Family (FnF) (Aharony et al., 2011). SNAPSHOT is publicly available in processed form and

contains multimodal sensing data from 200+ participants (e.g., skin conductance, acceleration, temperature, SMS/call metadata, mobility traces), with a subset also socially connected(Khalid et al., 2024). In contrast, FnF includes 130 participants and spans 15 months of raw mobile-sensing data (calls, SMS, Bluetooth, location), but is released only in raw form. To make FnF usable for graph-based Active Learning (AL), we developed a substantial preprocessing pipeline involving heterogeneous sensor cleaning, timestamp alignment, daily aggregation, symmetric dyad construction for communication events, and integration of survey-derived labels.

For both datasets, we construct graph networks using SMS/Call metadata; FnF also provides a friendship graph derived from survey responses. All resulting graphs will be released to support reproducible experimentation. In our experiments, results correspond to the SMS graph for SNAPSHOT and the friendship graph for FnF. Model inputs consist of daily sensor features, and model outputs are binary labels for mood and sleep duration. We divide the data into a held-out set (20%) and an AL pool (80%). During each AL iteration, k nodes are queried from the pool, and the base model is initially trained using data from the first L days. More details on preprocessing are provided in appendix F

Ethics and Data Use: Both SNAPSHOT and FnF contain sensors and survey data from human participants collected under the oversight of their respective institutional review boards. All data used in this work were provided in de-identified form and analyzed strictly in accordance with the usage terms specified by the dataset custodians. No additional data collection involving human subjects was conducted for this work.

6 Experimental Settings

GRAIL is designed with a flexible and systematic structure, allowing users to tailor experimental settings based on their application needs. Our evaluation spans 11 AL strategies, two longitudinal study datasets with multiple graphs(call/SMS etc), multiple node categories, several performance metrics (CPI, accuracy, macro-F1), and multiple burden and diversity measures (entropy, coverage, time-gaps, exertion), evaluated over 100 bootstrap iterations. Given this large number of experimental combinations, it is not feasible to report every result because of space constraints. Therefore, we present a representative subset of results in the main paper, summarizing key findings across both datasets and selecting whichever dataset most clearly illustrates each insight. The core performance–burden comparison in the context of network topology differences, which is central to our contributions, is shown for *both* datasets.

Experimental Design. GRAIL exposes several knobs that users can tune to match real-world deployment constraints. Key experiment design features include:

- **Training Period and Query Size (L, k).** L is the number of fully-labeled initial days used to bootstrap the model. After this warm-up, each day we query k new nodes, controlling the query budget and user-facing cost.
- **Bootstrapping.** We use 100 bootstrap iterations per AL strategy to obtain robust averages and quantify variability across runs.
- **Temporal (Streaming) Evaluation.** Models are updated incrementally as new labels arrive, mirroring real-world longitudinal deployments rather than static batch AL.

Our evaluation consists of five complementary sets of experiments, each designed to illuminate a different aspect of AL behavior in longitudinal social networks.

- **Headline Finding – When Does Graph-AL Beat Random?** We summarize, across all (strategy × dataset × node-category) cells, the fraction in which each strategy's mean CPI is at least one standard deviation above random sampling, and identify which strategies consistently provide value over random and which do not.
- **Temporal Analysis.** We examine how accuracy, F1, and CPI evolve across days to assess model stability, identify trends, and measure temporal generalization to unqueried nodes.

- **Sampling Diversity and Burden.** We quantify how widely each strategy explores the network by measuring entropy, coverage, time-gaps, and exertion to understand who gets queried, how often, and with what user-facing cost.
- **Performance–Burden Tradeoff.** We jointly analyze CPI and exertion on both datasets to determine which strategies achieve the best balance between predictive performance and user effort.
- **Network Topology Effects.** We evaluate how structural properties, such as degree distribution and community structure, influence AL behavior, highlighting when strategies benefit from or are hindered by the underlying graph structure.

6.1 Evaluation Rationale

Before presenting results, we make explicit the design rationale that links GRAIL's four node categories, four time-axes of comparison, and six core metrics. Table 2 what each comparison (e.g., "test same day," "train next day," "unqueried next day") is designed to measure.

Table 2: What each node category × time comparison is designed to measure, and which AL property it isolates.

Node Category	What it tests	AL property it isolates
Test (held-out, same day)	Generalization to entirely unseen users	Whether the strategy chose informative-enough queries to teach the model about users it never saw
Unqueried Same Day	In-pool generalization without label propagation through the same day	How well the strategy's queries generalize to other pool nodes within the current timestep – isolates spatial label propagation
Unqueried Next Day	In-pool generalization to the next timestep	Whether queried information transfers across time as features evolve – isolates temporal generalization
Train Next Day	Predicting the queried users' own next-day label	Whether queried users' own behavior is now predictable to the model – isolates the direct value of querying these specific users

The metrics, in turn, fall into three groups, each answering a distinct question: (i) *performance* metrics (Accuracy, micro/macro-F1, AUC, and our Cumulative Performance Index CPI) ask whether the model is correct; (ii) *diversity* metrics (Sampling Entropy, Coverage Ratio) ask whether queries are spread broadly across the network; and (iii) *burden* metrics (Average Time Gap, Within-Gap Percentage, Over-Exertion Score) ask whether individual users are queried at sustainable rates. CPI is the headline performance metric because it integrates per-day performance across the full study horizon rather than reporting a single endpoint, making it appropriate for longitudinal evaluation where day-to-day variability is real signal rather than noise. The burden metrics are deliberately positioned as proxies for query-cost-to-the-user; we do not claim they capture cognitive load or contextual availability, only sampling-rate equity, and we discuss this scope in Section 8.

7 Results

7.1 Headline Finding: When Does Graph-AL Beat Random Sampling?

A recurring concern in prior Graph-AL evaluations is that performance differences between AL strategies are small relative to bootstrap variability, raising the question of whether classical Graph-AL strategies actually beat random sampling in realistic longitudinal settings. We address this question explicitly. For each strategy s, dataset $d \in \{\text{SNAPSHOT}, \text{FnF}\}$, and node category $c \in \{\text{Test}, \text{Unq. Same}, \text{Unq. Next}, \text{Train Next}\}$, we

compute $\Delta(s, d, c) = \text{CPI}(s, d, c) - \text{CPI}(\text{Random}, d, c)$ from the bootstrap means in Tables 4 and 5. We count a cell as "beats random" if $\Delta \geq \sigma_{\text{Random}}(d, c)$, the bootstrap standard deviation of random sampling in that cell. Each strategy is evaluated across 8 cells total (2 datasets $\times$ 4 categories).

Table 3 summarizes the result. Three findings stand out:

Finding 1: A small subset of *hybrid* strategies are robust winners. AGE and Density beat random sampling in 6 of 8 cells and never underperform it significantly. Both combine signals from multiple sources: AGE blends centrality, uncertainty, and density, while Density blends embedding-space clustering with topology. No pure-uncertainty, pure-topology, or pure-embedding strategy matches their consistency.

Finding 2: GraphPartFar is volatile but capable of extreme gains under heterogeneous topology. On the more decentralized FnF graph, GraphPartFar achieves $\Delta = +0.27$ on Test (CPI 0.66 vs. random 0.39) and $\Delta = +0.43$ on Train-Next (0.97 vs. 0.54), the largest single-cell improvements observed in either dataset. However, it underperforms random in the Unqueried-Same-Day cell on FnF ($\Delta = -0.05$). The mechanism behind this volatility is analyzed in Section 7.6.

Finding 3: Uncertainty-only and topology-blind strategies offer little practical value over random. Pure-uncertainty methods (Entropy, Least-Confidence, Margin) and Coreset cluster around 3–4 cells beating random, with most gains concentrated on the SNAPSHOT dataset where the underlying graph is more uniform. On the heterogeneous FnF topology, these strategies are statistically indistinguishable from random in most cells. This is consistent with a recurring observation in prior Graph-AL evaluations that performance differences between AL strategies are often within bootstrap variability of random; the result holds for these strategies but not for the hybrids identified in Finding 1.

Table 3: Cells in which each strategy's mean CPI exceeds random sampling by at least one bootstrap standard deviation, across 4 node categories on each of 2 datasets ($n = 8$ cells per strategy). Strategies are sorted by total "beats random" cells. Computed from Tables 4 and 5.

Strategy	SNAPSHOT	FnF	Total	Notes
AGE	4/4	2/4	**6/8**	Robust; never significantly underperforms random
Density	4/4	2/4	**6/8**	Robust; matches AGE
GraphPartFar	3/4	2/4	5/8	Volatile; largest extreme wins (+0.27, +0.43) on FnF
FeatProp	3/4	1/4	4/8	Strong on SNAPSHOT, weak on heterogeneous FnF
Uncertainty (Entropy)	3/4	1/4	4/8	Topology-blind; SNAPSHOT-only gains
Uncertainty (Least-Conf.)	3/4	1/4	4/8	Similar to Entropy
PageRank	2/4	2/4	4/8	Consistent middle ground
Degree	2/4	2/4	4/8	Same pattern as PageRank
Coreset	3/4	0/4	3/8	Topology-blind; fails on FnF entirely
Uncertainty (Margin)	1/4	2/4	3/8	Inconsistent across datasets
GraphPart	1/4	1/4	2/8	Weakest of the graph-based hybrids

The Pareto-optimal strategies identified here (AGE, Density, GraphPartFar) are precisely those that combine structural and feature/uncertainty signals. The supporting evidence, per-cell CPI values, per-strategy burden, and the role of network topology, is presented in the remainder of Section 6, with the performance-burden Pareto front shown in Section 7.5 and the topology-driven mechanism behind GraphPartFar's volatility analyzed in Section 7.6.

7.2 Temporal Behavior Across Days

We first examine how performance evolves over time, since AL decisions directly shape both short-term adaptation and the model's ability to generalize to new and unqueried nodes. Figure 3 shows rolling accuracy for SNAPSHOT (sleep label) using a five-day window, and similar temporal patterns were observed across accuracy, macro-F1, and CPI.

