Detecting Illicit Activity on Cryptocurrency Networks

Abstract

Detecting illicit activity on cryptocurrency networks is challenging due to the high cost of expert labeling and the limited availability of reliable ground truth. Although blockchain data is publicly available, the scarcity of reliable labels limits the performance of supervised learning models. In this project, we investigate whether GNN combined with an Active Learning framework can improve the detection of illegal Bitcoin transactions under strict labeling constraints. Using the Elliptic dataset, we model and evaluate several architectures, including GCN, MLP, EvolveGCN and DySAT, alongside multiple AL query strategies such as entropy sampling, certainty-oriented minority class sampling (CMCS), and temporal-sequential acquisition.

Our goal is to measure how efficiently different querying policies identify informative nodes and improve minority-class performance, focusing on metrics robust to severe class imbalance such as F1-illicit and PR-AUC. Our results show that feature-based models outperform GNNs on Elliptic, while Active Learning reliably improves label efficiency.

Introduction

Cryptocurrencies have created a new type of financial system where transactions are open to everyone, visible to the public, and cannot be changed once recorded. Despite this openness, distinguishing between licit and illicit activity remains a central challenge for regulators, financial institutions, and law-enforcement agencies. Determining whether a transfer is linked to money laundering, fraud, or other criminal behavior typically requires costly manual investigation and specialized domain knowledge. As a result, high-quality labels are scarce and expensive, creating a substantial limitation for supervised machine-learning approaches.

Existing methods for blockchain transaction classification often rely on either feature-based models (e.g., MLPs, tree-based classifiers), or GNNs, which leverage the relational structure between transactions. While GNNs have demonstrated strong potential in capturing propagation patterns or suspicious fund flows, they still depend heavily on labeled data. In practice, most prior work assumes access to all labels in advance and does not account for the limited supervision available in real AML investigations.

To address this gap, our project explores a hybrid approach that combines GNN-based modeling with Active Learning. Rather than relying on a fixed set of labels, AL aims to identify the most informative transactions to label next, thus maximizing performance while minimizing labeling cost. We evaluate several AL query strategies and examine how they interact with different model classes.

This design allows us to isolate how much of the performance gain comes from the graph structure itself and how much comes from selective label acquisition – which leads us to the central research question of whether combining these two ideas can meaningfully improve illicit-transaction detection under limited supervision. Our study focuses on three main components:

1. We model the Elliptic dataset as a directed and undirected temporal graph and systematically compare multiple GNN architectures against non-graph baselines.
2. We integrate and analyze several Active Learning strategies tailored to imbalanced, temporally evolving graphs.

3. We quantify the improvement in performance per unit of labeling budget, focusing on minority-class metrics such as F1-illicit and PR-AUC.

- link to GitHub repository

Related Work

Throughout the project, we relied on several studies conducted in the field, focusing on approaches that used basic versions of Graph Neural Networks (GNNs), classical machine learning models, as well as more advanced models based on either GNNs or Active Learning (AL). The goal of these models is to detect illegal activities in the Bitcoin network.

The publication of the Elliptic dataset, which provides a large-scale transaction graph, marked a breakthrough in this research area. Weber et al. demonstrated how these data were used to compare classical machine learning models with graph-based models, showing that GCN achieved strong performance by leveraging the structural and temporal relationships within the graph – relationships that classical models fail to capture. These findings paved the way for deeper exploration of GNNs in the context of Anti-Money Laundering (AML) and illicit transaction detection. [1]

Following this line of research, later works shifted toward representation learning on dynamic graphs. The EvolveGCN model introduced a mechanism in which the parameters of a GCN evolve over time using an RNN, allowing the network to adapt to temporal changes without relying on fixed node embeddings. This approach demonstrated high performance in dynamic environments, making it particularly suitable for temporal graphs such as those representing cryptocurrency transactions. [2]

Another related approach is DySAT, which integrates self-attention mechanisms across both structural and temporal dimensions. DySAT learns node representations that capture both the influence of neighboring nodes and their historical evolution patterns over time. The model showed significant improvements in dynamic graph prediction tasks, especially in communication and rating networks represented as discrete time snapshots; however, the Elliptic dataset was not used in that study. [3]

Moving to AL, Lorenz et al. argued that detecting illicit activities is highly challenging in the absence of labels, since illicit transactions tend to blend within clusters of legitimate activity. However, they demonstrated that by applying Active Learning, comparable performance to supervised models can be achieved with as little as 5% labeled data, effectively simulating real-world conditions where manual labeling is limited. [4]

