Matcha: Mitigating Graph Structure Shifts with Test-Time Adaptation





Wenxuan Bao **Zhichen Zeng** **Zhining Liu** **Hanghang Tong** **Jingrui He**

University of Illinois Urbana-Champaign

{wbao4,zhichenz,liu326,htong,jingrui}@illinois.edu

Challenge: Distribution Shifts in Graphs

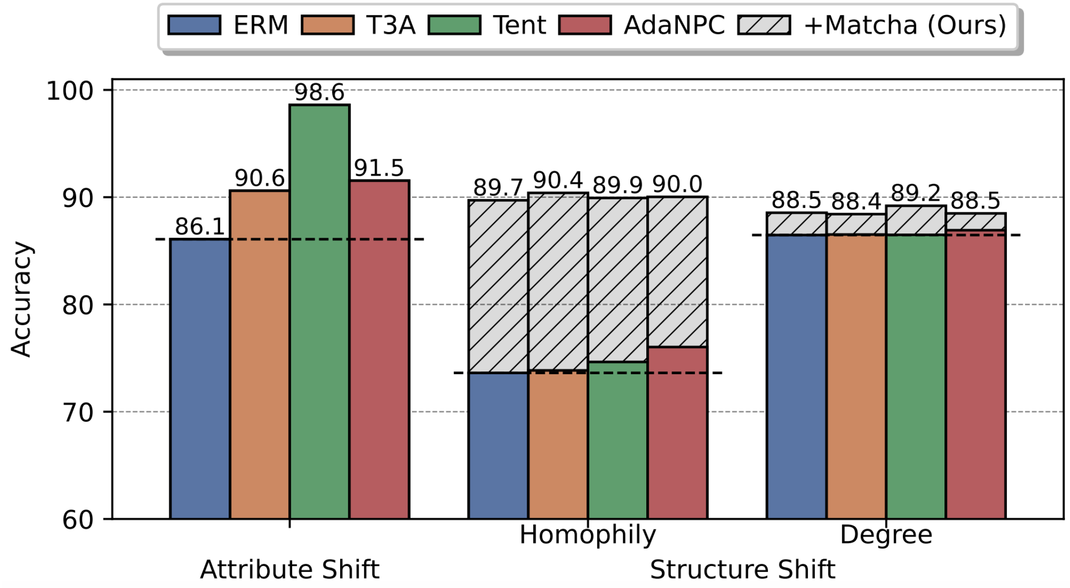
- Graph neural networks (GNNs) are vulnerable to distribution shifts.

- **Attribute shifts**: Node feature distributions are different.
  - LinkedIn: Share research & find jobs.
  - Instagram: Share trips & activities.
- **Structure shifts**: Node connectivity patterns are different.
  - LinkedIn: Follow more professional colleges.
  - Instagram: Follow more family & friends.
- Structure shifts include, but not limit to:
  - *Degree shift*: Changes in average node degree.
  - *Homophily shift*: Changes in average node homophily.

Test-Time Adaptation

- **Test-Time Adaptation (TTA)** addresses distribution shifts by adapting a source model to the target domain without access to source data.
- Many existing TTA methods (T3A, Tent, AdaNPC) are developed for images.
- These methods perform well under *attribute shifts*, but often fail under *structure shifts.*