The temporal curves separate strategies into three regimes: structure-aware methods (GraphPartFar, GraphPart, Density, AGE) achieve the highest and smoothest accuracy; uncertainty-driven methods rise

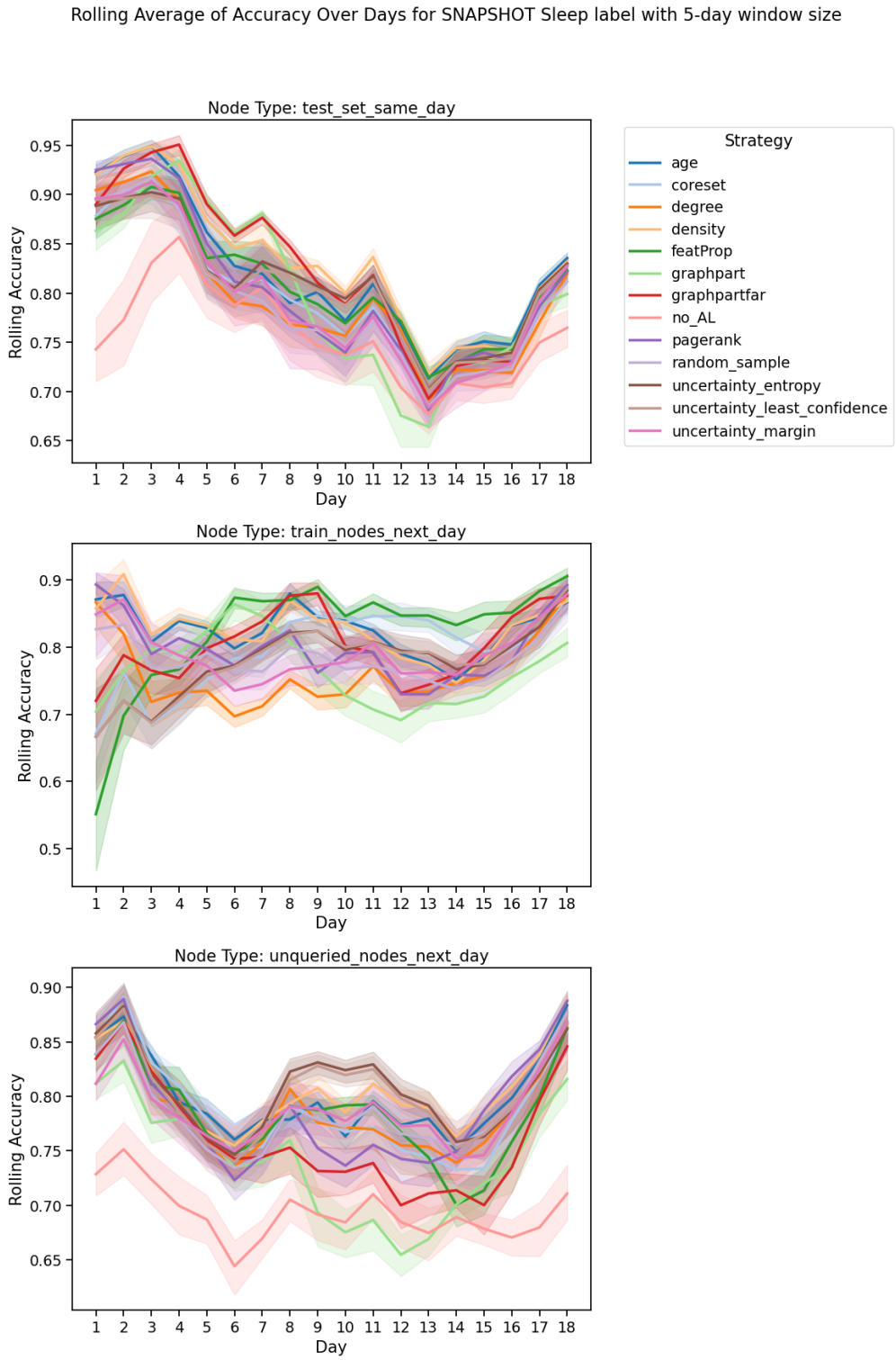

Figure 3: Rolling accuracy for SNAPSHOT sleep classification (L=8, k=9). Structure-aware and density-based methods achieve the most consistent long-term performance.

quickly but oscillate; and embedding-based methods (Coreset, FeatProp) are mid-range but very stable. The clearest separation appears on unqueried and next-day nodes, where Density and GraphPartFar reliably lead.

An important recovery-dynamic underlies these temporal curves and explains why some averaged comparisons across node categories appear flat in Tables 4 and 5. During the first L days, all nodes are labeled and the model trains on the full pool, producing an initial accuracy spike. Once active learning begins and only k nodes are queried per day, performance drops sharply across all node categories before climbing back as queried labels accumulate. Strategies recover at materially different rates: GraphPartFar, Density, and AGE recover within a handful of days, while Random Sampling and pure-uncertainty methods recover more slowly and exhibit periodic dips that we attribute to behavioral pattern shifts in the underlying sensing data. The implication for Tables 4 and 5 is that small differences between trained and unqueried nodes around the middle of the study window often reflect this recovery dynamic, not a failure of label propagation: the queried-node advantage is real but is partially masked when CPI is averaged over the full study horizon.

7.3 Node-Type Generalization Across Datasets

Having examined how performance evolves day by day, we now turn to the average response of each strategy when performance is aggregated over the full active learning cycle and over multiple bootstrap iterations. This perspective provides a "big picture" summary of how well each method generalizes across the different node categories in GRAIL's evaluation: held-out test nodes, next-day training nodes, and unqueried-next-day nodes. Tables 4 and 5 report the mean CPI for SNAPSHOT and FnF respectively.

For brevity, the full per-strategy, per-cell CPI matrices for SNAPSHOT and FnF are reported in Appendix Tables 4 and 5; the standout cells are: GraphPartFar and Density tie at 0.77 test CPI on SNAPSHOT, while AGE reaches 0.69 test CPI and GraphPartFar reaches 0.97 train-next-day CPI on FnF. We summarize the per-strategy patterns below.

Across both datasets, structure-aware and density-based strategies achieve the strongest overall performance, but the ranking differs between SNAPSHOT and FnF in ways that mirror the two graphs' topology. On SNAPSHOT, GraphPartFar and Density tie for the top test-set CPI (0.77), with AGE consistently competitive across all node types. On FnF, AGE leads on test-set CPI (0.69), while GraphPartFar exhibits extreme specialization (highest train-next-day CPI at 0.97 but lowest unqueried-same-day at 0.37). Uncertainty-based methods are inconsistent across datasets; embedding-based approaches (Coreset, FeatProp) show small gaps between node categories on both datasets but never reach top CPI. Per-strategy SNAPSHOT and FnF discussions, including the mechanism behind GraphPartFar's specialization, are reported in Appendix A. All differences are statistically significant (one-way ANOVA and Kruskal–Wallis, $p \ll 10^{-10}$ for both datasets).

7.4 Sampling Diversity, Coverage, and Burden

We next analyze how each AL strategy distributes its queries across users and days, averaged over 100 bootstrap iterations. We summarize burden with the over-exertion score — the fraction of queries issued to users whose time gap falls below a fixed threshold (e.g., three days). Figure 4 shows over-exertion at the 3-day threshold for FnF: uncertainty-based methods and some structure-focused strategies (notably GraphPartFar) impose the highest burden by repeatedly re-querying a subset of users, while AGE and GraphPart maintain relatively low burden alongside strong predictive performance. The same qualitative ordering holds for sampling entropy and node coverage and across average-time-gap metrics on both SNAPSHOT and FnF; full per-dataset and per-metric breakdowns (entropy/coverage, average time gap, and over-exertion at varying gap thresholds) are provided in Appendix B, Figures 9–14 and 10.

Sampling diversity and burden are highly strategy- and topology-dependent: structure-aware approaches concentrate queries on influential regions (strong performance, higher burden on central users); embedding-based and random strategies spread queries uniformly (lower burden, lower accuracy); and uncertainty-based methods over-query a difficult subset. These differences are invisible if accuracy is reported alone.

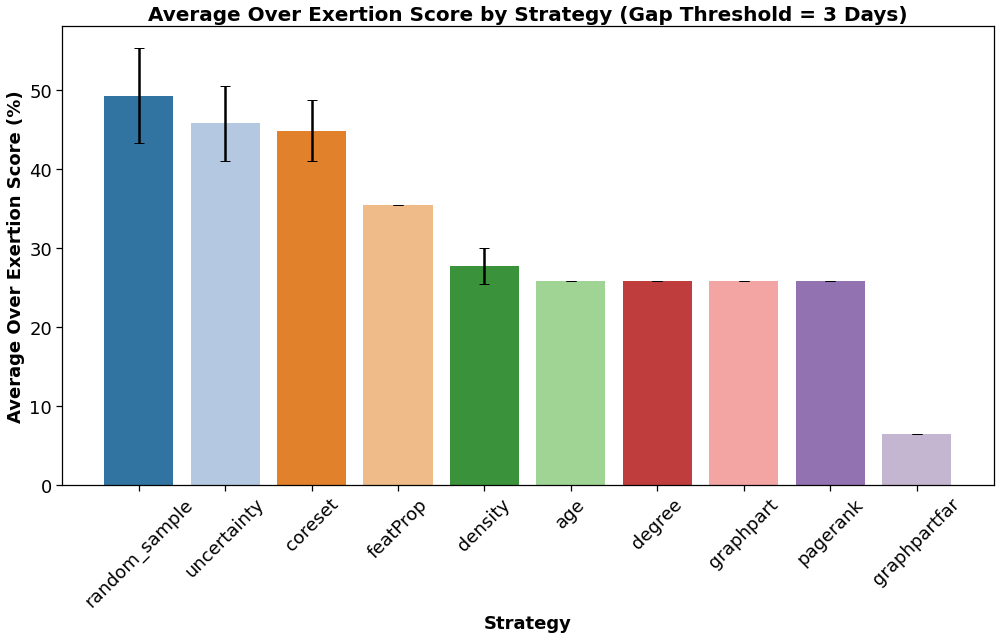

Figure 4: Average over-exertion score at a fixed 3-day gap threshold for sleep classification in the FnF dataset.

7.5 Performance–Burden Tradeoff Analysis

This subsection is the primary supporting evidence for the headline finding in Section 7.1: the same hybrid strategies that beat random in the most CPI cells (AGE, Density, GraphPartFar) also occupy the Pareto-optimal corner when performance and burden are jointly considered. Figure 5 plots CPI against per-user over-exertion for FnF; the ideal strategy occupies the upper-left corner (high CPI, low burden). AGE and GraphPartFar form a high-performance cluster, with AGE achieving the overall best trade-off. The SNAPSHOT counterpart (Appendix Figure 15) shows the same ordering with substantially tighter clustering due to SNAPSHOT's more uniform topology. The wider dispersion in FnF mirrors its more heterogeneous centrality distribution and motivates the topology analysis in the following subsection.

7.6 Network Topology Sensitivity

We now examine the mechanism behind the strategy/topology interactions identified above. The headline observation, visible in Figure 6, is that SNAPSHOT and FnF differ substantially in their centrality distributions: SNAPSHOT's SMS graph is comparatively uniform, while FnF's friendship graph is heavily skewed toward a small set of high-centrality hubs. This single difference explains most of the strategy-level divergence between the two datasets.

In SNAPSHOT, strategies like GraphPartfar show positive correlations with centrality metrics (betweenness, closeness, degree), maintaining sufficient time gaps between central nodes and achieving balanced sampling. Conversely, in FnF, GraphPartfar and GraphPart show negative correlations, frequently sampling central nodes due to the network's decentralized structure with diverse community sizes. This results in over-exertion and reduced time gaps. Per-strategy centrality–burden heatmaps for both datasets are provided in Appendix Figures 16 and 17, and a heatmap of mean normalized centrality preferences across strategies is provided in Appendix Figure 18.