Another relevant direction is minority-oriented Active Learning. Aggarwal et al. showed that standard uncertainty sampling tends to oversample majority classes in imbalanced datasets, and proposed strategies that explicitly prioritize samples predicted as minority. Following this insight, we incorporate Certainty-Oriented Minority Class Sampling (CMCS) as one of our acquisition functions to better handle the strong class imbalance in Elliptic.[5]

Unlike previous works that examined these approaches separately, our study explores the joint contribution of combining GNN and AL, aiming to improve both the accuracy and efficiency of detecting illegal cryptocurrency transactions.

Preliminaries and Problem Definition

We represent the Bitcoin transaction network as a directed temporal graph:
$G = \{G_1, G_2, \dots, G_T\}$, where each snapshot $G_t = (V_t, E_t, X_t)$ contains the set of transactions at time step t, directed edges representing fund flows, and node features. Each node $v \in V_t$ is associated with a binary label $y_v \in \{0,1\}$, where 1 indicates an illicit transaction. Only a small subset of nodes $L \subset V$ is initially labelled, while the rest remains unlabeled.

The learning task is node classification: given the temporal graph G and node features X, learn a function $f_\theta: V \to \hat{Y}$, that assigns each node v a predicted label $\hat{y}_v$. Importantly, the model is trained on the entire graph, allowing it to learn from both labeled and unlabeled nodes through message passing and neighborhood aggregation.

However, evaluation and comparison with the ground truth are performed only on the labeled nodes L, to enable quantitative assessment of classification performance. To mitigate the limited availability of labels, we incorporate an AL component that iteratively selects the most informative unlabeled nodes for manual annotation. This hybrid design – combining GNN with AL, aims to maximize performance while minimizing annotation cost.

Methodology

In this project, we evaluate how graph structure and temporal information contribute to illicit-transaction detection by comparing four model architectures: a non-graph MLP baseline, a static GCN, and two temporal GNNs (EvolveGCN-O and DySAT). This set of models allows us to isolate the added value of graph connectivity and dynamic modeling relative to feature-only learning. In addition, we evaluate multiple graph constructions – such as directed vs. undirected edges and local-only vs. full feature sets – to examine how the choice of graph representation affects model performance. These variants allow us to decouple structural information, neighborhood-derived statistics, and temporal dynamics.

To further study performance under limited supervision, we integrate an AL framework on top of each model. By applying multiple acquisition strategies, including certainty-based, minority-oriented, and time-aware methods – we examine how selective label querying interacts with graph-based representations.

We compare four models that differ in how they represent and learn from the graph's structure and temporal evolution.

i. MLP – serves as a non-graph baseline. It treats each transaction independently, relying solely on its node features without considering the network structure. This model allows us to evaluate the added value of incorporating graph topology into the learning process.
ii. GCN – extends convolutional operations from images to graphs. Each node updates its representation by aggregating features from its neighboring nodes – a process known as message passing. This enables the model to capture how a transaction's risk level depends on the behavior of adjacent transactions. GCN provides a strong baseline for modeling local structural dependencies, though it assumes the graph is static and does not evolve over time.
iii. EvolveGCN-O – introduces temporal awareness by allowing the model parameters themselves to evolve over time. Using a recurrent neural mechanism, the GCN's internal weights are updated dynamically across graph snapshots. This approach learns how patterns of information flow change – for instance, how illicit behaviors shift over different time periods – making it suitable for dynamic, real-world financial systems.

Code based on GitHub repository published in article [2]. [6]

iv. DySAT – employs self-attention mechanisms to learn which relationships in the graph and which time points are most relevant. It applies:
   - Structural attention: learning the relative importance of each neighbor at a given time step.
   - Temporal attention: identifying which past states are most informative for understanding current behavior.

   By combining these two forms of attention, DySAT captures long-range temporal dependencies and the contextual significance of connections – essential for uncovering hidden laundering patterns that unfold gradually.

Given the limited availability of labeled data in financial systems, we integrate an AL framework to optimize which transactions are selected for manual labeling. The process follows an iterative cycle:

i. Initialize with a small, labeled subset of transactions.
ii. Train the model on the labeled subset while performing message passing over the full graph.
iii. Evaluate informativeness of the unlabeled nodes.
iv. Select a batch of nodes according to a chosen acquisition strategy.
v. Add the selected samples to the labeled pool and repeat.

