A UNIFIED VIEW ON THE REPRESENTATIONAL POWER OF GRAPH CONVOLUTION, MESSAGE-PASSING NNS AND GRAPH TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

TODO
Guaranteed representational power + incorporate proper graph inductive bias.

Motivation: Xxx

Outline:

- A new framework
  - Unifying existings with this new framework;
- Representational power analysis
  - The upper bound of $\mathcal{L}$;
  - The Representational Power of Polynomial Filters;
  - The Representational Power of Truncate Filters;
  - The complementary between polynomial filter and truncate filter
    - The trade-off between representational power and complexity.
- Designing powerful and computational efficient $\mathbf{M}$;

Misc:

Predict (row-wise transformation) then propagate (column-wise transformation) paradigm

A more well-motivated explanation of required filters based on labels over exisitng low/high pass filter arguements. Lemma shows that $\mathcal{L}$ relates to graph structure, node features and labels.

1 INTRODUCTION

2 PRELIMINARIES

3 GRAPH LEARNING MODULE UNIFICATION

Given a graph G and node features $X \in \mathbb{R}^{n\times d}$, the operations in various graph learning modules can be unified into row-wise transformation f_W^{row} and column-wise transformation $f_{\mathbf{M}}^{\text{col}}$ on X,

$$Z = f_{\mathbf{M}}^{\text{col}} \circ f_W^{\text{row}}(X). \tag{1}$$

Here, regarding to specific graph learning modesl, $f_W^{\text{row}}(\mathcal{O}) = \sigma(\mathcal{O}W)$ can be a single- or multi-layer percetron applied on each row of X by right multiplying W, where $W \in \mathbb{R}^{d\times d'}$ is the learnable weights. $f_{\mathbf{M}}^{\text{col}}(\mathcal{O}) = \sigma(\mathbf{M}\mathcal{O})$ is applied on each column of X by left multiplying $\mathbf{M}$, where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is the matrix representation of G. $\mathbf{M}$ can also be learnable. Most importantly, $\mathbf{M}$ incorporates *graph inductive biases* to leverage topology information in learning node representations. For example, in GCN, $\mathbf{M} = \hat{A}$. In GAT, to utilize topology information, only entries corresponding to edges are learnable. The most fundamental difference of various graph learning modules is their

ways to incorporates graph inductive biases into $\mathbf{M}$. In some literatures, f_W^{row} and $f_{\mathbf{M}}^{\text{col}}$ refer to *predict* and *propagate* operations respectively (Klicpera et al., 2019), and different $\mathbf{M}$ result in different propagation scheme. [1]

3.1 Strategies of incorporating graph inductive biases into $\mathbf{M}$

$\mathbf{M}$ is learnable. Encoding topology information into $\mathbf{M}$ with various graph inductive biases.

Graph Convolution.

$$\mathbf{M} = g_{\boldsymbol{\alpha}}^{(k)}(\hat{L}) = U\text{diag}\left(g_{\boldsymbol{\alpha}}^{(k)}(\boldsymbol{\lambda})\right)U^\top = U\sum_{i=0}^{k}\boldsymbol{\alpha}_i\boldsymbol{\lambda}^i U^\top, \tag{2}$$

where $g_{\boldsymbol{\alpha}}^{(k)}$ denotes a k-degree polynomial with the polynomial coefficient $\boldsymbol{\alpha} \in \mathbb{R}^k$.

Message-passing NNs. MPNNs involve fixed $\mathbf{M}$, such as different aggregators SUM, MEAN, MAX/MIN, or learnable $\mathbf{M}$ such as GAT.

$$\mathbf{M} = \hat{A} \tag{3}$$

Neighbor-sampling, DropEdge.

Graph Transformers. Graph transformers incorporate graph inductive biases into $\mathbf{M}$ by proposing a series of positional encoding techniques.

$$\mathbf{M} = QK^\top + \text{Bias} \tag{4}$$

There is a lack of a unified understanding of the effectiveness of various graph learning modules. In this work, we fill this part by analyzing their representational power.

4 Representational Power Analysis

Following the setting in Xu et al. (2021) and Wang & Zhang (2022), we study the representational power, i.e. fitting ability, of the linear case of Equation 1 trained with the squared loss. For the standard semi-supervised node classification, it corresponds to

$$\begin{aligned}\mathcal{L} &= \|\mathbf{T}(\mathbf{Y} - \mathbf{MH})\|_F \\ &= \|\mathbf{TY} - \mathbf{TMH}\|_F \\ &= \|\mathbf{Y}^{\text{train}} - \mathbf{TMH}\|_F,\end{aligned} \tag{5}$$

where $\mathbf{Y} \in \{0,1\}^{n\times c}$ is one-hot encoding node labels with the number of classes c, $\mathbf{H} = f_W(\mathbf{X})$ and $\mathbf{T} \in \{0,1\}^{m\times n}$ is the mask of training set with the size m. [2] We use $\mathbf{Y}^*$ to denote valid or test set labels, then $\mathbf{Y} = [\mathbf{Y}^{\text{train}} \| \mathbf{Y}^*]$.