This topology-driven mechanism directly explains the most striking row in the per-cell CPI tables: on FnF, GraphPartFar achieves 0.97 on Train-Next-Day nodes while simultaneously achieving only 0.37 on Unqueried-Same-Day nodes (Appendix Table 5). The 0.97 is a real signal: aggressive hub-sampling means the queried nodes are precisely the high-influence nodes whose own next-day labels are most predictable from their own recent history. The 0.37 is the cost of that strategy: by concentrating queries on hubs, GraphPartFar

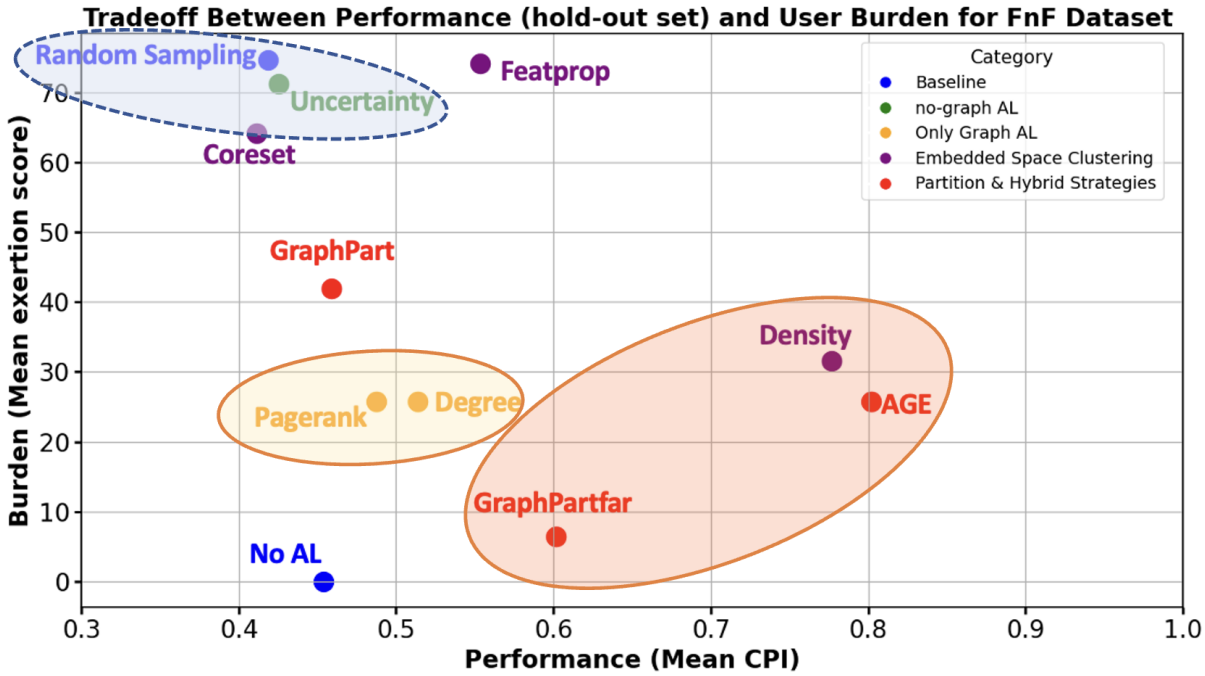

Figure 5: FnF performance–burden comparison.

starves the non-hub regions of the network of new labels, so unqueried nodes in modular FnF communities receive almost no label-propagation benefit on the same day. The same strategy on SNAPSHOT does not produce this divergence (0.77 Test, 0.73 Unq. Same, 0.77 Train-Next) because SNAPSHOT's centrality distribution is much more uniform (Figure 6), so hub-sampling does not concentrate queries away from any structural minority. This is the mechanism behind GraphPartFar's volatility flagged in Section 7.1 Finding 2: the strategy is highly effective when topology is uniform and dangerously imbalanced when topology is heterogeneous.

The tradeoff Figure 5 (and Appendix Figure 15) further support these observations. In SNAPSHOT, strategies like AGE and GraphPartfar form a high-performance, low-burden cluster, leveraging a uniform topology. In FnF, the same strategies exhibit higher variability, reflecting challenges from its heterogeneous structure. FeatProp and Coreset, less influenced by topology, maintain consistent performance across both datasets.

Overall, these findings emphasize the importance of aligning AL strategies with graph topology. Uniform networks like SNAPSHOT favor balanced sampling, while decentralized structures like FnF require careful tuning to avoid over-exertion of central nodes.

To quantitatively ground this mechanism, we computed Pearson correlations between standard centrality metrics (degree, betweenness, closeness, PageRank, harmonic, load, clustering) and two of GRAIL's burden metrics (Average Time Gap and Over-Exertion Score), per AL strategy on FnF sleep classification. The strongest correlations match the mechanism story: GraphPartFar correlates strongly negatively with clustering centrality under Average Time Gap ($r = -0.880$, $p \approx 10^{-10}$) and strongly positively with clustering under Over-Exertion Score ($r = +0.936$, $p \approx 10^{-14}$), reflecting its tendency to repeatedly sample tightly-clustered hub nodes with insufficient temporal spacing. Conversely, GraphPart and FeatProp show positive correlations under Average Time Gap and negative under Over-Exertion Score, indicating well-spaced sampling of central nodes. The full correlation matrices (only statistically significant entries, $p < 0.05$) are reported in Appendix Tables 6 and 7.

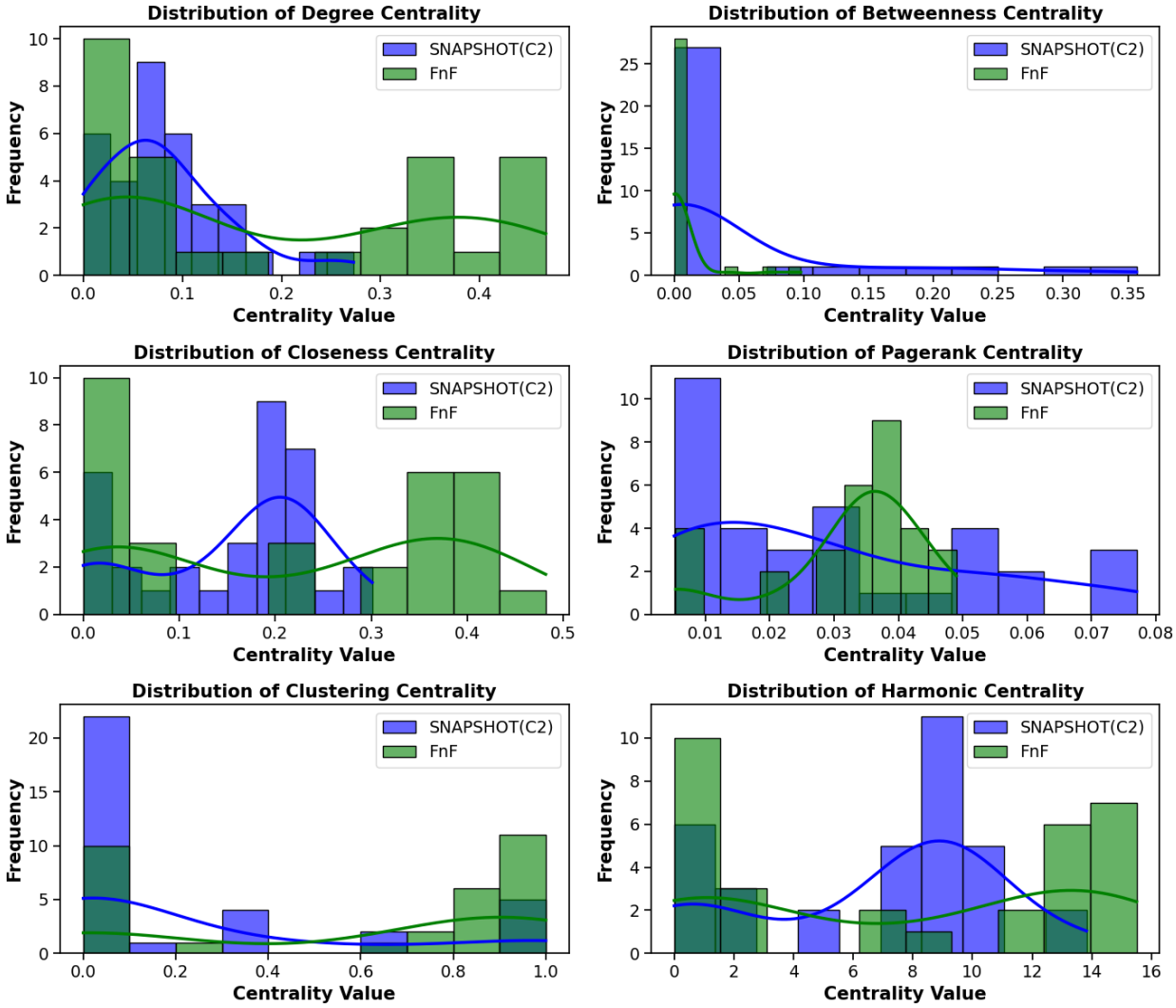

Figure 6: Network topology comparison between SNAPSHOT and FnF.

7.7 Sensitivity to Hyperparameters L and k

We also conduct a sensitivity analysis over the warm-up period L and per-day query budget k. Figure 7 and 8 sweep $k \in \{3, 5, 8, 10\}$ at fixed $L = 3$, reporting CPI on the unqueried pool and on the held-out test set respectively, on FnF mood classification using the friendship graph. The corresponding Macro-F1 panels and the full L-sweep (CPI and Macro-F1, both unqueried pool and held-out test) are provided in Appendix E, Figures 19–24.

The Pareto-optimality of AGE, GraphPart, and Density on held-out test CPI replicates across the full range of k, and the relative ordering of strategies on unqueried-pool generalization is stable as k grows (with mild saturation for the structure-aware methods at high k). AGE and FeatProp are insensitive to the warm-up period L, while No-AL and Random remain consistently low regardless of L. The headline finding from Section 7.1 therefore holds under varied hyperparameters; per-panel commentary and the full L-sweep are in Appendix E.

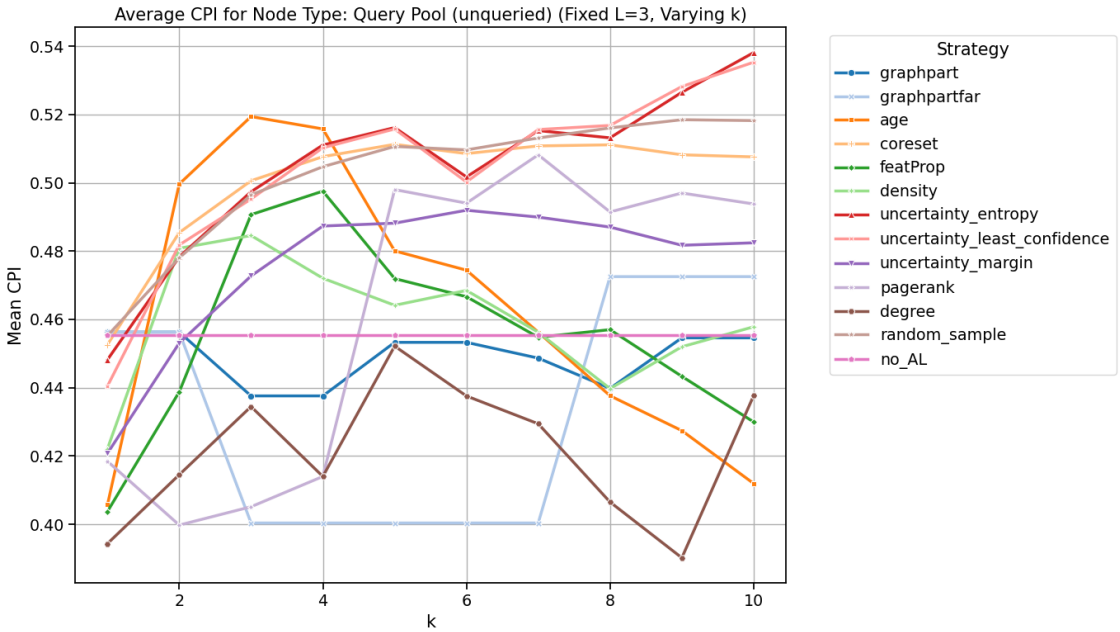

Figure 7: CPI on unqueried pool vs. k (FnF, $L = 3$).

