A. Proofs

Proposition 1. *Polynomial filters $g(\lambda) = \sum_{k=0}^{K} \theta_k \lambda^k$ are linearly stable with respect to any GSO where the spectrum lies in $[-1, 1]$. The stability constant is given by $C = \sum_{k=1}^{K} k|\theta_k|$.*

Proof. The proof technique is the same as in Kenlay et al. (2020a). Note that $\|\mathbf{L}\|_2 \leq 1$ and $\|\mathbf{L}_p\|_2 \leq 1$ so by Levie et al. (2019, Lemma 3) we know that $\|\mathbf{L}^k - \mathbf{L}_p^k\| \leq k\|\mathbf{E}\|_2$. Using this and the the triangle inequality we have:

$$\|g(\mathbf{L}) - g(\mathbf{L}_p)\|_2 = \|\sum_{k=1}^{K} \theta_k(\mathbf{L}^k - \mathbf{L}_p^k)\|_2 \leq \sum_{k=1}^{K} |\theta_k| \|\mathbf{L}^k - \mathbf{L}_p^k\|_2 \leq \sum_{k=1}^{K} k|\theta_k| \|\mathbf{E}\|_2. \qquad \square$$

Proposition 2. *The low-pass filter $g(\lambda) = (1 + \alpha\lambda)^{-1}$ is linearly stable with respect to the normalised Laplacian matrix. The constant is given by $C = \alpha$.*

Proof. Let $\mathbf{X} = \mathbf{I}_n + \alpha\mathbf{L}$ and $\mathbf{Y} = \mathbf{I}_n + \alpha\mathbf{L}_p$ where $\mathbf{L}, \mathbf{L}_p$ are normalised Laplacian matrices. Then we have:

$$\|f(\mathbf{L}) - f(\mathbf{L}_p)\|_2 = \|\mathbf{X}^{-1} - \mathbf{Y}^{-1}\|_2 = \|\mathbf{X}^{-1}(\mathbf{Y} - \mathbf{X})\mathbf{Y}^{-1}\|_2 \leq \|\mathbf{X}^{-1}\|_2 \|\mathbf{Y}^{-1}\|_2 \|\mathbf{X} - \mathbf{Y}\|_2. \tag{15}$$

Note that $\mathbf{X}$ has eigenvalues in the interval $[1, 1 + 2\alpha]$, so $\mathbf{X}^{-1}$ has eigenvalues in the interval $[(1 + 2\alpha)^{-1}, 1]$. This holds similarly for $\mathbf{Y}$. The matrices $\mathbf{X}^{-1}$ and $\mathbf{Y}^{-1}$ are symmetric and positive definite so their operator norm is the largest eigenvalue $\|\mathbf{X}^{-1}\|_2 = \|\mathbf{Y}^{-1}\|_2 = 1$. Thus,

$$\|f(\mathbf{L}) - f(\mathbf{L}_p)\|_2 \leq \|\mathbf{X}^{-1}\|_2 \|\mathbf{Y}^{-1}\|_2 \|\mathbf{X} - \mathbf{Y}\|_2 \leq \|\mathbf{X} - \mathbf{Y}\|_2 = \alpha\|\mathbf{L} - \mathbf{L}_p\|_2. \qquad \square$$

Lemma 1. *Let $\alpha_u \in [0, 1)$. Then the following holds:*

$$\sum_{v \in \mathcal{R}_u} |\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{d'_u d'_v}}| \leq \sum_{v \in \mathcal{R}_u} \left(\frac{\alpha_u}{1 - \alpha_u}\right) \frac{1}{\sqrt{d_u d_v}} \tag{16}$$

$$\leq \left(\frac{\alpha_u}{1 - \alpha_u}\right) \frac{d_u - \Delta_u^-}{\sqrt{d_u \delta_u}}. \tag{17}$$

Proof. The main part of proving this Lemma is proving the following identity:

$$|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{d'_u d'_v}}| \leq \left(\frac{\alpha_u}{1 - \alpha_u}\right) \frac{1}{\sqrt{d_u d_v}}. \tag{18}$$

To prove Eq. (18), we will maximise the left hand size bearing in mind that the constraint $\alpha_u \in [0, 1)$ limits the possible values d'_u and d'_v can take. To maximise the left hand size we will consider it as a function of Δ_u and Δ_v and use partial derivatives to reason where the maxima is.

