41st IEEE International Conference on Data Engineering
HONG KONG SAR, CHINA | MAY 19 – 23, 2025

KAIST AI
Kim Jaechul Graduate School

Simple yet Effective Node Property Prediction on Edge Streams under Distribution Shifts


Jongha Lee


Taehyung Kwon

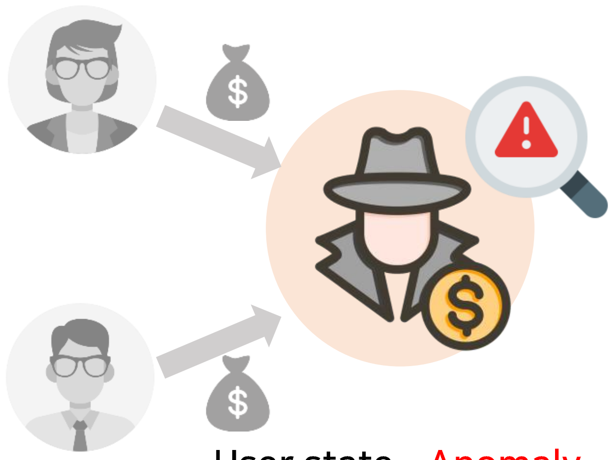

Heechan Moon


Kijung Shin

Node Properties in Real-world Networks

Various characteristics of entities can be represented as node properties

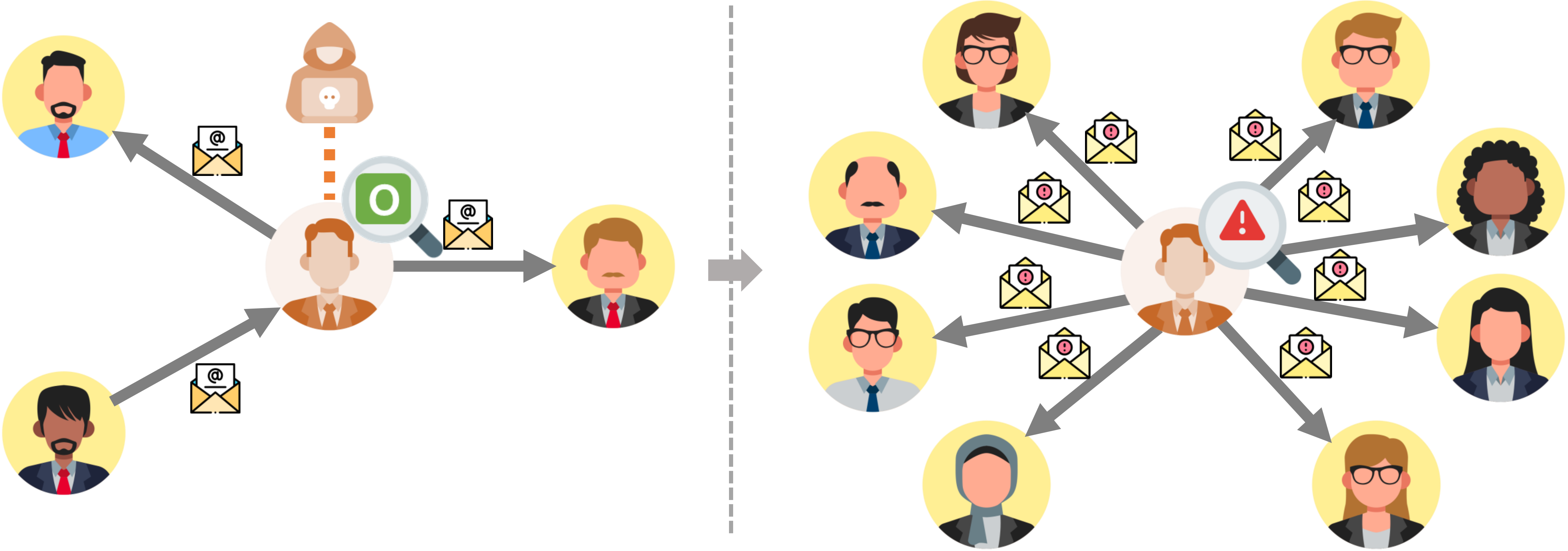
- Predicting these properties can be valuable for many real-world applications
  - Anomalous states in financial networks (anomaly detection)
  - Users' preferred song categories for next week on a streaming site (recommendation)



Challenges in Predicting Node Properties

Node properties dynamically change in real-world networks

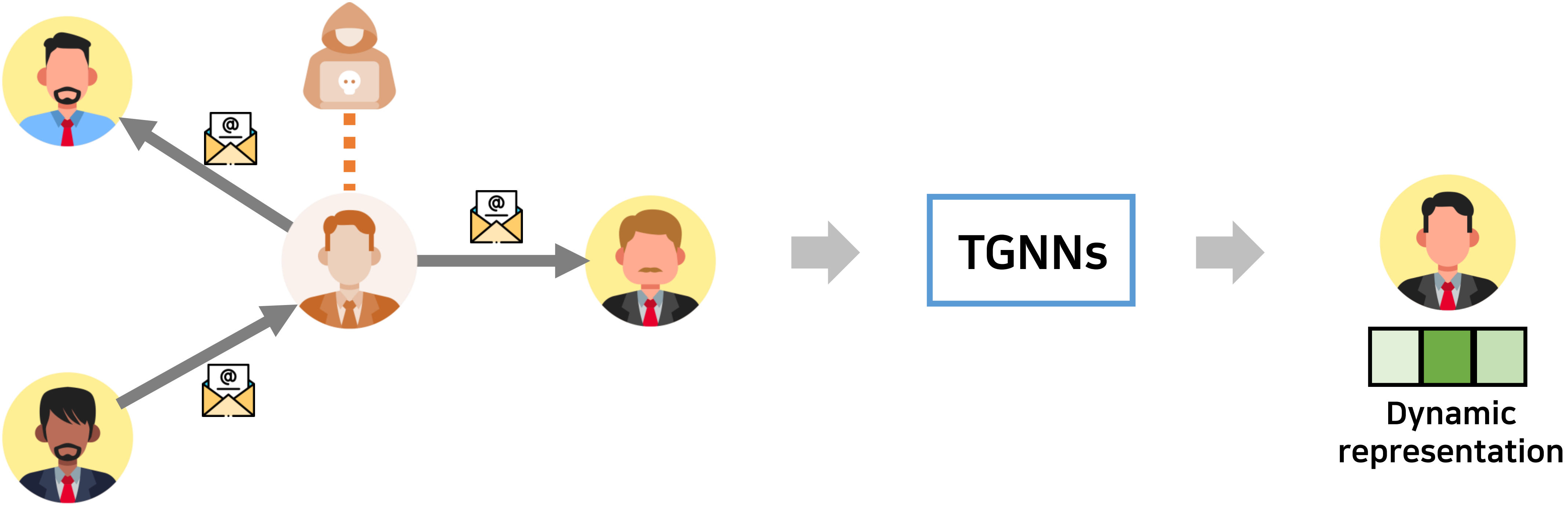
- Many real-world networks evolve over time, with emerging interactions
  - Previous methods based on static graphs become less effective and efficient in such a case



Challenges in Predicting Node Properties

Temporal Neural Networks (TGNNs) can be used for node property prediction

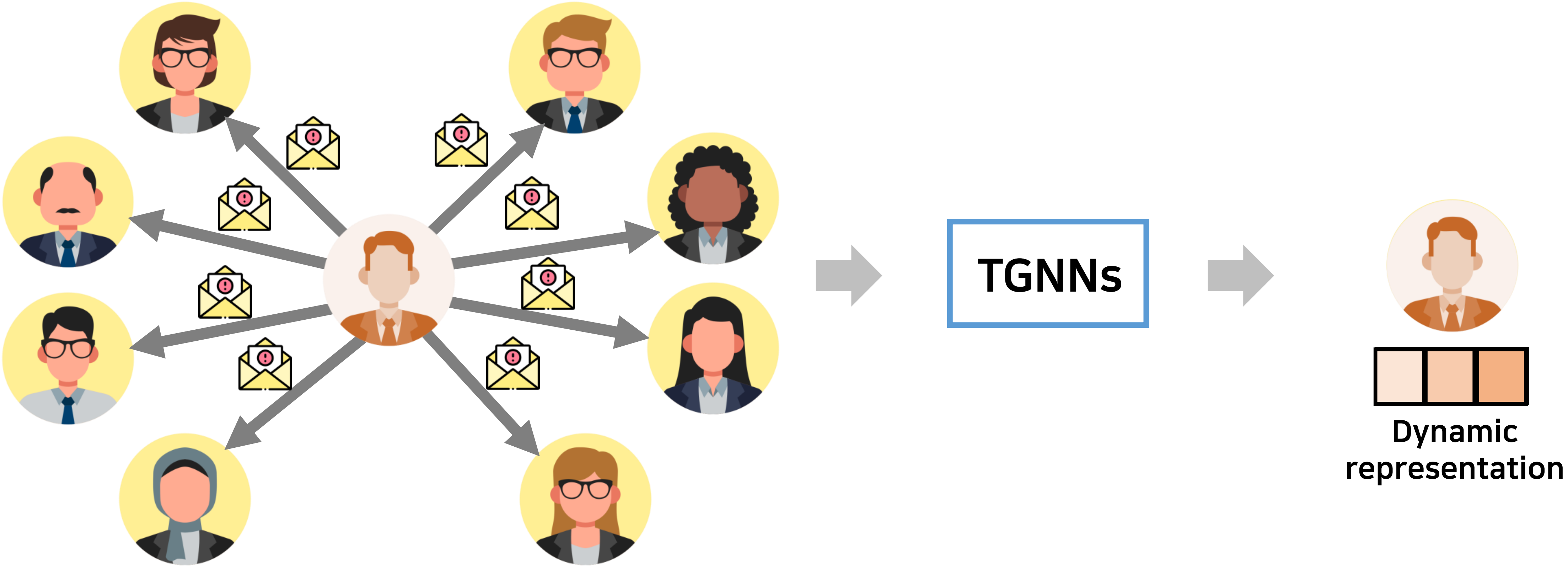
- These models can predict dynamically changing node properties by incrementally update node representations that capture complex temporal and structural patterns



Challenges in Predicting Node Properties

Temporal Neural Networks (TGNNs) can be used for node property prediction

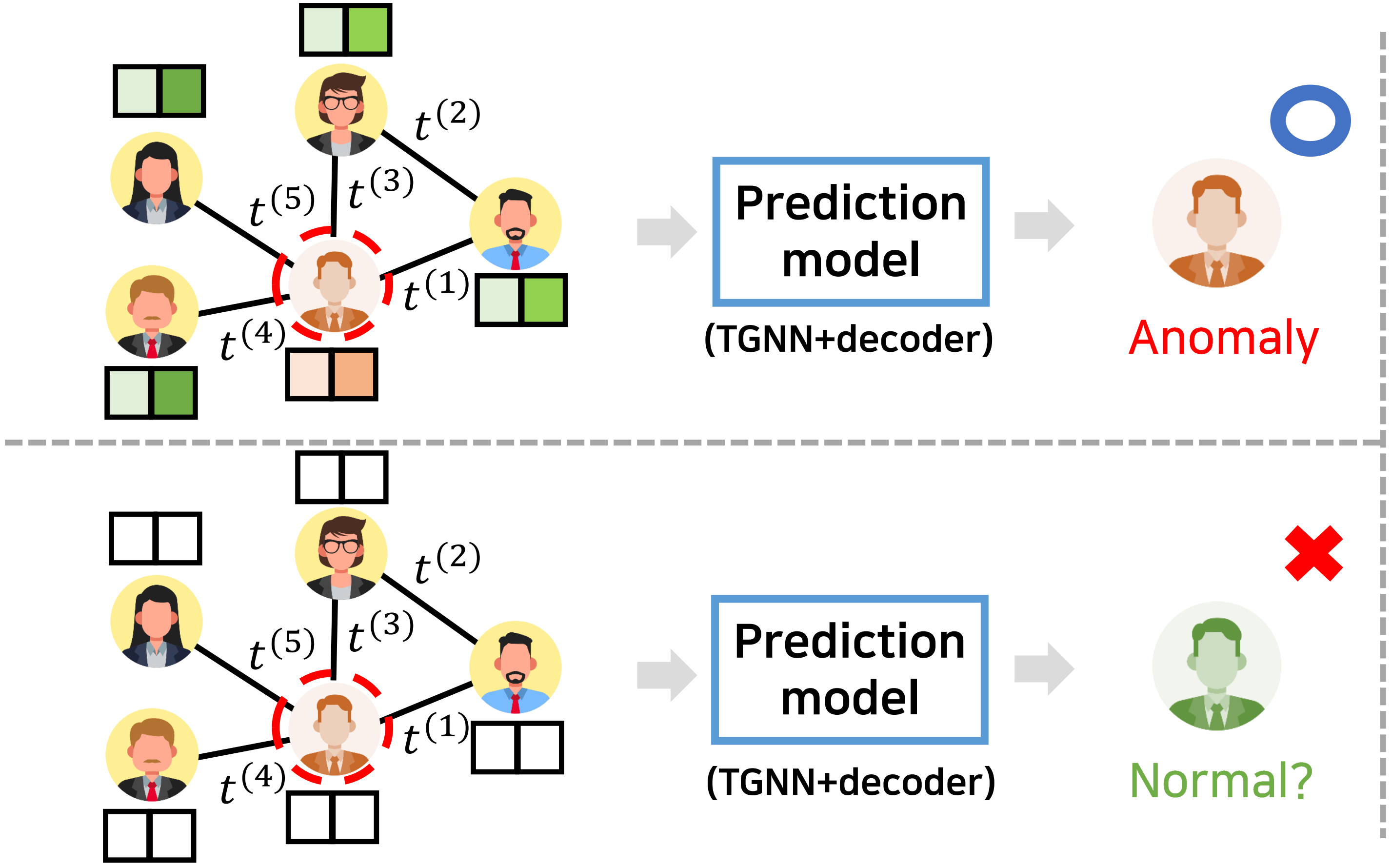
- These models can predict dynamically changing node properties by incrementally update node representations that capture complex temporal and structural patterns

