# CAFIN Supplementary Technical Appendix

**Arvindh Arun**[a;*], **Aakash Aanegola**[a], **Amul Agrawal**[a], **Ramasuri Narayanam**[b] **and Ponnurangam Kumaraguru**[a]

[a]IIIT, Hyderabad
[b]Adobe Research

**Abstract.** The following supplementary material contains additional results and analysis which are not included in the main paper for brevity.

## 1 Datasets

We use datasets from different domains to rigorously test CAFIN's performance.

**Citation networks.** We use the two most standard citation network benchmark datasets - Cora [3], and CiteSeer [1]. In both the datasets, nodes correspond to manuscripts, and the undirected edges to citations. The node feature vector for both datasets is a word dictionary indicating the distribution of words in each paper's abstract, and the node label is the domain of the paper.

**Social networks.** We use the Twitch (EN) dataset [5] to study CAFIN's efficacy on social networks. The nodes in this dataset correspond to Twitch streamers, and the edges to mutual followers. Node features contain a representation of the games played by the streamer. The node label is binary and indicates if the streamer streams mature content.

**E-commerce co-purchase networks.** We use two datasets - Amazon Photos (AMZN-P) and Amazon Computers (AMZN-C) [8]. In both datasets, the nodes correspond to products, and the edges connect co-purchased products. Node features contain the product reviews as bag-of-words, and the node label indicates the product category.

**Protein interaction networks.** Protein-Protein Interactions (PPI) [9] contains a collection of graphs (24), each an interaction network within different human tissues. Node features contain positional gene sets, motif gene sets, and immunological signatures, and the node label is a 121-dimensional vector corresponding to the gene ontology sets.

## 2 Stricter Sampling

GraphSAGE [2] requires positive and negative samples for each node for contrastive learning. They use a random node from the graph as the negative sample. We impose stricter constraints for selecting negative samples by leveraging the precomputed pairwise distances. We define a minimum distance threshold (NEG_MIN_DIST) for a node to qualify as a negative sample, i.e., the chosen negative sample must atleast be NEG_MIN_DIST hops away from the node ($d(u, v_n) \geq$ NEG_MIN_DIST). This ensures that the positive and negative samples are mutually exclusive, thus further improving the contrastive learning.

* Email: arvindh.a@research.iiit.ac.in

## 3 Landmark-based Approximate Distance Method Error Rates

CAFIN requires pairwise distances to operate and to impose a stricter sampling strategy; thus, it needs to be precomputed. The best time complexity (without approximations) for it is $O(|V|^2)$, where $|V|$ is the number of nodes in the graph. The time complexity becomes a bottleneck when we try to scale CAFIN to large graphs. To address this issue, we explore a fast approximate distance method.

As a part of our ablation study, we try out *landmark-based* distance method for approximate distance computation [4]. This method involves selecting a subset of nodes as "landmarks" and precomputing the distances of each node to those landmarks. When CAFIN requires the distance between a pair of nodes at runtime, it can be estimated quickly by combining the precomputed distances .

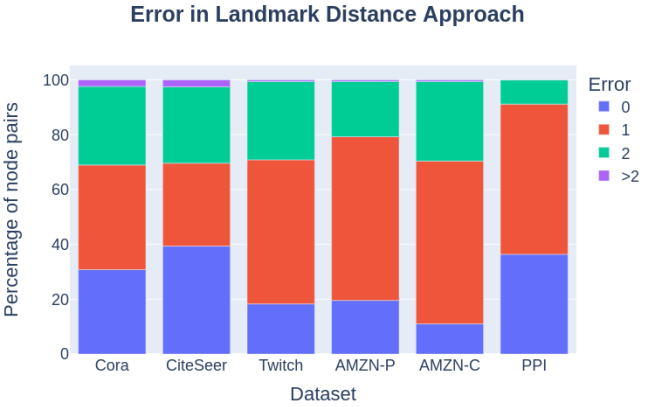


**Figure 1**: **Error in Landmark-based Distance Approach -** The figure illustrates the error of the Landmark Distance approach in calculating pairwise distances of nodes for various datasets. The color of the bars represents the error value in the predicted distance. An error of $x$ denotes that the predicted distance deviates from the actual distance by $x$.

