You Can Have Better Graph Neural Networks by Not Training Weights at All: Finding Untrained GNNs Tickets

Tianjin Huang, Tianlong Chen, Meng Fang, Vlado Menkovski, Jiaxu Zhao, Lu Yin, Yulong Pei, Decebal Constantin Mocanu, Zhangyang Wang, Mykola Pechenizkiy, Shiwei Liu

Раевская Алеся

Основная идея

- В графовых нейтронных сетях существуют необученные подсети (лотерейные билеты), которые предсказывают результат с почти таким же качеством, как и обученные
- Генерируется сеть (одной из архитектур) с рандомными начальными весами и из нее выбирается хорошо работающая подсеть

Что такое гипотеза о лотерейном билете?

- В большой нейронной сети, найдется подсеть сильно меньше размером, которая выдает сравнимые по качеству результаты
- Такая сеть требует меньше ресурсов для обучения

Что произойдет если мы найдем такую сеть?

- Это будет новым способом получения рабочей GNN и стандартное обучение перестанет быть незаменимым
- Это продвинет исследования в наличие необучаемых рабочих нейронных сетей и в остальных областях

Как происходит поиск необученной НС?

- A - матрица ребер
- X - матрица признаков вершин
- m - маска

$$min_{m \in \{0,1\}^{|\theta|}} L(g(A, X; \theta \odot m), y)$$

Старые способы получения нетренированных сетей

- Edge-popup
  - Вычислим popup-score дальше оставляем топ k% весов в слое с самыми высокими popup-score
- Разреженность (sparsity) устанавливается в начале и не меняется в процессе обучения
- Слои юниформные - на каждом одинаковая sparsity

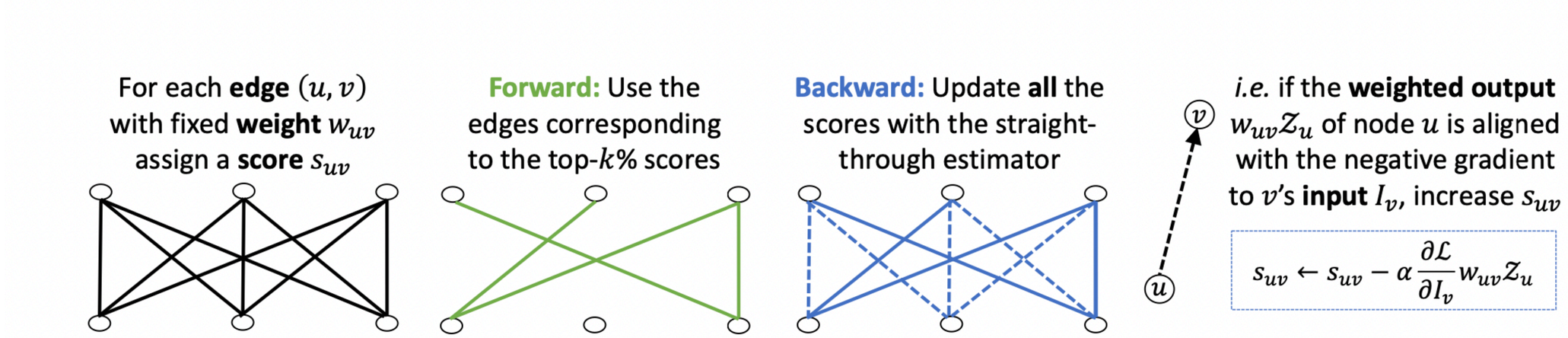


Figure 2. In the `edge-popup` Algorithm, we associate a score with each edge. On the forward pass we choose the top edges by score. On the backward pass we update the scores of all the edges with the straight-through estimator, allowing helpful edges that are "dead" to re-enter the subnetwork. We never update the value of any weight in the network, only the score associated with each weight.

Как происходит изменение sparsity?

- s_i - начальная sparsity
- s_f - конечная sparcity
- s_t - sparsity на шаге t
- t_0 - начальное время
- Δt - время между шагами спарсификации

