A SUPPLEMENTARY MATERIAL

A.1 PROOF OF THEOREM 3.1

Proof. We give the proof in two steps.

Step 1: For a fixed W, Solve optimal Z^* as a function of W: When fixing W as constant, the problem becomes quadratic and convex. There is a unique solution, given by first-order optimal condition. Let ℓ denote the objective function as given in (5). Its gradient can be calculated as

$$\frac{\partial \ell}{\partial Z} = 2(I + \alpha\tilde{L})Z - 2XW \ . \tag{14}$$

Setting (14) to 0 leads to the solution $Z^* = (I + \alpha\tilde{L})^{-1}XW$.

Step 2: Replace Z with Z^*, Solve optimal W^*: Substituting Z in objective ℓ with $Z^* = (I + \alpha\tilde{L})^{-1}XW$, we reduce the optimization to

$$\min_{W, W^TW=I} \ \|X - (I+\alpha\tilde{L})^{-1}XWW^T\|_F^2 \ + \ \alpha \operatorname{Tr}\left[W^TX^T(I+\alpha\tilde{L})^{-1}\tilde{L}(I+\alpha\tilde{L})^{-1}XW\right] . \tag{15}$$

For this part only, let $M = (I + \alpha\tilde{L})^{-1}$ to simplify notation. We can show that (15) is equivalent to

$$\begin{aligned} \min_{W, W^TW=I} \quad & \operatorname{Tr}(XX^T + MXWW^TWW^TX^TM) \\ & - 2\operatorname{Tr}(MXWW^TX^T) + \alpha \operatorname{Tr}(W^TX^TM\tilde{L}MXW) \end{aligned} \tag{16}$$

Using the cyclic property of (Tr)ace (and plugging $(I + \alpha\tilde{L})^{-1}$ for M back), we can write it as (see Supp. A.2 for detailed derivation.)

$$\max_{W, W^TW=I} \quad \operatorname{Tr}\left[W^TX^T(I + \alpha\tilde{L})^{-1}XW\right] . \tag{17}$$

Based on the spectral theorem of PSD matrices, the optimal solution W^* of problem (17) is the combination of eigenvectors, associated with the largest c eigenvalues of the graph-revised covariance matrix $X^T(I + \alpha\tilde{L})^{-1}X$. □

A.2 DERIVATION FROM EQ. (16) TO EQ. (17)

For this part only, let $A = (I + \alpha\tilde{L})^{-1}$ to simplify the notation. We can show that (15) is equivalent to

$$\begin{aligned} & \min_{W, W^TW=I} \quad \operatorname{Tr}(XX^T) - 2\operatorname{Tr}(AXWW^TX^T) \\ & + \operatorname{Tr}(AXWW^TWW^TX^TA) + \alpha \operatorname{Tr}(W^TX^TA\tilde{L}AXW) \\ \equiv & \max_{W, W^TW=I} \quad 2\operatorname{Tr}(AXWW^TX^T) - \operatorname{Tr}(AXWW^TX^TA) \\ & \qquad - \alpha \operatorname{Tr}(W^TX^TA\tilde{L}AXW) \end{aligned} \tag{18}$$

Using the cyclic property of (Tr)ace, we can write

$$\begin{aligned} & \max_{W, W^TW=I} \quad 2\operatorname{Tr}(W^TX^TAXW) - \operatorname{Tr}(W^TX^TAAXW) \\ & \qquad - \alpha \operatorname{Tr}(W^TX^TA\tilde{L}AXW) \\ & \max_{W, W^TW=I} \quad \operatorname{Tr}\left[W^TX^T(2A - AA - A(\alpha\tilde{L})A)XW\right] \\ & \max_{W, W^TW=I} \quad \operatorname{Tr}\left[W^TX^T\big(A + \{I - A(I + \alpha\tilde{L})\}A\big)XW\right] \\ & \max_{W, W^TW=I} \quad \operatorname{Tr}\left[W^TX^T(I + \alpha\tilde{L})^{-1}XW\right] \end{aligned} \tag{19}$$

where the objective simplifies upon replacing A with $(I + \alpha\tilde{L})^{-1}$.