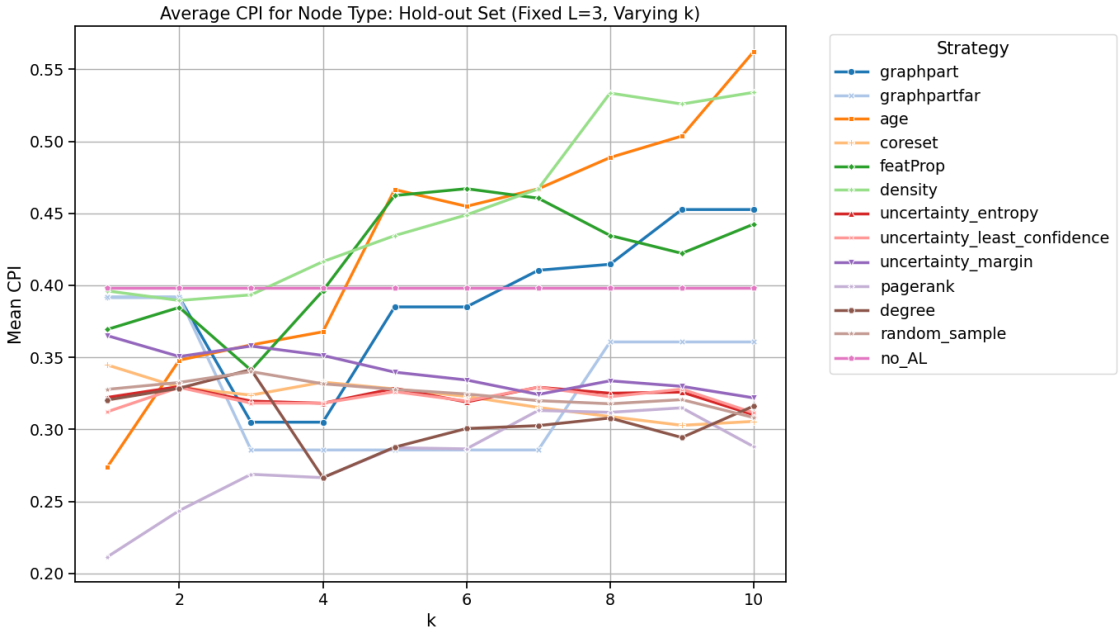

Figure 8: CPI on hold-out test vs. k (FnF, $L = 3$).

Summary

The headline finding from Section 7.1 — that AGE and Density beat random sampling in 6 of 8 evaluated cells while no pure-uncertainty, pure-topology, or pure-embedding strategy matches their consistency — is corroborated across the other subsections. Structure-aware strategies achieve strong CPI but concentrate queries on central users; embedding-based methods distribute queries broadly but never reach top CPI; uncertainty-based methods over-query difficult nodes, with CPI gains over random falling within bootstrap variability across most FnF categories. The Pareto-optimal corner is occupied by hybrid strategies that combine multiple signals; pure single-signal strategies are systematically dominated. These findings underscore that active learning for longitudinal social networks requires simultaneous consideration of structure, burden, diversity, and generalization, since focusing solely on accuracy obscures critical differences in how strategies allocate labeling effort across participants.

8 Limitations

Although this study presents significant advances in AL for longitudinal sensing on social networks , there are several limitations primarily arising from the current state of available datasets and technological constraints.

- **Dataset Diversity and Static Graphs:** As discussed upfront in Section 1, GRAIL is architected to support evolving graph topologies but the two real-world datasets evaluated here use static graph structures with longitudinally evolving features and labels. Both datasets focus on health monitoring with multi-modal sensor data; public datasets that combine rich sensing features with a longitudinal record of evolving social ties remain extremely rare.

- **Sample Size and Recruitment Constraints:** The datasets used have limited participant numbers (200+ for SNAPSHOT and 130 for FnF) due to the high user burden in health sensing studies. Recruiting socially connected participants further restricted dataset size.

- **Scalability and Larger Networks:** GRAIL's scalability was not fully tested on larger, complex networks due to the absence of suitable datasets, limiting our ability to assess AL strategy performance in such scenarios.

- **Fixed Querying Intervals:** Our experiments employed a uniform daily query interval, which may not be optimal for user engagement or model performance. Adaptive query intervals based on user behavior or model uncertainty could be explored in future work.

- **Simplified User Burden Measurement:** User burden was measured primarily by sampling frequency and time-gap, overlooking cognitive load, availability, or context. A multi-dimensional burden model could provide a more user-centric evaluation.

- **Custom Preprocessing for Specific Datasets:** The preprocessing pipeline was tailored to SNAPSHOT and FnF, ensuring data quality but limiting direct applicability to other datasets without adaptation. However, GRAIL's modular design allows adjustments for other domains.

9 Conclusion

We introduced GRAIL, a benchmark for graph AL on dynamic, time-series data, revealing how graph-based strategies like GraphPartFar and AGE achieve superior performance while balancing user burden. Our results highlight the critical impact of network topology on strategy effectiveness, and our proposed metrics (CPI, entropy, coverage) provide a robust foundation for evaluating graph AL methods. GRAIL is publicly available, allowing researchers to compare and identify the most suitable AL strategies for their own dynamic graph applications.

References

Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and mobile computing*, 7(6):643–659, 2011.

Anonymous. Grail dataset preprocessing and feature engineering repository. `https://osf.io/4h8df/?view_only=feee28b3b4224721b3e6d83955f9961f`, 2025.

Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active learning for networked data. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 79–86, 2010.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085*, 2017.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL `https://arxiv.org/abs/1706.02216`.

Maryam Khalid and Akane Sano. Sleep contagion dataset, 2023. URL `https://github.com/comp-well-org/SPAND_Dataset/tree/main`.

Maryam Khalid, Elizabeth B Klerman, Andrew W McHill, Andrew JK Phillips, and Akane Sano. Sleepnet: Attention-enhanced robust sleep prediction using dynamic social networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(1):1–34, 2024.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. URL `https://arxiv.org/abs/1609.02907`.

Yang Li, Yu Liu, Wei-Gang Cui, Yu-Zhu Guo, Hui Huang, and Zhong-Yi Hu. Spatio-temporal-spectral hierarchical graph convolutional network with semisupervised active learning for patient-specific seizure prediction. *IEEE Transactions on Cybernetics*, 52(11):12189–12204, 2022.

Jiaqi Ma, Ziqiao Ma, Joyce Chai, and Qiaozhu Mei. Partition-based active learning for graph neural networks. *arXiv preprint arXiv:2201.09391*, 2022.

Cristopher Moore, Xiaoran Yan, Yaojia Zhu, Jean-Baptiste Rouquier, and Terran Lane. Active learning for node classification in assortative and disassortative networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 841–849, 2011.

Pratheeksha Nair and Zhi Wen. A study on active learning for graphs. 2020.

Mark Newman. Networks: An introduction, 2010.

Florence Regol, Soumyasundar Pal, Yingxue Sun, and Mark Coates. Active learning on attributed graphs via graph cognizant logistic regression and preemptive query generation. In *International Conference on Machine Learning (ICML)*, 2020.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 1070–1079, 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL `https://arxiv.org/abs/1710.10903`.

Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Active learning for graph neural networks via node feature propagation. *arXiv preprint arXiv:1910.07567*, 2019.

Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Graph-based active learning with structural uncertainty. In *Conference on Neural Information Processing Systems (NeurIPS) – Workshop*, 2021.

Wentao Zhang, Wenxiao Zheng, Yexin Li, and Tong Yu. A survey on graph-based active learning for dynamic networked data. *ACM Computing Surveys*, 2022.

A Full Per-Cell CPI Matrices

This appendix reports the full per-strategy, per-node-category CPI matrices underlying Table 3 and Section 7.1. Tables 4 and 5 report mean CPI for SNAPSHOT and FnF respectively, with bootstrap standard deviations in parentheses. Bold values mark the best strategy in each column.

Table 4: CPI on SNAPSHOT (Sleep, L=8, k=9). Higher is better. Bold values mark the best strategy in each column.

AL Strategy	Test	Unq. Same	Unq. Next	Train Next
random_sample	0.72 (0.03)	0.70 (0.03)	0.74 (0.03)	0.75 (0.03)
no AL (baseline)	0.71 (0.03)	0.67 (0.04)	0.65 (0.05)	N/A
graphpart	0.75 (0.02)	0.71 (0.03)	0.70 (0.03)	0.73 (0.04)
graphpartfar	**0.77 (0.01)**	0.73 (0.02)	0.72 (0.03)	0.77 (0.02)
age	0.76 (0.02)	0.75 (0.01)	0.76 (0.02)	0.78 (0.02)
coreset	0.74 (0.03)	**0.76 (0.02)**	0.74 (0.02)	0.78 (0.03)
featProp	0.75 (0.02)	0.75 (0.02)	0.74 (0.03)	**0.81 (0.02)**
density	**0.77 (0.01)**	0.75 (0.02)	**0.78 (0.02)**	0.78 (0.02)
uncertainty entropy	0.75 (0.02)	0.75 (0.02)	0.76 (0.03)	0.76 (0.03)
uncertainty least confidence	0.75 (0.03)	0.74 (0.02)	0.76 (0.03)	0.76 (0.03)
uncertainty margin	0.74 (0.04)	0.70 (0.05)	0.75 (0.04)	0.75 (0.03)
pagerank	0.74 (0.02)	0.74 (0.02)	0.74 (0.02)	0.76 (0.02)
degree	0.74 (0.02)	0.75 (0.02)	0.74 (0.03)	0.72 (0.02)

Table 5: CPI on FnF (Sleep, L=6, k=5). Higher is better. Bold values mark the best strategy in each column.