$$s_t = s_f + (s_i - s_f)(1 - \frac{t - t_0}{n\Delta t})$$

$$t \in \{t_0, t_0 + \Delta t, \ldots, t_0 + n\Delta t\}$$

В этой статье

- s_i = 0
- Δt - время на одну эпоху

Архитектуры и датасеты

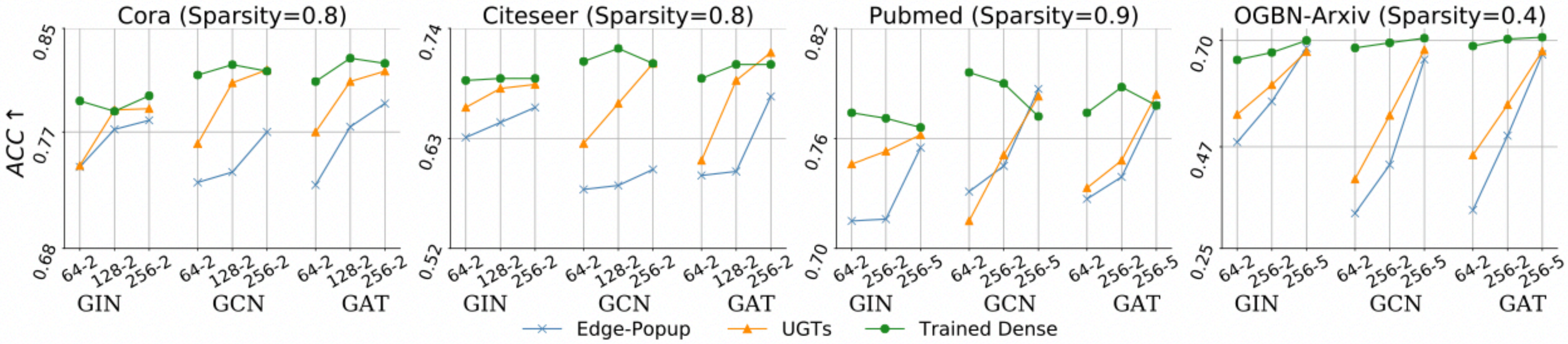
- Архитектуры: GCN, GIN, GAT

Table 2: Graph datasets statistics.

DataSets	#Graphs	#Nodes	#Edges	#Classes	#Features	Metric
Cora	1	2708	5429	7	1433	Accuracy
Citeseer	1	3327	4732	6	3703	Accuracy
Pubmed	1	19717	44338	3	3288	Accuracy
OGBN-Arxiv	1	169343	1166243	40	128	Accuracy
Texas	1	183	309	5	1703	Accuracy
OGBN-Products	1	24449029	61859140	47	100	Accuracy
OGBG-molhiv	41127	25.5(Average)	27.5(Average)	2	-	ROC-AUC
OGBG-molbace	1513	34.1(Average)	36.9(Average)	2	-	ROC-AUC

Результаты

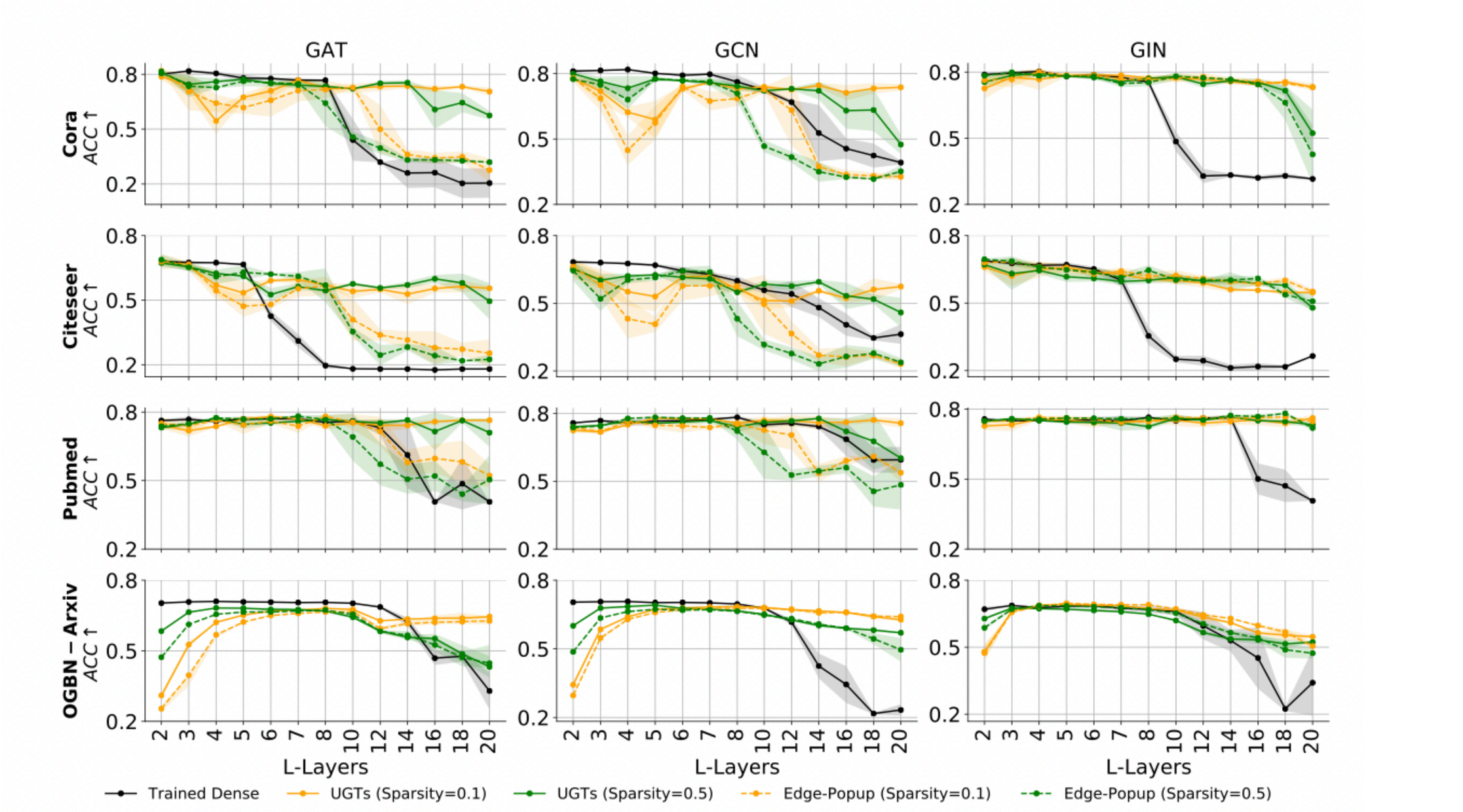
- С увеличением размера сети UGTs могут достичь сравнимой точности
- Edge-popup тоже стремится



