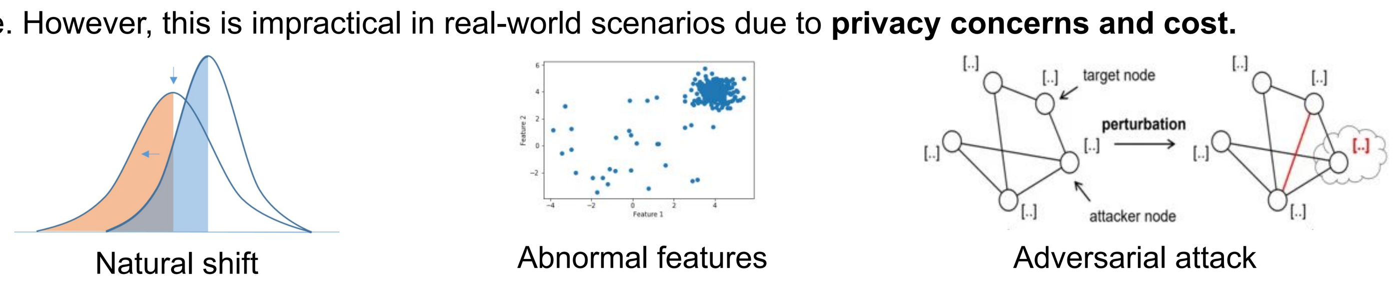
KAIST

Dynamic Uncertainty Minimization with Adaptive Batch Normalization Statistic (DUMBS)

Sorn Chottananurak   Settasit Murichan

Abstract

Graph neural networks (GNNs) demonstrate strong representation learning performance on training and testing data when both are drawn from the same distribution. However, **their performance degrades significantly when there is a distribution shift between training and testing data**, such as abnormal features and adversarial structure attacks. Traditional approaches suggest using labeled data from both the training and test-time distributions to address this issue. However, this is impractical in real-world scenarios due to **privacy concerns and cost.**