Query Strategies:

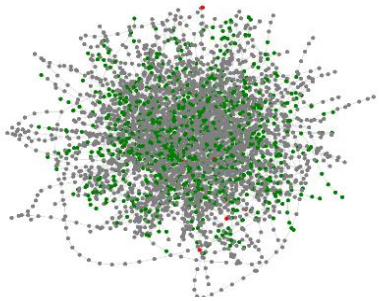
i. Random Sampling (Baseline): selects unlabeled nodes uniformly at random and serves as a control condition for evaluating whether more sophisticated acquisition strategies provide an advantage.
ii. Entropy Sampling: selects nodes with the highest predictive uncertainty, under the assumption that uncertain samples lie near the decision boundary and therefore provide maximal information gain when labeled.
iii. CMCS (Certainty-Oriented Minority Class Sampling): prioritizes high-certainty samples predicted as illicit, based on the observation that in highly imbalanced datasets, uncertain nodes are often dominated by the majority class. CMCS increases the likelihood of acquiring informative minority-class examples, which are crucial for improving illicit-class performance.
iv. Sequential (Temporal-Aware): selects unlabeled nodes in chronological blocks, ensuring that the model only acquires information available at that point in time. This mimics real investigative workflows and respects the temporal constraints of the Elliptic dataset, where future information cannot be used retroactively. Based on CMCS inside blocks.

Our approach captures both structural and temporal dependencies in transaction data.

Experimental Setup

We use the Elliptic dataset, a publicly available Bitcoin transaction graph published in Kaggle. The dataset contains 203,769 transactions, 234,355 directed edges, where each edge represents the flow of Bitcoin from one transaction to the next. Each node is described by 166 features: 94 local transaction-level features (e.g., number of inputs/outputs, fees, volumes, timestamps), and 72 aggregated one-hop neighbors features, constructed by summarizing statistics of adjacent transactions. Most preprocessing (including normalization and handling of missing values) is performed by Elliptic prior to release. Because of intellectual property restrictions, the dataset providers did not release details about how the features were created. Only the final standardized feature vectors are available, without information on what each feature represents.

The dataset spans 49 discrete time steps, each representing roughly two weeks of blockchain activity, transactions within a step form a connected component and no edges exist between steps, yielding 49 independent temporal snapshots. Each node is labeled as *licit* (21%), *illicit (2%)*, or *unknown (77%)*.



Example to individual time step graph:

For all experiments, the data is split chronologically to prevent future-data leakage and to match real investigative constraints: training time steps 1–34, validation time steps 35–41, testing time steps 42–49. This ensures that models are trained only on information available up to a given point in time, enabling a realistic evaluation of temporal generalization.

To evaluate the robustness of our models, we construct multiple versions of the transaction graph.

First, we generate both directed and undirected graphs: directed edges preserve the actual flow of Bitcoin, while undirected edges allow information to propagate in both directions, capturing suspicious neighborhood patterns even when the risk signal does not follow the monetary flow.

Second, we consider two feature configurations: a local-only representation and the full feature set including aggregated neighbor information. This allow us to assess whether incorporating graph-derived features provides additional predictive value.

To assess the contribution of graph structure and temporal modeling, we evaluate four architectures: MLP, GCN, EvolveGCN-O, and DySAT. MLP serves as our non-graph baseline, allowing us to measure the benefit of incorporating relational information.

For the Active Learning component, we evaluate four acquisition strategies: Random Sampling, Entropy Sampling, CMCS and a Sequential strategy based on chronological label acquisition. Random and Entropy serve as standard AL baselines, while CMCS and Sequential provide minority-aware and time-aware alternatives. For each architecture, we include its "passive" version as the baseline for comparing against its AL version.

About hyperparameters, all models were optimized with Adam, using configuration values specified in the project's YAML files. MLP used two hidden layers (128–64) with a learning rate of 0.05 and dropout p=0.5; GCN used 64 hidden units with lr=0.02; EvolveGCN-O used 128 hidden units with lr=0.001; and DySAT followed the recommended settings from its original paper (128-dimensional structural embeddings, 4 temporal attention heads, and L=5 layers). For passive learning, all models were trained with a maximum of 170 epochs and patience=15. For Active Learning, runtime constraints required shorter training schedules: up to 50 epochs per AL round for MLP, GCN, and EvolveGCN-O. (DySAT was excluded from AL due to its significantly higher computational cost)

The AL setup used a batch size of 100, a total labeling budget of 1000, an initial seed batch of 20 samples (randomly selected while ensuring at least one illicit node), and class-weighted cross-entropy computed from the labeled set at each round.