The new preprocessing step only requires computing the distance of each node to landmarks, which can be achieved by doing a breadth-first search from each landmark node. The time complexity for the same is $O(|V| \cdot l)$, where $l$ is the number of landmark nodes.

**Table 1**: **Results of ablation studies for Node Classification:** CAFIN-P and CAFIN-N show decreases in performance when compared to CAFIN. Even for cases where high II is observed, either the CV is very high or the CA is high. Results with * were run with AdaBoost [6] classifier instead of LinearSVC.

| | (a) CAFIN-N | | | (b) CAFIN-P | | |
|---|---|---|---|---|---|---|
| **Dataset** | **II ↑ (CV ↓)** | **CA ↑** | **T ↓** | **II ↑ (CV ↓)** | **CA ↑** | **T ↓** |
| Cora | 35.23% (8.50%)* | -1.20% | 0.29 | -10.10% (7.44%) | -2.90% | INF |
| CiteSeer | 75.61% (0.00%) | -15.43% | 0.15 | -55.48% (10.39%) | -6.28% | INF |
| Twitch | 32.24% (51.86%)* | 0.10% | 2.05 | 10.57% (7.02%) | -1.76% | 6.3 |
| AMZN-P | 55.37% (30.98%)* | -7.35% | 2.09 | 16.92% (15.98%) | -17.90% | 6.93 |
| AMZN-C | 88.83% (48.28%)* | -1.36% | 4.82 | 42.54% (12.48%) | -2.94% | 10.09 |
| PPI | 73.71% (4.70%) | -4.12% | 3.26 | 6.29% (10.63%) | -1.05% | 38.16 |

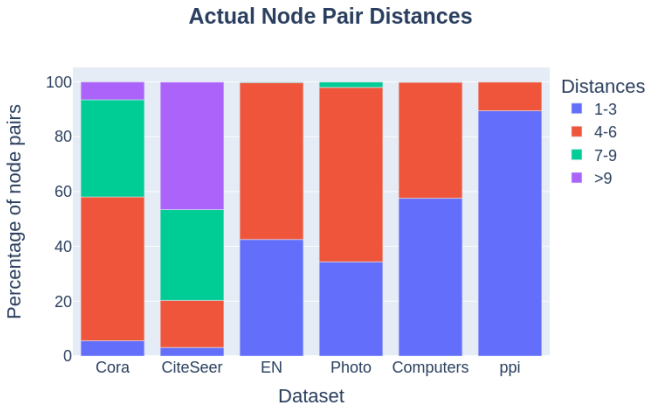


**Figure 2**: **Actual Node Distances -** The figure illustrates the correct node distances for each dataset. The bar's color represents the range the actual distance lies in (for example the blue part of the bar represents nodes with distances 1-3).

Let $d(u, v)$ be the actual distance between nodes $u$ and $v$ in the graph. It is important to note that the distance in graphs is a metric that satisfies the triangle inequality. That is, given any three nodes $x$, $y$, and $z$, the following inequality holds,

$$d(x, z) \leq d(x, y) + d(y, z) \quad (1)$$

$$d(x, z) \geq |d(x, y) - d(y, z)| \quad (2)$$

Note that if $y$ belongs to one of the shortest paths from $x$ to $z$, then the inequality 1 holds with equality. In order to compute the pairwise distances during runtime, we use the precomputed distances of nodes $u$ and $v$ from the landmark nodes and the triangle inequality. Let $i$ be an arbitrary landmark node, and thus based on 1 and 2, for any two nodes $u$ and $v$, we have,

$$\max_i |d(u, i) - d(v, i)| \leq d(u, v) \leq \min_i \{d(u, i) + d(i, v)\}$$

In other words, the true distance between nodes $d(u, v)$ lies in the range $[lb, ub]$, where $lb = \max_i |d(u, i) - d(v, i)|$ and $ub = \min_i \{d(u, i) + d(i, v)\}$. Any value in the range $[lb, ub]$ works as an approximation for $d(u, v)$. We use $ub$ as an approximation as [4] suggests that it is the best in most cases. Therefore, the complexity of computing distance during runtime for a pair of nodes is $O(l)$.