The left hand side of Eq. (18) can be written as a function of the change in degree of node u and v

$$f(\Delta_u, \Delta_v) = |\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{(d_u + \Delta_u)(d_v + \Delta_v)}}| \tag{19}$$

on the domain $\Omega = [-\alpha_u d_u, \alpha_u d_u] \times [-\alpha_u d_v, \alpha_u d_v]$. Because the domain is such that $\Omega \subset (0, d_{\max}]^2$, where $d_{\max}$ is the largest degree of all nodes in graph $\mathcal{G}$, this function has no poles. The function f has zeros in its domain and the function is non-differentiable at these points but differentiable away from these points. Since the function is non-negative but not identically zero, the local maximas are strictly positive. Therefore we can consider critical points of the function away from the zeros. We consider the partial derivative of f with respect to Δ_u. The chain rule gives us:

$$\frac{\partial f(\Delta_u, \Delta_v)}{\partial \Delta_u} = \text{sign}(f(\Delta_u, \Delta_v)) \frac{\partial}{\partial \Delta_u} \left(\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{(d_u + \Delta_u)(d_v + \Delta_v)}}\right) \tag{20}$$

$$= \text{sign}(f(\Delta_u, \Delta_v)) \frac{\partial}{\partial \Delta_u} \left(\frac{-1}{\sqrt{(d_u + \Delta_u)(d_v + \Delta_v)}}\right). \tag{21}$$

Because we only consider the function f away from its zeros, $\text{sign}(f(\Delta_u, \Delta_v))$ will be non-zero. Therefore, the partial derivative of f with respect to Δ_u is zero if and only if the partial derivative on the right hand side of Eq. (21) is zero. By setting $z = (d_u + \Delta_u)(d_v + \Delta_v)$ we use the chain rule again to obtain:

$$\frac{\partial}{\partial \Delta_u} \frac{-1}{\sqrt{(d_u + \Delta_u)(d_v + \Delta_v)}} = \frac{\partial}{\partial z} \frac{-1}{\sqrt{z}} \frac{\partial z}{\partial \Delta_u} = \frac{1}{2z^{3/2}}(d_v + \Delta_v) = \frac{d_v + \Delta_v}{2\big((d_u + \Delta_u)(d_v + \Delta_v)\big)^{3/2}}. \tag{22}$$

The partial derivative for f with respect to Δ_u is zero if and only if the right hand side of Eq. (22) is zero. One can see this partial derivative is zero if and only if $d_v + \Delta_v = 0$, but our constraints are such that $d_v + \Delta_v \geq d_v - \alpha_u d_v > 0$. This holds by symmetry for the other partial derivative. Since there are no local maximas in the domain of the function and the function is continuous, the maxima must lie on the boundary of the domain which we write as $\delta\Omega$ (Spivak, 2018, Theorem 2.6).

The boundary $\delta\Omega$ describes a closed rectangle. One can parameterise the sides of this rectangle by either fixing $\Delta_u \in \{-\alpha d_u, \alpha d_u\}$ or by fixing $\Delta_v \in \{-\alpha d_v, \alpha d_v\}$. Without loss of generality consider fixing $\Delta_u = -\alpha d_u$ to give a 1D function in Δ_v describing a side of the boundary of this rectangle. The derivative of this 1D function is exactly the partial derivative of f with respect to Δ_v. We proved that both partial derivatives are non-zero in Ω, so in particular the derivative of this 1D function is non-zero and thus the maxima must lie on its boundary (Spivak, 2018, Theorem 2.6). The boundary of a side are the two corner end points in the rectangle adjacent to the side. By considering all four sides we can deduce that the maximum of the function must lie on one or more of the four corners. In the following we will prove that the function has a single maximum and show it takes this value at the corner $(-\alpha_u d_u, -\alpha_u d_v)$.