A.3 Derivation of Equivalence in Eq. (9)

$$
\begin{aligned}
&\max_{\mathbf{z}} && \big[\mathrm{corr}(Y,\mathbf{z})\big]^T\big[\mathrm{corr}(Y,\mathbf{z})\big]\mathrm{var}(\mathbf{z}) \\
\equiv &\max_{\mathbf{z}} && \mathrm{var}(Y)\big[\mathrm{corr}(Y,\mathbf{z})\big]^T\big[\mathrm{corr}(Y,\mathbf{z})\big]\mathrm{var}(\mathbf{z}) && (20)\\
\equiv &\max_{\mathbf{z}} && \big[\mathrm{cov}(Y,\mathbf{z})\big]^T\big[\mathrm{cov}(Y,\mathbf{z})\big] && (21)\\
& && \text{where } \mathrm{cov}(Y,\mathbf{z}) = \sqrt{\mathrm{var}(Y)}\mathrm{corr}(Y,\mathbf{z})\sqrt{\mathrm{var}(\mathbf{z})} \\
\equiv &\max_{\mathbf{z}} && \big[Y^T\mathbf{z}\big]^T\big[Y^T\mathbf{z}\big] && (22)\\
\equiv &\max_{\mathbf{z}} && \mathbf{z}^TYY^T\mathbf{z} && (23)
\end{aligned}
$$

Note that in (20) we added the term $\mathrm{var}(Y)$ without affecting the optimization problem as it is with respect to $\mathbf{z}$.

A.4 Dataset Statistics

Table 4: Statistics of used datasets.

Dataset	#Nodes	#Edges	#Features	#Classes	Train/Val./Test
Cora	2,708	5,429	1,433	7	5.2%/18.5%/36.9%
CiteSeer	3,327	4,732	3,703	6	3.6%/15%/30%
PubMed	19,717	44,338	500	3	0.3%/2.5%/5%
Arxiv	169,343	1,166,243	128	40	54%/18%/28%
Products	2,449,029	61,859,140	100	47	8%/2%/90%

Datasets used in the experiments are presented in Table 4. Cora, CiteSeer, and PubMed can be downloaded in Pytorch Geometric Library Fey & Lenssen (2019). Arxiv and Products can be accessed in `https://ogb.stanford.edu/`.

A.5 Hyperparameter Configurations

We setup hyperparameters pool for each dataset, presented in Table 5. All methods use the *same* pool. The only exception is GPCA, as GPCA is just a 1-layer shallow model which can be trained with lager learning rate; we use 0.1 learning rate for it on all datasets.

Table 5: Hyperparameters pool for each dataset, includes learning rate (LR), weight decay (WD), number of layers (#Layers), hidden size, dropout, α, and β. For Arxiv and Products, weight decay is set as 0 because the dataset is large and no overfit happened. Same reason for choosing smaller dropout rate for them.

Dataset	LR	WD	#Layers	Hidden
Cora	0.001	[0.0005, 0.005, 0.05]	[2, 3, 5, 10, 15]	[128, 256]
CiteSeer	0.001	[0.0005, 0.005, 0.05]	[2, 3, 5, 10, 15]	[128, 256]
PubMed	0.001	[0.0005, 0.005, 0.05]	[2, 3, 5, 10, 15]	[128, 256]
Arxiv	0.005	0	[2, 3, 5, 10, 15]	[128, 256]
Products	0.001	0	[2, 3, 5, 10, 15]	[128, 256]

Dataset	Dropout	α	β
Cora	[0, 0.5]	[1, 5, 10, 20, 50]	[0, 0.1, 0.2]
CiteSeer	[0, 0.5]	[1, 5, 10, 20, 50]	[0, 0.1, 0.2]
PubMed	[0, 0.5]	[1, 5, 10, 20, 50]	[0, 0.1, 0.2]
Arxiv	[0, 0.2]	[1, 5, 10, 20, 50]	0
Products	[0, 0.1]	[1, 5, 10, 20, 50]	0