Figure 1 captures the absolute errors in the distance approximations across datasets. Figure 2 contains the graph's distribution of actual node distances. These two graphs show that the landmark method for distance approximation provides good approximations but has large error rates at times, further accentuating the robustness of CAFIN.

As expected, the performance of this method depends on the chosen subset of landmark nodes. A perfectly chosen subset can give accurate results for all distance pairs. Literature suggests picking nodes that have a high degree or high closeness centrality because they are more likely to be present in the shortest path of most pairs of nodes. As we aim to mitigate bias induced by a node's centrality, we do not use the above to maintain the pipeline bias-free. There is a trade-off between the performance and the computational complexity based on the number of landmarks chosen. More landmarks will guarantee stricter approximations but at the cost of more computation. Therefore, we select $l = 100$ landmark nodes for all datasets, which fared well on the trade-off through experimentation.

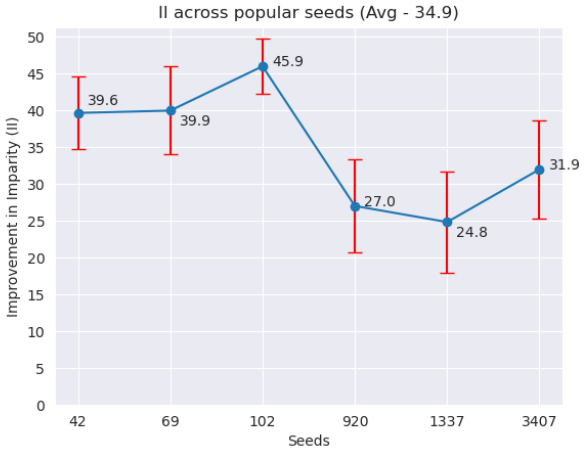


**Figure 3**: **II vs popular seeds -** Positive II is observed for various popular seeds for Link Predcition on Twitch dataset.

Since we use the upper bound $ub$ of triangle inequality to predict distance, our predicted distance can only be greater or equal to the actual distance. We increase the NEG_MIN_DIST value by 1 or 2 to account for the off by 1 or 2 error after noting the trend from Figure 1.

## 4 Node Classification

From Table 1(a) and Table 1(b), we can observe that neither CAFIN-N nor CAFIN-P performs well consistently for Node Classification. It either compromises II, CA, or the robustness of the values. It is also important to note that the embeddings generated by CAFIN-N for Cora, AMZN-P, and AMZN-C were very homogenous. We used a more robust classifier with higher representative power, AdaBoost, instead of our current one (LinearSVC) for the downstream task.

Although we observe better performances for a couple of datasets, CAFIN is the preferred choice due to its balance of stability and performance.

## 5 Influence of Seeds

GraphSAGE is known to be stochastic [7] due to the various random components it contains. Even then, CAFIN shows a consistent positive II (34.9% on average for Twitch dataset) across popular seeds[1] as seen in Figure 3. The values are averaged over 100 runs and we also plot the first standard deviation interval. Figure 3 shows that CAFIN is robust, and the results are reproducible across seeds, withstanding the stochastic nature of GraphSAGE.

## 6 Note on PPI's performance

Table 3 in the Main paper shows that PPI is extremely fast computationally and deviates from the trend of preprocessing time increasing with $|V|$ and $|E|$. The performance can be attributed to the inherent structure of the dataset and how we exploit it to increase performance. The dataset comprises 24 disconnected subgraphs, each possessing a disjoint subset of the nodes in the graph. We perform pairwise distance computations on each subgraph individually and parallelly. It leads to a dramatic speedup as the pairwise distance computation is now proportional to $|V|$. The memory constraints are also relaxed as we split and store the pairwise distances for each component. Similar performance can be expected for any dataset with multiple connected components that can be processed in parallel.