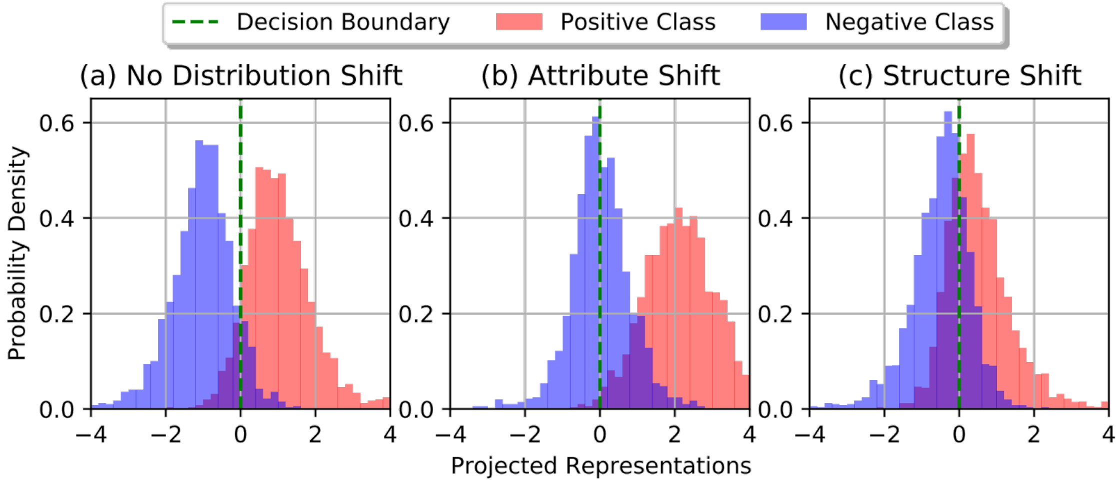
- Why does this performance gap exist?
- How can we enhance the performance of TTA under graph structure shifts?



[1] Yusuke Iwasawa, Yutaka Matsuo. Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization. NeurIPS 2021.
[2] Dequan Wang, et al. Tent: Fully Test-Time Adaptation by Entropy Minimization. ICLR 2021.
[3] Yifan Zhang, et al. AdaNPC: Exploring Non-Parametric Classifier for Test-Time Adaptation. ICML 2023.

Why Generic TTA Fails on Structure Shifts?

- We visualize the distribution of node representations (projected to 1-D).



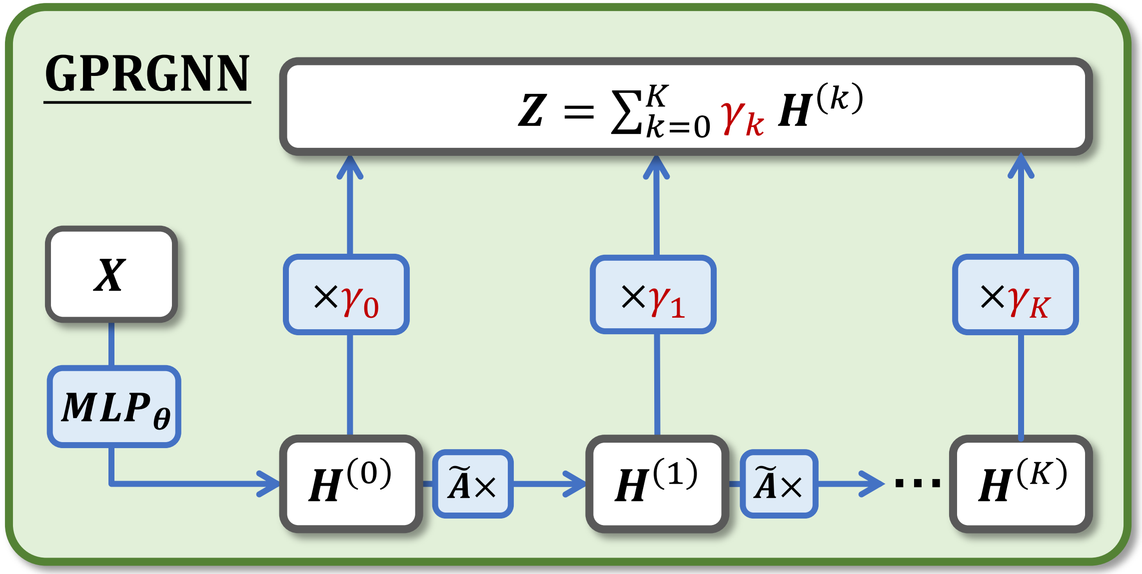
Histograms of Projected Representations Under Different Distribution Shifts

- **Attribute shifts and structure shifts have different impact patterns!**
  - *Attribute shifts introduce classifier bias*:
    - Node representations remain discriminative.
    - Can be handled by adjusting the decision boundary.
  - *Structure shifts introduce representation degradation:*
    - Node representations are overlapping.
    - Cannot be handled by adjusting the decision boundary.