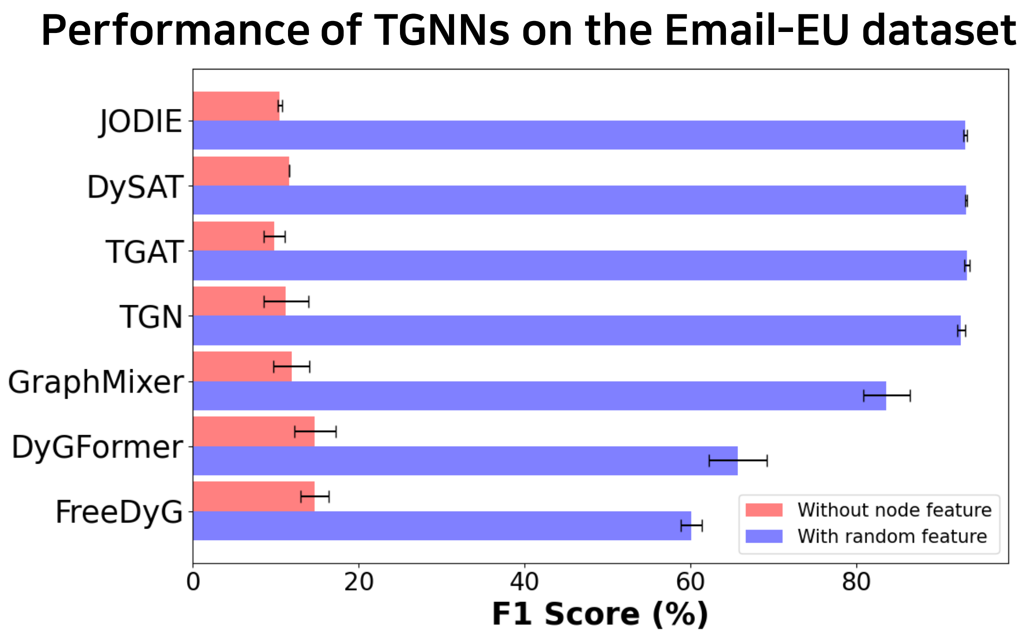


Challenges in Predicting Node Properties

However, performance of TGNNs drops when node features are absent

- TGNNs can be less effective without proper node features in node property prediction



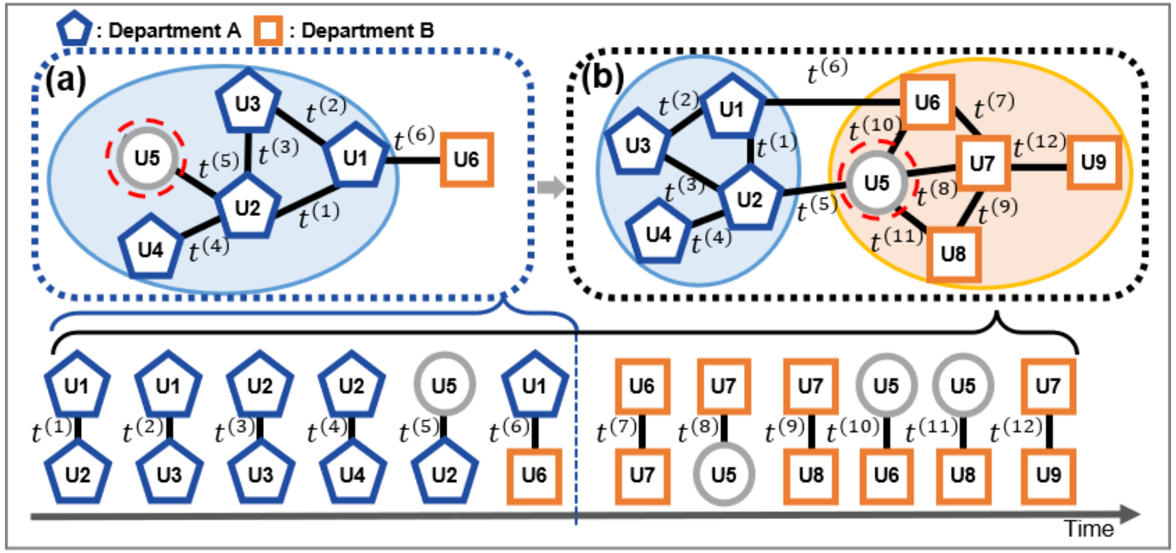


Challenges in Predicting Node Properties

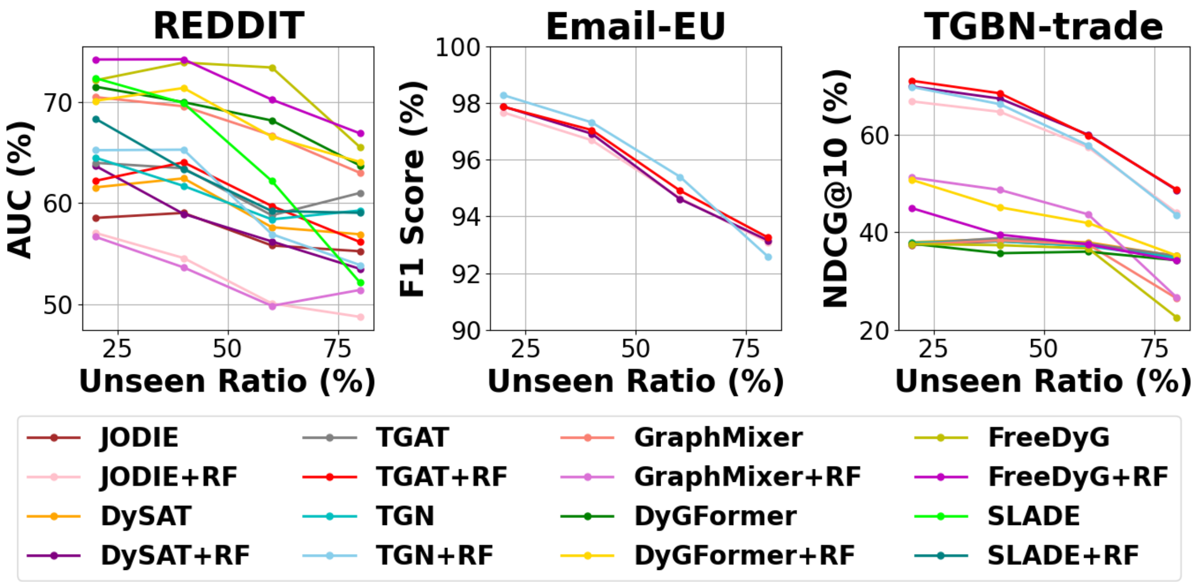
TGNNs are vulnerable to distribution shifts in real-world networks

- Many TGNN models employ complex architectures such as RNNs and attention modules
- However, complex model architectures can be vulnerable to distribution shifts

Example of distribution shifts in a collaboration network



Performance degradations of TGNNs under distribution shift

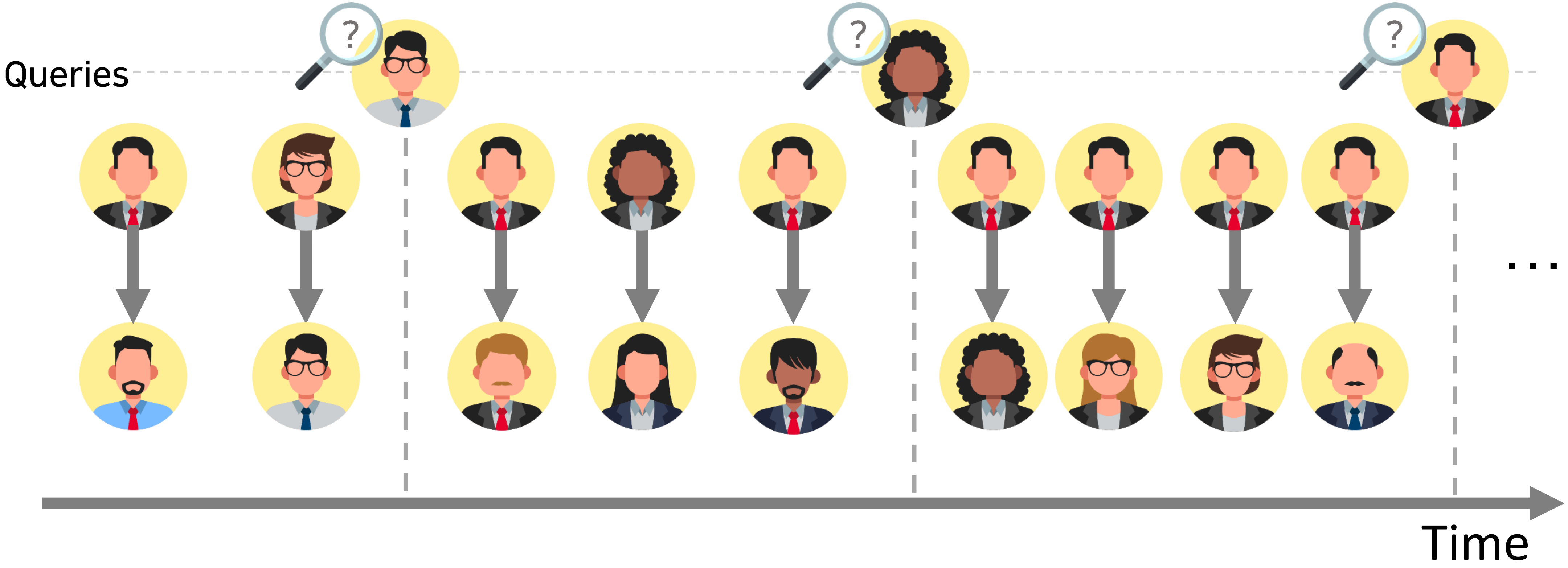


Contents

Problem Definition

Node Property Prediction in Edge Streams

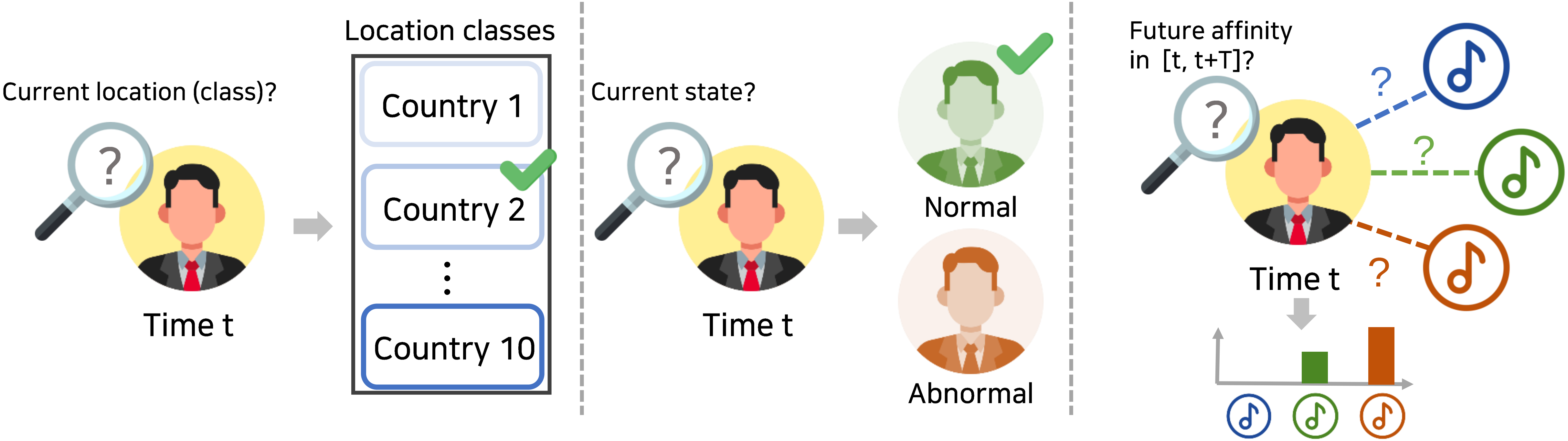
- To predict the node property of each query node in edge streams
- Only past edges can be utilized for predicting the current node properties