We first show that the function maps two of the corners to the same value:

$$f(-\alpha_u d_u, \alpha_u d_v) = \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{(d_u - \alpha_u d_u)(d_v + \alpha_u d_v)}}\right| \tag{23}$$

$$= \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{(1-\alpha_u)(1+\alpha_u)}\sqrt{d_u d_v}}\right| \tag{24}$$

$$= \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{(d_u + \alpha_u d_u)(d_v - \alpha_u d_v)}}\right| = f(\alpha_u d_u, -\alpha_u d_v). \tag{25}$$

Next, we show that $f(-\alpha_u d_u, -\alpha_u d_v) \geq f(-\alpha_u d_u, \alpha_u d_v)$. We do this by proving that $f(-\alpha_u d_u, -\alpha_u d_v) - f(-\alpha_u d_u, \alpha_u d_v)$ is non-negative. Writing this out we have that:

$$f(-\alpha_u d_u, -\alpha_u d_v) - f(-\alpha_u d_u, \alpha_u d_v) = \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{(1-\alpha_u)\sqrt{d_u d_v}}\right| - \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{\sqrt{(1-\alpha_u^2) d_u d_v}}\right|. \tag{26}$$

Since $(1 - \alpha_u)$ and $\sqrt{1 - \alpha_u^2}$ both lie in the interval $(0, 1]$, we know that both the values inside the absolute values are non-positive. By negating the values inside and removing the absolute value we get that this is equal to:

$$f(-\alpha_u d_u, -\alpha_u d_v) - f(-\alpha_u d_u, \alpha_u d_v) = \left(\frac{1}{(1-\alpha_u)\sqrt{d_u d_v}} - \frac{1}{\sqrt{d_u d_v}}\right) - \left(\frac{1}{\sqrt{(1-\alpha_u^2) d_u d_v}} - \frac{1}{\sqrt{d_u d_v}}\right) \tag{27}$$

$$= \frac{1}{(1-\alpha_u)\sqrt{d_u d_v}} - \frac{1}{\sqrt{(1-\alpha_u^2) d_u d_v}}. \tag{28}$$

This quantity is non-negative if and only if $(1 - \alpha_u) \leq \sqrt{(1 - \alpha_u^2)}$. By squaring both sides and rearranging we get that this is true if and only if $\alpha_u \in [0, 1]$ which holds true by our assumption.

Finally, we show that $f(-\alpha_u d_u, -\alpha_u d_v) \geq f(\alpha_u d_u, \alpha_u d_v)$. Similar to before we show the following is non-negative:

$$f(-\alpha_u d_u, -\alpha_u d_v) - f(\alpha_u d_u, \alpha_u d_v) = \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{(1-\alpha_u)\sqrt{d_u d_v}}\right| - \left|\frac{1}{\sqrt{d_u d_v}} - \frac{1}{(1+\alpha_u)\sqrt{d_u d_v}}\right| \tag{29}$$

$$= \left(\frac{1}{(1-\alpha_u)\sqrt{d_u d_v}} - \frac{1}{\sqrt{d_u d_v}}\right) - \left(\frac{1}{\sqrt{d_u d_v}} - \frac{1}{(1+\alpha_u)\sqrt{d_u d_v}}\right) \tag{30}$$

$$= \frac{1}{\sqrt{d_u d_v}}\left(\frac{1}{1-\alpha_u} + \frac{1}{1+\alpha_u} - 2\right) \tag{31}$$

$$= \frac{1}{\sqrt{d_u d_v}}\left(\frac{-2\alpha_u^2}{(\alpha_u - 1)(\alpha_u + 1)}\right). \tag{32}$$

The above is non-negative for $\alpha_u \in [0, 1)$. This proves that $f(-\alpha_u d_u, -\alpha_u d_v)$ is a maxima of the four corners hence a global maxima of the function f. We show that the inequality in Eq. (18) holds and is tight by showing equality holds when the left hand side takes its largest value among all possible values of d'_u and d'_v (equivalently Δ_u and Δ_v):

$$\max_{\substack{\Delta_u \in [-\alpha d_u, \alpha d_u] \\ \Delta_v \in [-\alpha d_v, \alpha d_v]}} f(\Delta_u, \Delta_v) = f(-\alpha_u d_u, -\alpha_u d_v) = \frac{1}{(1-\alpha_u)\sqrt{d_u d_v}} - \frac{1}{\sqrt{d_u d_v}} = \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{1}{\sqrt{d_u d_v}}. \tag{33}$$