Remark: This phenomenon is also supported by theory in our paper!

Adapt the Hop-Aggregation Parameters

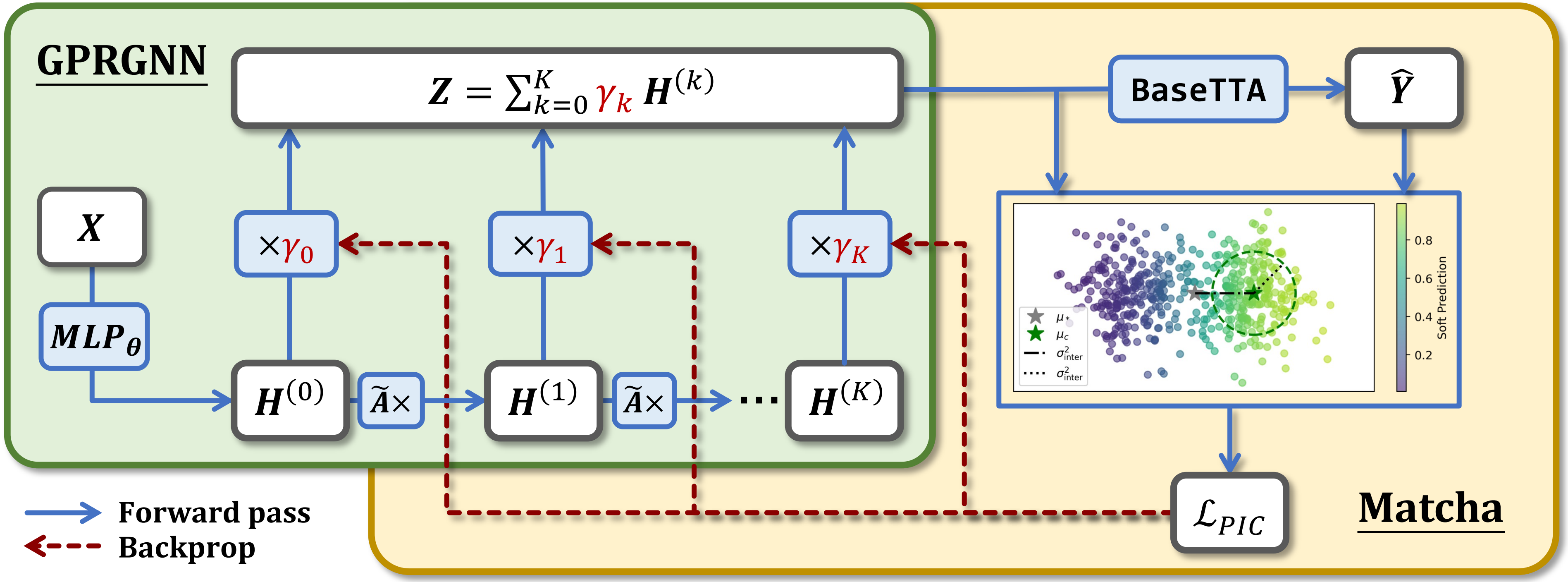
- Many GNN architectures have hop-aggregation parameters:
  - Control how GNNs integrate node features with neighbor information across different hops.
  - Example: $\boldsymbol{\gamma} = [\gamma_0, \cdots, \gamma_K]$ in GPRGNN.
- Structure shifts does not affect $\boldsymbol{H}^{(0)}$, but change the signal-to-noise ratio in $\boldsymbol{H}^{(1)}, \cdots, \boldsymbol{H}^{(K)}$.
  - **The hop-aggregation parameters $\boldsymbol{\gamma}$ should be adjusted accordingly!**



[1] Eli Chien, et al. Adaptive Universal Generalized PageRank Graph Neural Network. ICLR 2021.