Lemma 1. *Given $\mathcal{L} = \|\mathbf{T}(\mathbf{Y} - \mathbf{MH})\|_F$, suppose $\mathbf{H}$ has no missing frequency components over all channels and frequency profiles, i.e. $\boldsymbol{u}_i^\top \boldsymbol{h}_j \neq 0$, and $\|\boldsymbol{h}_j\|_2 \leq \eta$ for all $i \in [n], j \in [c]$, then:*

$$\mathcal{L} \leq \sqrt{m}n\eta \sum_{i=1}^{c}\left\|\text{vec}_j\left(\frac{\boldsymbol{u}_j^\top[\boldsymbol{y}_i^{train}\|\boldsymbol{y}_i^*]}{\boldsymbol{u}_j^\top \boldsymbol{h}_i}\right) - \boldsymbol{t}\right\|_2, \tag{6}$$

where $\mathbf{M} = \mathbf{U}\text{diag}(\boldsymbol{t})\mathbf{U}^\top$.

[1] In the multi-layer settings, they apply multi-layer f_W^{row} and $f_{\mathbf{M}}^{\text{col}}$ in an alternate manner, i.e. $Z^{(k)} = f_{\mathbf{M}}^{\text{col}(k)}(f_W^{\text{row}(k)}(Z^{(k-1)}))$. There are also GNNs applying decouple architectures where node features are prcessed by the multi-layer f_W^{row} and $f_{\mathbf{M}}^{\text{col}}$ individually (Klicpera et al., 2019; Wu et al., 2019; Klicpera et al., 2019; Liu et al., 2020; Zhu & Koniusz, 2020; Zhang et al., 2021). σ in $f_{\mathbf{M}}^{\text{col}}$ and f_W^{row} share the same role that introduces nonlinearity to the transformation. Some works study linear GNNs which correspond to removing σ in Equ. 1 (Wu et al., 2019; Xu et al., 2021; Wang & Zhang, 2022; Liu et al., 2021; 2022).

[2] Note that although $\mathbf{Y}$ in Equation 5 involves valid or test set labels, it does not lead to label leakage as these labels are masked by $\mathbf{T}$ and do not affect $\mathcal{L}$.

Due to semi-supervised settings, the derived upper bound of $\mathcal{L}$ in Equation 6 indicates a connection with valid or test set labels $\mathbf{Y}^*$, which makes it inconsistent with a practical scenario. However, it provides valuable insights by showing that no matter the assignments of $\mathbf{Y}^*$, an expressive filter $\boldsymbol{t}$ can always achieve a lower upper bound than an inexpressive filter by approximating $\mathrm{vec}_j\left(\frac{\boldsymbol{u}_j^\top[\boldsymbol{y}_i^{\text{train}}\|\boldsymbol{y}_i^*]}{\boldsymbol{u}_j^\top \boldsymbol{h}_i}\right)$ with smaller errors.

We prove Lemma 1 in Appendix A. As $H = f_W(X)$ and $\mathbf{M} = U\mathrm{diag}(\boldsymbol{t})U^\top$, Lemma 1 indicates the upper bound of $\mathcal{L}$ is controlled by both $f_{\mathbf{M}}^{\text{col}}$ and f_W^{row}. This is consistent with our empirical understanding that the predictions of nodes should consider both the node features and topology information.

f_W^{row} is generally implemented as single- or multi-layer percetron whose representational power is guaranteed by universal approximation theorem (Hornik et al., 1989; Cybenko, 1989). In contrast, the design of the transformation $f_{\mathbf{M}}^{\text{col}}$ is more challenging since apart from maintaining the representational power, it also needs to effectively encode topology information into the learned node representations, which is also mentioned as graph inductive biases (Ma et al., 2023). This highlights the difficulty of graph learning. Next, we study the effects of $f_{\mathbf{M}}^{\text{col}}$ to the fitting ability.

4.1 The Representational Power of Polynomial Filters

Filter expressiveness in graph convolution has drawn extensive research interests. A common understanding is that a more expressive filter can better filtering signal patterns over different frequency profiles thus achieves better performance (Chien et al., 2021; He et al., 2021; Wang & Zhang, 2022; Yang et al., 2022; Bo et al., 2022). Here, we study how filter expressiveness relates to the representational power of models.