The first inequality (Eq. (16)) in Lemma 1 follows immediately from this. The second inequality (Eq. (17)) comes from noting that $d_v \geq \delta_u \implies 1/\sqrt{d_u d_v} \leq 1/\sqrt{d_u \delta_u}$ and that $|\mathcal{R}_u| = d_u - \Delta_u^-$ giving:

$$\sum_{v \in \mathcal{R}_u} \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{1}{\sqrt{d_u d_v}} \leq \sum_{v \in \mathcal{R}_u} \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{1}{\sqrt{d_u \delta_u}} = \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{d_u - \Delta_u^-}{\sqrt{d_u \delta_u}}. \tag{34}$$

□

Theorem 1. *Let $\alpha_u \in [0, 1)$ for all nodes $u \in \mathcal{V}$. Then the following holds:*

$$\|\mathbf{E}\|_2 \leq \max_{u \in \mathcal{V}} \left\{ \frac{\Delta_u^-}{\sqrt{d_u \delta_u}} + \frac{\Delta_u^+}{\sqrt{d'_u \delta'_u}} + \left(\frac{\alpha_u}{1-\alpha_u}\right) \frac{d_u - \Delta_u^-}{\sqrt{d_u \delta_u}} \right\}.$$

Proof. If $\alpha_u \in [0, 1)$ for all nodes then Eq. (10) also holds for all nodes. Substituting Eq. (10) into Eq. (5) gives the desired result. □

B. Random graph models

In this work we consider graphs without isolated nodes for simplicity. Although the bounds do not require graphs to be connected, we always consider the unperturbed graphs to be connected. To achieve this in practice we use rejection sampling, i.e., sampling from the random graph models until we sample a connected graph. We describe the random graph models we use in our experiments in detail below. Where available we use implementations provided by the NetworkX and PyGSP libraries. Summary statistics describing the degree distribution as well as distance between nodes are given in Table S1.

Graph	Mean degree	Degree standard deviation	Average shortest path length	Diameter	Degree correlation
K-Reg	3.00	0.00	4.83	8.77	N/A
WS	4.00	0.62	5.07	10.58	-0.02
ENZYMES	4.02	1.01	5.35	13.60	0.02
PROTEINS_full	3.89	1.03	6.66	17.37	0.06
SBM	4.22	1.87	3.53	7.43	-0.02
Assortative	4.64	2.02	4.21	10.79	0.80
ER	4.66	2.04	3.15	6.41	-0.02
BA	5.82	4.72	2.59	4.43	-0.16

Table S1. Summary statistics averaged across the generated graphs. The first two columns give the average and standard deviation of the degree sequence. The average shortest path length and diameter (largest shortest path length) measure node connectivity. The degree correlation measures assortativity.

Erdős-Rényi Erdős-Rényi graphs are constructed by independently including each possible edge between any pair of nodes with probability p (Gilbert, 1959). For sufficiently large graphs the graph will be connected with high probability if $p > \ln n/n$ and disconnected if $p < \ln n/n$. We thus chose $p = \ln n/n$. The degree distribution of Erdős-Rényi graphs is approximately a Poisson distribution.

Barabási–Albert Barabási–Albert graphs are randomly generated scale-free (with power-law degree distributions) graphs which are constructed using a preferential attachment mechanism (Albert & Barabási, 2002). The graphs are constructed using parameters n, the number of nodes, and m, the number of edges that are preferentially attached. An initial graph is

given by a star graph on $m+1$ nodes. Then, until the graph has n nodes the following step is repeated. A new node is introduced and connected to m existing nodes with probabilities proportional to their degrees. Barabási–Albert graphs are connected by construction.

Watts-Strogatz Watts-Strogatz graphs with appropriate parameters exhibit small-world properties such as small average path lengths and high clustering (Watts & Strogatz, 1998). The graph begins as a ring lattice which is obtained by taking a cycle of nodes and connecting each node to its K nearest neighbours. Then, each edge is rewired independently with probability p, which means deleting edge $i \sim j$ and adding edge $i \sim j'$ such that $j' \notin \{i, j\}$ is selected uniformly at random. We connected each node to its $K = 4$ neighbours for the initial configuration and rewire each edge with probability $p = 0.1$.