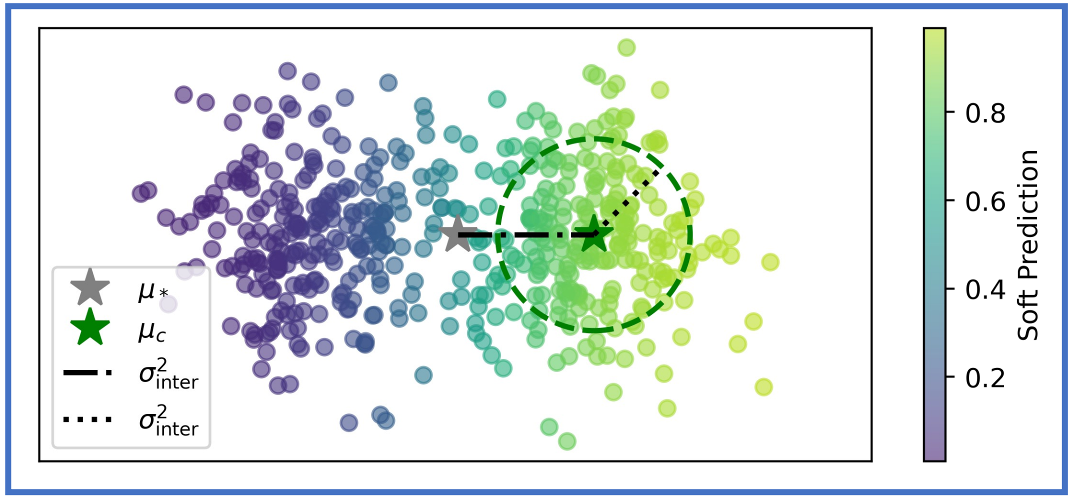
Matcha: Overview

- We propose *Matcha* to enhance the performance of generic TTA methods by adjusting the hop-aggregation parameters.



Prediction-Informed Clustering Loss

- We proposed a new loss function: prediction-informed clustering (PIC) loss



$\mathcal{L}_{\text{PIC}} = \frac{\sigma^2_{\text{intra}}}{\sigma^2_{\text{intra}} + \sigma^2_{\text{inter}}}$, where

- Intra-class variance: $\sum_{i=1}^{M} \sum_{c=1}^{C} \widehat{\boldsymbol{Y}}_{i,c} \|\boldsymbol{z}_i - \boldsymbol{\mu}_c\|_2^2$
- Inter-class variance: $\sum_{c=1}^{C} \left(\sum_{i=1}^{M} \widehat{\boldsymbol{Y}}_{i,c}\right) \|\boldsymbol{\mu}_c - \boldsymbol{\mu}_*\|_2^2$
- Centroid for class c: $\boldsymbol{\mu}_c = \frac{\sum_{i=1}^{M} \widehat{\boldsymbol{Y}}_{i,c} \boldsymbol{z}_i}{\sum_{i=1}^{M} \widehat{\boldsymbol{Y}}_{i,c}}$ Centroid for all nodes: $\boldsymbol{\mu}_* = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{z}_i$

- Intuition
  - Small intra-class variance σ^2_{intra}, large inter-class variance σ^2_{inter}

Integration of Generic TTA Methods

- Integrate generic TTA algorithms to handle structure shift and attribute shift simultaneously

Algorithm 1 `Matcha`

`Matcha` (target graph $\mathcal{T}$, featurizer $f_{\boldsymbol{\theta},\boldsymbol{\gamma}}$, classifier $g_{\boldsymbol{w}}$, baseline TTA method `BaseTTA`)

1: **for** epoch $t = 1$ to T **do**

2: Apply generic TTA:
$\hat{\boldsymbol{Y}} \leftarrow \texttt{BaseTTA}(\mathcal{T}, f_{\boldsymbol{\theta},\boldsymbol{\gamma}}, g_{\boldsymbol{w}})$

3: Update hop-aggregation parameters:
$\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \eta \nabla_{\boldsymbol{\gamma}} \mathcal{L}(\mathcal{T}, f_{\boldsymbol{\theta},\boldsymbol{\gamma}}, g_{\boldsymbol{w}}, \hat{\boldsymbol{Y}})$

4: **return** $\hat{\boldsymbol{Y}} \leftarrow \texttt{BaseTTA}(\mathcal{T}, f_{\boldsymbol{\theta},\boldsymbol{\gamma}}, g_{\boldsymbol{w}})$