As shown in Equ. 6, $\boldsymbol{t}$ is shared over all c channels, for a given channel $\boldsymbol{y}^* \in \mathbb{R}^n$ and the corresponding $\boldsymbol{h} \in \mathbb{R}^n$, we have

$$\begin{aligned}\mathcal{L} &= \|T\left(\boldsymbol{y}^* - \mathbf{M}\boldsymbol{h}\right)\|_2 \\ &\leq \sqrt{mn}\eta \left\| \left(U^\top \boldsymbol{h}\right)^{-1} \odot \left(U^\top \boldsymbol{y}^*\right) - \boldsymbol{t} \right\|_2 .\end{aligned} \tag{7}$$

In graph convolution, $\boldsymbol{t}$ is interpreted as a filter and is generally approximated by polynomials. Although there are extensive studies on approximation abilities, there is a lack of the understanding of the approximation objective. Equ. 7 fills this gap from the perspective of the representational power. Specifically, we use $\boldsymbol{t}$ to approximate $(U^\top \boldsymbol{h})^{-1} \odot (U^\top \boldsymbol{y}^*)$, and $\|(U^\top \boldsymbol{h})^{-1} \odot (U^\top \boldsymbol{y}^*) - \boldsymbol{t}\|_2$ is the approximation error. A smaller $\mathcal{L}$ requires a smaller approximaton error. In the context of filter studies, the desired filter relates to graph topology U, input signals $\boldsymbol{h}$ and the label distribution $\boldsymbol{y}^*$. U and $\boldsymbol{y}^*$ together reflect the homophily of a graph. So the desired filter also relates to the graph homophily, which is consistent with our intuition.

Let $\epsilon = \|(U^\top \boldsymbol{h})^{-1} \odot (U^\top \boldsymbol{y}^*) - \boldsymbol{t}\|_2$ denote the approximation error of polynomials. Then according to Equ. 7, we have

$$\mathcal{L} \leq \sqrt{mn}\eta\epsilon. \tag{8}$$

A high-degree polynomial achieves smaller approximation error, i.e. $\epsilon_{g^{(k+1)}} \leq \epsilon_{g^{(k)}}$, thus a high-degree polynomial corresponds to higher representational power. The perfect fitting requires a degree-$(n-1)$ polynomial which is also known as the universal filter approximator (He et al., 2021; Yang et al., 2022; Wang & Zhang, 2022; Bo et al., 2022). Now, we align filter expressiveness with the representaitonal power. Although theoretically we can increase the degree of polynomials to consistently improve the expressiveness, unfortunately, due to the limited computational precision and numerical instability, the high-degree polynomial are intractable in implementations (Yang et al., 2022).

4.2 The Representational Power of Truncate Filter

Due to the limited expressiveness of polynomials, some recent work choose to learn the filter with more expressive MLP or Transformer (Lingam et al., 2022; Yang et al., 2022; Bo et al., 2022). Correspondingly, following the formulation of Equ. 1, $\mathbf{M} = U\mathrm{diag}(\mathrm{MLP}(\boldsymbol{\lambda}))U^\top$ or

$\mathbf{M} = U\text{diag}(\text{Transformer}(\boldsymbol{\lambda}))U^\top$ respectively. The limitation is that it requires eigendecomposition computation to obtain U and $\boldsymbol{\lambda}$ first, which is intractable on large-scale graphs. Luckily, the results show that utilizing partial spectrum information of the graph matrix can still obtain competitive prediction performance. Thus they apply truncate eigendecompostion which achieves a good balance between prediction performance and computation efficiency on large graphs. We call such partial eigendecomposition-based methods truncate filter.

We use $\tilde{\boldsymbol{\lambda}}^{(k)} \in \mathbb{R}^k$ and $\tilde{U}^{(k)} \in \mathbb{R}^{n\times k}$ to denote the top k spectrum information obtained by truncate eigendecomposition, and $\tilde{U}^{(n-k)} \in \mathbb{R}^{n\times(n-k)}$ to denote the remaining $n-k$ eigenvectors.

Proposition 1. *When* $\mathbf{M} = \tilde{U}^{(k)}\text{diag}(\text{MLP}(\tilde{\boldsymbol{\lambda}}^{(k)}))\tilde{U}^{(k)\top}$*, we have*

$$\mathcal{L} \leq \sqrt{mn}\eta\left(\epsilon + \left\|\left(\tilde{U}^{(n-k)\top}\boldsymbol{h}\right)^{-1} \odot \left(\tilde{U}^{(n-k)\top}\boldsymbol{y}^*\right)\right\|_2\right), \tag{9}$$

where $\epsilon > 0$ *denotes the approximation error of MLP.*

We prove Proposition 1 in Appendix B. Proposition 1 shows that a larger k achieves better fitting ability. When $k = n$, $\mathcal{L} \leq \sqrt{mn}\eta\epsilon$, where ϵ relates to the expressiveness of MLP.