**Table 2**: **Results of t-test for Node Classification:** CAFIN-P and CAFIN-N and CAFIN-P show a statistically significant change in distribution hence verifying the validity of our results.

| Dataset | CAFIN-N | CAFIN-P |
|---|---|---|
| Cora | <0.00001 | 0.293251 |
| CiteSeer | <0.00001 | <0.00001 |
| Twitch | <0.00001 | <0.00001 |
| AMZN-P | <0.00001 | <0.00001 |
| AMZN-C | <0.00001 | 0.361 |
| PPI | <0.00001 | <0.00001 |

## 7 Statistical significance of results

We conducted a t-test for statistical significance on the study results. We observe that in almost all the subject groups, the p-value is less than 0.00001. This indicates that the distributions are, in fact, independent of each other and that our ablation studies are statistically significant. Table 2 enumerates the p-values for CAFIN-N and CAFIN-P for every dataset (with respect to CAFIN) for the node classification task.

[1] Weights & Biases 2022 statistics - https://twitter.com/weights_biases/status/1612829576618442754

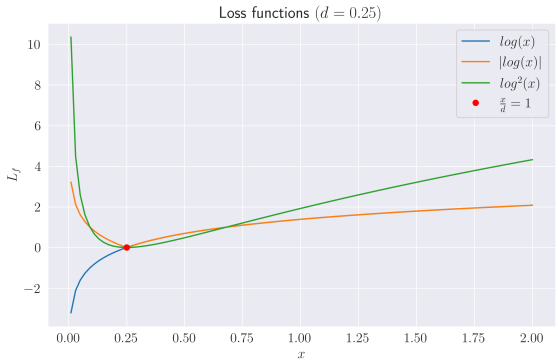


**Figure 4**: **Loss Formulations -** $log^2(x)$, a smooth function, achieves its minima at the required value

## 8 Intuition behind the loss function

Equation 5 details the loss formulation, and the goal is to make the actual distance in the graph directly proportional to the distance in the embedding space. Since the $log$ function is naturally monotonic, we square the formulation to ensure that the minima occurs when the actual distance is equal to the embedding distance and the loss doesn't push the embedding distance to zero for all pairs of nodes (if we don't square the formulation, having an embedding distance of 0 would result in the least loss).

Considering $x$ to be the embedding distance and $d$ to be the actual node distance in the graph, we ideally want the minima to occur when $x = d$. We notice that this happens for both $log^2$ and $|log|$. As $|log|$ has a point of non-differentiability at $x = d$, and because smooth functions are preferred for gradient descent, we choose $log^2$ for our formulation. We have demonstrated the same in Figure 4.

## References


[1] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence, 'Citeseer: an automatic citation indexing system', in *Digital library*, (1998).

[2] William L. Hamilton, Rex Ying, and Jure Leskovec, 'Inductive representation learning on large graphs', in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, p. 1025–1035, Red Hook, NY, USA, (2017). Curran Associates Inc.

[3] Andrew McCallum, Kamal Nigam, Jason D. M. Rennie, and Kristie Seymore, 'Automating the construction of internet portals with machine learning', *Information Retrieval*, **3**, 127–163, (2000).

[4] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis, 'Fast shortest path distance estimation in large networks', in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 867–876, (2009).

[5] Benedek Rozemberczki, Carl Allen, and Rik Sarkar, 'Multi-Scale attributed node embedding', *Journal of Complex Networks*, **9**(2), (05 2021). cnab014.

[6] Robert E. Schapire, 'A brief introduction to boosting', IJCAI'99, p. 1401–1406, San Francisco, CA, USA, (1999). Morgan Kaufmann Publishers Inc.

[7] Tobias Schumacher, Hinrikus Wolf, Martin Ritzert, Florian Lemmerich, Jan Bachmann, Florian Frantzen, Max Klabunde, Martin Grohe, and Markus Strohmaier, 'The effects of randomness on the stability of node embeddings', in *PKDD/ECML Workshops*, (2020).

[8] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation, 2018.

[9] Marinka Zitnik and Jure Leskovec, 'Predicting multicellular function through multi-layer tissue networks', *Bioinformatics*, **33**(14), i190–i198, (07 2017).