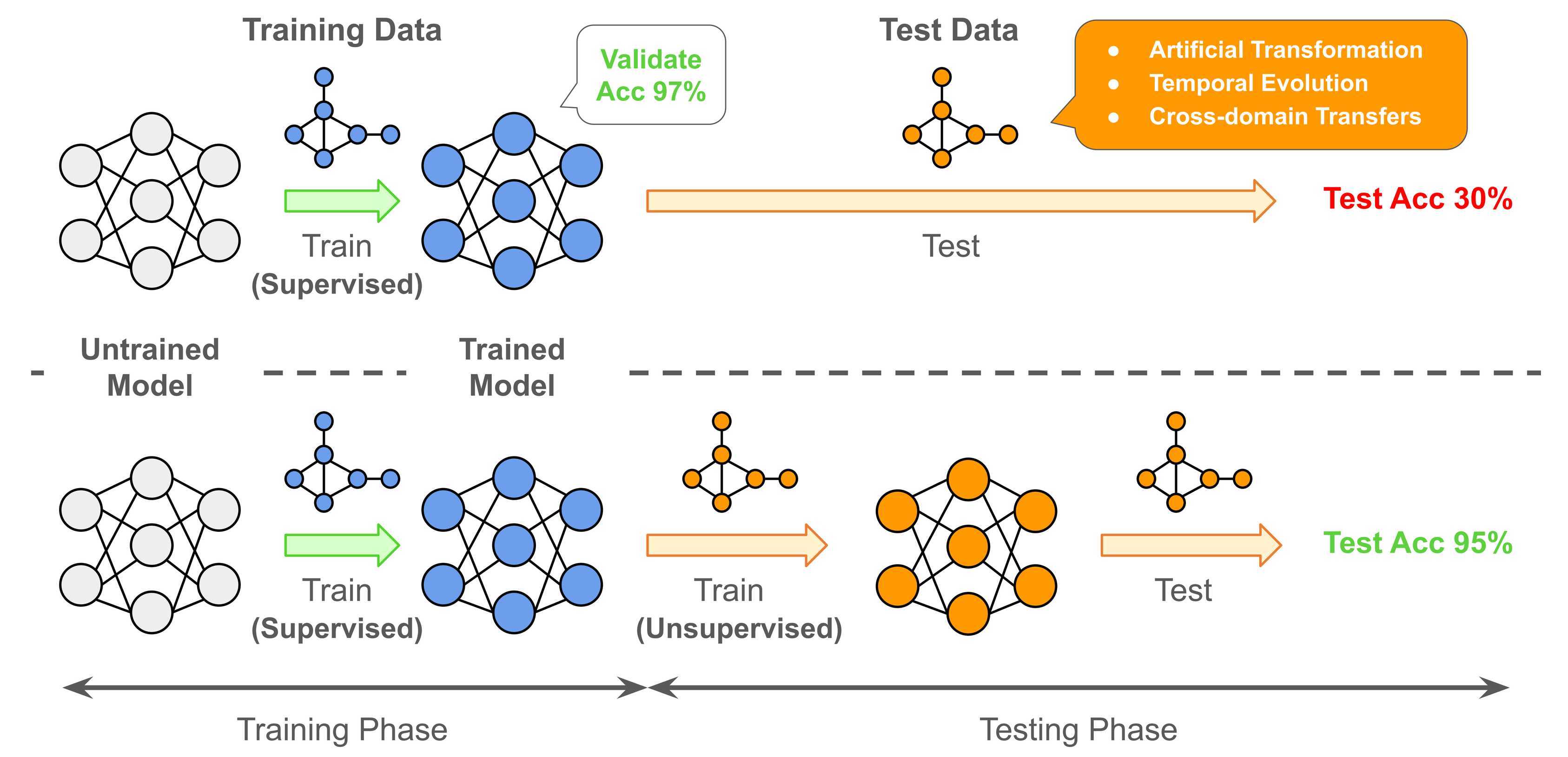


GTrans [4], the **data-centric approach to adapt graph data at test time** is proposed to improve performance under the domain shift. However, solely focusing on data adaptation without optimizing the modeling aspect and batch normalization statistic can lead to suboptimal results. Recognizing this opportunity, we propose a method named **DUMBS: Dynamic Uncertainty Minimization with Adaptive Batch Normalization Statistic** which **concurrently minimizes the uncertainty of submodels generated from Monte Carlo dropout inference and optimizes batch normalization statistics during test time.** This approach aims to fully leverage all available optimization parameters of the model to better mitigate the effects of domain shift and enhance overall adaptation performance. Remarkably, **DUMBS achieves better performance compare to baselines** as well as surpasses state-of-the-art method such as GTrans on the majority of dataset used.