4.3 Polynomial Filter VS Truncate Filter

On the one hand, polynomial filters have drawn extensive research interests, and many spectral GNNs have been propposed based on different polynomials. On the other hand, truncate filters show its competitive performance in recent work. However, there is a lack of a detailed understanding of the difference between them. The representational power analysis provides a way to fill this gap. Specifically, we can compare the power of two filters by comparing which one achieves smaller loss bound. We use $\mathcal{L}_{\text{poly.}}$ to denote the loss of polynomial filter as shown in Equ. 8, and $\mathcal{L}_{\text{trunc.}}$ to denote the loss of polynomial filter as shown in Equ. 9. Both $\mathcal{L}_{\text{poly.}}$ and $\mathcal{L}_{\text{trunc.}}$ are related to the approximation abilities of approximators and the availability of frequency profiles (FPs) as shown in Fig. 1.

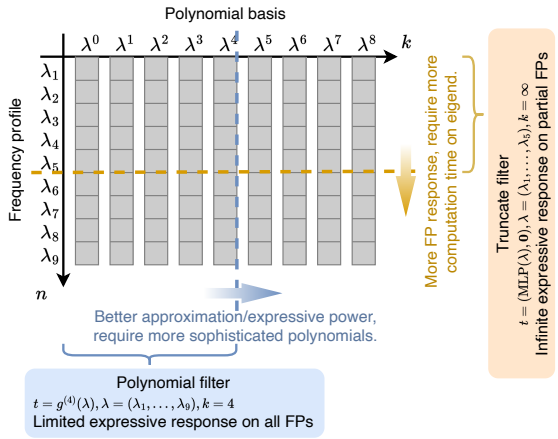

Figure 1: The comparison between polynomial filter and truncate filter.

Polynomial filters are implemented as the polynomial of the entire spectrum, thus they can operate all frequency profiles. While the limitation is that the polynomials as approximators whose approximation abilities are bounded by the degree of polynomials. Generally, the applicable degree of polynomials is far from the desired infinite approximation power, i.e. $k = n$. Therefore, although polynomial filters have all profiles response, the approximation error can be large, and this leads to a larger bound of $\mathcal{L}_{\text{poly.}}$. In contrast, truncate filters conduct truncate eigendecomposition and can only

operate partial spectrum, thus they can only operate partial frequency profiles. For the missing frequency profiles, it corresponds to $\|(\tilde{U}^{(n-k)\top}\boldsymbol{h})^{-1} \odot (\tilde{U}^{(n-k)\top}\boldsymbol{y}^*)\|_2$ as in Equ. 9. The positive side is that for the available frequency profiles, truncate frilter can achieve infinite approximation power (which correponds to $k = n$) with powerful approximators like MLP or Transformer. Therefore, truncate filters are complementary to polynomial filters that well handle the approximation error but only on the partial profiles. If the label information has no frequency components in the missing profiles, i.e. $(\tilde{U}^{(n-k)\top}\boldsymbol{y}^*) = \mathbf{0}$, truncate filters can achieve much smaller loss bound.

5 Designing Powerful and Computational Efficient M

Following the principle of graph convolution, the most powerful model requires the filter approximator to have infinite approximation power over all frequency profiles. This objctive is much challenging as discussed in Sec. 4.3. Also, we have shown that various efforts in improving graph convolutions can be unified into imrpoving the representational power. Naturally, this motivates us to alter the objective to imrpoving the representational power. Let $\bar{\mathcal{L}} = \sqrt{m}\|\boldsymbol{y}^* - \mathbf{M}\boldsymbol{h}\|_2$ denote the upper bound of $\mathcal{L}$.

Proposition 2. *Given a group of $k+1$ linearly independent vectors, let $U^{(k)} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k) \in \mathbb{R}^{n\times k}$, $U^{(k+1)} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k+1}) \in \mathbb{R}^{n\times(k+1)}$ and $\mathbf{M}^{(k)} = U^{(k)}\mathrm{diag}(\boldsymbol{\alpha})U^{(k)\top}$. Then we have*

$$\min_{\boldsymbol{\alpha}} \bar{\mathcal{L}}_{\mathbf{M}^{(k+1)}} \leq \min_{\boldsymbol{\alpha}} \bar{\mathcal{L}}_{\mathbf{M}^{(k)}}. \tag{10}$$

When $k = n$, we have $\min_{\boldsymbol{\alpha}} \bar{\mathcal{L}}_{\mathbf{M}^{(k)}} = 0$.