Problem Definition

Subtasks of Node Property Prediction in Edge Streams

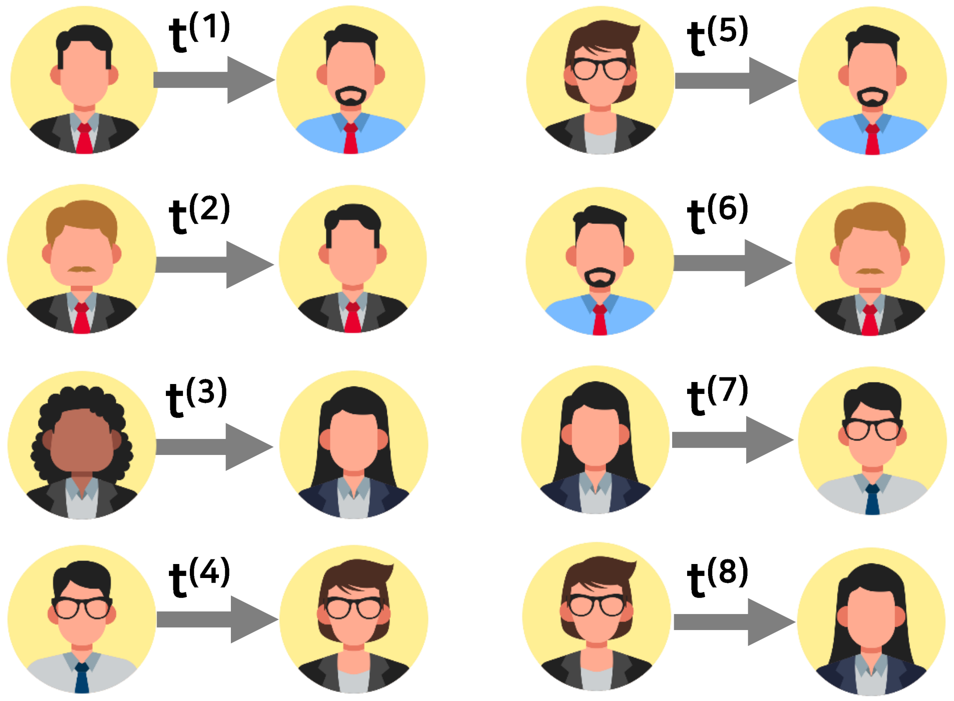
- Dynamic Node Classification
- Dynamic Anomaly Detection
- Node Affinity Prediction



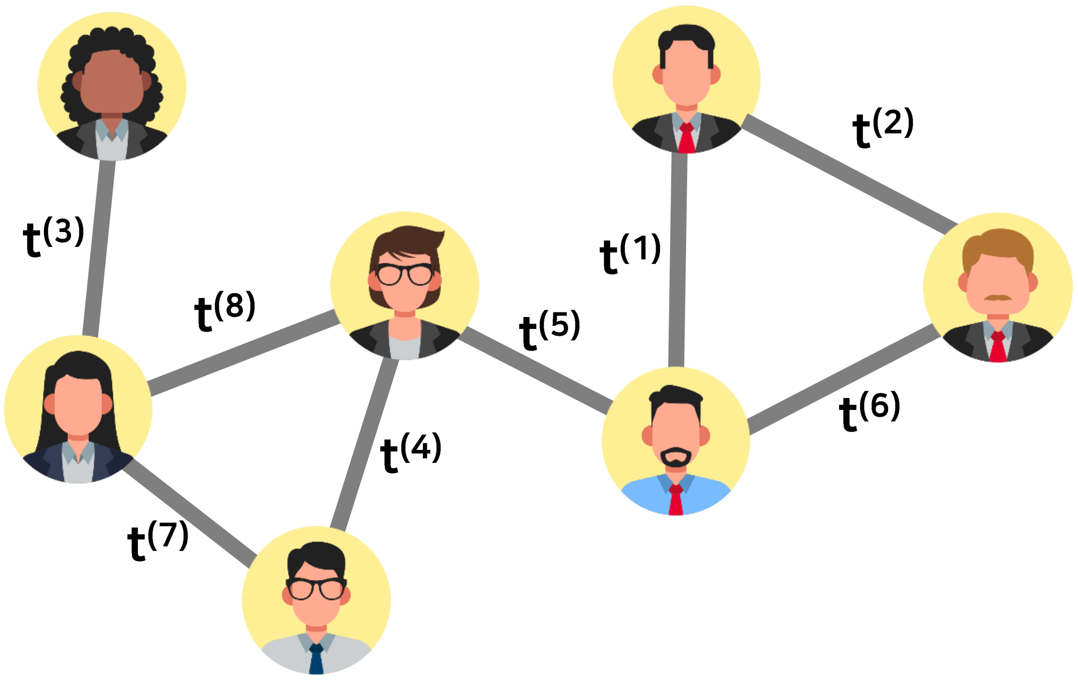
Our Graph Model for Edge Stream: CTDG

CTDG (Continuous Time Dynamic Graph)-based methods

- Incrementally update the graph by adding time information as a new edge arrives
- Allow for incremental algorithms to minimize time delay



Edge Stream

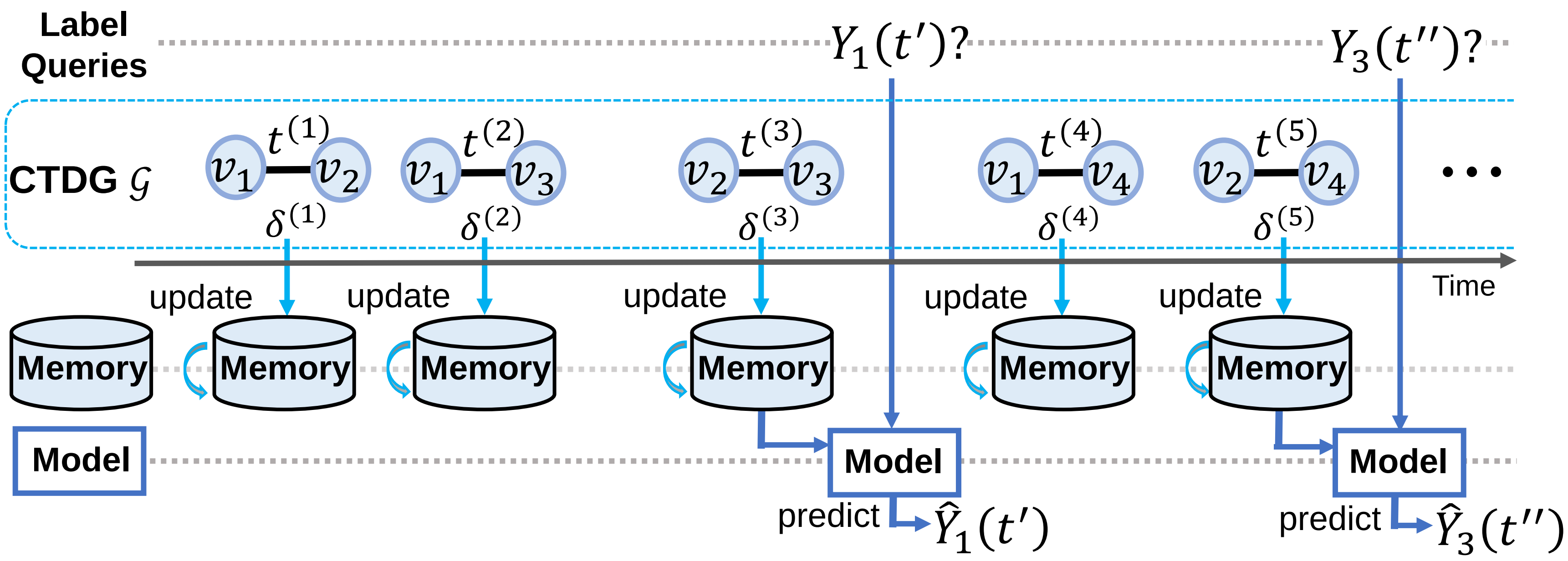


Continuous Time Dynamic Graph

Our Graph Model for Edge Stream: CTDG

Node Property Prediction Process using CTDGs

- Specifically, temporal edges appear over time and are used to update the memory
- Based on the updated memory, the model make predictions for incoming property queries

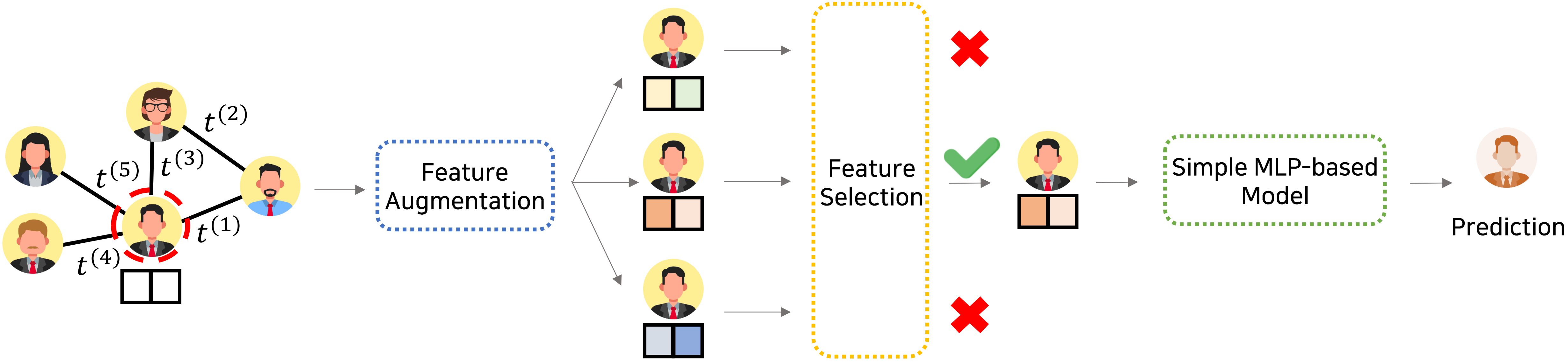


Contents

Proposed Method: SPALSH

Simple node Property prediction via representation Learning with Augmented features under distribution SHifts

- Effective in predicting node properties using proposed augmented node features
- Accurately and efficiently select the proper feature augmentation schemes
- Lightweight MLP-based model that is highly efficient and robust under distribution shifts

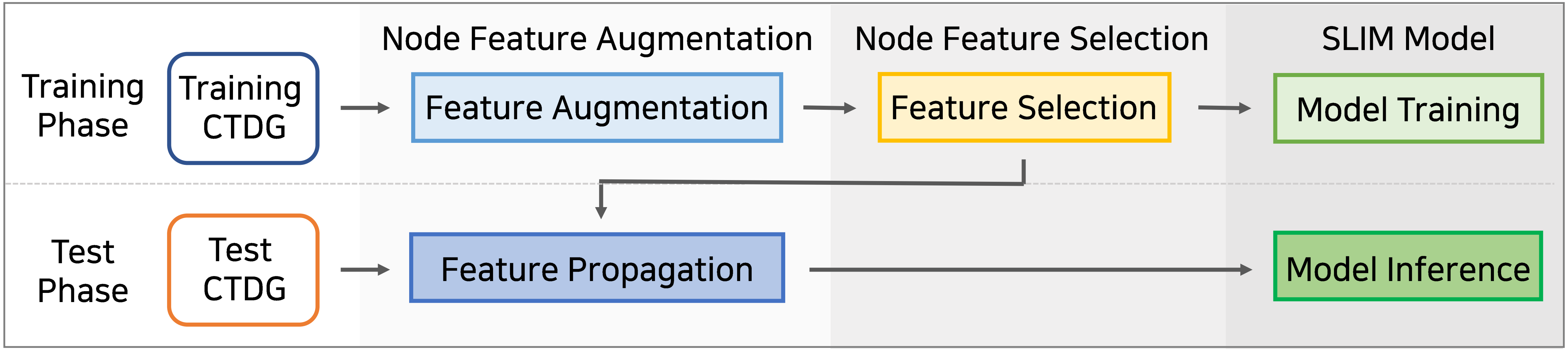


Overview of SPLASH

(1) Node Feature Augmentation

(2) Node Feature Selection

(3) SLIM Model (Our Proposed TGNN Model)