K-regular A K-regular graph is one such that the degree of every node is K. We use the algorithm described in Steger & Wormald (1999) and implemented in NetworkX. For a fixed K, the algorithm samples graphs with n nodes all of degree K almost uniformly at random in the sense that the distribution approaches uniform in the limit $n \to \infty$.

Assortative An assortative graph is one such that there is a high positive correlation between the degree of end points along edges. To generate assortative graphs we use a variant of the Xulvi-Brunet & Sokolov (XBS) Algorithm (Xulvi-Brunet & Sokolov, 2004) applied to Erdős-Rényi graphs. The XBS algorithm iteratively rewires edges in the graph to increase assortativity whilst keeping the degree sequence fixed. At each step two edges, corresponding to four nodes, are chosen uniformly at random. With probability p, these edges are rewired in a way that increases the assortativity. Otherwise, the edges are rewired randomly. In our experiments we set this probability to $p = 1$, meaning the assortativity increases in each step. In the original algorithm a graph can become disconnected; we discard an iteration if it disconnects the graph. Instead of running the algorithm for a fixed number of steps we run the algorithm until the degree correlation is at least 0.8. We describe the algorithm in Algorithm 1.

Algorithm 1 XBS Algorithm

Input: An initial graph $\mathcal{G}$, assortative threshold a
repeat
  Initialize $\mathcal{G}_{temp} \leftarrow \mathcal{G}$.
  Sample edges $u \sim v$ and $u' \sim v'$ from $\mathcal{G}_{temp}$ such that u, v, u', v' are unique
  Delete edges $u \sim v$ and $u' \sim v'$ from $\mathcal{G}_{temp}$
  Connect the two nodes in $\mathcal{G}_{temp}$ from $\{u, v, u', v'\}$ with the highest degree
  Connect the two nodes in $\mathcal{G}_{temp}$ from $\{u, v, u', v'\}$ with the lowest degree
  if $\mathcal{G}_{temp}$ is connected **then**
    $\mathcal{G} \leftarrow \mathcal{G}_{temp}$
  end if
until Assortativity of $\mathcal{G}$ more than or equal to a
Output: Perturbed graph $\mathcal{G}$

C. Perturbation strategies

We make use of four random strategies (Add, Delete, Add/Delete and Rewire), one strategy which gives perturbations according to an optimisation based search strategy (PGD), and a strategy which is based on the theory described in Section 5 (Robust). All strategies delete or add a fixed number of edges. The rewiring operation is depicted graphically in Fig. S2. We describe Robust and PGD in more detail.

Robust The robust strategy works by iteratively building the perturbed graph. Consider two unique nodes u and v, then we say an edge is flipped between them if an edge $u \sim v$ exists and is deleted, or if the edge does not exist and is added. A single iteration consists of finding nodes u and v that have not been flipped in previous iterations such that $\|\mathbf{E}\|_1$ is minimal. The steps of the algorithm are described in Algorithm 2.

PGD Projected gradient descent is a variant of gradient descent where after each gradient update a projection operation is applied. This strategy aims to find adversarial examples and follows closely the strategy described in Xu et al. (2019), with three modifications. The first modification is that during the sampling procedure (Xu et al., 2019, Algorithm 1 (step 5)), we

(a) Before rewiring (b) After rewiring

Figure S1. In the rewiring operation the red edges are deleted and the blue edges are added. The degree of each node remains the same.