Since the classification task involves highly imbalanced data, choosing an appropriate decision threshold is critical. Rather than relying on the default value of 0.5, we compute model probabilities for each node and determine an optimal threshold that maximizes the F1 score of the illicit class on the validation set.

Formally, given the predicted probability $\hat{p}_i$ for each node i, the model assigns a positive (illicit) label if $\hat{p}_i \geq \tau$, where τ is the decision threshold. We evaluate F1 across a range of thresholds $\tau \in [0,1]$ and select:

$$\tau^* = \arg\max_{\tau} F1_{\text{illicit}}(\tau)$$

This ensures that the model's decision boundary is calibrated to balance precision and recall specifically for the minority class. The chosen threshold τ^* from the validation set is then used for evaluation on the test set and across all Active Learning iterations to maintain consistency.

Model evaluation is performed exclusively on labeled test nodes, since only these provide reliable ground truth. We report three complementary metrics:

i. F1 Score (Illicit Class): Used to evaluate all four models under standard (non-AL) training. Measures the harmonic mean of precision and recall for the minority (illicit) class. This metric is crucial due to the extreme class imbalance (2% illicit), where accuracy is not informative.
ii. PR-AUC (Precision–Recall Area Under Curve): reflects performance under severe class imbalance. Reflects the model's ability to identify illicit nodes at different confidence thresholds.
iii. Performance vs. Labeling Budget – In the Active Learning setting, we track F1-illicit as a function of the number of labeled samples acquired. This allows us to directly compare how quickly different querying strategies improve minority-class detection under identical annotation costs.

Together, these metrics capture both predictive performance under class imbalance and the efficiency of label usage in the Active Learning setting.

Evaluation

Passive Learning Results:

We first evaluate all four architectures under standard passive learning. Table 1 shows that the MLP baseline achieves the highest illicit-class F1 (0.526 on average), as well achieve the best PR-AUC, substantially outperforming all graph-based models. This suggests that the Elliptic dataset's engineered node features already encode most of the discriminative signal, matching observations from prior work [1]. In contrast, GNNs underperform: GCN achieves only 0.282, EvolveGCN-O reaches 0.326, and DySAT performs worst (0.242). These results indicate that message passing may dilute strong local features, particularly in noisy or weakly informative graph structures.

Table 1: Passive Model Comparison

Model	Rank	Test F1	AUPRC
MLP	2.5	0.526	0.450
EvolveGCN	8.75	0.326	0.273
GCN	10.5	0.282	0.238
DySAT	12.25	0.242	0.250

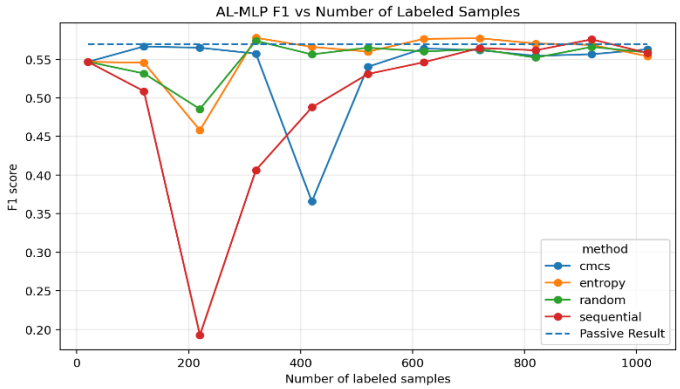
Effect of Feature Set (LOCAL vs. ALL):

The effect of using the full feature set differs across models. MLP and DySAT perform slight better with LOCAL features: MLP treats the aggregated neighbor statistics as noise, while DySAT already learns rich neighborhood and temporal patterns through attention, making the engineered aggregates redundant. GCN shows almost no difference between LOCAL and ALL, as both feature sets provide similar value under its simple mean-based aggregation. In contrast, EvolveGCN performs better with ALL features (+15%): although it applies the same GCN layer within each snapshot, its temporal mechanism updates weights over time rather than learning explicit structural patterns, making it more dependent on the richer neighborhood information provided by the aggregated features. These trends are shown in Appendix Figure 1.