Models are trained on every configuration across HP pools and picked based on validation performance. We use the Adam optimizer for all models. Learning rate is first manually tuned for each dataset to achieve stable training, and the same learning rate is fixed for all models—we empirically observed that learning rate is sensitive to datasets but insensitive to models. For GPCA and GPCANET, number of power iterations in Eq. (**??**) is always set to 5. All experiments use the maximum training epoch as 1000 and repeat 5 times. Detailed configuration of HPs can be found in Supp. A.5. We mainly use a single GTX-1080ti GPU for small datasets CORA, CITESEER, and PUBMED. RTX-3090 GPU is used for ARXIV and PRODUCTS.

Mini-batch training. As nodes are not independent, GNN is mostly trained in full-batch under semi-supervised setting. We use full-batch training for all datasets except PRODUCTS, which is too large to fit into GPU memory during training. ClusterGCN Chiang et al. (2019), a subgraph based mini-batch training algorithm, is used to train GCN and GPCANET. For evaluation, we still use full-batch since a single forward pass can be conducted without memory issues. Initialization is also employed in full-batch.

Fair evaluation. Instead of picking the hyperparameter configurations manually, reported (test) performance is based on the *best* configuration selected using validation performance, where all models leverage the *same* hyperparameter pools. Further, each configuration from the pool is conducted 5 times to reduce randomness.

A.6 GPCA WITH VARYING α

Table 6: Performance of unsupervised GPCA ($\beta = 0$) for varying α w.r.t. mean test accuracy and standard deviation (in parentheses). GPCA (best α) selects $\alpha \in \{1, 5, 10, 20, 50\}$ based on validation, whereas GPCA with specific α uses the specified fixed α.

	CORA	CITESEER	PUBMED	ARXIV	PRODUCTS
GPCA (BEST α)	81.10 (0.00)	71.80 (0.75)	78.78 (0.36)	71.86 (0.18)	79.23 (0.14)
GPCA-α=1	72.57 (0.79)	70.90 (0.58)	76.92 (0.30)	65.47 (0.26)	73.65 (0.07)
GPCA-α=5	80.95 (0.17)	71.80 (0.75)	**79.40** (0.29)	70.69 (0.11)	78.66 (0.09)
GPCA-α=10	**82.23** (0.58)	71.65 (0.53)	78.78 (0.36)	71.37 (0.09)	**79.24** (0.09)
GPCA-α=20	82.05 (0.54)	**72.15** (0.47)	78.15 (0.50)	**71.86** (0.18)	79.23 (0.14)
GPCA-α=50	81.10 (0.00)	71.50 (0.32)	78.00 (0.19)	71.48 (0.15	78.92 (0.10)

A.7 CONFIGURATIONS FOR EXPERIMENTS OF 1∼3-LAYER GPCANET

The goal is train a shallow GPCANET with tunable α (β=0 is used), we setup different α pool for different number of layers, because the effect of increasing α is the same to increasing number of layers (shown in Figure 1). We report the pool for α for each layer in Table 7. For other parameters we use the same setting mentioned in Table 5.

Table 7: Pool of α for 1∼3-layer GPCANET, same across all datasets.

# LAYERS	POOL OF α
1-LAYER	[10, 20, 30]
2-LAYER	[3, 5, 10]
3-LAYER	[1, 2, 3, 5]

A.8 GPCANET-INIT'S ROBUSTNESS FOR ADDITIONAL DATASETS

Histogram of test set accuracy over 100 runs for GCN initialized by Xavier-initialization and GPCANET-initialization in CORA (Figure 3), CITESEER (Figure 4), and PUBMED (Figure 5). We have ignored PRODUCTS as it takes too long to run 100 times, but the result should be similar.