We prove Proposition 2 in Appendix C. In graph convlution, the columns of U is considered as the bases of graph Fourier transform and are orthogonal to each other. However, from the perspective of improving the representational power, the orthogonality is unnecessary, and one can apply a larger k to consistently improve the representational power as long as all $\boldsymbol{u}_i$ are linearly independent. Benefit from relaxing the orthogonality requirement, we can make the highest representational power implementable.

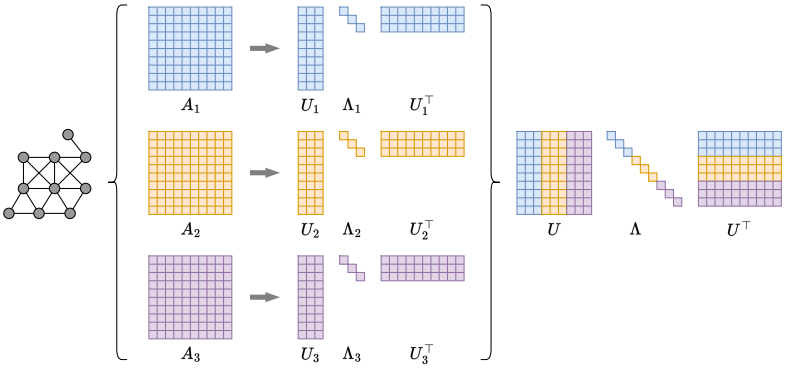

Figure 2: Xxx.

6 Experiments

7 Conclusion

References

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=n6jl7fLxrP.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *NeurIPS*, 2021.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations (ICLR)*, 2019.

Vijay Lingam, Manan Sharma, Chanakya Ekbote, Rahul Ragesh, Arun Iyer, and Sundararajan Sellamanickam. A piece-wise polynomial filtering approach for graph neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 412–452. Springer, 2022.

Juncheng Liu, Kenji Kawaguchi, Bryan Hooi, Yiwei Wang, and Xiaokui Xiao. Eignn: Efficient infinite-depth graph neural networks. *Advances in Neural Information Processing Systems*, 34: 18762–18773, 2021.

Juncheng Liu, Bryan Hooi, Kenji Kawaguchi, and Xiaokui Xiao. Mgnni: Multiscale graph neural networks with implicit layers. In *Advances in Neural Information Processing Systems*, 2022.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 338–348, 2020.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. *arXiv preprint arXiv:2305.17589*, 2023.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. *ICML*, 2022.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6861–6871. PMLR, 2019.

Keyulu Xu, Mozhi Zhang, Stefanie Jegelka, and Kenji Kawaguchi. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *International Conference on Machine Learning*, pp. 11592–11602. PMLR, 2021.

Mingqi Yang, Yanming Shen, Rui Li, Heng Qi, Qiang Zhang, and Baocai Yin. A new perspective on the effects of spectrum in graph neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Shanzhuo Zhang, Lihang Liu, Sheng Gao, Donglong He, Xiaomin Fang, Weibin Li, Zhengjie Huang, Weiyue Su, and Wenjin Wang. Litegem: Lite geometry enhanced molecular representation learning for quantum property prediction, 2021.

Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations*, 2020.

A PROOF OF LEMMA 1

Proof.