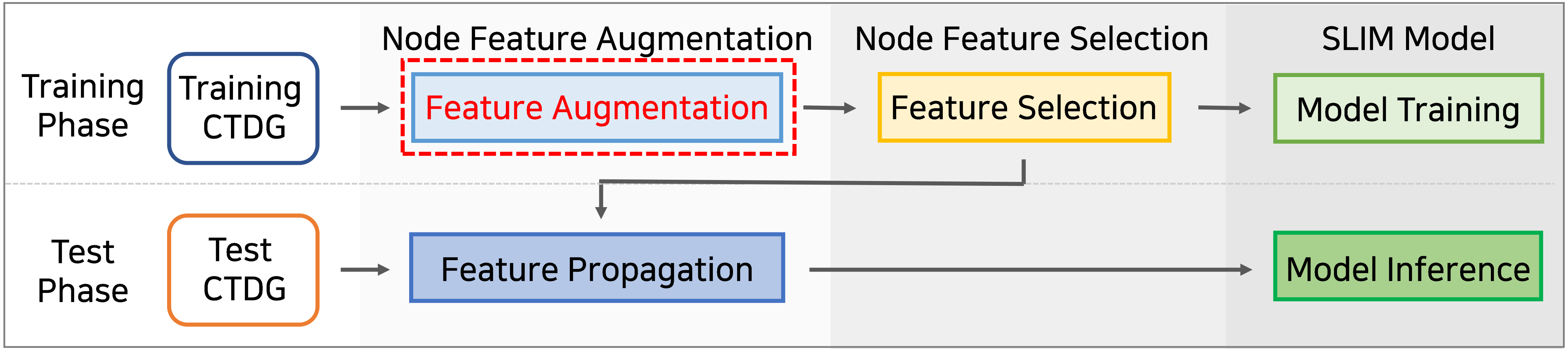


Overview of SPLASH

(1) Node Feature Augmentation

(2) Node Feature Selection

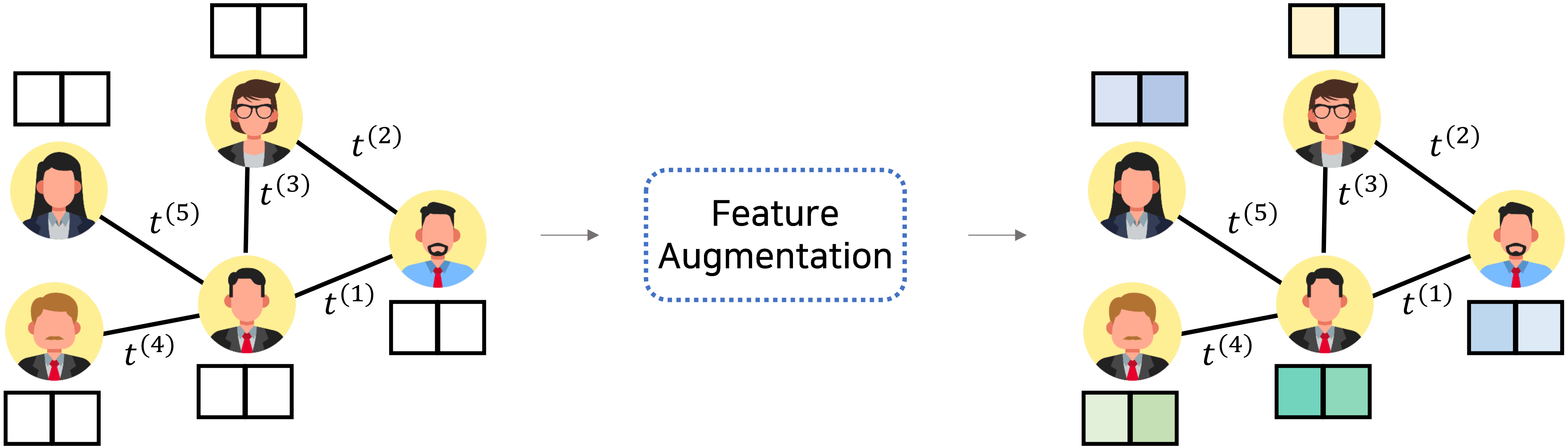
(3) SLIM Model (Our Proposed TGNN Model)



Feature Augmentation (Training Phase)

Goal: to augment the node features of seen nodes within the training graph

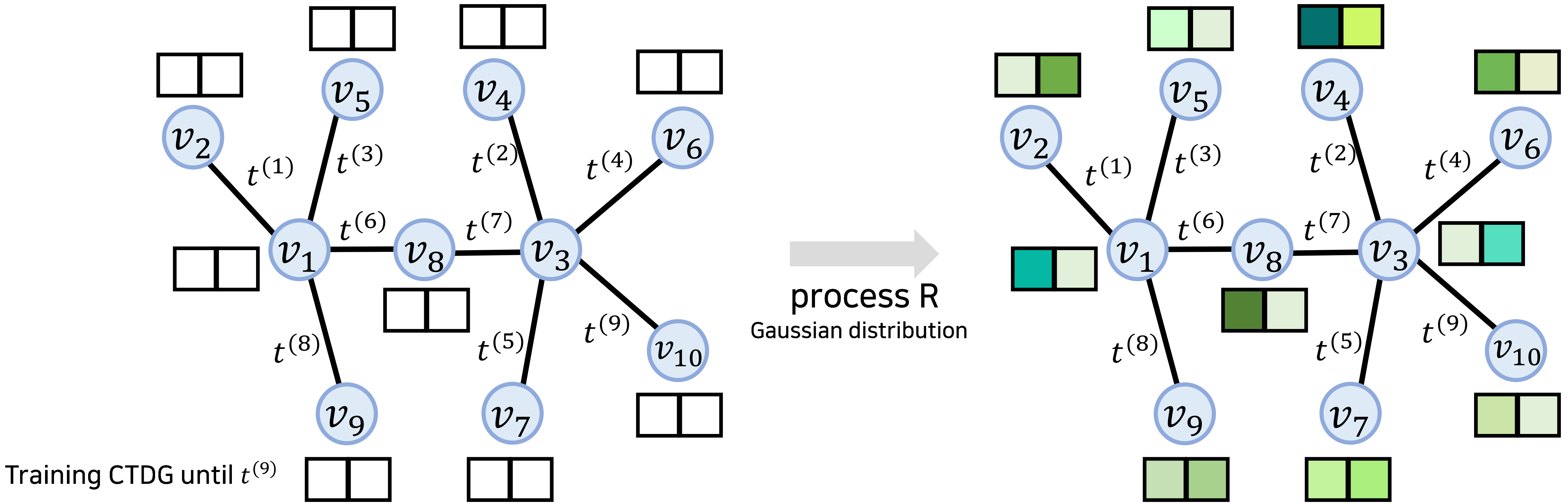

: Seen Nodes



Feature Augmentation (Training Phase)

Random Feature Augmentation

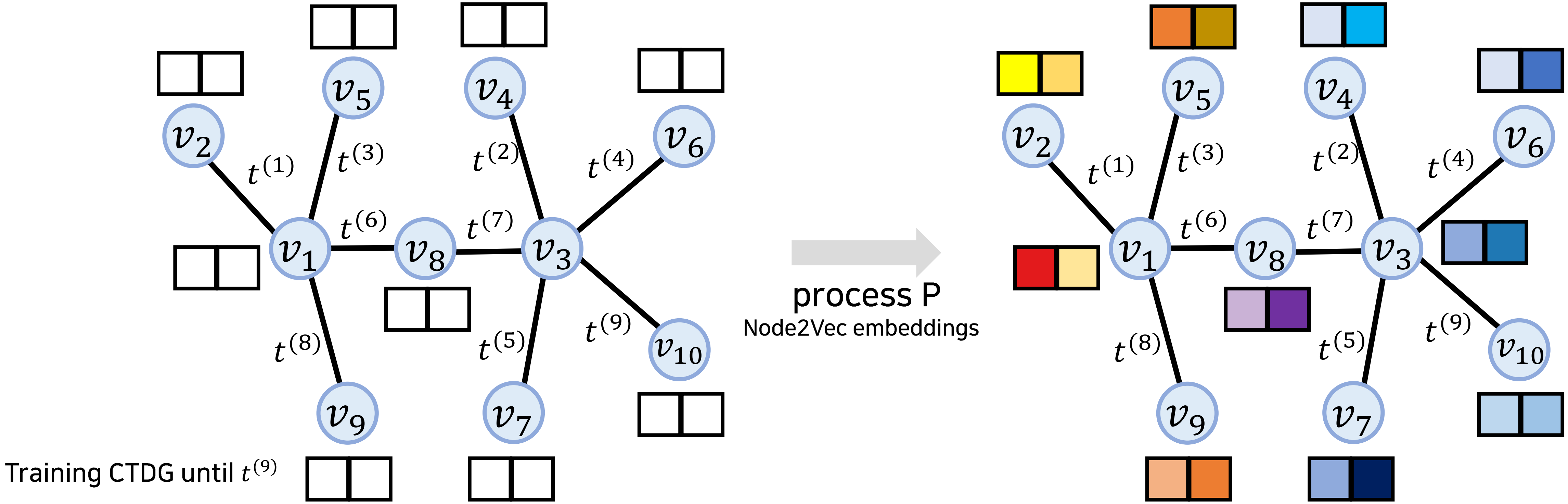
- This process aims to encode stable and absolute positions of seen nodes



Feature Augmentation (Training Phase)

Positional Feature Augmentation

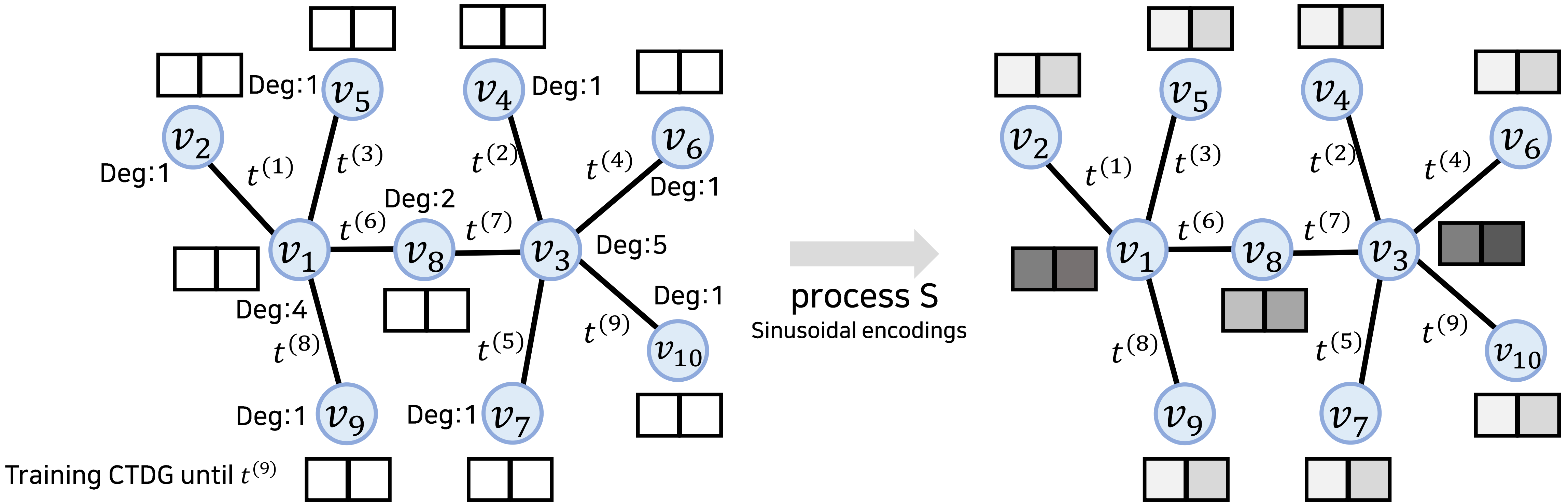
- This process aims to encode stable and relative positions of seen nodes



Feature Augmentation (Training Phase)