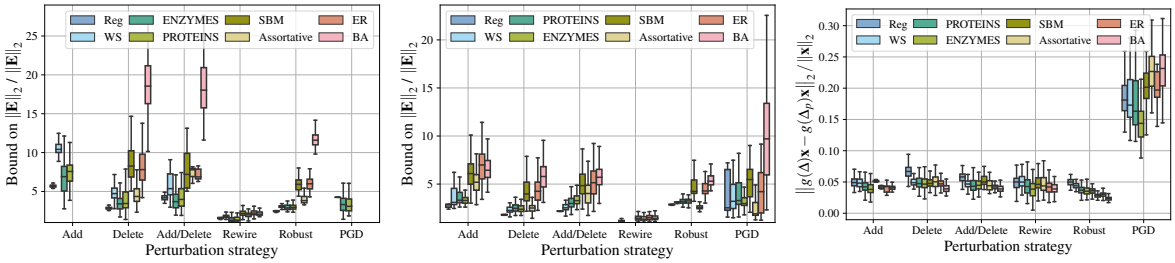
Algorithm 2 Robust Algorithm

Input: An initial graph $\mathcal{G}$, a budget B
Initialize set of flipped edges $\mathcal{F} \leftarrow \emptyset$.
while $|\mathcal{F}| < B$ **do**
  Initialize candidate edge e
  Initialize candidate edge norm $f \leftarrow \infty$
  for Potential edge $e' \in \{1, \dots, n\}^2 \setminus \mathcal{F}$ **do**
    Initialize $\mathcal{G}_{temp} \leftarrow \mathcal{G}$
    Flip edge e' in $\mathcal{G}_{temp}$
    Calculate normalised Laplacian matrix $\mathbf{L}_{temp}$ of $\mathcal{G}_{temp}$
    if $\|\mathbf{L}_{temp}\|_1 < f'$ **then**
      $e \leftarrow e'$
      $f \leftarrow \|\mathbf{L}_{temp}\|_1$
    end if
  end for
  Flip edge e in $\mathcal{G}$
  $F \leftarrow F \cup \{e\}$
end while
Output: Perturbed graph $\mathcal{G}$

not only discard perturbed graphs that are over budget, but also those containing isolated nodes.. The second modification is that we perform gradient ascent on the relative error between the original signal (before noise is added) and the denoised signal (estimate). This is different from the negative cross-entropy or Carlili-Wagner loss functions used in Xu et al. (2019) which are better suited for classification problems. Finally, our filter involves the inverse of a matrix which can become singular making it not possible to propagate gradients. We describe how we calculate the gradient used in Xu et al. (2019, Algorithm 2 (step 3)) in Algorithm 3. The variables $\mathbf{A}', \mathbf{s}$ and $\mathbf{S}$ are described further in Xu et al. (2019). We include additional steps 3 and 4 to improve the stability of the filtering operation (step 6). Step 4 projects negative values to 0 and values above 1 to 1. Step 6 makes use of the lower–upper (LU) decomposition for solving linear systems. We use a learning rate of $\eta_t = 200/\sqrt{t}$, and implement $T = 200$ iterations and $K = 250$ random trials.

Algorithm 3 Numerically stable gradient calculation

Input: Adjacency matrix $\mathbf{A}$, probability vector $\mathbf{s}$, target signal $\mathbf{y}$, noisy signal $\mathbf{x}$.
$\mathbf{A}' = \mathbf{A} + (\bar{\mathbf{A}} - \mathbf{A}) \circ \mathbf{S}$ (Xu et al., 2019, Eq. 4) where $\mathbf{S}$ is the matrix form of $\mathbf{s}$
$\mathbf{A}' = \mathbf{A}' + \mathrm{diag}(\mathcal{N}(\mathbf{0}_n, 10^{-3} \times \mathbf{I}_n))$
$\mathbf{A}' = \mathrm{clamp}(\mathbf{A}')$
Calculate Laplacian $\mathbf{L}'$ of $\mathbf{A}'$
Calculate denoised signal $\hat{\mathbf{y}} = (\mathbf{I}_n + \mathbf{L}')^{-1}\mathbf{x}$
Calculate relative error loss $\mathcal{L} = \|\mathbf{y} - \hat{\mathbf{y}}\|/\|\mathbf{y}\|$
Calculate gradient $\nabla_s \mathcal{L}$
Output: $\nabla_s \mathcal{L}$



(a) Larger graphs ($n = 500$) (b) Smaller perturbation (c) Lower SNR ($-10dB$)