1

Motivation

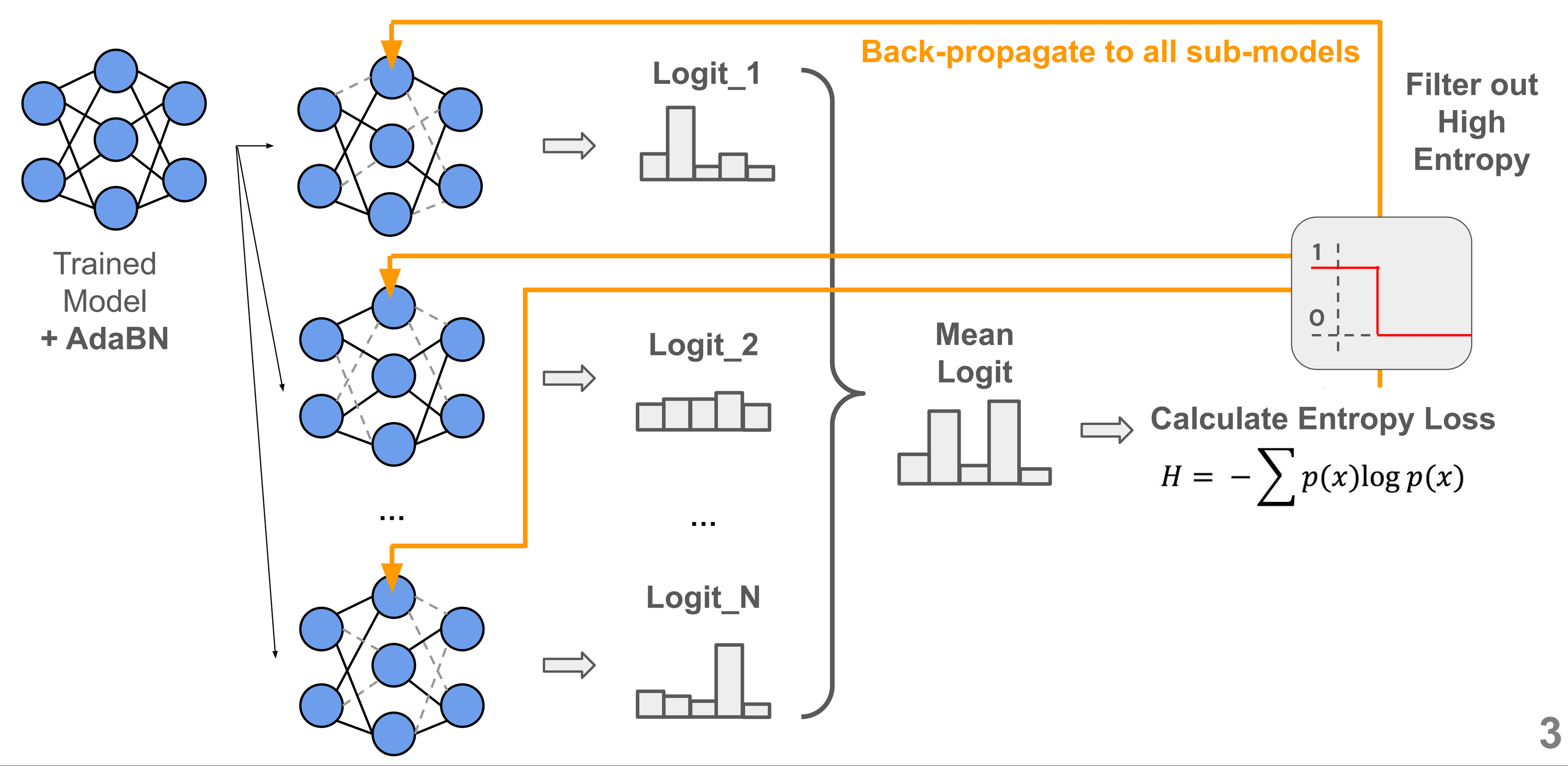
Distribution Shift problem is solved by Test-Time Training!



2

Proposed Method

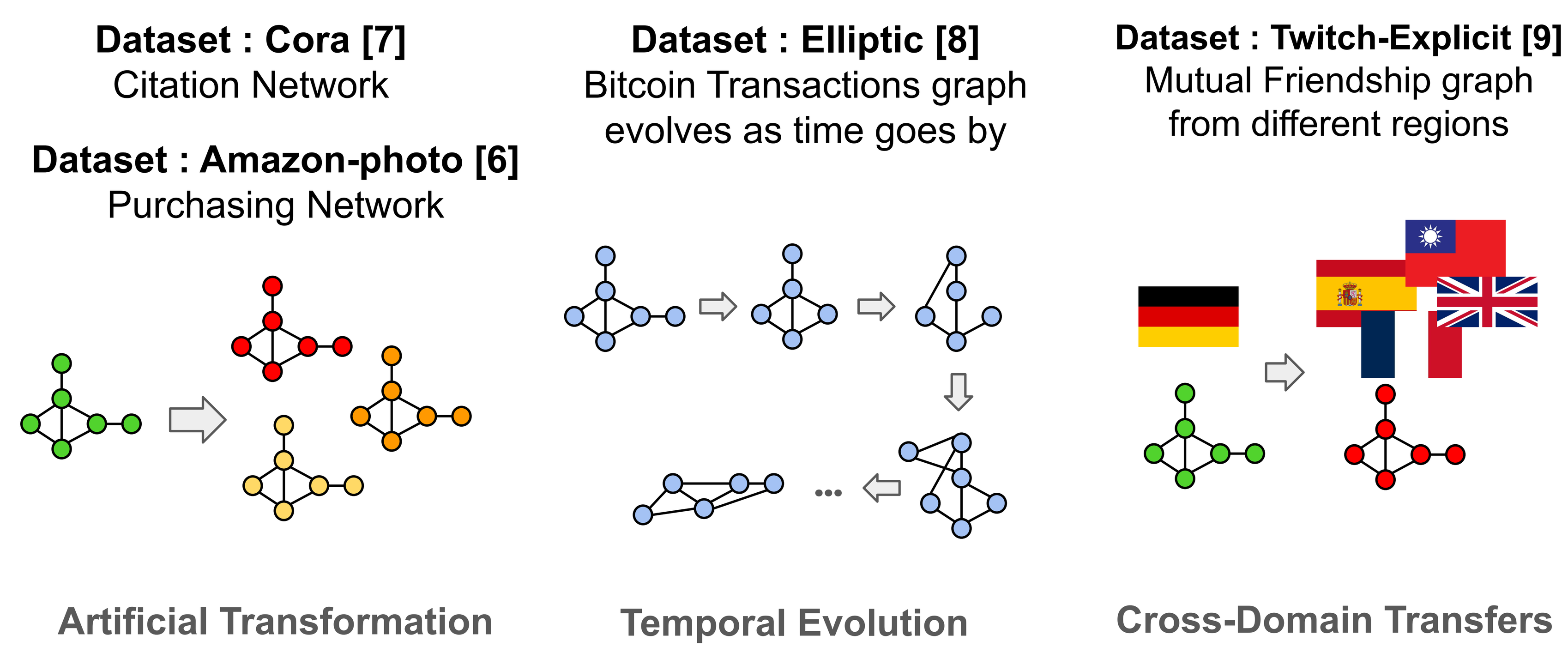
DUMPS: Dynamic Uncertainty Minimization with Adaptive Batch Normalization Statistic