Structural Feature Augmentation

- This process aims to encode dynamic structural patterns of seen nodes

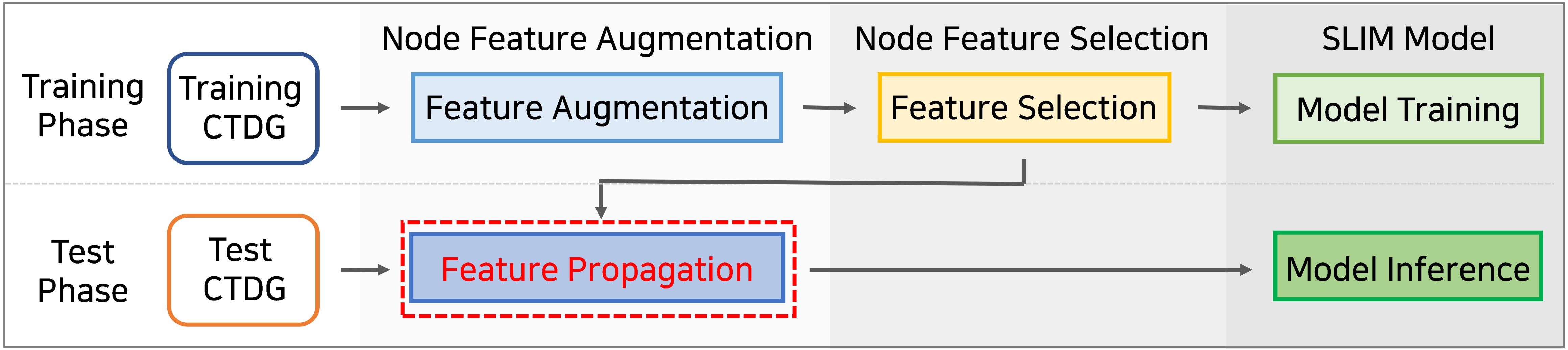


Overview of SPLASH

(1) Node Feature Augmentation

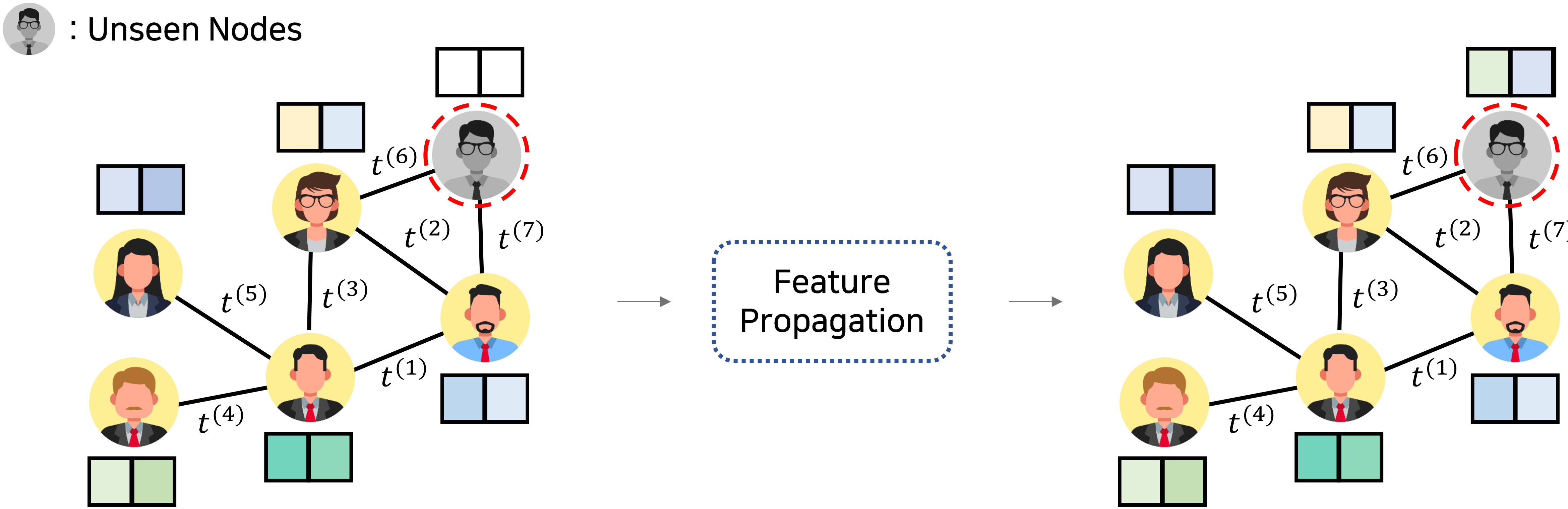
(2) Node Feature Selection

(3) SLIM Model (Our Proposed TGNN Model)



Feature Propagation (Test Phase)

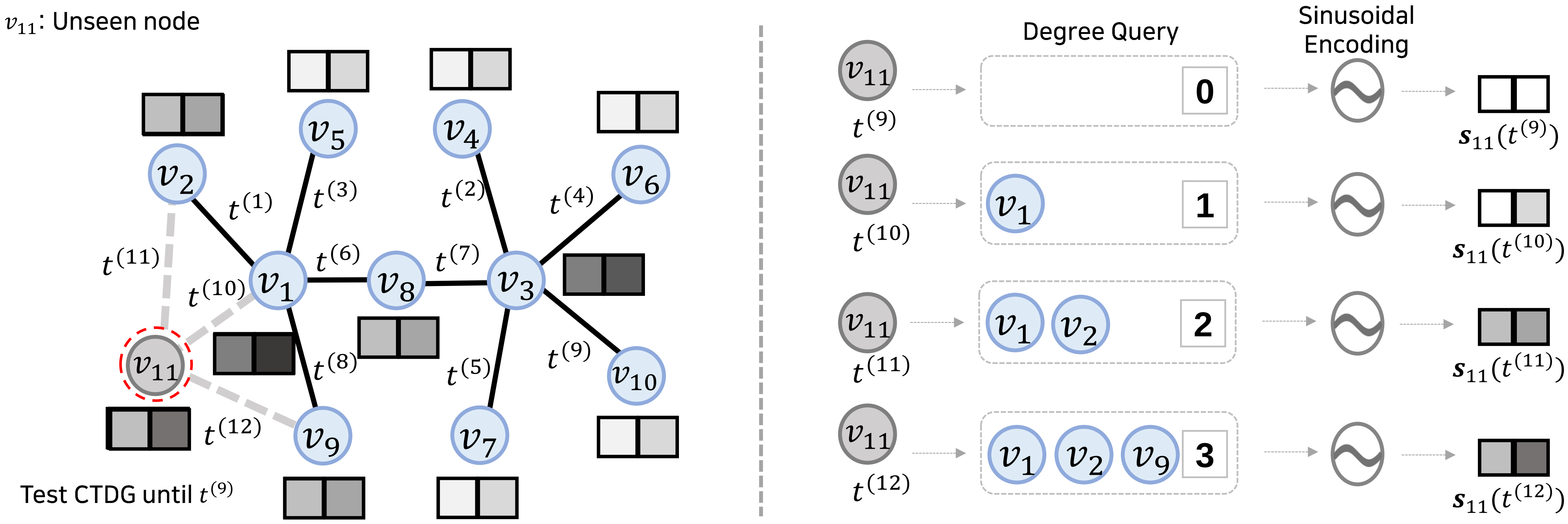
Goal: to generate the node features of unseen nodes in the test phase



Feature Propagation (Test Phase)

Structural Feature Augmentation for Unseen Nodes

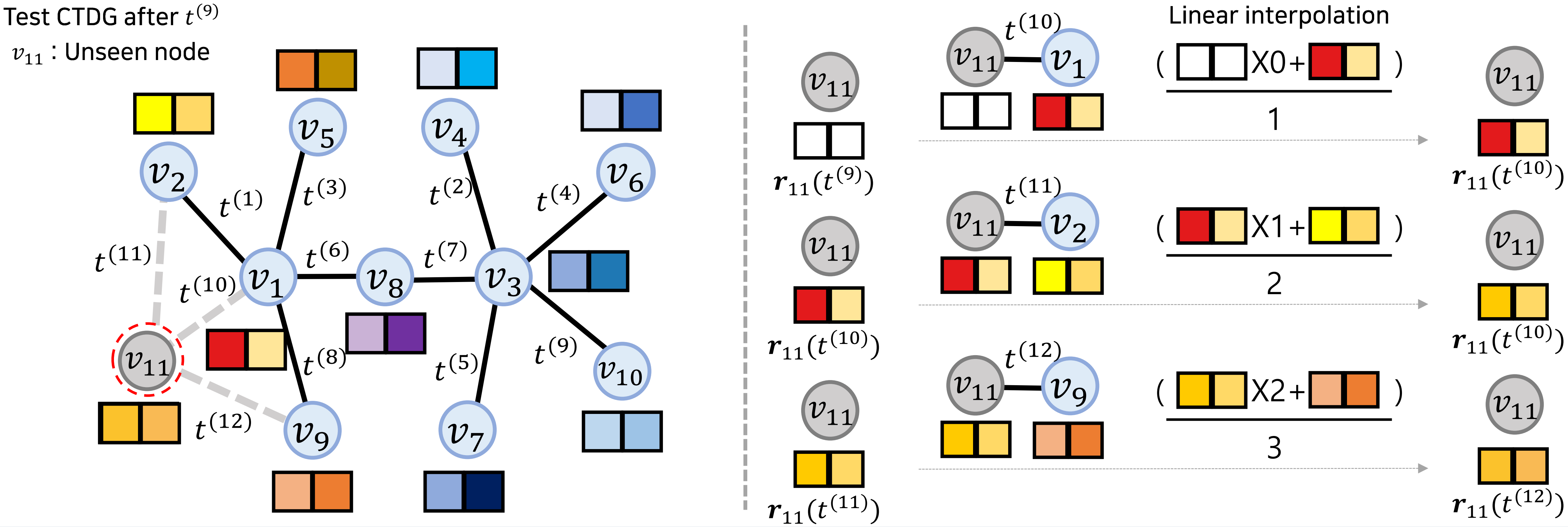
- The same process S used in the training phase is applied to unseen nodes



Feature Propagation (Test Phase)

Positional Feature Augmentation for Unseen Nodes

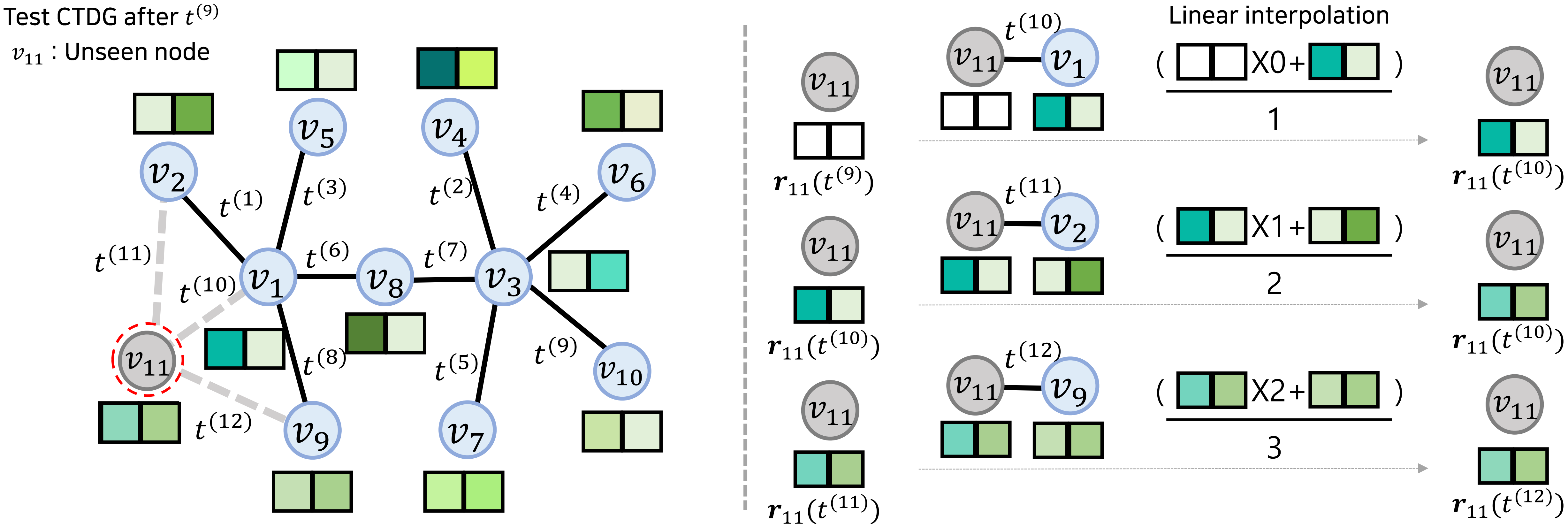
- This process aims to represent the node feature for the unseen node using a simple linear interpolation of neighboring nodes' features



Feature Propagation (Test Phase)

Random Feature Augmentation for Unseen Nodes

- As the random features assigned to unseen nodes cannot be trained, the same linear interpolation is used instead for unseen nodes