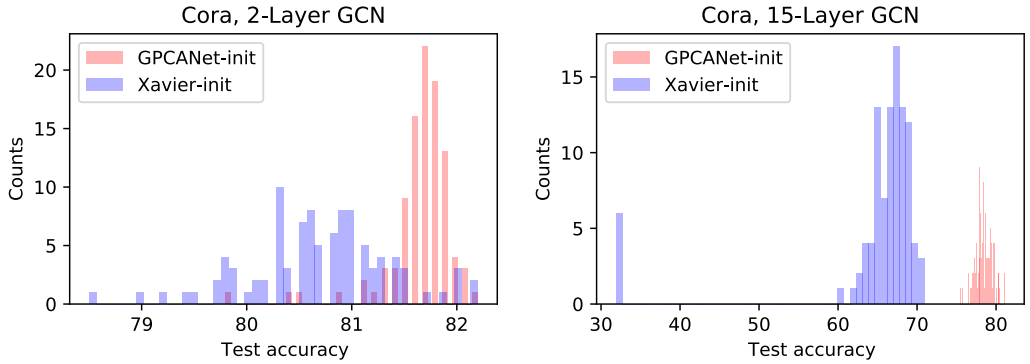

Figure 3: Comparison between Xavier-init and GPCANET-init in terms of test accuracy robustness over 100 seeds on CORA.

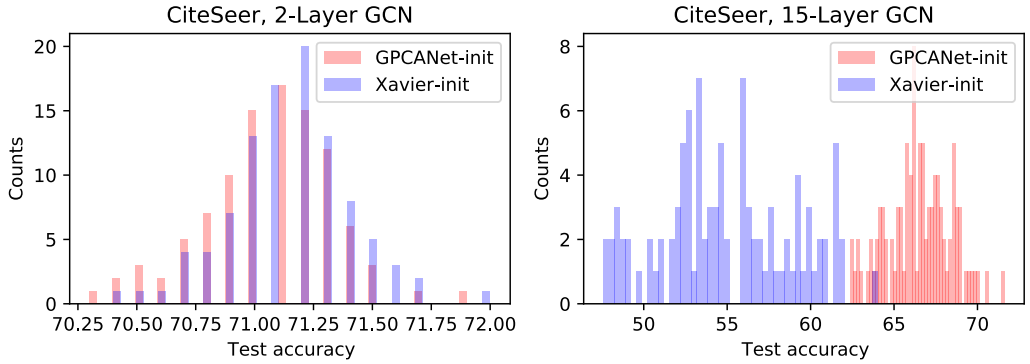

Figure 4: Comparison between Xavier-init and GPCANET-init in terms of test accuracy robustness over 100 seeds on CITESEER.

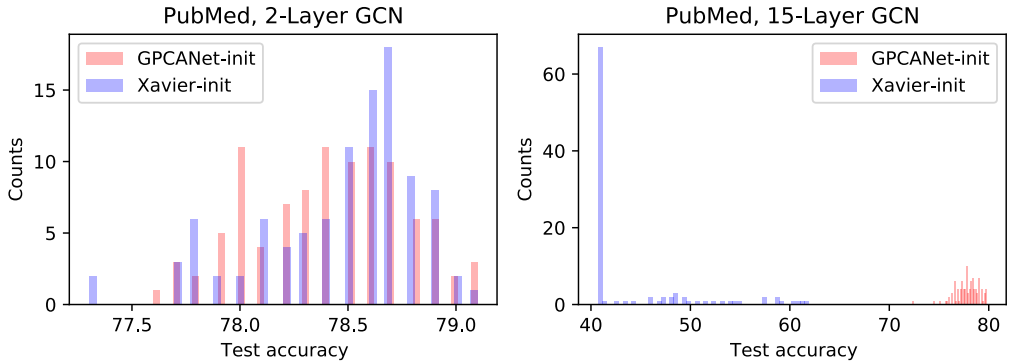

Figure 5: Comparison between Xavier-init and GPCANET-init in terms of test accuracy robustness over 100 seeds on PUBMED.