$$
\begin{aligned}
\mathcal{L} &= \|\mathbf{T}(\mathbf{Y}-\mathbf{M}\mathbf{H})\|_F \\
&\leq \|\mathbf{T}\|_F \|\mathbf{Y}-\mathbf{M}\mathbf{H}\|_F \\
&= \sqrt{m}\left\|\mathbf{U}\mathbf{U}^\top\mathbf{Y}-\mathbf{U}\mathrm{diag}(\boldsymbol{t})\mathbf{U}^\top\mathbf{H}\right\|_F \\
&\leq \sqrt{m}\|\mathbf{U}\|_F\left\|\mathbf{U}^\top\mathbf{Y}-\mathrm{diag}(\boldsymbol{t})\mathbf{U}^\top\mathbf{H}\right\|_F \\
&= \sqrt{m}\sqrt{n}\sqrt{\sum_{i=1}^{c}\|\mathbf{U}^\top\boldsymbol{y}_i-\mathrm{diag}(\boldsymbol{t})\mathbf{U}^\top\boldsymbol{h}_i\|_2^2} \\
&\leq \sqrt{m}\sqrt{n}\sum_{i=1}^{c}\left\|\mathbf{U}^\top\boldsymbol{y}_i-\mathrm{diag}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\boldsymbol{t}\right\|_2 \\
&= \sqrt{m}\sqrt{n}\sum_{i=1}^{c}\left\|\mathrm{diag}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\left(\mathrm{diag}^{-1}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\mathbf{U}^\top\boldsymbol{y}_i-\boldsymbol{t}\right)\right\|_2 \\
&\leq \sqrt{m}\sqrt{n}\sum_{i=1}^{c}\left\|\mathbf{U}^\top\boldsymbol{h}_i\right\|_2\left\|\mathrm{diag}^{-1}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\mathbf{U}^\top\boldsymbol{y}_i-\boldsymbol{t}\right\|_2 \\
&\leq \sqrt{m}\sqrt{n}\sum_{i=1}^{c}\left\|\mathbf{U}^\top\right\|_F\|\boldsymbol{h}_i\|_2\left\|\mathrm{diag}^{-1}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\mathbf{U}^\top\boldsymbol{y}_i-\boldsymbol{t}\right\|_2 \\
&= \sqrt{m}n\sum_{i=1}^{c}\|\boldsymbol{h}_i\|_2\left\|\mathrm{diag}^{-1}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\mathbf{U}^\top\boldsymbol{y}_i-\boldsymbol{t}\right\|_2 \\
&\leq \sqrt{m}n\eta\sum_{i=1}^{c}\left\|\mathrm{diag}^{-1}\left(\mathbf{U}^\top\boldsymbol{h}_i\right)\mathbf{U}^\top\boldsymbol{y}_i-\boldsymbol{t}\right\|_2 \\
&= \sqrt{m}n\eta\sum_{i=1}^{c}\left\|\mathrm{vec}_j\left(\frac{\boldsymbol{u}_j^\top\boldsymbol{y}_i}{\boldsymbol{u}_j^\top\boldsymbol{h}_i}\right)-\boldsymbol{t}\right\|_2 \\
&= \sqrt{m}n\eta\sum_{i=1}^{c}\left\|\mathrm{vec}_j\left(\frac{\boldsymbol{u}_j^\top[\boldsymbol{y}_i^{\text{train}}\|\boldsymbol{y}_i^*]}{\boldsymbol{u}_j^\top\boldsymbol{h}_i}\right)-\boldsymbol{t}\right\|_2
\end{aligned}
\tag{11}
$$

□

B PROOF OF PROPOSITION 1

Assume the k-truncate eigendecomposition is applied. Correspondingly,

$$
\begin{aligned}
\mathbf{M} &= \tilde{U}^{(k)}\mathrm{diag}\left(\mathrm{MLP}\left(\tilde{\boldsymbol{\lambda}}^{(k)}\right)\right)\tilde{U}^{(k)\top}+\tilde{U}^{(n-k)}\mathrm{diag}(\mathbf{0})\tilde{U}^{(n-k)\top} \\
&= U\mathrm{diag}\left(\left(\mathrm{MLP}\left(\tilde{\boldsymbol{\lambda}}^{(k)}\right),\mathbf{0}\right)\right)U^\top \\
&= U\mathrm{diag}(\boldsymbol{t})U^\top,
\end{aligned}
\tag{12}
$$

where $\boldsymbol{t} = (\mathrm{MLP}(\tilde{\boldsymbol{\lambda}}), \mathbf{0}) \in \mathbb{R}^n$. Let $\epsilon > 0$ denotes the approximation error of MLP. Then

$$\begin{aligned}
\mathcal{L} &\leq \sqrt{m}n\eta \left\| \left(U^\top \boldsymbol{h}\right)^{-1} \odot \left(U^\top \boldsymbol{y}^*\right) - \boldsymbol{t} \right\|_2 \\
&= \sqrt{m}n\eta \left\| \left(U^\top \boldsymbol{h}\right)^{-1} \odot \left(U^\top \boldsymbol{y}^*\right) - \left(\mathrm{MLP}\left(\tilde{\boldsymbol{\lambda}}\right), \mathbf{0}\right) \right\|_2 \\
&= \sqrt{m}n\eta \sqrt{\left\| \left(\tilde{U}^\top \boldsymbol{h}\right)^{-1} \odot \left(\tilde{U}^\top \boldsymbol{y}^*\right) - \mathrm{MLP}\left(\tilde{\boldsymbol{\lambda}}\right) \right\|_2^2 + \left\| \left(\tilde{U}'^\top \boldsymbol{h}\right)^{-1} \odot \left(\tilde{U}'^\top \boldsymbol{y}^*\right) - \mathbf{0} \right\|_2^2} \\
&\leq \sqrt{m}n\eta \sqrt{\epsilon^2 + \left\| \left(\tilde{U}'^\top \boldsymbol{h}\right)^{-1} \odot \left(\tilde{U}'^\top \boldsymbol{y}^*\right) \right\|_2^2} \\
&\leq \sqrt{m}n\eta \left(\epsilon + \left\| \left(\tilde{U}'^\top \boldsymbol{h}\right)^{-1} \odot \left(\tilde{U}'^\top \boldsymbol{y}^*\right) \right\|_2 \right).
\end{aligned} \tag{13}$$