Effect of Graph Direction (LOCAL vs. ALL):

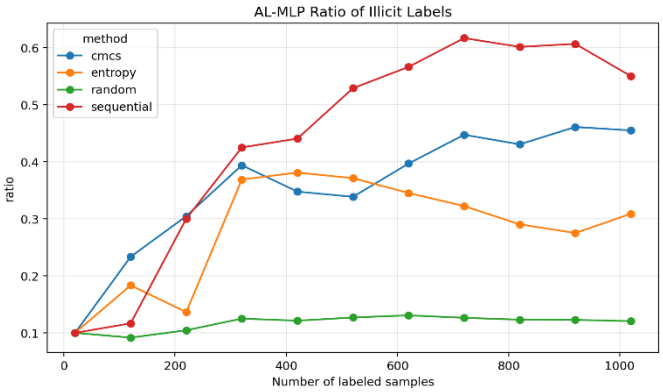
DySAT shows a clear advantage (+10%) when using a *directed* graph, as the direction of money flow provides a meaningful temporal and structural cue that its attention mechanisms can effectively exploit. In contrast, both GCN and EvolveGCN perform better on *undirected* graphs: allowing information to flow in both directions enlarges their effective neighborhoods and supplies richer structural context than the sparse, one-way connectivity of the directed version. The MLP baseline, which ignores graph structure entirely, is only minimally affected. These trends are shown in Appendix Figure 2.

Active Learning Results:



Plot 1: MLP F1 Score by labeled samples

First, we used the local feature set and direct graph mode to reduce permutations of the experiment. Overall, the Active Learning experiments demonstrate clear value across all three models (DySAT excluded): with relatively few labeled samples, all strategies can reach the passive-learning F1, confirming that AL is effective for this task. Among the models, MLP (plot 1) shows the fastest and cleanest convergence, stabilizing around its passive baseline after 600–800 labeled nodes. In contrast, both GCN and EvolveGCN-O (appendix figures 3-4) remain highly unstable across rounds, making it difficult to draw strong conclusions about convergence; however, EvolveGCN-O consistently surpasses its passive baseline under all strategies, likely because AL provides smaller and temporally coherent labeled subsets that better suit its evolving temporal architecture. Regarding the acquisition strategies themselves, MLP exhibits almost no difference between methods, and the instability in GCN and EvolveGCN-O prevents identifying a reliably superior strategy. Overall, the results suggest that AL improves label efficiency across models, but differences between AL methods are not robust enough to support strong comparative claims.



Plot 2: Ratio of illicit labels in MLP model

From the illicit-ratio analysis, in all Ratio plots (MLP- plot 2, GCN and EvolveGCN appendix figures 5-6) both CMCS and Sequential (based on cmcs) clearly succeed at their intended goal: they aggressively surface illicit nodes throughout the AL rounds, as reflected by the sharp rise in their illicit-label ratios. However, the intuitive hypothesis that such artificially accelerated class balancing would translate into higher F1 does not hold. The summary table (table 3 in appendix) shows the opposite trend: artificially forcing balance does not improve model performance, and in fact often destabilizes it. Instead, Random shows the strongest positive correlations of label balance with F1. This suggests that models benefit more from a "naturally" evolving labeled distribution than from an over-corrected or aggressively rebalanced one. In other words, exposing the model to illicit nodes is necessary, but excessive or abrupt enrichment does not yield better performance and may even hinder it.

Conclusion, Limitations and Future Work

Our results show that feature-based modeling consistently outperforms graph-based approaches on the Elliptic dataset. The MLP baseline achieves the highest and most stable performance, likely because the engineered features already encode key neighborhood information, leaving little additional signal for GNNs to extract. Although we initially hypothesized that temporal GNNs combined with AL might better exploit the graph's evolving structure, this expectation didn't materialize: both GCN and EvolveGCN-O exhibited substantial volatility and struggled to leverage the graph effectively. Despite this, Active Learning itself proved highly effective-across all models, AL enabled reaching passive-learning F1 with far fewer labeled samples, demonstrating strong label-efficiency benefits even when the underlying architecture is unstable. A surprising outcome is that strategies designed to aggressively rebalance the minority class didn't translate into improved predictive performance, suggesting that overly manipulated label distributions may interfere with the models' learning dynamics.