AL Strategy	Test	Unq. Same	Unq. Next	Train Next
random_sample	0.39 (0.03)	0.42 (0.02)	0.41 (0.02)	0.54 (0.04)
no AL (baseline)	0.45 (0.05)	0.41 (0.03)	0.40 (0.02)	N/A
graphpart	0.36 (0.04)	**0.48 (0.01)**	0.43 (0.01)	0.41 (0.03)
graphpartfar	0.66 (0.00)	0.37 (0.00)	0.42 (0.02)	**0.97 (0.01)**
age	**0.69 (0.03)**	0.43 (0.01)	0.42 (0.01)	0.58 (0.02)
coreset	0.41 (0.03)	0.44 (0.01)	0.43 (0.02)	0.55 (0.04)
featProp	0.45 (0.02)	0.44 (0.01)	0.38 (0.01)	0.46 (0.03)
density	**0.68 (0.03)**	0.44 (0.01)	0.42 (0.01)	0.58 (0.03)
uncertainty entropy	0.42 (0.05)	0.47 (0.02)	0.44 (0.02)	0.52 (0.04)
uncertainty least confidence	0.41 (0.05)	0.46 (0.02)	0.43 (0.02)	0.53 (0.04)
uncertainty margin	0.45 (0.03)	0.43 (0.03)	**0.45 (0.03)**	0.47 (0.04)
pagerank	0.44 (0.03)	0.45 (0.02)	0.44 (0.02)	0.62 (0.04)
degree	0.44 (0.03)	0.45 (0.02)	0.44 (0.02)	0.62 (0.04)

B Sampling Diversity, Coverage, and Burden — Additional Plots

This appendix complements the FnF over-exertion figure (Figure 4) shown in the main text with: sampling entropy and node coverage for both datasets, the average time gap metric for both datasets, and over-exertion

at varying gap thresholds on FnF. The same qualitative strategy ordering holds across all panels: FeatProp and Random achieve the broadest coverage; structure-aware strategies (GraphPartFar, GraphPart, Density) concentrate queries; uncertainty-based methods consistently impose the highest burden through repeated re-querying.

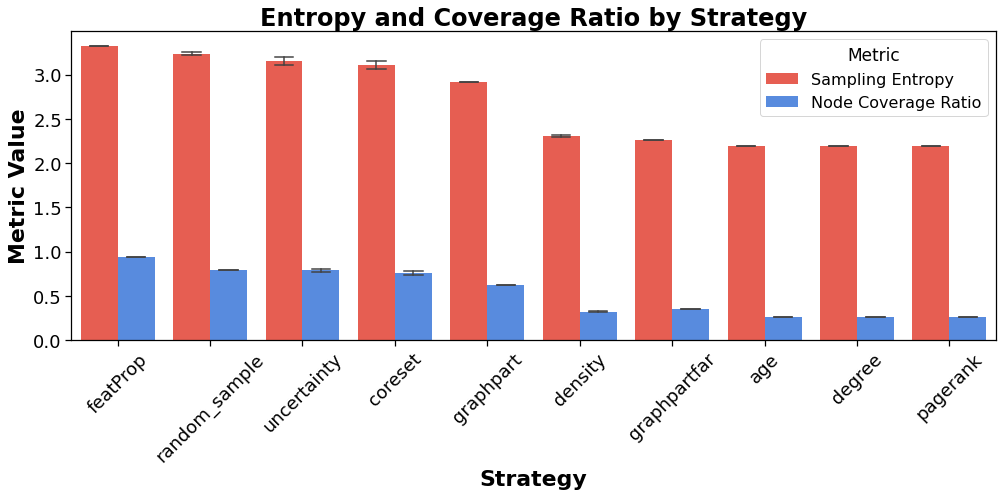

Figure 9: Sampling entropy and node coverage ratio for SNAPSHOT sleep classification across AL strategies.

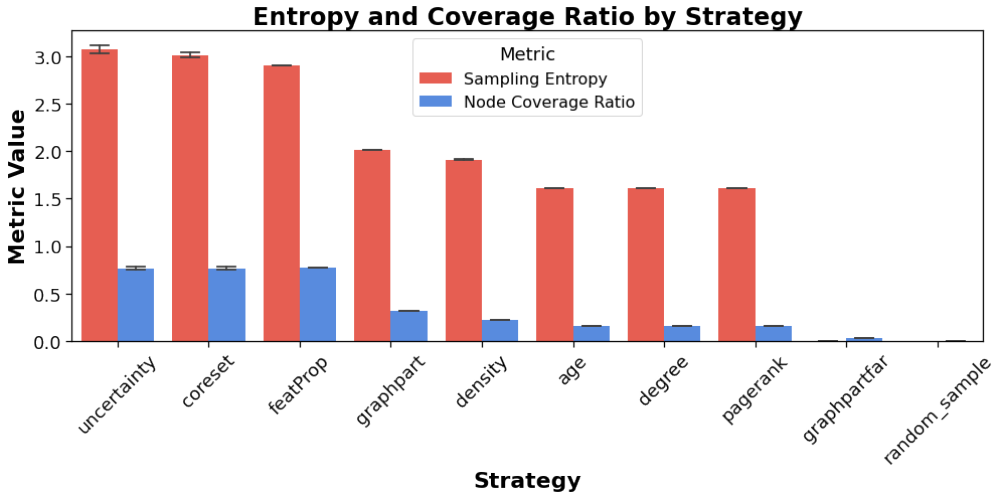

Figure 10: Sampling entropy and node coverage ratio for sleep classification in the FnF dataset.

C Performance–Burden Tradeoff — SNAPSHOT

The SNAPSHOT counterpart of Figure 5 is provided here for completeness. The qualitative ordering of strategies is the same as in FnF: graph-based hybrids (AGE, GraphPartfar) occupy the high-performance/moderate-burden corner; pure uncertainty methods sit in the high-burden cluster; and No-AL has the lowest burden but moderate performance. The dispersion is tighter than in FnF because SNAPSHOT's centrality distribution is more uniform (cf. Figure 6).

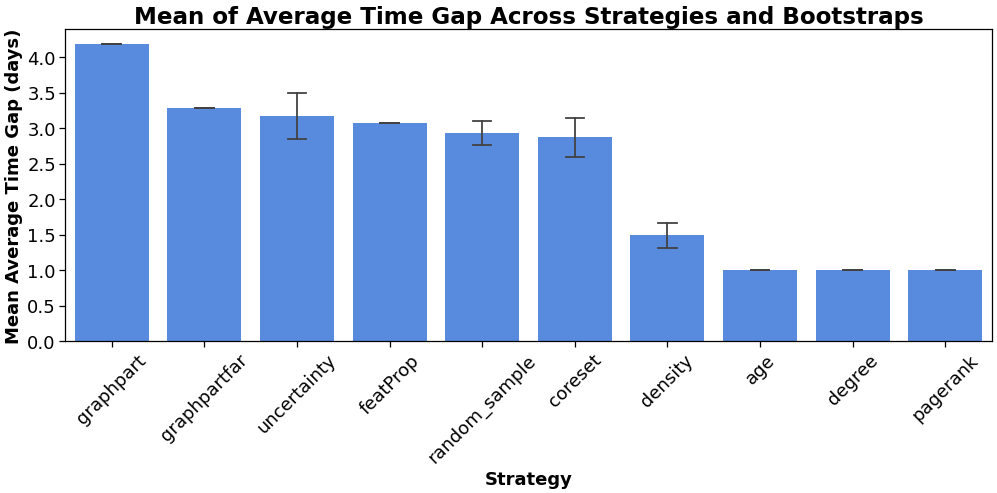

Figure 11: Mean average time gap between repeated queries for SNAPSHOT sleep classification. Longer gaps indicate lower user burden.

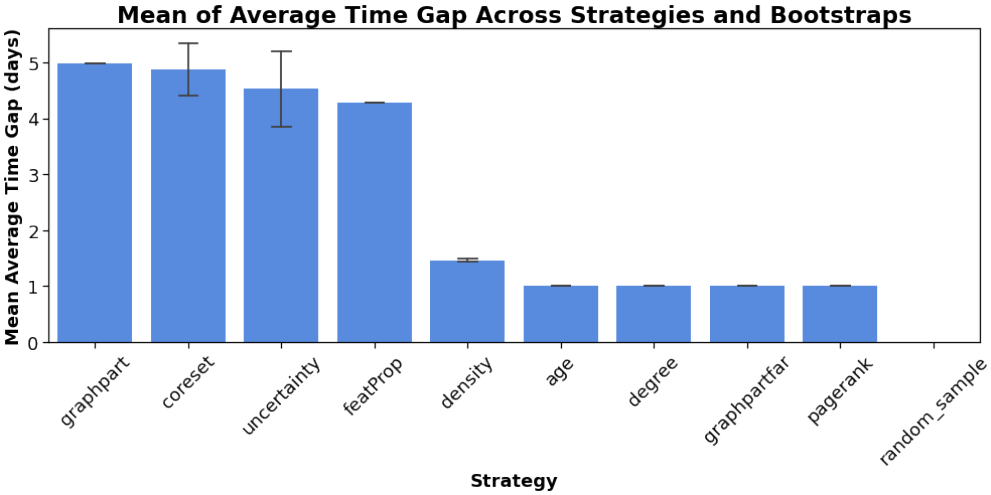

Figure 12: Mean average time gap between repeated queries for sleep classification in the FnF dataset.

D Centrality–Burden Correlations

This appendix reports the per-strategy centrality–burden heatmaps for SNAPSHOT and FnF (Figures 16 and 17), the mean normalized centrality preference across strategies on FnF (Figure 18), and the full Pearson correlation tables underlying the quantitative claims in Section 7.6.

Pearson Correlations

Tables 6 and 7 report Pearson correlations (with p-values) between standard centrality metrics (degree, betweenness, closeness, PageRank, harmonic, load, clustering) and two of GRAIL's burden metrics (Average Time Gap and Over-Exertion Score), per AL strategy on FnF sleep classification. Only statistically significant correlations ($p < 0.05$) are shown. GraphPartFar's strong negative correlations with degree, closeness, harmonic, and clustering under Average Time Gap (e.g., $r = -0.880$, $p \approx 10^{-10}$ with clustering) and strong positive correlations with the same centralities under Over-Exertion Score (e.g., $r = +0.936$, $p \approx 10^{-14}$ with clustering) directly reflect its tendency to repeatedly sample tightly-clustered hub nodes with insufficient temporal spacing.

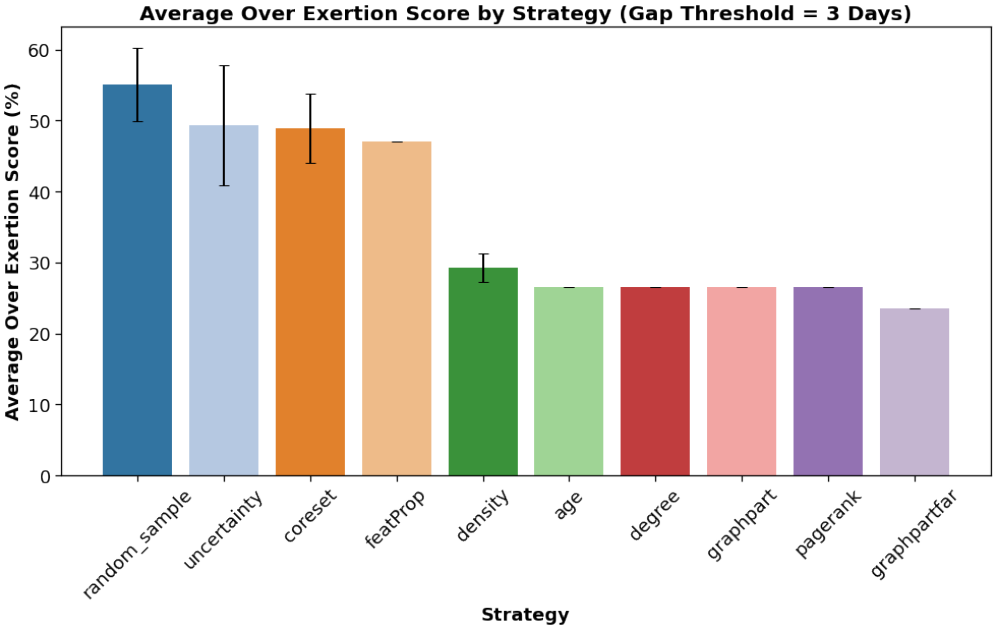

Figure 13: Average over-exertion score at a fixed 3-day gap threshold for SNAPSHOT sleep classification. Lower values correspond to lower burden.