3

Setting and Dataset

Out-Of-Distribution (OOD) Setting for Node Classification Task



4

Empirical Evaluation : Main Experiment

DUMBS significantly outperforms all S.O.T.A. baselines in GCN

Backbone	Method	Dataset				Average
		Amz-Photo [6]	Cora [7]	Elliptic [8]	Twitch-E [9]	
GCN	ERM	93.79 (±0.97)	91.59 (±1.44)	50.90 (±1.51)	59.89 (±0.50)	74.04±(1.11)
	DropEdge [1]	92.11 (±0.31)	81.01 (±1.33)	53.96 (±4.91)	59.95 (±0.39)	71.76±(1.74)
	TENT [2]	94.03 (±1.07)	91.87 (±1.36)	51.71 (±2.00)	59.46 (±0.55)	74.27±(1.25)
	EERM [3]	94.05 (±0.40)	87.21 (±0.53)	53.96 (±0.65)	59.85 (±0.85)	73.77±(0.61)
	GTrans [4]	94.13 (±0.77)	94.66 (±0.63)	55.88 (±3.10)	60.42 (±0.86)	76.27±(1.34)
	DUMPS (ours)	**96.18** (±0.27)	**98.31** (±0.21)	**62.20** (±1.17)	**60.45** (±0.39)	**79.29**±(0.51)

5

Empirical Evaluation : Additional Experiment

Robustness of DUMBS is generalized to SAGE and GAT architecture!

Backbone	Method	Dataset				Average
		Amz-Photo [6]	Cora [7]	Elliptic [8]	Twitch-E [9]	
GCN	ERM	93.79 (±0.97)	91.59 (±1.44)	50.90 (±1.51)	59.89 (±0.50)	74.04 (±1.11)
	DropEdge [1]	92.11 (±0.31)	81.01 (±1.33)	53.96 (±4.91)	59.95 (±0.39)	71.76 (±1.74)
	TENT [2]	94.03 (±1.07)	91.87 (±1.36)	51.71 (±2.00)	59.46 (±0.55)	74.27 (±1.25)
	EERM [3]	94.05 (±0.40)	87.21 (±0.53)	53.96 (±0.65)	59.85 (±0.85)	73.77 (±0.61)
	GTrans [4]	94.13 (±0.77)	94.66 (±0.63)	55.88 (±3.10)	60.42 (±0.86)	76.27 (±1.34)
	DUMPS (ours)	**96.18** (±0.27)	**98.31** (±0.21)	**62.20** (±1.17)	**60.45** (±0.39)	**79.29** (±0.51)
SAGE	ERM	95.09 (±0.60)	99.67 (±0.14)	56.12 (±4.47)	62.06 (±0.09)	78.24 (±1.33)
	DropEdge [1]	92.61 (±0.56)	95.85 (±0.30)	52.38 (±3.11)	62.14 (±0.12)	75.75 (±1.02)
	TENT [2]	95.72 (±0.43)	99.80 (±0.10)	55.89 (±4.87)	62.09 (±0.09)	78.38 (±1.37)
	EERM [3]	95.57 (±0.13)	98.77 (±0.14)	58.20 (±3.55)	62.11 (±0.12)	78.66 (±0.99)
	GTrans [4]	96.91 (±0.68)	99.45 (±0.13)	60.81 (±5.19)	62.15 (±0.13)	79.83 (±1.53)
	DUMPS (ours)	**99.33** (±0.25)	**99.99** (±0.02)	**62.48** (±5.07)	**62.41** (±0.14)	**81.05** (±1.37)
GAT	ERM	96.30 (±0.79)	94.81 (±1.44)	65.36 (±2.70)	58.53 (±1.00)	78.75 (±1.48)
	DropEdge [1]	90.70 (±0.29)	76.91 (±1.55)	63.78 (±2.39)	58.89 (±1.01)	72.57 (±1.31)
	TENT [2]	95.99 (±0.46)	95.91 (±1.14)	66.07 (±1.66)	58.33 (±1.18)	79.08 (±1.11)
	EERM [3]	95.57 (±1.32)	85.00 (±0.96)	58.14 (±4.71)	**59.84** (±0.71)	74.64 (±1.93)
	GTrans [4]	96.67 (±0.74)	96.37 (±1.00)	**66.43** (±2.57)	58.59 (±1.07)	79.52 (±1.35)
	DUMPS (ours)	**98.10** (±0.74)	**99.72** (±0.12)	62.15 (±1.15)	58.42 (±1.40)	**79.60** (±0.85)