Our study has several limitations. The most significant is runtime: computational constraints restricted both the number of AL rounds and the number of epochs per round, especially for the graph-based models, which likely contributed to their noisy learning curves and prevented us from including DySAT-the strongest temporal GNN-in the AL experiments. In addition, all results are based on a single dataset (Elliptic), a fixed temporal split, and essentially single runs per configuration.

These limitations naturally suggest several directions for future work. The first step is to enable long-horizon AL experiments with richer training budgets and more powerful temporal architecture such as DySAT, to test whether graph-based models can close the gap to feature-based baselines when adequately optimized. A second line of work is to expand the experimental evaluation itself: increasing the number of AL rounds, running multiple random seeds, and exploring alternative temporal splits and acquisition budgets. These additions would reduce variance, improve stability, and allow stronger statistical conclusions-particularly for the graph-based models, whose learning curves were highly sensitive to training noise.

Perhaps the most intriguing direction concerns the minority-class problem itself. Our results suggest that aggressively forcing minority–majority parity does not improve-and can even harm-model performance on Elliptic, raising the broader question of which models or acquisition strategies are inherently capable of detecting minority structure in noisy temporal graphs. A valuable next step is therefore to study which architecture, loss formulations, or AL policies naturally excel at discovering rare illicit patterns without relying on externally imposed balance. Another important extension is to evaluate whether the same behavior appears in other fraud detection graphs or whether Elliptic is a particularly extreme case. Such cross-dataset validation could clarify whether the failure of aggressive balancing is dataset-specific or reflects a general property of minority learning in graph-based AL settings.

References

1. Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*.
2. Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., ... & Leiserson, C. (2020, April). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 04, pp. 5363-5370).
3. Sankar, A., Wu, Y., Gou, L., Zhang, W., & Yang, H. (2018). Dynamic graph representation learning via self-attention networks. *arXiv preprint arXiv:1812.09430*.
4. Lorenz, J., Silva, M. I., Aparício, D., Ascensão, J. T., & Bizarro, P. (2020, October). Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity. In *Proceedings of the first ACM international conference on AI in finance* (pp. 1-8).
5. Aggarwal, U., Popescu, A., & Hudelot, C. (2021, January). Minority class oriented active learning for imbalanced datasets. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 9920-9927). IEEE.
6. GitHub repository of EvolveGCN model

Appendix

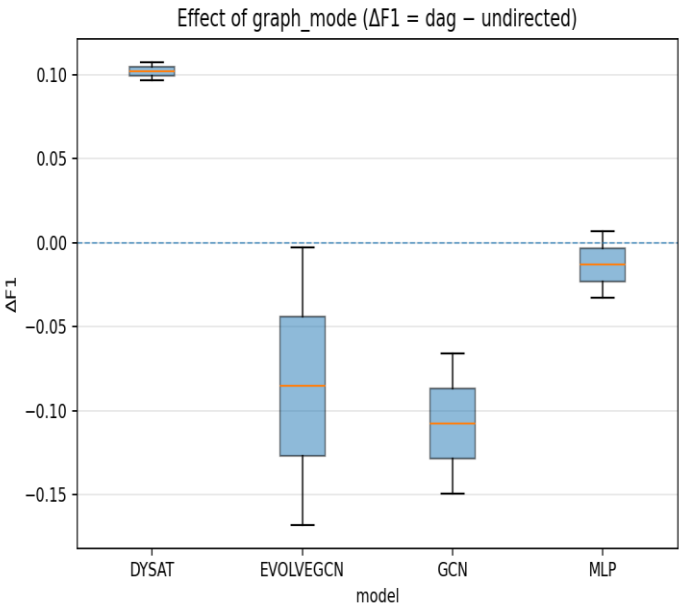


Figure 1: Boxplot presenting graph direction influence on models score. Above 0: directed graph improve the model. Below 0: undirected graph is favored.

