We would like to thank the reviewers for providing detailed and constructive reviews. Please find below our responses.

(1). **Why are the rational polynomial filters proposed in our work preferable for convolution operations? Why can our proposed work resolve the issue "narrow frequency bands" and improve the localization?**

The rational polynomial filters proposed in our work have two key components: (i) *feed-forward filtering* which performs the k-hop localization as polynomial filters; (ii) *feedback filtering* which filters out errors/noises in the output of feed-forward filtering to improve the localization. The feedback filtering component can only be supported by our proposed filters, not any other existing spectral filters. Moreover, unlike other rational polynomial filters, our filters have guaranteed stability, i.e., rational polynomial coefficients are learned in a stable way as discussed in Section 3.4, because the pole of our filters always lies in the unit circle of the z-plane.

“Narrow frequency bands" is an issue existing in the current spectral filters, including both polynomial and rational polynomial filters (e.g. Chebyshev and Calyley), because they only accept a small band of frequencies of the Laplacian. However, our proposed filters resolve this issue using a cut-off frequency technique, i.e., accepts frequencies higher than a certain low cut-off frequency value and attenuates frequencies lower than that cut-off value. Thus, our proposed filtrs can accept a wider range of frequencies of the Laplacian and capture better characteristic properties of a graph.

(2). **Why is Eq. (7) a valid approximation for Eq. (6)?**
An ARMA filter of Eq. (6) can filter a graph signal x by altering its frequency response which has the form as presented in Eq. (8). Then, according to Proposition 1 in [16], we also know that a feedback-looped filter using the approximation in Eq. (7) has the same frequency response as described in Eq. (8) under a stability condition $||\alpha\psi||_\infty \leq \gamma$ and $\gamma < 1$. This stability condition is required in Eq. (11) and discussed in Section 3.4 in detail.

(3). **Why is a regularization term μ introduced for the spectral convolution layer in Eq. (12)?**
The reason for introducing a regularization term μ is to alleviate the overfitting issue. Specifically, we use the unit-norm constraint technique to restrict parameters of all layers in a small range and the kernel regularization technique to penalize the parameters in each convolution layer during the training. In doing so, we can prevent the generation of spurious features and thus improve accuracy of the prediction. This is our contribution on designing spectral convolution layers suitable for the proposed filters. The theoretical discussion in Section 3.4 is still applicable in this case.

(4). **In the following, we clarify the questions relating to experiments. All the related results and the detailed analysis will be provided in our final version and the supplementary materials of the paper.**

In Section 4.2, we have compared our proposed filters with self-attention, (i.e. DF-ATT) against GAT which uses Chebyshev filters and self-attention. The results in Table 4 show that DF-ATT outperforms GAT over all four datasets. Additionally, we have conducted experiments on comparing DFNet (our proposed filters+DenseBlock) with GCN+DenseBlock and GAT+DenseBlock, as well as comparing our proposed filters with Chebyshev, GCN and Cayley. The results below show that our proposed filters perform best, no matter whether the dense architecture is used.

Model	Cora	Citeseer	Pubmed	NELL
GCN+DenseBlock	82.7 ±0.5	71.3 ±0.3	81.5 ±0.5	66.4 ±0.3
GAT+Dense Block	83.8 ±0.3	73.1 ±0.3	81.8 ±0.3	-
DFNet (ours)	**85.2 ±0.5**	**74.2 ±0.3**	**84.3 ±0.4**	**68.3 ±0.4**

Model	Cora	Citeseer	Pubmed
Chebyshev	81.2	69.8	74.4
GCN	81.5	70.3	79.0
Cayley	81.9	-	-
Feedback-looped (ours)	**82.6 ±0.3**	**71.5 ±0.4**	**81.7 ±0.6**

For our models, hyperparameters were initially selected using the orthogonalization technique (a randomized search strategy). Then, we used the validation dataset to select the best model. Thus, the best (p,q) for the validation dataset is the same as the (p,q) used for the test dataset.

We have conducted experiments on our proposed filters with and without adding the μ term in Eq. (12). The table below shows that, adding the μ term in Eq. (12) improves the performance on all four datasets.

Model	Cora	Citeseer	Pubmed	NELL
Without adding the μ term in Eq. (12)	84.2 ±0.3	73.1 ±0.4	83.1 ±0.3	67.4 ±0.4
With adding the μ term in Eq. (12)	**85.2 ±0.5**	**74.2 ±0.3**	**84.3 ±0.4**	**68.3 ±0.4**

We have benchmarked the performance of our DFNet model against the models in [23]. All experiments were repeated 10 times and the same hyperparameter settings in Section 4.2 were used for DFNet. The table below shows that DFNet performs significantly better than all the models over the dataset Cora, including AdaLNet proposed in [23].

Training Split	Chebyshev	GCN	GAT	LNet	AdaLNet	DFNet
5.2% (standard split used in previous works [7, 19, 31])	78.0 ±1.2	80.5 ±0.8	82.6 ±0.7	79.5 ±1.8	80.4 ±1.1	**85.2 ±0.5**
3% (random split as in [23])	62.1 ±6.7	74.0 ±2.8	56.8 ±7.9	76.3 ±2.3	77.7 ±2.4	**80.5 ±0.4**
1% (random split as in [23]	44.2 ±5.6	61.0 ±7.2	48.6 ±8.0	66.1 ±8.2	67.5 ±8.7	**69.5 ±2.3**
0.5% (random split as in [23]	33.9 ±5.0	52.9 ±7.4	41.4 ±6.9	58.1 ±8.2	60.8 ±9.0	**61.3 ±4.3**