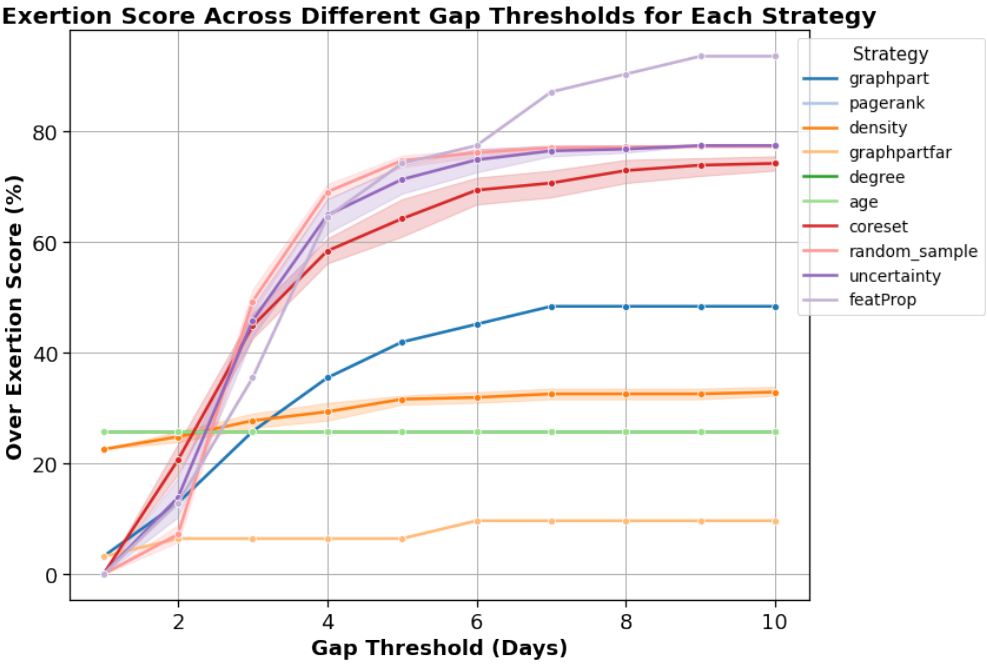

Figure 14: Variation of over-exertion score across different gap thresholds for sleep classification in the FnF dataset.

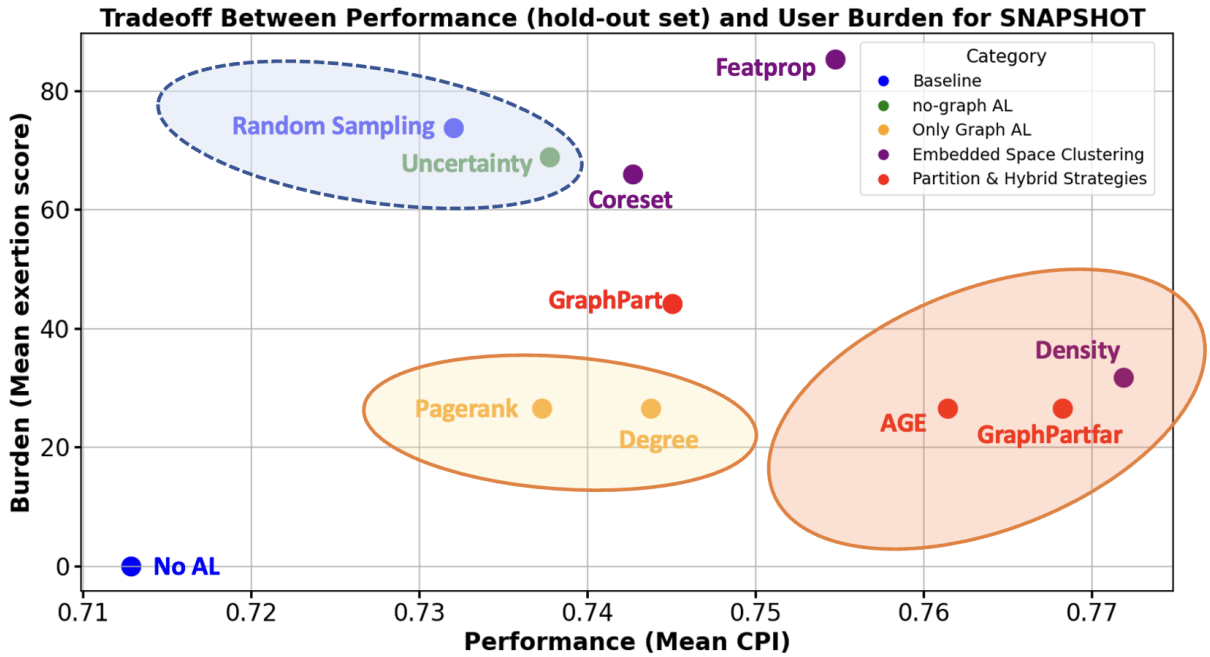

Figure 15: SNAPSHOT performance–burden comparison.

Table 6: Significant Pearson correlations (centrality × Average Time Gap) on FnF sleep classification. Positive values indicate the strategy produces longer time gaps when sampling more central nodes; negative values indicate the strategy samples central nodes with shorter gaps (over-querying).

Strategy	**Centrality**	**Pearson** r	p**-value**
GraphPart	degree	0.341	1.95×10^{-5}
GraphPart	betweenness	0.489	2.15×10^{-10}
GraphPart	closeness	0.391	7.58×10^{-7}
GraphPart	PageRank	0.428	4.67×10^{-8}
GraphPart	harmonic	0.317	7.87×10^{-5}
GraphPart	load	0.489	2.15×10^{-10}
GraphPart	clustering	-0.240	3.07×10^{-3}
GraphPartFar	degree	-0.556	1.43×10^{-3}
GraphPartFar	closeness	-0.614	3.06×10^{-4}
GraphPartFar	harmonic	-0.578	8.32×10^{-4}
GraphPartFar	clustering	-0.880	1.42×10^{-10}
Coreset	closeness	0.205	1.52×10^{-3}
Coreset	harmonic	0.209	1.20×10^{-3}
Coreset	clustering	0.303	2.05×10^{-6}
Random	PageRank	-0.142	4.70×10^{-4}
Uncertainty	degree	-0.206	1.31×10^{-3}
Uncertainty	PageRank	-0.210	1.07×10^{-3}
FeatProp	betweenness	0.213	2.56×10^{-4}
FeatProp	PageRank	-0.230	7.77×10^{-5}
FeatProp	load	0.213	2.56×10^{-4}

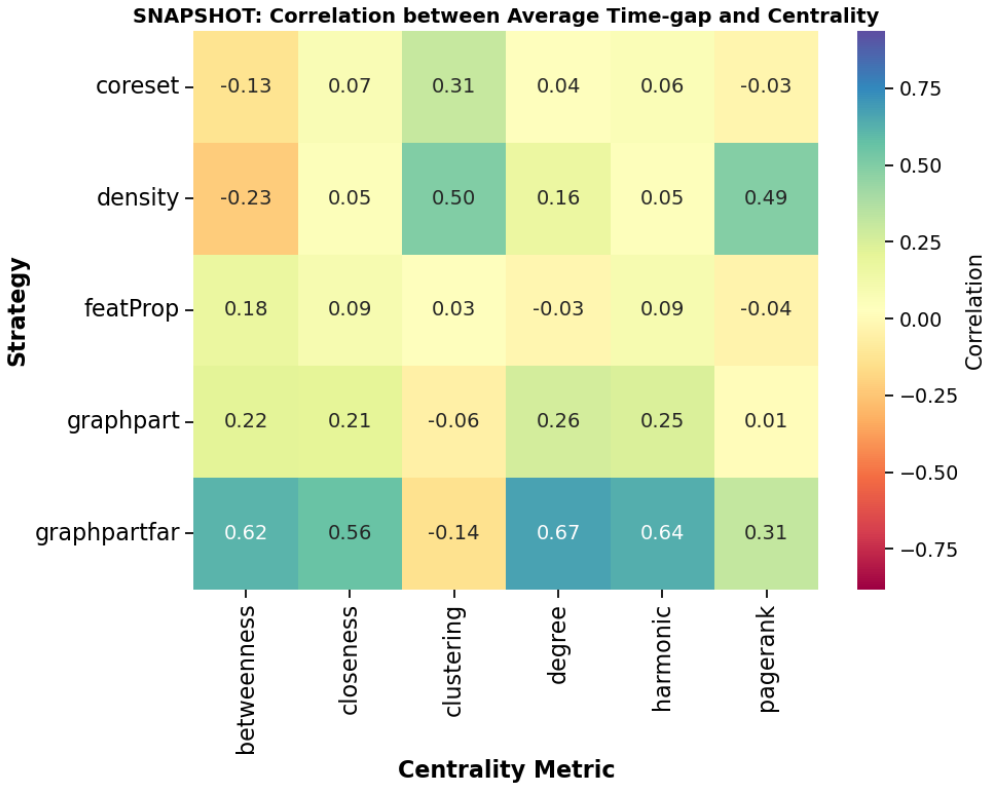

Figure 16: Correlation between centralities and burden for SNAPSHOT sleep classification for Cohort 2 using the SMS Graph (L=8, k=9, gap=5).

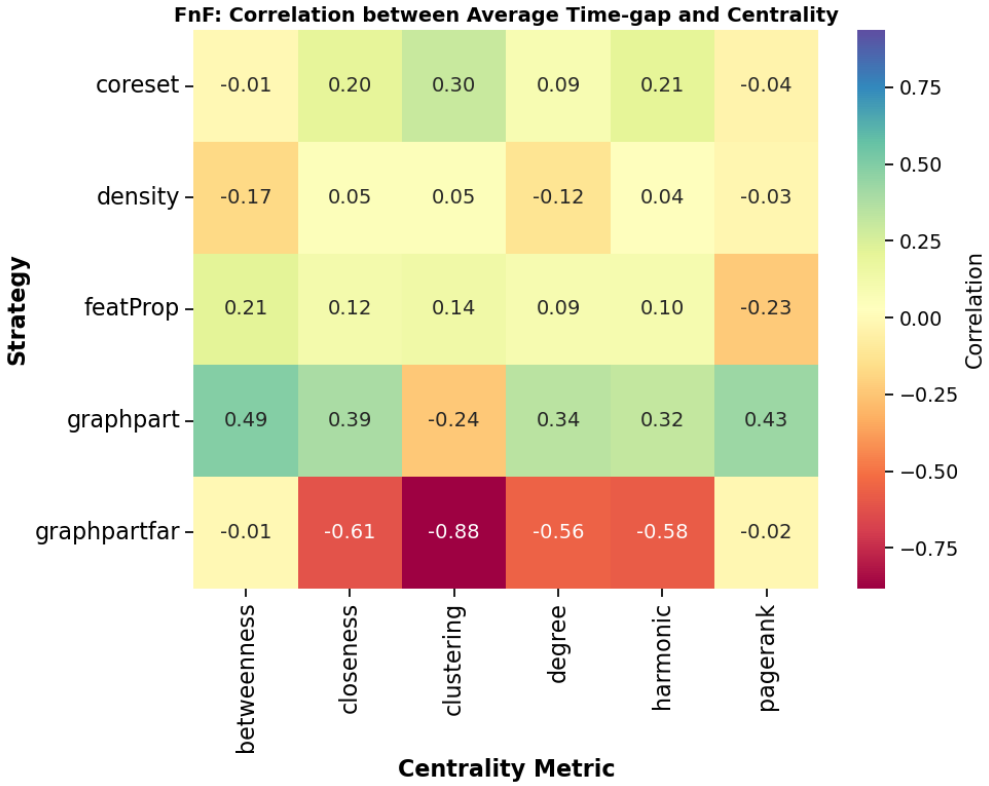

Figure 17: Correlation between centralities and burden for FnF sleep classification using the Friendship Graph (L=6, k=8, gap=5).