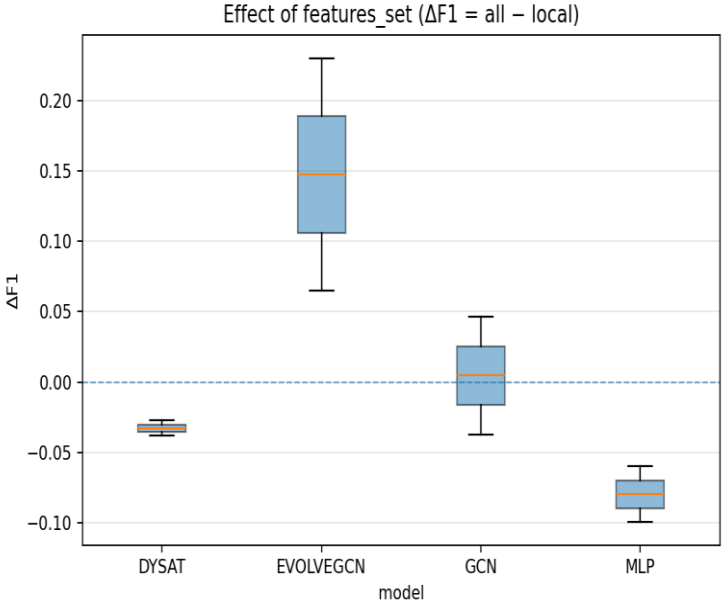


Figure 2: Boxplot presenting feature set influence on models score. Above 0: all feature sets improve the model. Below 0: local feature set is favored.