C Proof of Proposition 2

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \bar{\mathcal{L}}_{\mathbf{M}^{(k+1)}} &= \min_{\boldsymbol{\alpha}} \sqrt{m} \left\| \boldsymbol{y}^* - \mathbf{M}^{(k+1)} \boldsymbol{h} \right\|_2 \\
&= \min_{\alpha_0, \ldots \alpha_{k+1}} \sqrt{m} \left\| \boldsymbol{y}^* - \sum_{i=1}^{k+1} \boldsymbol{\alpha}_i U_i U_i^\top \boldsymbol{h} \right\|_2 \\
&\leq \min_{\alpha_0, \ldots \alpha_k | \alpha_{k+1}=0} \sqrt{m} \left\| \boldsymbol{y}^* - \sum_{i=1}^{k+1} \boldsymbol{\alpha}_i U_i U_i^\top \boldsymbol{h} \right\|_2 \\
&= \min_{\alpha_0, \ldots \alpha_k} \sqrt{m} \left\| \boldsymbol{y}^* - \sum_{i=1}^{k} \boldsymbol{\alpha}_i U_i U_i^\top \boldsymbol{h} \right\|_2 \\
&= \min_{\boldsymbol{\alpha}} \sqrt{m} \left\| \boldsymbol{y}^* - \mathbf{M}^{(k)} \boldsymbol{h} \right\|_2 \\
&= \min_{\boldsymbol{\alpha}} \bar{\mathcal{L}}_{\mathbf{M}^{(k)}}
\end{aligned} \tag{14}$$

When $k = n$, $\mathbf{M}^{(n)} \boldsymbol{h} = U \mathrm{diag}(\boldsymbol{\alpha}) U^\top \boldsymbol{h} = U(\boldsymbol{\alpha} \odot U^\top \boldsymbol{h})$. Let $\boldsymbol{x} = \boldsymbol{\alpha} \odot U^\top \boldsymbol{h}$. Consider the linear system $U\boldsymbol{x} = \boldsymbol{y}^*$. As $\mathrm{rank}(U) = n$, a solution $\boldsymbol{x}_0$ always exists. Suppose $U_i^\top \boldsymbol{h} \neq 0$ for all $i \in [n]$. Correspondingly, $\boldsymbol{\alpha}_0 = \boldsymbol{x}_0 \odot (U^\top \boldsymbol{h})^{-1}$. Therefore, $\min_{\boldsymbol{\alpha}} \bar{\mathcal{L}}_{\mathbf{M}^{(n)}} = \bar{\mathcal{L}}_{\mathbf{M}^{(n)}}\big|_{\boldsymbol{\alpha}_0} = 0$.

D Proof of Proposition 2

For any $\boldsymbol{\alpha} \in \mathbb{R}^{k+1}$, we have

$$\begin{aligned}
\bar{\mathcal{L}}_{\mathbf{M}^{(k+1)}} &= \sup \left\| \boldsymbol{y}^* - \mathbf{M}^{(k+1)} \boldsymbol{h} \right\|_2 \\
&= \left\| \boldsymbol{y}^* - \sum_{i=1}^{k+1} \boldsymbol{\alpha}_i U_i U_i^\top \boldsymbol{h} \right\|_2 \\
&\leq \left\| \boldsymbol{y}^* - \sum_{i=1}^{k+1} \boldsymbol{\alpha}_i U_i U_i^\top \boldsymbol{h} \right\|_2 \\
&= \left\| \boldsymbol{y}^* - \sum_{i=1}^{k} \boldsymbol{\alpha}_i U_i U_i^\top \boldsymbol{h} \right\|_2 \\
&= \left\| \boldsymbol{y}^* - \mathbf{M}^{(k)} \boldsymbol{h} \right\|_2 \\
&= \bar{\mathcal{L}}_{\mathbf{M}^{(k)}}
\end{aligned} \tag{15}$$

E MISC

H is channel-independent. $\boldsymbol{t}$ is shared over different channels. There are also channel-independent filter design where each signal channel learns an individual filter (Yang et al., 2022; Wang & Zhang, 2022; Bo et al., 2022). Correspondingly, the loss is $\mathcal{L} \leq \sqrt{m}n\eta \sum_{i=1}^{c} \left\| \left(U^\top H_i\right)^{-1} \odot \left(U^\top Y_i^*\right) - \mathbf{T}_i \right\|_F$ with $\mathbf{T} \in \mathbb{R}^{n \times c}$.