Что такое over-smoothing и как помогут UGTs?

- Качество GNN не улучшается, когда мы увеличиваем количество слоев
- Так как в процессе обмена информацией между вершинами они становятся слишком похожи друг на друга

Что такое over-smoothing и как помогут UGTs?



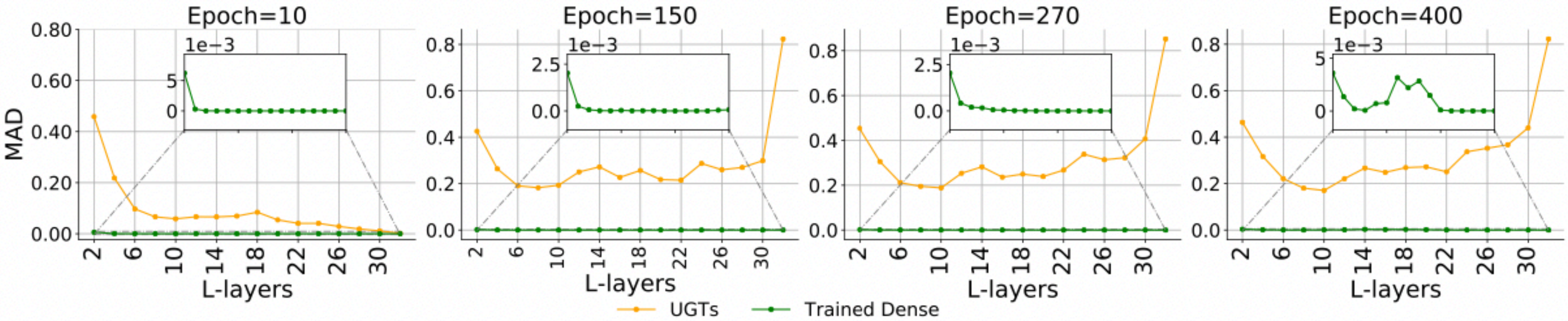
Сравнение с другими state-of-the-art техниками

- Что это за техники можно почитать тут

	Cora		Citeseer		Pubmed	
N-Layers	16	32	16	32	16	32
Trained Dense GCN	21.4	21.2	19.5	20.2	39.1	38.7
+Residual	20.1	19.6	20.8	20.90	38.8	38.7
+Jumping	76.0	75.5	58.3	55.0	75.6	75.3
+NodeNorm	21.5	21.4	18.8	19.1	18.9	18
+PairNorm	55.7	17.7	27.4	20.6	71.3	61.5
+DropNode	27.6	27.6	21.8	22.1	40.3	40.3
+DropEdge	28.0	27.8	22.9	22.9	40.6	40.5
UGTs-GCN	**77.3 ± 0.9**	**77.5 ± 0.8**	**61.1±0.9**	**56.2±0.4**	**77.6±0.9**	**76.3±1.2**

