BLISS: Bandit Layer Importance Sampling Strategy for Efficient Training of GNN

Omar Alsaqa, Thi Linh Hoang,
Muhammed Fatih Balin

NEURAL INFORMATION PROCESSING SYSTEMS

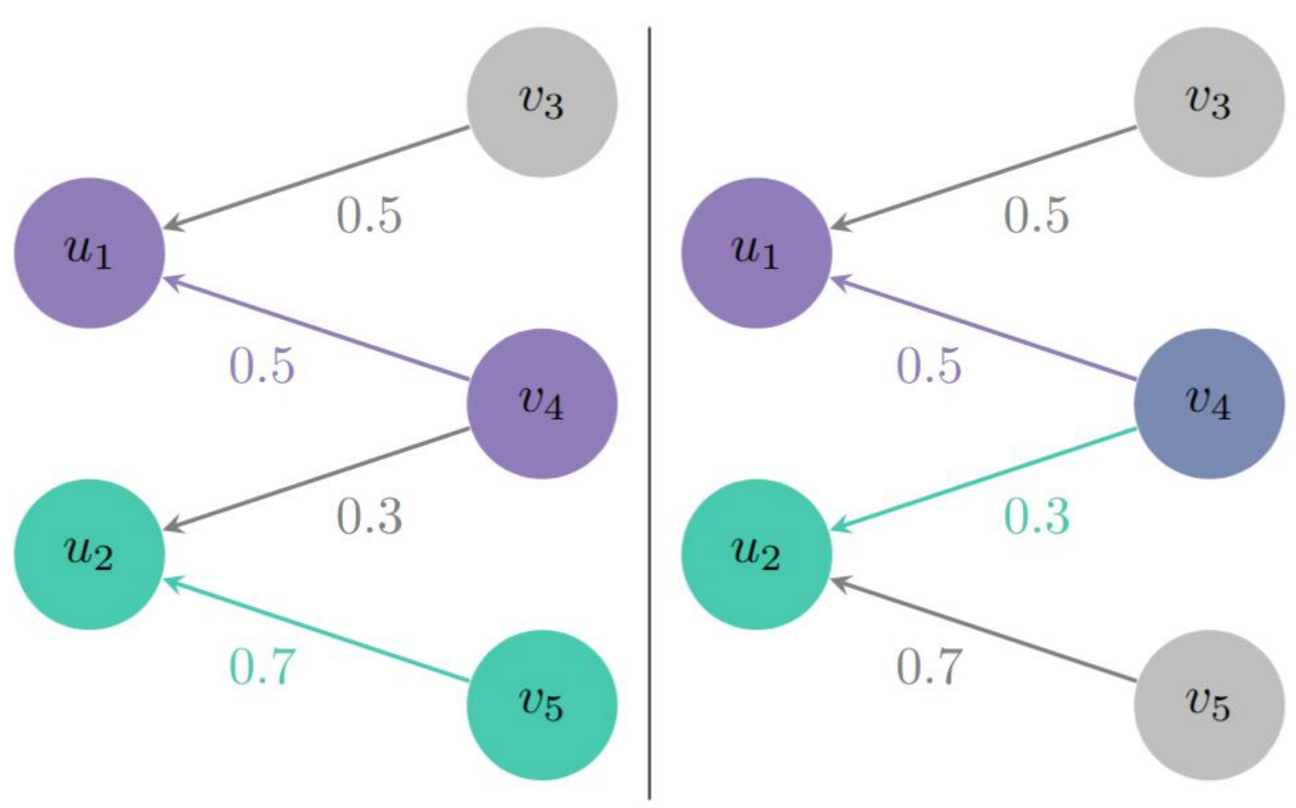
Motivation: Overcoming the GNN Scalability Bottleneck

Graph Neural Networks (GNNs) are powerful models for learning from graph-structured data, but they struggle to scale to large graphs. The core challenge is the "neighborhood explosion," where the number of nodes required for message passing grows exponentially with network depth. This creates prohibitive memory and computational costs, making full-batch training infeasible. Efficient neighbor sampling is therefore essential.

- **Node-wise Sampling (e.g., GraphSAGE):** Samples neighbors for each node individually. Can be redundant and computationally inefficient.
- **Layer-wise Sampling (e.g., LADIES):** Jointly samples a single set of nodes at each layer. More efficient but existing methods use static importance scores that don't adapt during training.
- **Sub-graph Sampling (e.g., GraphSAINT):** Trains on smaller induced subgraphs. Risks losing crucial global context.

Background: Layer-wise Sampling

Layer-wise sampling methods aim to create an unbiased estimator for neighborhood aggregation by sampling nodes across an entire layer. The goal is to select the most informative nodes to minimize the variance of the estimator, leading to more stable and accurate training. Our work, BLISS, introduces a dynamic approach to this paradigm.



Visual comparison of node-wise sampling (left), which can have high redundancy, versus layer-wise sampling (right), which provides broader, more efficient coverage.

Proposed Method: The BLISS Algorithm

BLISS frames neighbor selection as a **multi-armed bandit problem** where each neighbor is an "arm." By balancing exploration (discovering new informative nodes) and exploitation (sampling known important nodes), BLISS dynamically learns the optimal sampling policy during training. The core process is a four-step loop:

1. Select: Use the bandit policy to determine layer-wise node sampling probabilities p_j.
2. Aggregate: Perform GNN message passing using the sampled neighbors.
3. Reward: Calculate a reward r_{ij} based on a neighbor's contribution to the node representation.
4. Update: Feed the reward back to the bandit (EXP3) to update the policy for the next iteration.