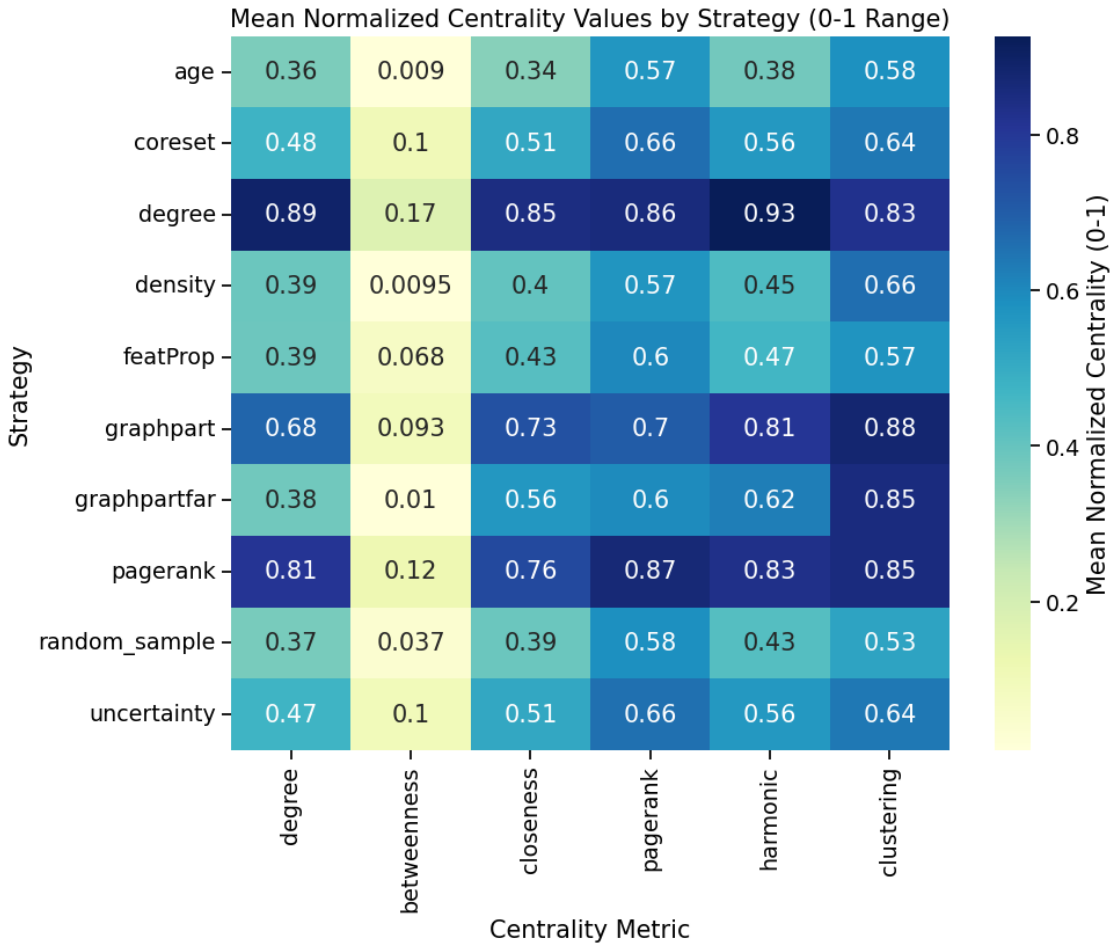

Figure 18: Mean normalized centrality values across different strategies for sleep classification for FnF dataset

E Hyperparameter Sensitivity — Additional Panels

This appendix provides the Macro-F1 panels for the k-sweep and the full L-sweep (CPI and Macro-F1, on both the unqueried pool and the held-out test set) corresponding to Section 7.7. The qualitative findings stated in the main text replicate across all panels: AGE, GraphPart, and Density improve as k increases on the held-out test set; uncertainty-based and Coreset strategies are most robust on the unqueried pool; and AGE and FeatProp are most stable across the full range of L.

F Dataset Details

Additional FnF preprocessing details: FnF required substantially more preprocessing and feature engineering effort because the dataset is released entirely in heterogeneous, raw-sensor formats with no unified schema. Each sensing stream (calls, SMS, Bluetooth, accelerometer, and location) contains irregular timestamps, missing segments, device-specific artifacts, and inconsistent logging conventions across users and phones.

To construct a usable daily time series, we first performed multi-stream cleaning that removed duplicated entries, reordered events with inconsistent timestamps, filtered negative or zero-duration events, repaired broken location tuples, and merged fragmented call/SMS logs generated by unstable device states. Because different phones recorded events with different clock drift and sampling granularities, all timestamps were

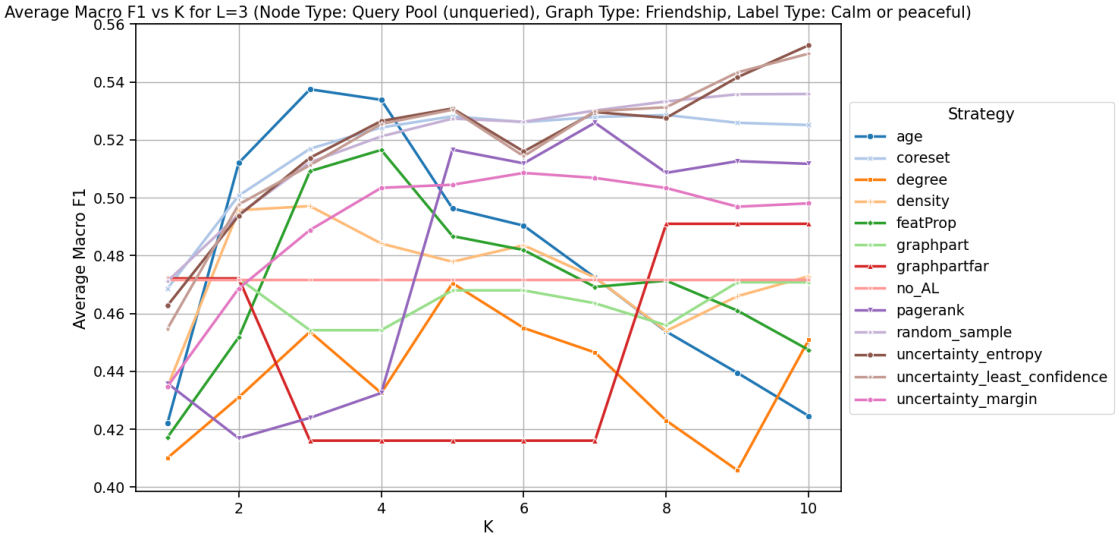

Figure 19: Macro-F1 on unqueried pool vs. k (FnF, $L = 3$).

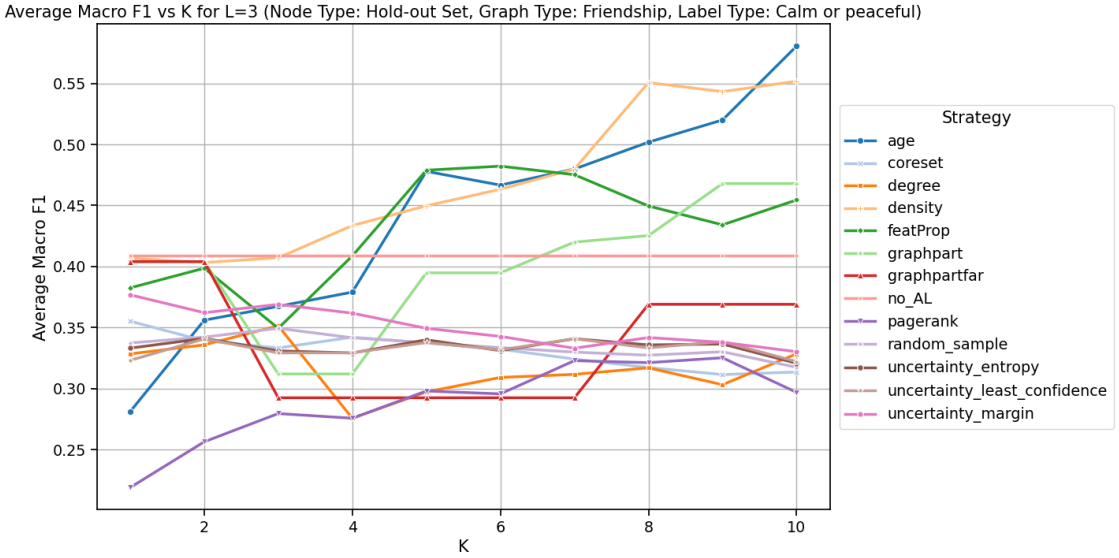

Figure 20: Macro-F1 on hold-out test vs. k (FnF, $L = 3$).