- In each optimization step,
  - First apply base TTA algorithm to get predictions $\{\widehat{\boldsymbol{Y}}_{i,c}\}$
  - Compute PIC loss with $\{\widehat{\boldsymbol{Y}}_{i,c}\}$ to optimize node representation

- Synergy between representation quality and prediction accuracy:
  - Better prediction → better pseudo-class for PIC loss, improving representation
  - Better representation quality → better prediction

Experiments: Handle Various Structure Shifts

- Matcha consistently enhances the performance of base TTA methods
  - Homo (homophilious), hetero (heterophilious), high (high degree), low (low degree)

Table 1: Accuracy (mean ± s.d. %) on CSBM with structure shifts and attribute shifts.

Method	Homophily shift		Degree shift		Attribute + homophily shift		Attribute + degree shift	
	homo → hetero	hetero → homo	high → low	low → high	homo → hetero	hetero → homo	high → low	low → high
ERM	73.62 ± 0.44	76.72 ± 0.89	86.47 ± 0.38	92.92 ± 0.43	61.06 ± 1.67	72.61 ± 0.38	77.63 ± 1.13	73.60 ± 3.53
+ Matcha	89.71 ± 0.27	90.68 ± 0.26	88.55 ± 0.44	93.78 ± 0.74	85.34 ± 4.68	74.70 ± 0.99	78.29 ± 1.41	73.86 ± 4.20
T3A	73.85 ± 0.24	76.68 ± 1.08	86.52 ± 0.44	92.94 ± 0.37	65.77 ± 2.11	72.92 ± 0.90	80.89 ± 1.28	81.94 ± 3.24
+ Matcha	90.40 ± 0.11	90.50 ± 0.24	88.42 ± 0.60	93.83 ± 0.41	88.49 ± 0.58	79.34 ± 1.85	81.82 ± 1.36	82.12 ± 4.03
Tent	74.64 ± 0.38	79.40 ± 0.57	86.49 ± 0.50	92.84 ± 0.18	74.42 ± 0.41	79.57 ± 0.40	86.05 ± 0.33	93.06 ± 0.24
+ Matcha	89.93 ± 0.16	**91.26 ± 0.08**	**89.20 ± 0.20**	**94.88 ± 0.09**	**90.12 ± 0.07**	**91.15 ± 0.20**	**87.76 ± 0.16**	**95.04 ± 0.06**
AdaNPC	76.03 ± 0.46	81.66 ± 0.17	86.92 ± 0.38	91.15 ± 0.39	63.96 ± 1.31	76.33 ± 0.71	77.69 ± 0.91	76.24 ± 3.06
+ Matcha	90.03 ± 0.33	90.36 ± 0.67	88.49 ± 0.31	92.84 ± 0.57	85.81 ± 0.30	77.63 ± 1.55	78.41 ± 1.03	76.31 ± 3.68
GTrans	74.01 ± 0.44	77.28 ± 0.56	86.58 ± 0.11	92.74 ± 0.13	71.60 ± 0.60	74.45 ± 0.42	83.21 ± 0.25	89.40 ± 0.62
+ Matcha	89.47 ± 0.20	90.31 ± 0.31	87.88 ± 0.77	93.23 ± 0.52	88.88 ± 0.38	76.87 ± 0.66	83.41 ± 0.16	89.98 ± 0.93
SOGA	74.33 ± 0.18	83.99 ± 0.35	86.69 ± 0.37	93.06 ± 0.21	70.45 ± 1.71	76.41 ± 0.79	81.31 ± 1.03	88.32 ± 1.94
+ Matcha	89.92 ± 0.26	90.69 ± 0.27	88.83 ± 0.32	94.49 ± 0.23	88.92 ± 0.28	90.14 ± 0.33	87.11 ± 0.28	93.38 ± 1.06
GraphPatcher	79.14 ± 0.62	82.14 ± 1.11	87.87 ± 0.18	93.64 ± 0.45	64.16 ± 3.49	76.98 ± 1.04	76.99 ± 1.43	73.31 ± 4.48
+ Matcha	**91.28 ± 0.28**	90.66 ± 0.15	88.01 ± 0.18	93.88 ± 0.69	89.99 ± 0.41	87.94 ± 0.39	78.43 ± 1.84	77.86 ± 4.14

Experiments on Real-World Setting

- Syn-Cora and Syn-Products
  - Only homophily shift
- Twitch-E and OGB-Arxiv
  - Natural attribute and structure shift
  - We randomly delete homophilic edges to inject more homophily and degree shifts
- Matcha also improves the model performance

Table 2: Accuracy on real-world datasets.