Key Equations:

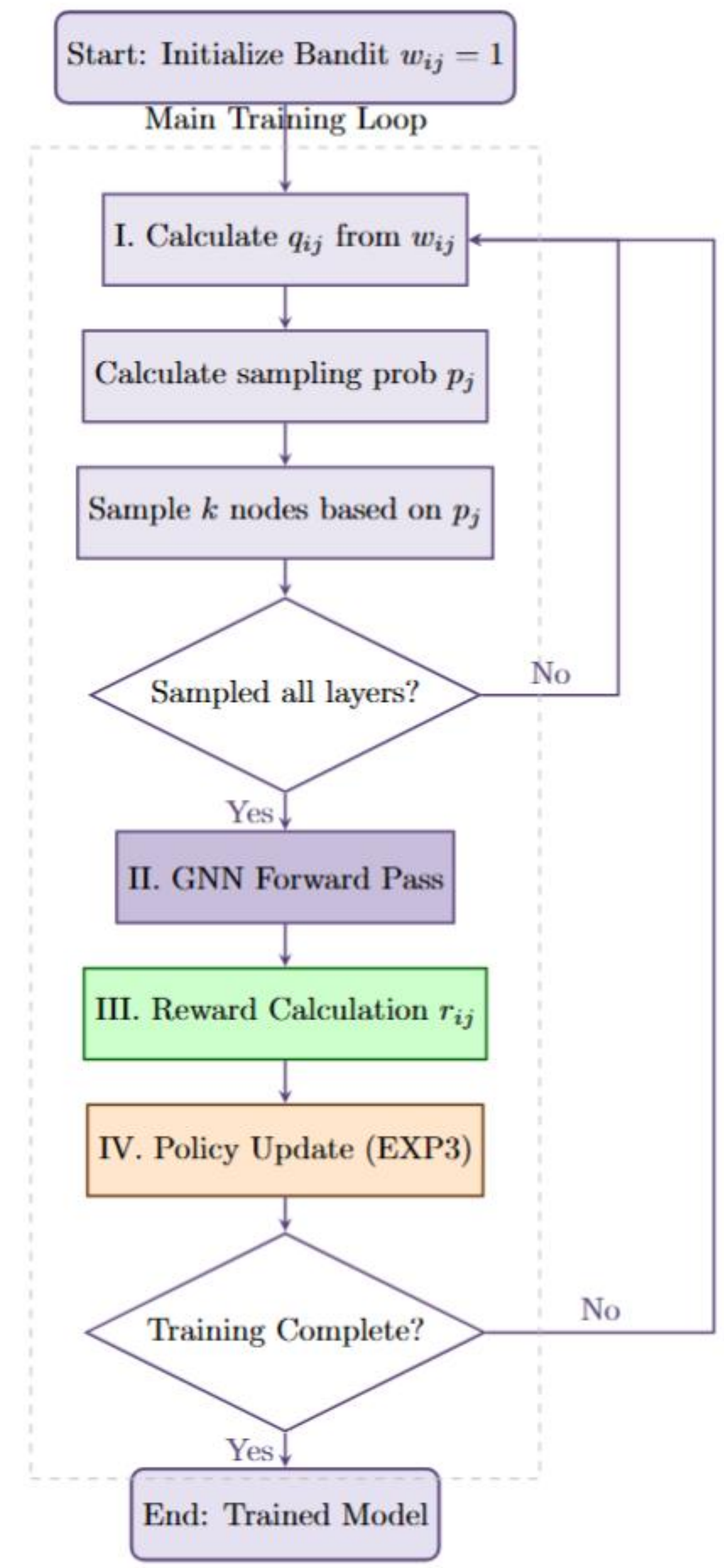
$$p_j = \sqrt{\sum_i \left(\frac{q_{ij}}{\sum_k q_{ik}}\right)^2}$$

$$h_i = \frac{1}{k}\sum_{s=1}^{k} \frac{\alpha_{ijs}}{q_{ijs}} \hat{h}_{j_s}$$

$$r_{ij} = \frac{\alpha_{ij}^2}{k \cdot q_j^2} \parallel h_j \parallel_2^2$$

Node Classification F1-Score (%)

Dataset	Sampler	Test GAT	Test SAGE
citeseer	BLISS	0.706 ± 0.002	0.580 ± 0.032
	PLADIES	0.683 ± 0.005	0.601 ± 0.017
cora	BLISS	0.813 ± 0.004	0.795 ± 0.009
	PLADIES	0.809 ± 0.003	0.772 ± 0.014
flickr	BLISS	0.511 ± 0.002	0.503 ± 0.002
	PLADIES	0.507 ± 0.005	0.505 ± 0.001
pubmed	BLISS	0.731 ± 0.007	0.597 ± 0.057
	PLADIES	0.718 ± 0.013	0.557 ± 0.042
reddit	BLISS	0.949 ± 0.001	0.962 ± 0.000
	PLADIES	0.950 ± 0.001	0.962 ± 0.000
yelp	BLISS	0.540 ± 0.002	0.529 ± 0.005
	PLADIES	0.539 ± 0.002	0.502 ± 0.009



References

Will Hamilton, et al. (2017). "Inductive representation learning on large graphs." In: Neural Information Processing Systems (NeurIPS).

Petar Velickovic, et al. (2018) "Graph attention networks". In: 6th International Conference on Learning Representations (ICLR).

Difan Zou, et al. (2019). "Layer-dependent importance sampling for training deep and large graph convolutional networks". In: Advances in neural information processing systems 32.

Muhammed Fatih Balin, et al. (2023). "Layer-neighbor sampling — defusing neighborhood explosion in gnns" In: Advances in Neural Information Processing Systems, volume 36.