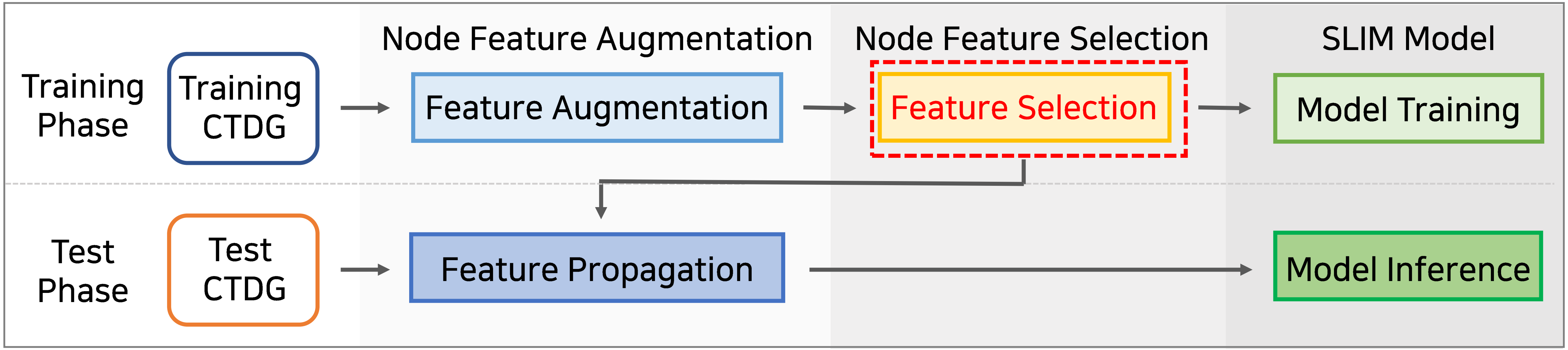


Overview of SPLASH

(1) Node Feature Augmentation

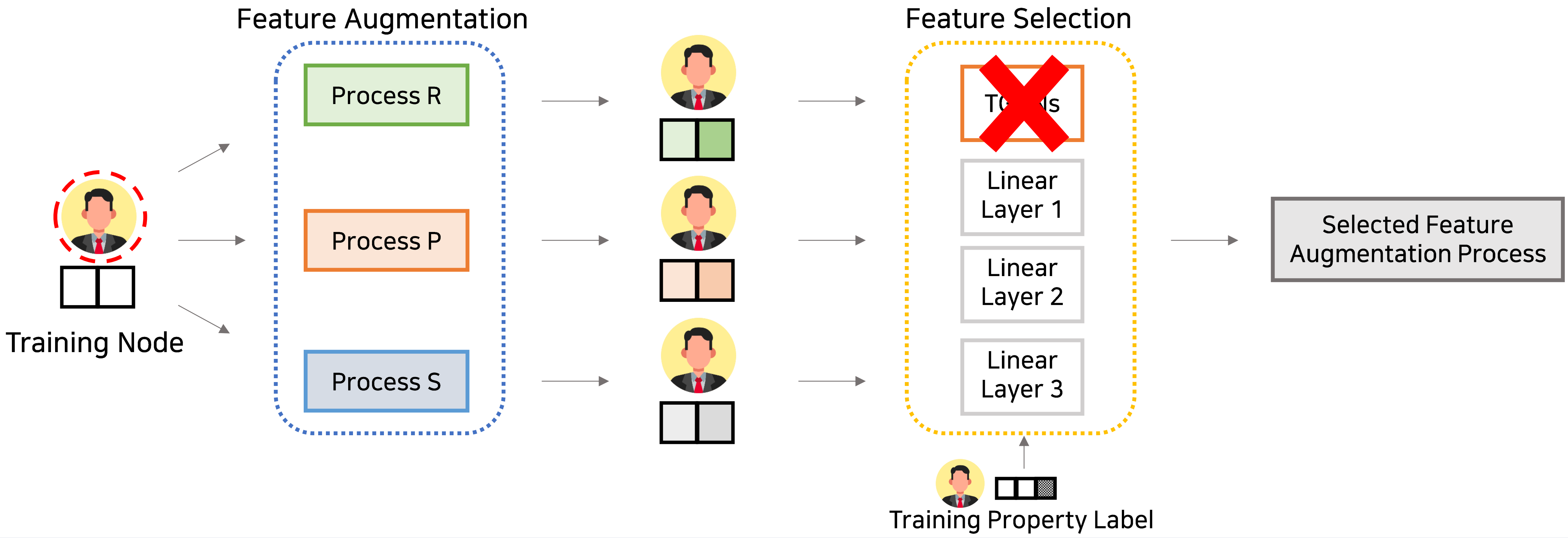
(2) Node Feature Selection

(3) SLIM Model (Our Proposed TGNN Model)



Node Feature Selection

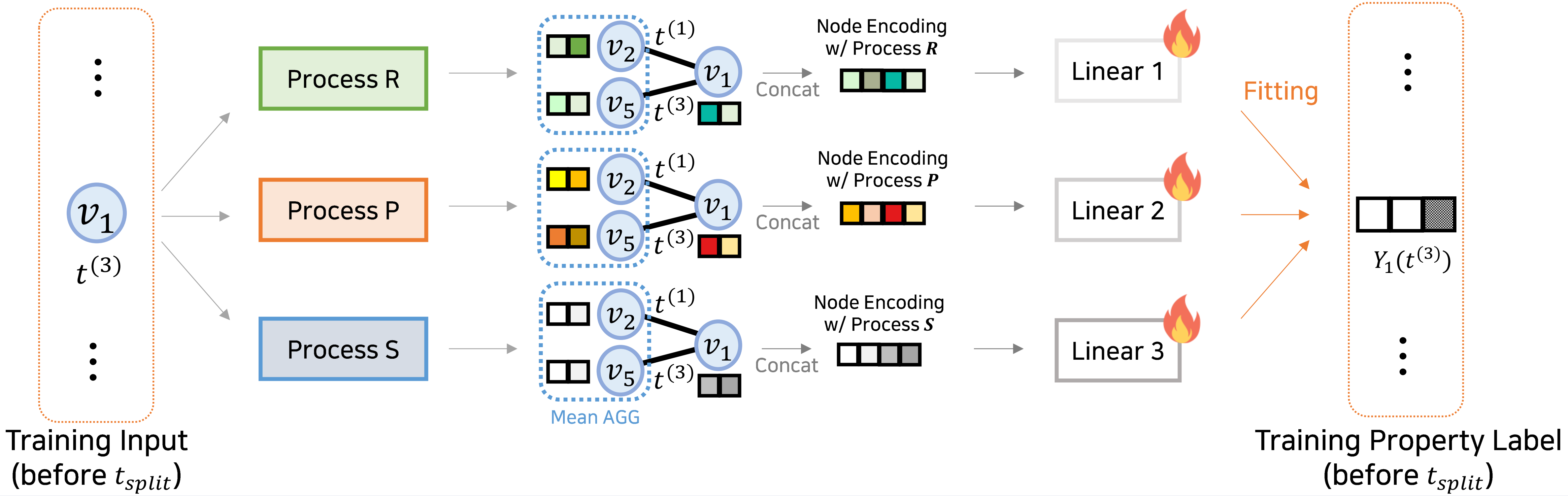
Goal: to efficiently identify a feature augmentation process that is effective for property label prediction



Node Feature Selection

Fitting with Node Encodings

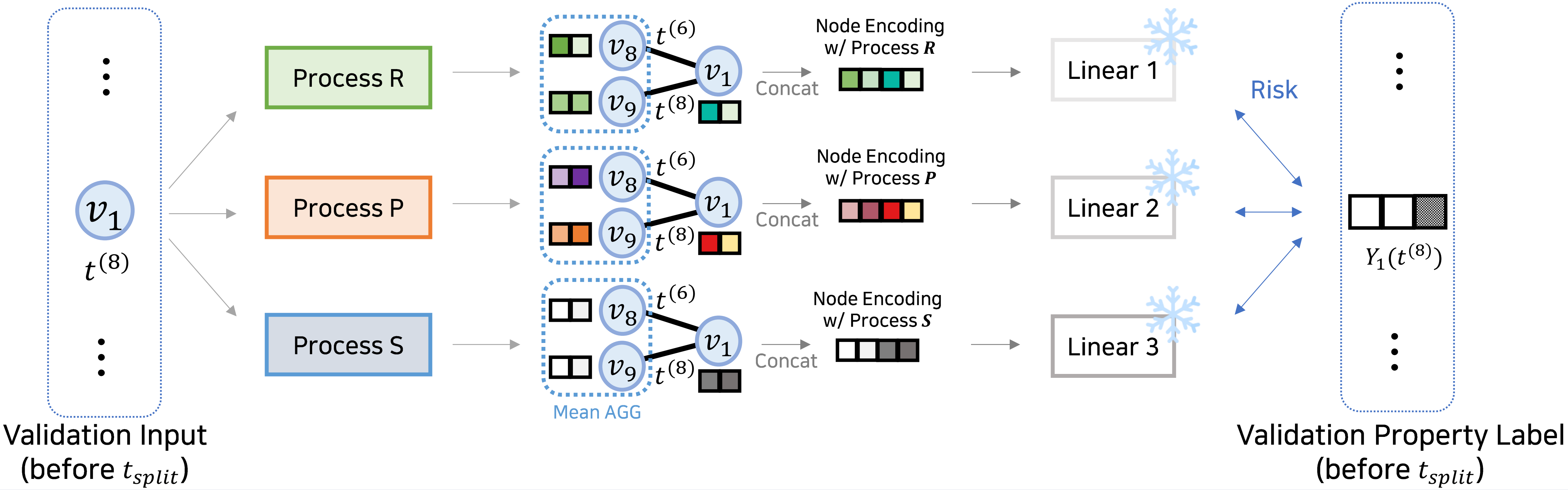
- Each augmentation process generates node encodings using recent neighbors, followed by training the corresponding linear model to fit the property labels



Node Feature Selection

Evaluation Using the Validation Set

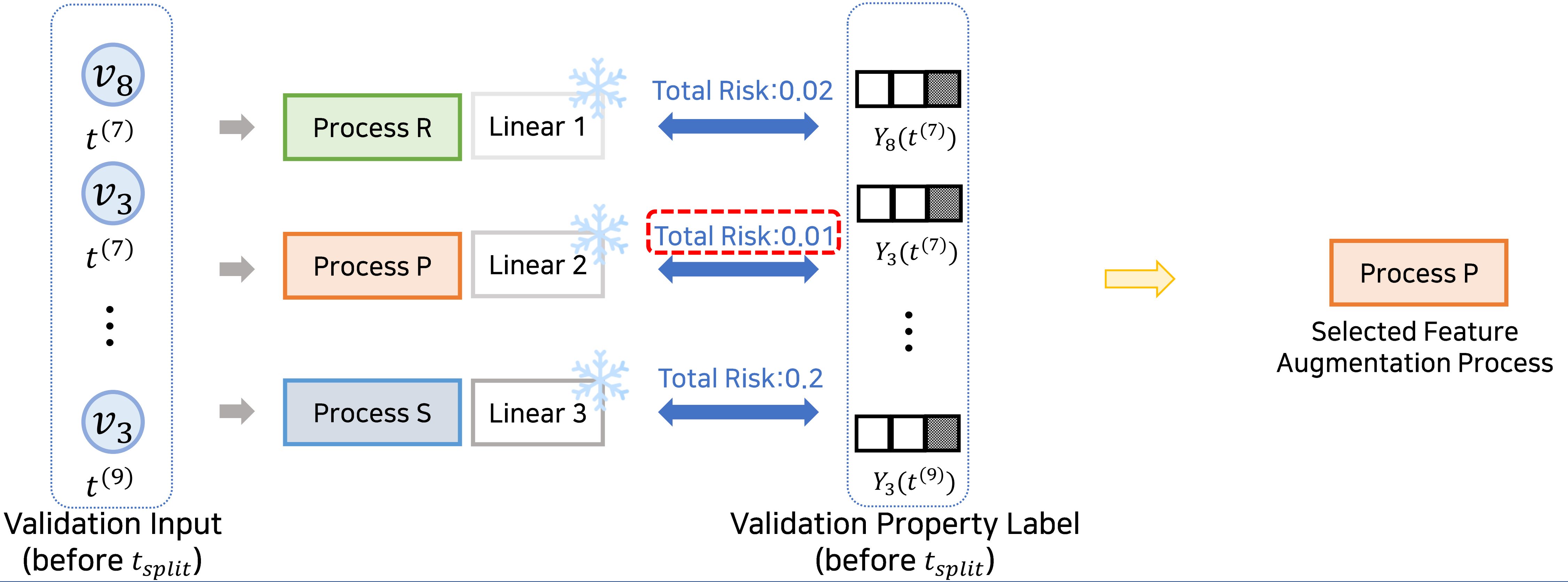
- The risk of each augmentation process with the trained linear model is evaluated on the validation set



Node Feature Selection

Augmentation Process Selection

- The feature augmentation process with the lowest sum of risks on the validation set is selected