E.1 EXPLORING GRAPH REPRESENTATION SPACE

According to spectral theorem $\mathbf{M} = U\mathrm{diag}(\boldsymbol{t})U^\top$, a graph matrix representation $\mathbf{M}$ provides a unique tuple $(U, \boldsymbol{t})$. $(U, \boldsymbol{t})$ decides the fitting ability. Improving fitting ability while maintaining graph topology information.

$$
\begin{aligned}
\mathcal{L} &= \|T\left(Y^* - \mathbf{M}H\right)\|_F \\
&\leq \|T\|_F \|Y^* - \mathbf{M}H\|_F \\
&= \|T\|_F \left\| Y^* - \sum_{i=1}^{n} \boldsymbol{\lambda}_i U_i U_i^\top H \right\|_F \\
&= \|T\|_F \left\| \sum_{i=1}^{n} U_i U_i^\top Y^* - \sum_{i=1}^{n} \boldsymbol{\lambda}_i U_i U_i^\top H \right\|_F \\
&= \|T\|_F \left\| \sum_{i=1}^{n} U_i U_i^\top \left(Y^* - \boldsymbol{\lambda}_i H\right) \right\|_F \\
&\leq \|T\|_F \sum_{i=1}^{n} \left\| U_i U_i^\top \left(Y^* - \boldsymbol{\lambda}_i H\right) \right\|_F \\
&= \|T\|_F \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{c} \left\| U_i U_i^\top \left(Y_j^* - \boldsymbol{\lambda}_i H_j\right) \right\|_F^2} \\
&= \|T\|_F \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{c} \left(U_i^\top \left(Y_j^* - \boldsymbol{\lambda}_i H_j\right)\right)^2 \|U_i\|_F^2} \\
&= \|T\|_F \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{c} \left(U_i^\top \left(Y_j^* - \boldsymbol{\lambda}_i H_j\right)\right)^2}
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
\mathcal{L} &= \|T\left(Y^* - \mathbf{M}H\right)\|_F \\
&\leq \|T\|_F \|Y^* - \mathbf{M}H\|_F \\
&= \sqrt{m}\left\|UU^\top Y^* - U\mathrm{diag}\left(\boldsymbol{t}\right)U^\top H\right\|_F \\
&\leq \sqrt{m}\|U\|_F \left\|U^\top Y^* - \mathrm{diag}\left(\boldsymbol{t}\right)U^\top H\right\|_F \\
&= \sqrt{m}\sqrt{n}\sqrt{\sum_{j=1}^{c}\left\|U^\top Y_j^* - \mathrm{diag}\left(\boldsymbol{t}\right)U^\top H_j\right\|_F^2} \\
&= \sqrt{m}\sqrt{n}\sqrt{\sum_{j=1}^{c}\left\|U^\top Y_j^* - \mathrm{diag}\left(U^\top H_j\right)\boldsymbol{\lambda}\right\|_F^2} \\
&= \sqrt{m}\sqrt{n}\sqrt{\sum_{j=1}^{c}\left\|\mathrm{diag}\left(U^\top H_j\right)\left(\mathrm{diag}^{-1}\left(U^\top H_j\right)U^\top Y_j^* - \boldsymbol{\lambda}\right)\right\|_F^2} \\
&\leq \sqrt{m}\sqrt{n}\sqrt{\sum_{j=1}^{c}\|U^\top H_j\|_F^2\left\|\left(U^\top H_j\right)^{-1}\odot\left(U^\top Y_j^*\right) - \boldsymbol{\lambda}\right\|_F^2} \\
&\leq \sqrt{m}\sqrt{n}\sqrt{\sum_{j=1}^{c}\|U^\top\|_F^2\|H_j\|_F^2\left\|\left(U^\top H_j\right)^{-1}\odot\left(U^\top Y_j^*\right) - \boldsymbol{\lambda}\right\|_F^2} \\
&= \sqrt{m}n\sqrt{\sum_{j=1}^{c}\|H_j\|_F^2\left\|\left(U^\top H_j\right)^{-1}\odot\left(U^\top Y_j^*\right) - \boldsymbol{\lambda}\right\|_F^2} \\
&\leq \sqrt{m}n\sum_{j=1}^{c}\|H_j\|_F\left\|\left(U^\top H_j\right)^{-1}\odot\left(U^\top Y_j^*\right) - \boldsymbol{\lambda}\right\|_F
\end{aligned}
\tag{17}
$$