Method	Syn-Cora	Syn-Products	Twitch-E	OGB-Arxiv
ERM	65.67 ± 0.35	37.80 ± 2.61	56.20 ± 0.63	41.06 ± 0.33
+ Matcha	78.96 ± 1.08	69.75 ± 0.93	56.76 ± 0.22	41.74 ± 0.34
T3A	68.25 ± 1.10	47.59 ± 1.46	56.83 ± 0.22	38.17 ± 0.31
+ Matcha	78.40 ± 1.04	69.81 ± 0.36	56.97 ± 0.28	38.56 ± 0.27
Tent	66.26 ± 0.38	29.14 ± 4.50	58.46 ± 0.37	34.48 ± 0.28
+ Matcha	78.87 ± 1.07	68.45 ± 1.04	$\mathbf{58.57} \pm \mathbf{0.42}$	35.20 ± 0.27
AdaNPC	67.34 ± 0.76	44.67 ± 1.53	55.43 ± 0.50	40.20 ± 0.35
+ Matcha	77.45 ± 0.62	71.66 ± 0.81	56.35 ± 0.27	40.58 ± 0.35
GTrans	68.60 ± 0.32	43.89 ± 1.75	56.24 ± 0.41	41.28 ± 0.31
+ Matcha	$\mathbf{83.49} \pm \mathbf{0.78}$	$\mathbf{71.75} \pm \mathbf{0.65}$	56.75 ± 0.40	41.81 ± 0.31
SOGA	67.16 ± 0.72	40.96 ± 2.87	56.12 ± 0.30	41.23 ± 0.34
+ Matcha	79.03 ± 1.10	70.13 ± 0.86	56.62 ± 0.17	41.78 ± 0.34
GraphPatcher	63.01 ± 2.29	36.94 ± 1.50	57.05 ± 0.59	41.27 ± 0.87
+ Matcha	80.99 ± 0.50	69.39 ± 1.29	57.41 ± 0.53	$\mathbf{41.83} \pm \mathbf{0.90}$

Experiments: Visualization

- Matcha successfully restores the quality of node representations under structure shifts

- Better representations result in higher accuracy

Loss	Homophily shift		Degree shift	
	hom $\rightarrow$ het	het $\rightarrow$ hom	hi $\rightarrow$ lo	lo $\rightarrow$ hi
(None)	73.6 ± 0.4	76.7 ± 0.9	86.5 ± 0.4	92.9 ± 0.4
Entropy	75.9 ± 0.7	90.0 ± 0.2	86.8 ± 0.3	93.8 ± 0.7
PseudoLabel	77.3 ± 3.0	89.4 ± 0.2	86.7 ± 0.3	93.7 ± 0.7
$\sigma^2_{\text{intra}} - \sigma^2_{\text{inter}}$	76.1 ± 0.4	72.4 ± 0.7	82.6 ± 1.0	92.9 ± 0.4
PIC (Ours)	$\mathbf{89.7 \pm 0.3}$	$\mathbf{90.7 \pm 0.3}$	$\mathbf{88.6 \pm 0.4}$	$\mathbf{93.8 \pm 0.7}$

Key Takeaways

- Focusing on graph test-time adaptation (TTA), we find that **attribute shifts and structure shifts have different impact patterns**, which limit the performance of generic TTA algorithms.

- We propose ***Matcha*, adjusting the hop-aggregation parameters in GNNs**.
  - Address structure shifts effectively
  - Compatible to generic TTA algorithms to handle attribute shifts

- Our experiments show that *Matcha* improves model performance across different types of structure shifts.