6

Ablation Study

Ablation Study to show impacts of different components of DUMPS

Backbone	Components	Dataset				Average
		Amz-Photo [6]	Cora [7]	Elliptic [8]	Twitch-E [9]	
GCN	No Adapt	93.79 (±0.97)	91.59 (±1.44)	50.90 (±1.51)	59.89 (±0.50)	74.04 (±1.11)
	AdaBN	95.98 (±0.14)	94.15 (±0.51)	56.42 (±1.21)	58.91 (±0.55)	76.37 (±0.60)
	AdaBN + MC	96.05 (±0.23)	97.14 (±0.12)	60.19 (±1.14)	60.23 (±0.21)	78.40 (±0.43)
	AdaBN + MC + EntThreshold	**96.18** (±0.27)	**98.31** (±0.21)	**62.20** (±1.17)	**60.45** (±0.39)	**79.29** (±0.51)

7

Conclusion & Reference

Conclusion

GNNs tend to yield unsatisfying performance when the presented data is sub-optimal. While approaches like GTrans [4] that focus on data adaptation at test time can lead to a better performance on distribution shift data, solely prioritizing this aspect can lead to sub-optimal outcomes. To address this, we propose **DUMBS: Dynamic Uncertainty Minimization with Adaptive Batch Normalization Statistic** that concurrently minimizes the uncertainty of submodels generated from Monte Carlo dropout inference and optimizes batch normalization statistics during test time. Experimental results on out-of-distribution (OOD) dataset for node classification task have demonstrated the effectiveness of our method. DUMBS outperforming various state-of-the-art method in GCN baseline as well as achieving generalization in SAGE and GAT architecture.

Reference

[1] Yu Rong., et al, "DropEdge: Towards Deep Graph Convolutional Networks on Node Classification," in International Conference on Learning Representations, 2020.
[2] Wang, D., et al, "Tent: Fully Test-Time Adaptation by Entropy Minimization," in International Conference on Learning Representations, 2021
[3] Qitian Wu., et al, "Handling Distribution Shifts on Graphs: An Invariance Perspective," in International Conference on Learning Representations (ICLR), 2022.
[4] Wei Jin., et al, "Empowering Graph Representation Learning with Test-Time Graph Transformation," in The Eleventh International Conference on Learning Representations , 2023.
[5] T. Kipf, M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in International Conference on Learning Representations (ICLR), 2017.
[6] Z. Yang, W. Cohen, R. Salakhudinov, "Revisiting Semi-Supervised Learning with Graph Embeddings," in Proceedings of The 33rd International Conference on Machine Learning, 2016, pp. 40–48.
[7] Oleksandr Shchur., et al, "Pitfalls of Graph Neural Network Evaluation," 2019.
[8] A. Pareja, “EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs”, AAAI, vol. 34, no. 04, pp. 5363-5370, Apr. 2020.
[9] Derek Lim., et al, "New Benchmarks for Learning on Non-Homophilous Graphs," 2021.

8