MAD - сравнение гладкости

- UGTs сильно менее гладкие в процессе обучения, что и хорошо



TSNE - визуализации

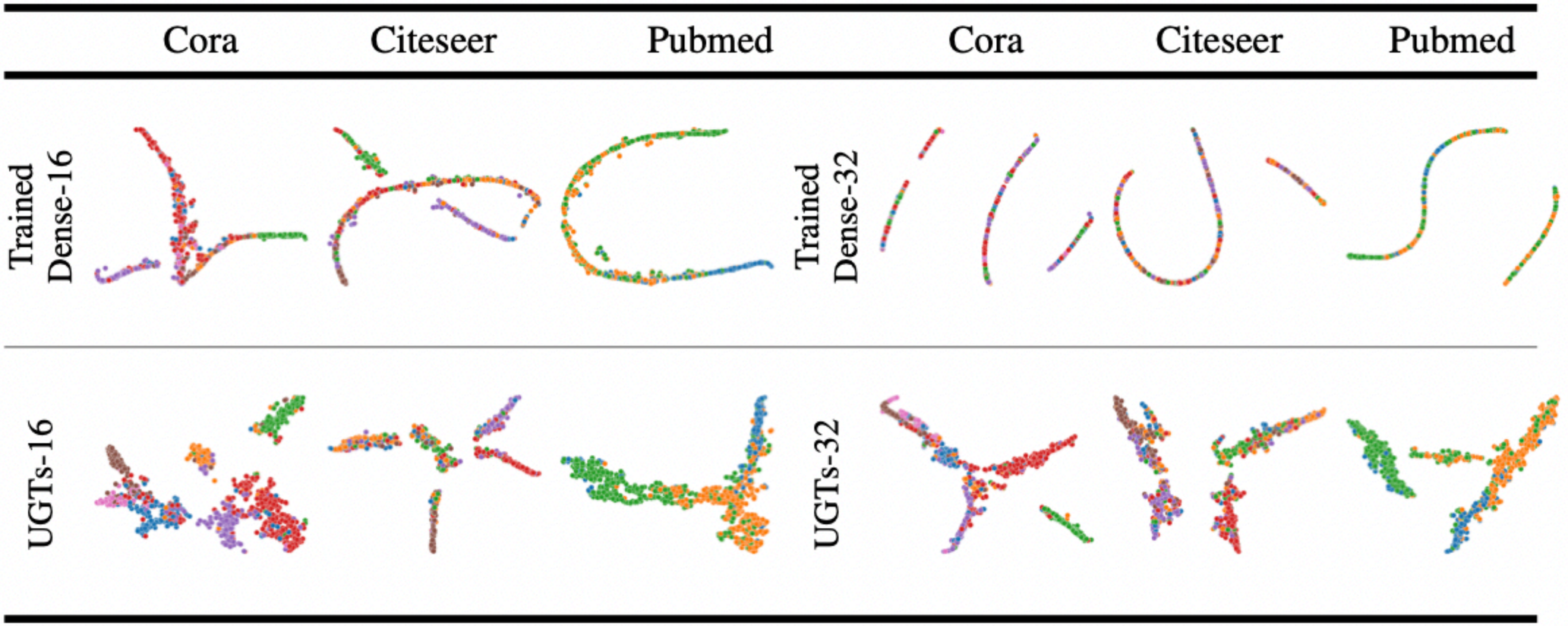


Figure 3: TSNE visualization of node representations learned by densely trained GCN and UGTs. Ten classes are randomly sampled from OGBN-Arxiv for visualization. Model depth is set as 16 and 32 respectively; width is set as 448. See Appendix B.1 for GAT architecture.

Другие эксперименты

- Более подробно исследовали эффект sparsity для UGTs
    - UGTs постоянно находит нетренированную подсеть для большого диапазона sparsity
    - Постоянно побеждает edge-popup
- UGTs хорошо работает на детекции аномалий
- Посмотрели на задачи графового уровня и другие датасеты

Выводы

- Необученная хорошо работающая подсеть существует
- Работает хорошо на разной sparsity
- Метод работает лучше, чем Edge-popup почти всегда
- Побеждает проблему чрезмерного сглаживания (over-smoothing)
- Получает сравнимые (а часто сильно лучшие) результаты, чем уже исследованные техники для борьбы со сглаживанием
- Также применимо к другим задачам


ВСЁ!