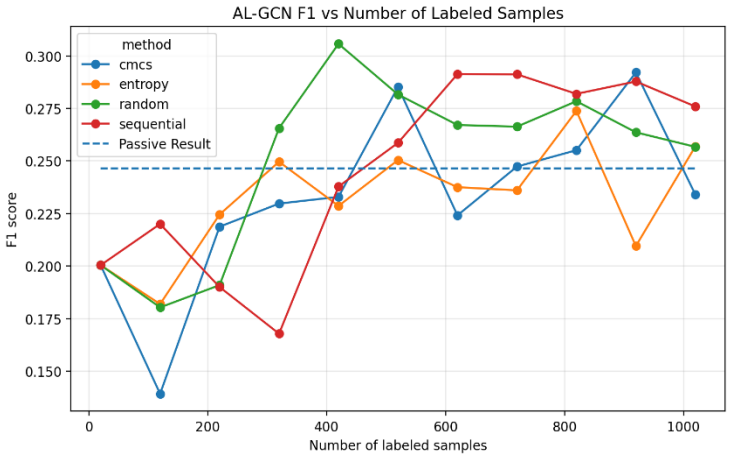


Figure 3: Plot of GCN Active Learning F1 results

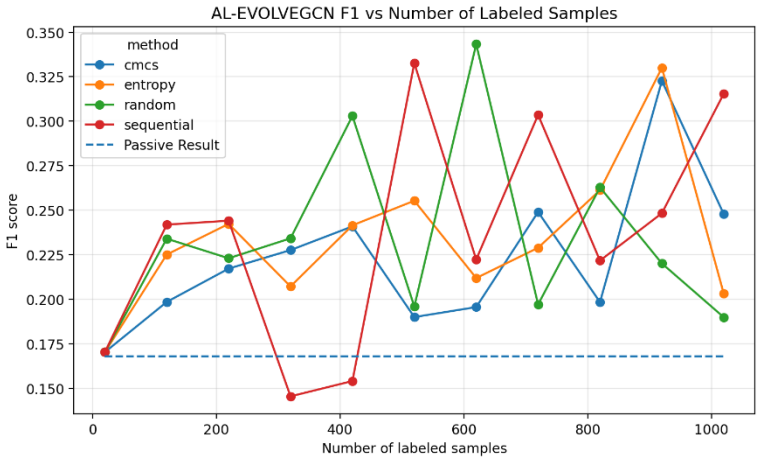


Figure 4: Plot of Evolve-GCN Active Learning F1 Results

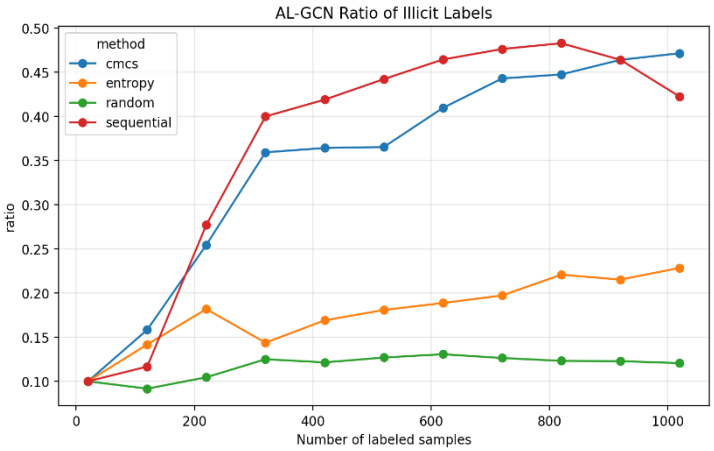


Figure 5: Ratio of Illicit Labels in GCN Active Learning

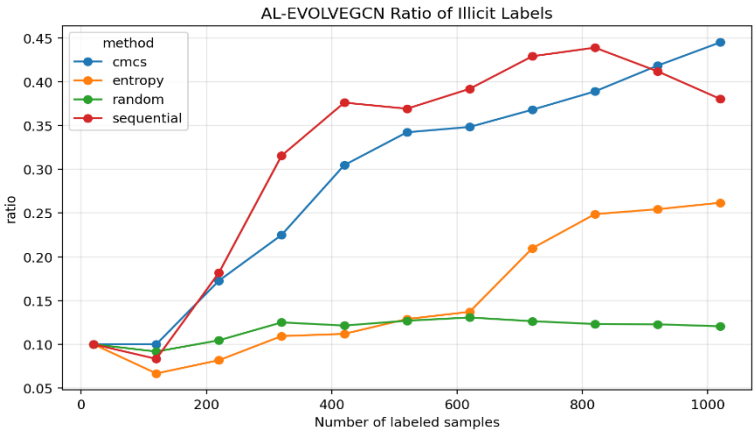


Figure 6: Ratio of Illicit Labels in EvolveGCN Active Learning

Table 2: Active Learning Method Comparison

Method	Rank	Test F1	AUPRC
sequential	4.667	0.383	0.301
cmcs	6.667	0.348	0.253
entropy	7.333	0.338	0.235
random	7.333	0.335	0.249

Table 2: Mean Rank (out of all AL runs) of ever AL method, with their F1 Test result.

Table 3: AL Methods vs Ratio

Method	Partial Corr.
random	0.592
entropy	0.218
cmcs	0.055
sequential	-0.273

Table 3: Partial correlation between F1 and the labeled-set balance ratio. Positive values indicate that a method benefits from more balanced labeled sets. Negative values indicate that increasing balance harms the method's performance. Values near zero imply no meaningful relationship.

model	features_set	graph_mode	best_val_f1	test_f1	auprc	best_epoch
MLP	local	dag	0.8576	0.5695	0.439	37
MLP	local	undirected	0.8605	0.5629	0.4893	23
MLP	all	undirected	0.759	0.5029	0.4739	18
MLP	all	dag	0.7368	0.4701	0.3971	10
EVOLVEGCN	all	undirected	0.5227	0.4009	0.3101	46
EVOLVEGCN	all	dag	0.6136	0.3982	0.3185	37
GCN	all	undirected	0.5294	0.3586	0.3154	50
EVOLVEGCN	local	undirected	0.612	0.3361	0.3832	74
GCN	local	undirected	0.5934	0.3122	0.2921	65
DYSAT	local	dag	0.5088	0.3073	0.3828	53
DYSAT	all	dag	0.5	0.2799	0.3488	43
GCN	local	dag	0.513	0.2464	0.198	49
DYSAT	local	undirected	0.36	0.2105	0.1266	2
GCN	all	dag	0.4273	0.2093	0.147	37
DYSAT	all	undirected	0.3405	0.1722	0.1403	1
EVOLVEGCN	local	dag	0.3504	0.168	0.0799	11

Table 4: Passive Learning Results by permutation

model	method	features_set	graph_mode	best_val_f1	test_f1	auprc
AL-MLP	cmcs	local	dag	0.8576	0.5629	0.414
AL-MLP	random	local	dag	0.8454	0.5588	0.4395
AL-MLP	sequential	local	dag	0.8557	0.558	0.4478
AL-MLP	entropy	local	dag	0.8544	0.5539	0.3854
AL-EVOLVEGCN	sequential	local	dag	0.5701	0.3155	0.2614
AL-GCN	sequential	local	dag	0.5605	0.2759	0.1935
AL-GCN	entropy	local	dag	0.5228	0.2569	0.2086
AL-GCN	random	local	dag	0.5058	0.2567	0.193
AL-EVOLVEGCN	cmcs	local	dag	0.4792	0.2479	0.1317
AL-GCN	cmcs	local	dag	0.508	0.234	0.2142
AL-EVOLVEGCN	entropy	local	dag	0.4684	0.2032	0.1108
AL-EVOLVEGCN	random	local	dag	0.3698	0.19	0.1142

Table 5: Active Learning Results by Model and Method