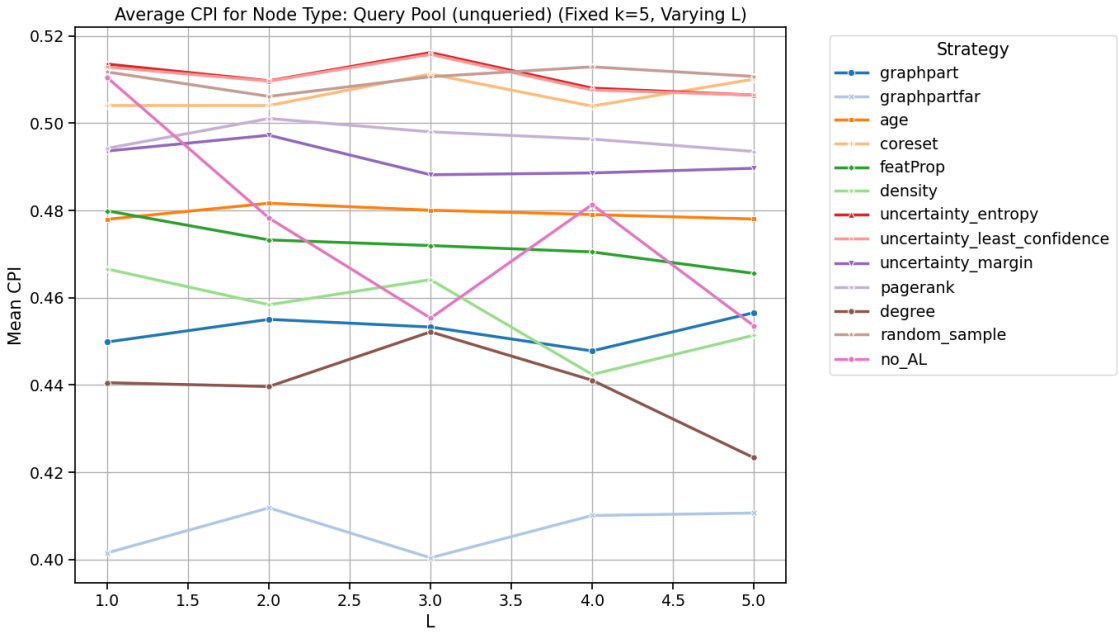

Figure 21: CPI on unqueried pool vs. L (FnF, $k = 5$).

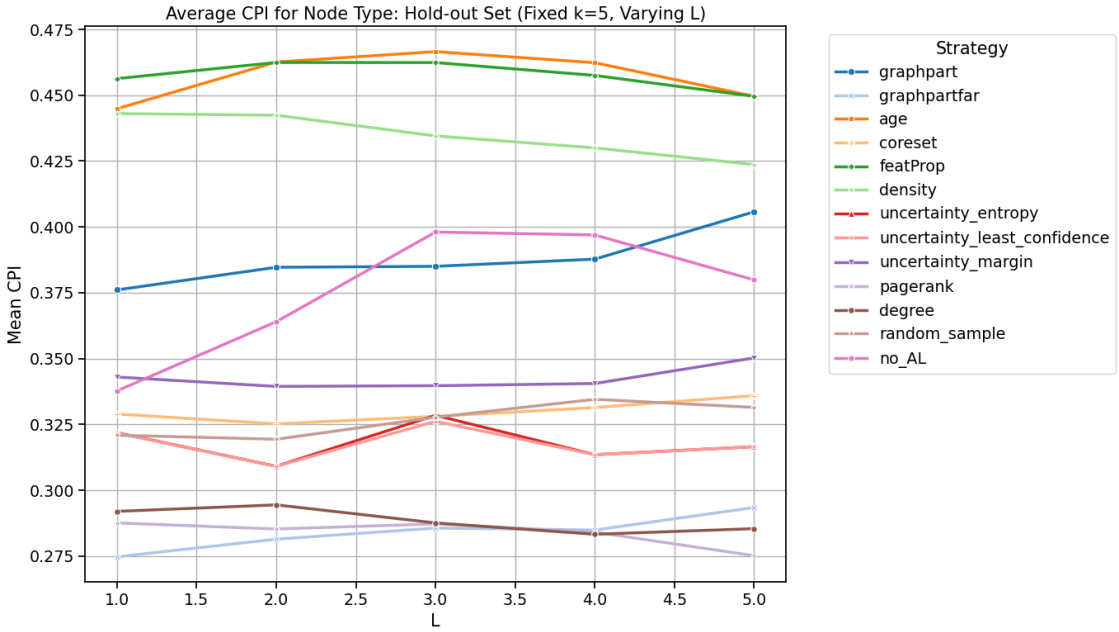

Figure 22: CPI on hold-out test vs. L (FnF, $k = 5$).

Table 7: Significant Pearson correlations (centrality × Over-Exertion Score) on FnF sleep classification. Positive values indicate higher over-exertion on more central nodes; negative values indicate lower over-exertion on central nodes (i.e., the strategy avoids over-querying hubs).

Strategy	**Centrality**	**Pearson** r	p**-value**
GraphPart	degree	-0.316	8.16×10^{-5}
GraphPart	betweenness	-0.538	1.26×10^{-12}
GraphPart	closeness	-0.418	1.00×10^{-7}
GraphPart	PageRank	-0.264	1.08×10^{-3}
GraphPart	harmonic	-0.340	2.08×10^{-5}
GraphPart	load	-0.538	1.26×10^{-12}
GraphPartFar	degree	0.441	1.48×10^{-2}
GraphPartFar	closeness	0.504	4.50×10^{-3}
GraphPartFar	harmonic	0.464	9.75×10^{-3}
GraphPartFar	clustering	0.936	3.54×10^{-14}
Coreset	degree	-0.196	2.49×10^{-3}
Coreset	closeness	-0.270	2.57×10^{-5}
Coreset	harmonic	-0.272	2.12×10^{-5}
Coreset	clustering	-0.333	1.57×10^{-7}
FeatProp	degree	-0.140	1.73×10^{-2}
FeatProp	betweenness	-0.308	8.98×10^{-8}
FeatProp	closeness	-0.240	3.51×10^{-5}
FeatProp	PageRank	0.211	2.95×10^{-4}
FeatProp	harmonic	-0.224	1.18×10^{-4}
FeatProp	load	-0.308	8.98×10^{-8}
FeatProp	clustering	-0.227	9.46×10^{-5}

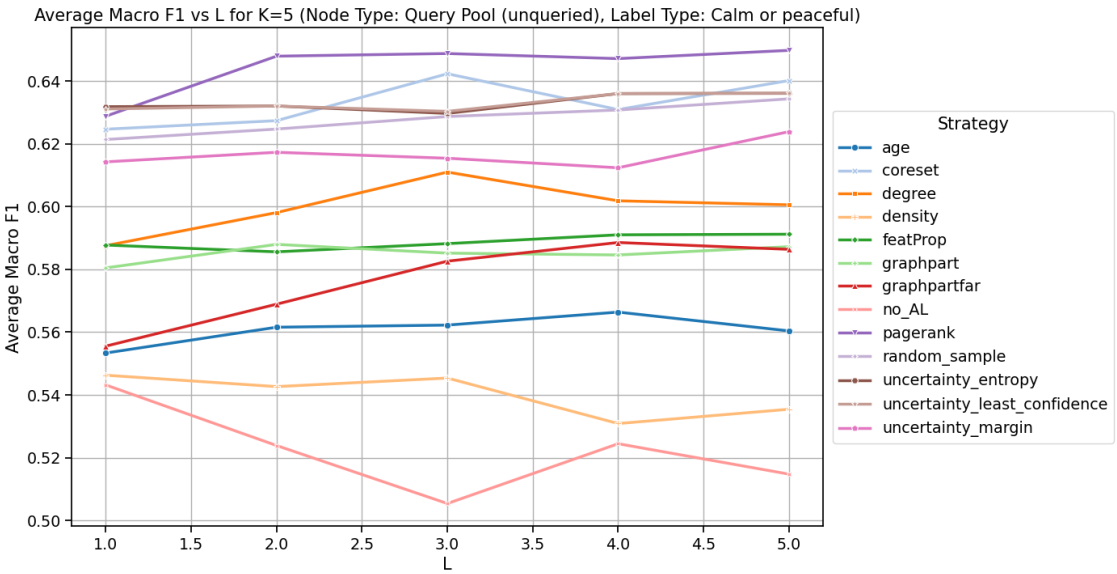

Figure 23: Macro-F1 on unqueried pool vs. L (FnF, $k = 5$).

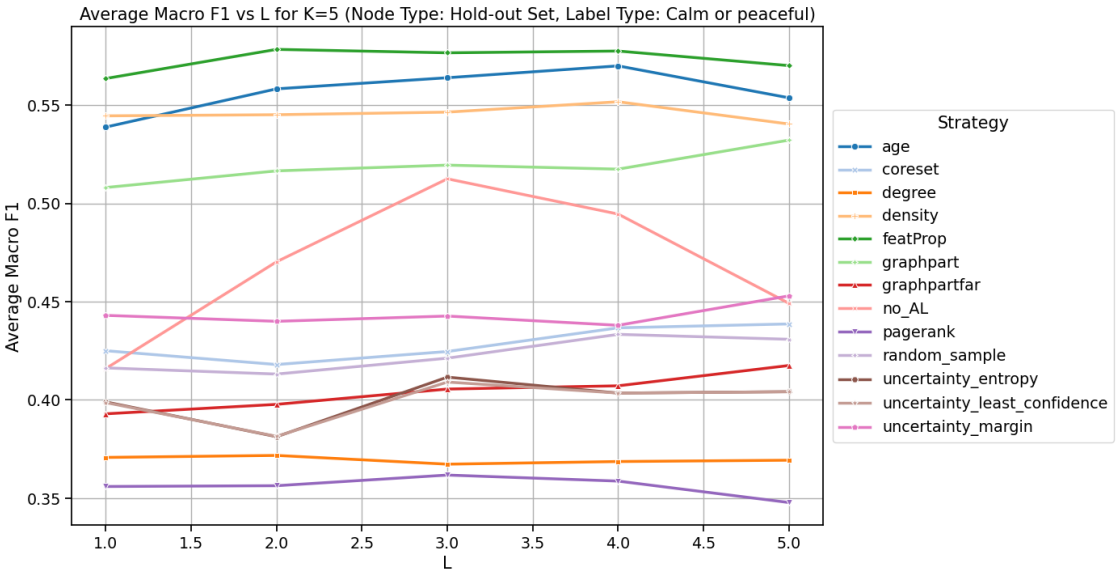

Figure 24: Macro-F1 on hold-out test vs. L (FnF, $k = 5$).

normalized into a unified timezone and corrected for multi-hour offsets by detecting improbable overnight activity spikes.

Next, we performed temporal alignment of all streams onto a shared daily grid. This required resolving high-frequency jitter by merging events that occurred within small temporal gaps (5–30 seconds depending on modality), applying rolling-window smoothing to continuous signals such as accelerometer magnitude, and interpolating short missing segments for modalities where gaps were caused by OS-level batching (e.g., Bluetooth scans). Continuous sensor streams were then aggregated into daily descriptors such as activity histograms, acceleration moments, encounter counts, unique contacts, mobility entropy, call/SMS frequency, inter-event intervals, and screen-use statistics. Because different modalities exhibit different dynamic ranges and distributions, we applied modality-specific normalization using either z-normalization or min–max scaling depending on distributional skew.

Graph construction required an additional layer of processing. For communication-based edges, we built symmetric dyads by detecting reciprocal call/SMS exchanges in a sliding window, collapsing multi-edge temporal interactions into weighted undirected edges. FnF also provides a survey-derived friendship graph, which is considerably denser and more stable over time; we use this graph in our evaluations because it offers a more reliable representation of the long-term social structure.

Finally, all behavioral labels (sleep duration and mood) were independently cleaned, aligned to the daily window, and binarized. FnF's survey timestamps do not always coincide with sensor logs, so labels were matched to sensor days using nearest-neighbor temporal alignment with bounded offset thresholds. To support cross-dataset comparability, sleep and mood labels were mapped to the same binary scheme used for SNAPSHOT. Additional details on feature-engineering and full preprocessing are provided in the accompanying repository, enabling readers to reproduce our pipeline and experiment with alternative graph constructions and feature sets.