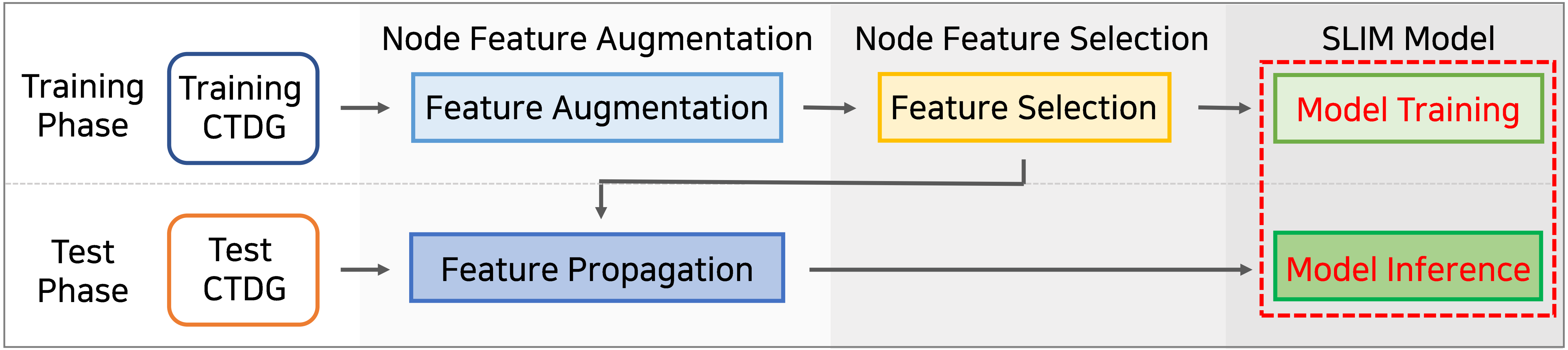


Overview of SPLASH

(1) Node Feature Augmentation

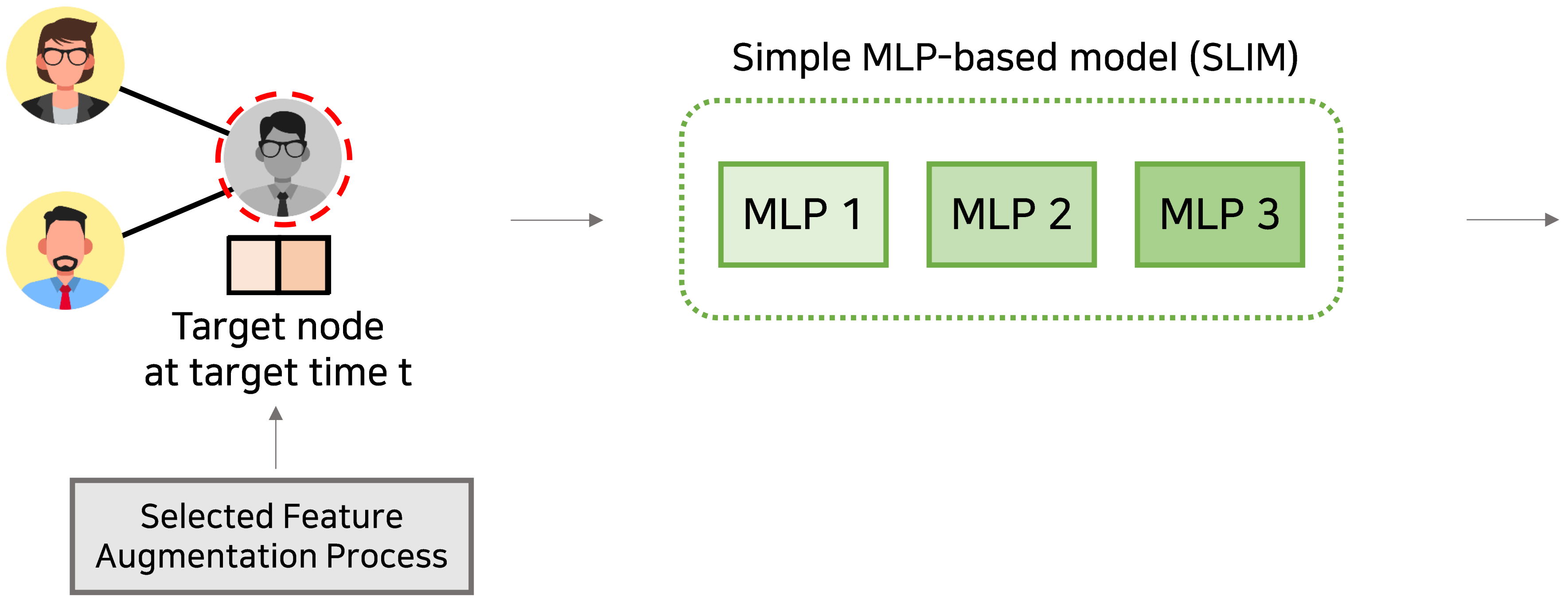
(2) Node Feature Selection

(3) SLIM Model (Our Proposed TGNN Model)



SLIM Model

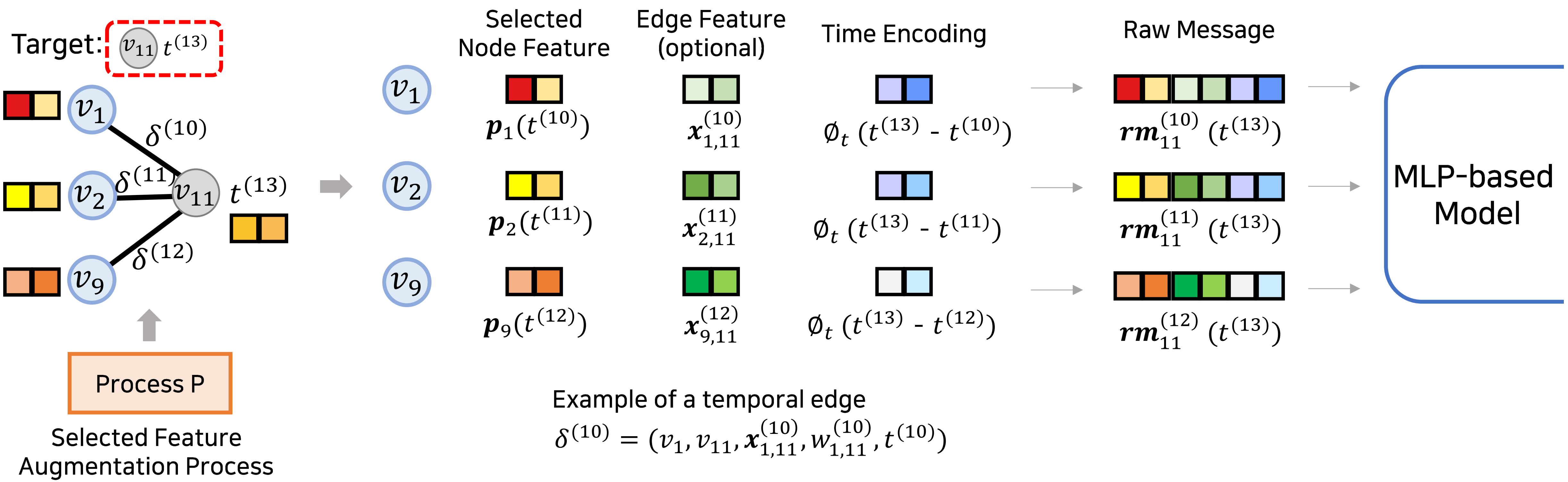
Goal: to efficiently and effectively predict node properties using a simple MLP-based model with augmented node features



SLIM Model

Message Encoding Module

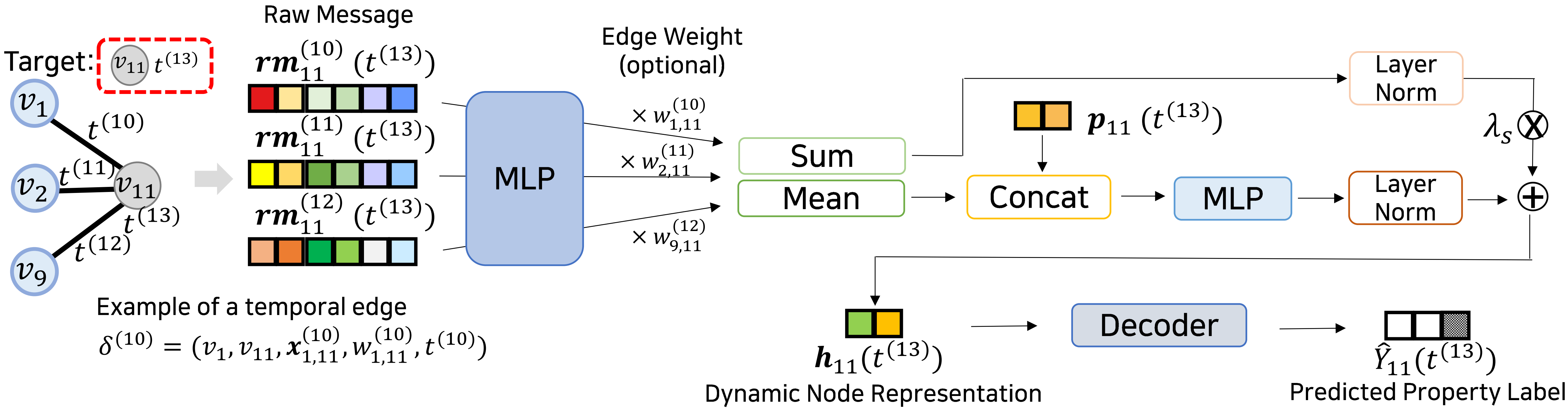
- SLIM first generates messages from recent temporal edges using selected augmentation process and MLPs



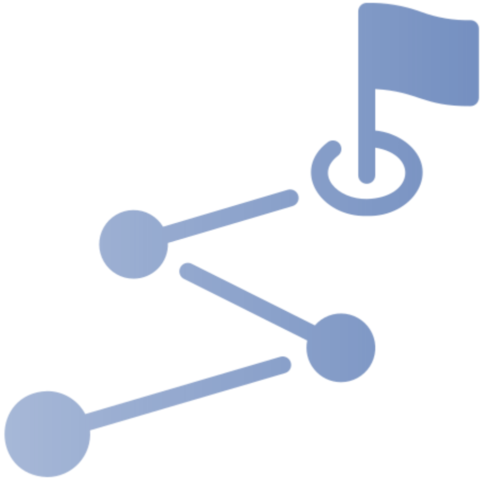
SLIM Model

Proposed SLIM Model Architecture

- SLIM computes the latest representation of a given target node by aggregating the messages in the previous module and predicts its property label



Contents



Research Questions

RQ1) Accuracy & Generalization

- ✓ How accurately does SPLASH predict node properties under distribution shifts?

RQ2) Efficiency & Scalability

- ✓ How efficient and scalable is SPLASH?

RQ3) Qualitative Analysis

- ✓ Does SPLASH outperform other baselines in qualitative evaluation?

Experimental Settings

Datasets

- ✓ 3 social networks (Wikipedia, Reddit, MOOC) for dynamic anomaly detection
- ✓ Email network (Email-EU) and event network (GDELT) for dynamic node classification
- ✓ Trade network (tgbn-trade) and music platform network (tgbn-genre) for node affinity prediction

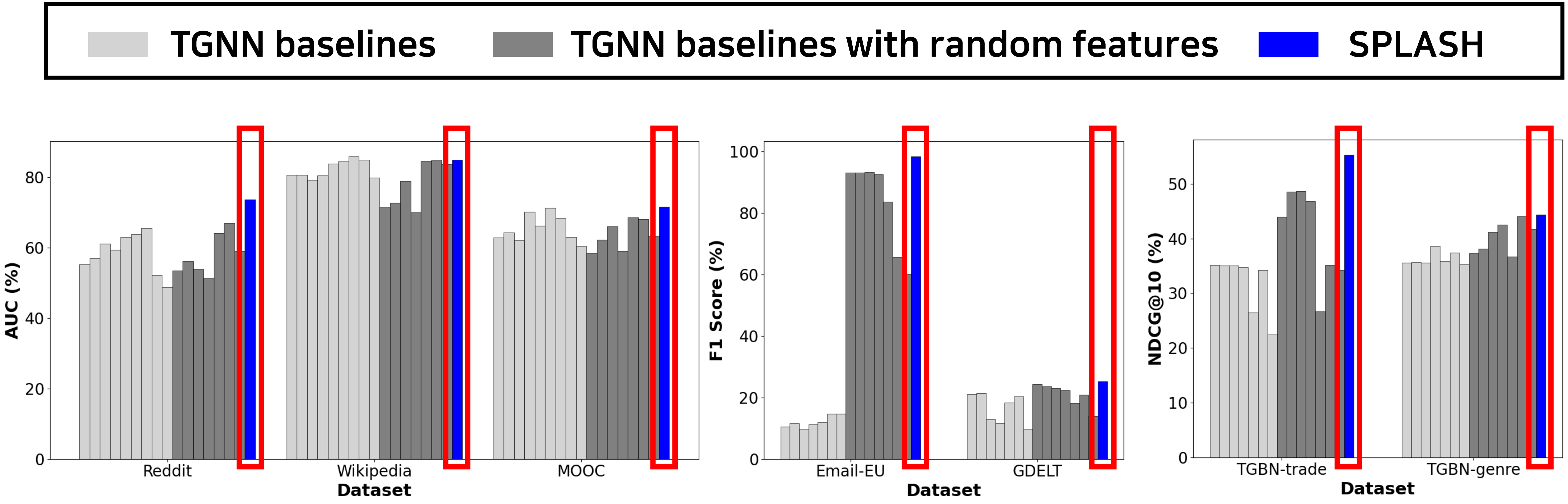
Baselines

- ✓ 8 TGNNs: JODIE, DySAT, TGAT, TGN, GraphMixer, DyGFormer, FreeDyG, SLADE
- ✓ 8 TGNNs with random features: JODIE+RF, DySAT+RF, TGAT+RF, TGN+RF, GraphMixer+RF, DyGFormer+RF, FreeDyG+RF, SLADE+RF

RQ1) Accuracy & Generalization

SPLASH outperforms other baselines in node property prediction

- Including TGNNs without node features and TGNNs using random features

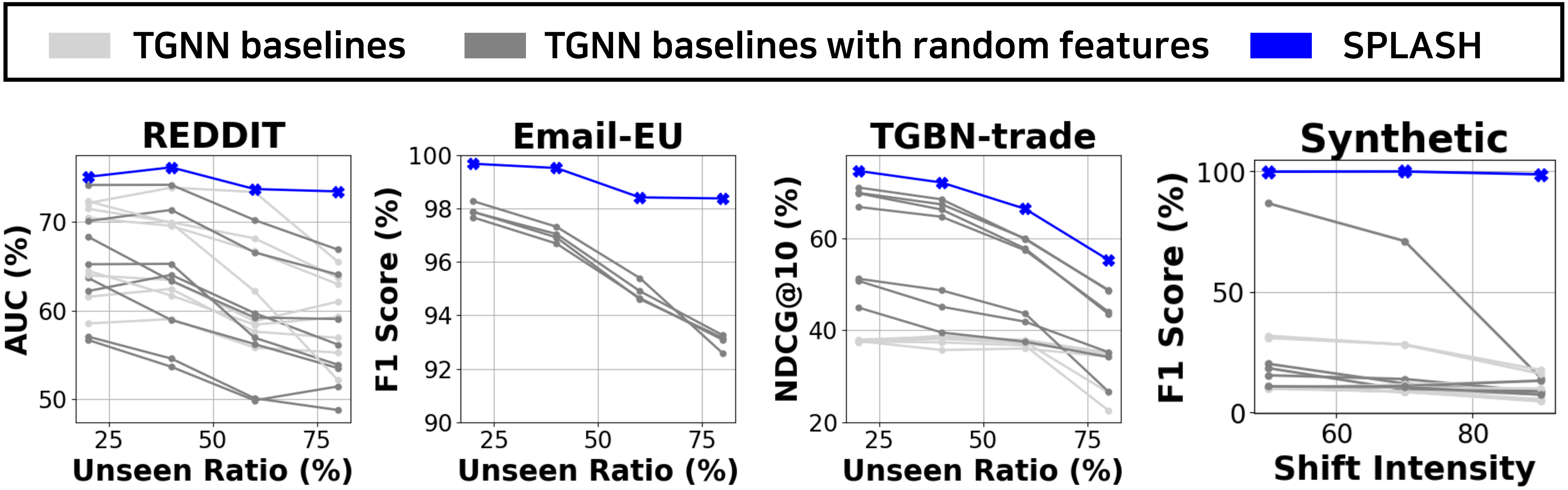


Dynamic anomaly detection Dynamic node classification Node affinity prediction

RQ1) Accuracy & Generalization

SPLASH shows better generalization capabilities compared to other baselines

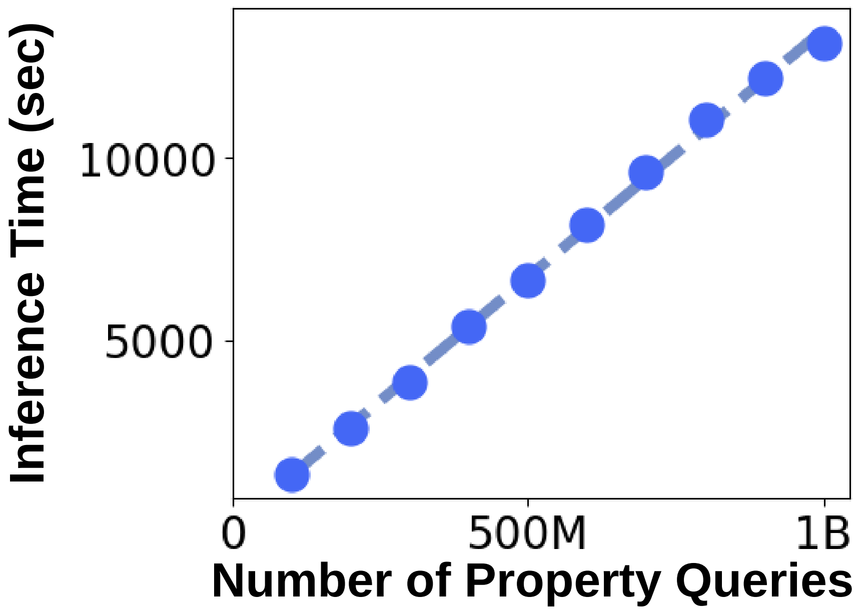
- As the distribution shift becomes more severe, the performance gap widens

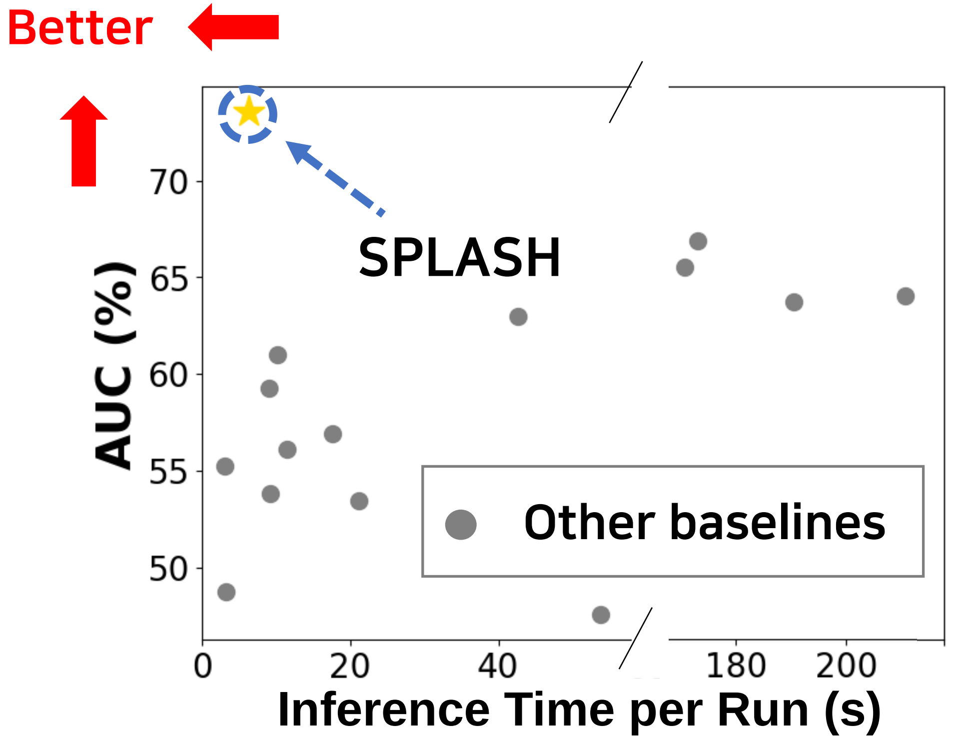


RQ2) Efficiency & Scalability

SPLASH enables efficient and scalable node property prediction

- Maintains a constant inference time per property query regardless of graph size
- Offers the best trade-off between performance and inference time

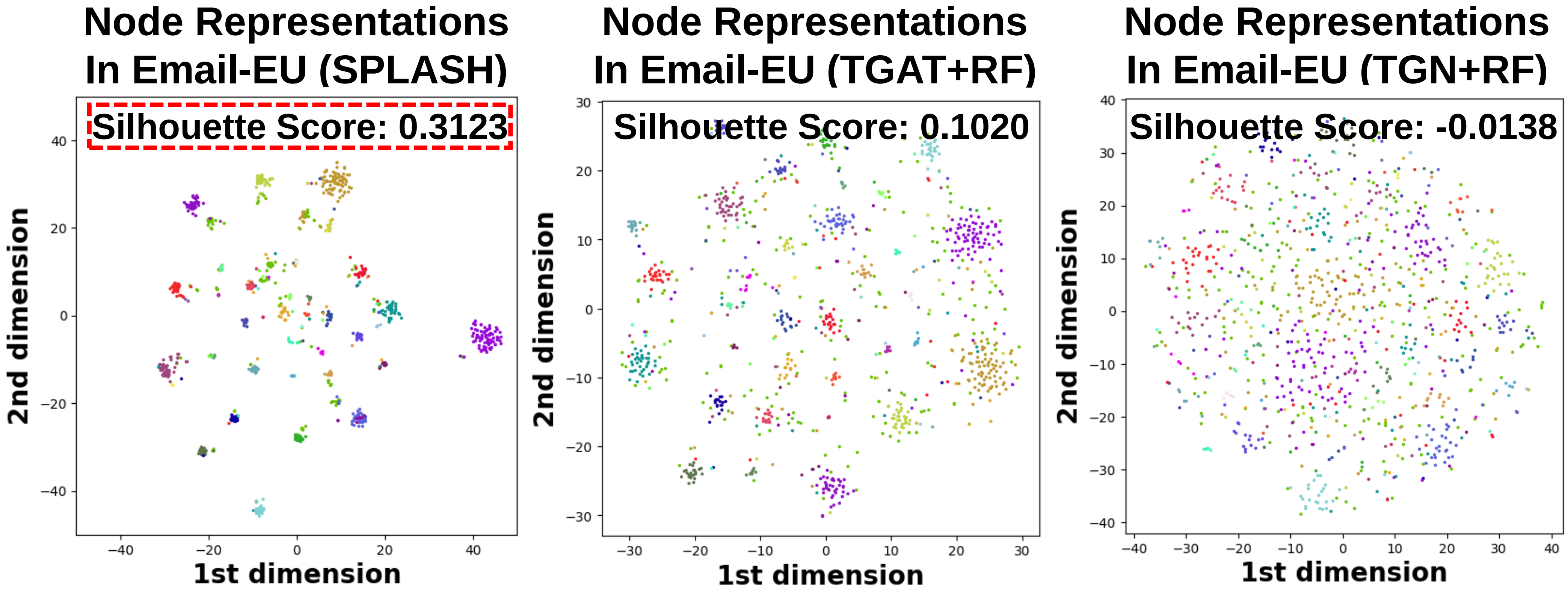




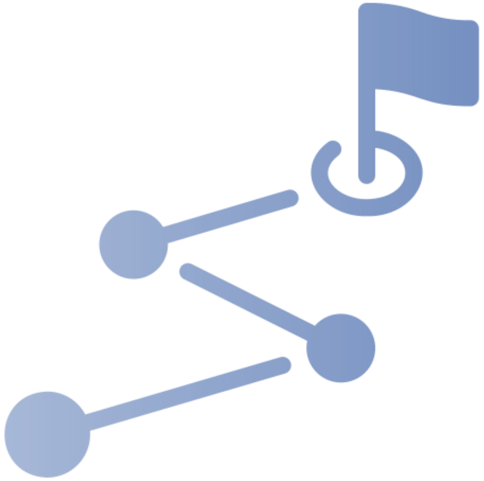
RQ3) Qualitative Analysis

SPLASH shows qualitatively better performance than other baselines

- In the Email EU dataset, which has a static class property, SPLASH generates the most cohesive clusters for each class compared to other baselines



Contents



Conclusion

Github: https://github.com/jhsk777/SPLASH

- We propose SPLASH, a simple yet effective method for node property prediction in edge streams under distribution shifts
  - ✓ Fast & Lightweight: SPLASH uses only MLP layers, enabling fast inference
  - ✓ Effective: SPLASH outperforms other baselines in node property prediction
  - ✓ Robust: SPLASH shows the smallest performance drop as the distribution shift intensifies