Figure S2. Further results for varying experimental conditions

D. Additional results

D.1. Experimental setup

We experiment with changing the size of the random graphs, the size of the perturbation and the SNR ratio, whilst keeping other experimental variables fixed. We present the bound given by Theorem 1 for $n = 500$ in Fig. S2a. As expected, we find our bounds to be looser for large graphs. Furthermore, a larger proportion of experiments are invalid more often due a larger number of edge edits and more opportunities for the assumption of Theorem 1 to be violated. Nevertheless, our bounds are still valuable in practice as in the graph classification setting graphs with 100 or less nodes are common. For larger graphs, we approximate the Robust strategy with a sampling scheme as exhaustively searching for an edge at each step became computationally prohibitive. The sampling scheme alters step 6 in Algorithm 2 such that instead of considering every possible edge flip, we randomly sample 50 edges present in the graph, and 50 pairs of nodes which are not present as edges in the graph.

The bound in Theorem 1 tends to be tighter for a smaller perturbation level of $\lfloor 2\% \cdot |\mathcal{E}| \rfloor$ edge edits (Fig.S2b). [3]. Our approach to generating synthetic signals for the random graphs are entangled with the graph structure. This is reasonable since an implicit assumption behind using graph filters is that the signal and graph are related. Nevertheless, by varying SNR we can control the degree of such entanglement, and we show how the relative output distance changes for a lower SNR of -10dB (Fig. S2c, patterns are similar for a higher SNR of 10dB). We consider the relative output distance here rather than the bound in Theorem 1. This is because all perturbation strategies apart from PGD are invariant to the level of noise.

D.2. How close are the relative output distance to the filter distance?

In Section 4 we have established a relationship between the relative output distance in Eq. (2) and the filter distance in Eq. (3). The looseness of the bound given in Eq. (3) in shown in Fig. S3. As we can see the PGD strategy gives rise to a tighter inequality compares to other strategies.

D.3. How tight is the linear stability bound?

Linearly stable filters have the property that there exists an upper bound on the filter distance which is linearly proportional to the error norm. In the previous works of Levie et al. (2019) and Kenlay et al. (2020a) it has also been shown experimentally that the relationship is linear in practice. The low-pass filter we use in experiments has a stability constant of one meaning $\|f(\mathbf{L}) - f(\mathbf{L}_p)\|_2 \leq \|\mathbf{E}\|_2$. Our experiments (Fig. S4) show the looseness of this bound to be at least 1.5 in all our experiments.

D.4. Validity of experiments

As mentioned in Section 6, some experiments give $\alpha_u > 1$ for some node u in the graph meaning that strictly speaking the bound of Theorem 1 cannot be applied. Therefore, we discard the results of these experiments from our experimental analysis for figures relating to the bound. Explicitly we drop invalid experiments for the following figures: Fig. 1, Fig. 2, the bottom right pane of Fig. 3, Fig. S2b, Fig. S2c and the right pane of Fig. S7. Table S2 shows the proportion out of 100

[3] With the smaller perturbation level, we have that for some graphs in the real-world data sets $\lfloor 2\% \cdot |\mathcal{E}| \rfloor < 4$, meaning a rewire operations cannot take place within the budget of perturbation, hence the abscene of the bars in the plot.

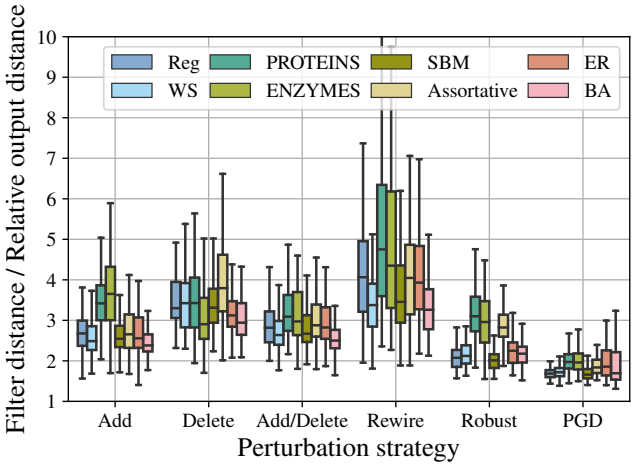


Figure S3. Looseness of the bound relating the relative output distance to the filter distance.

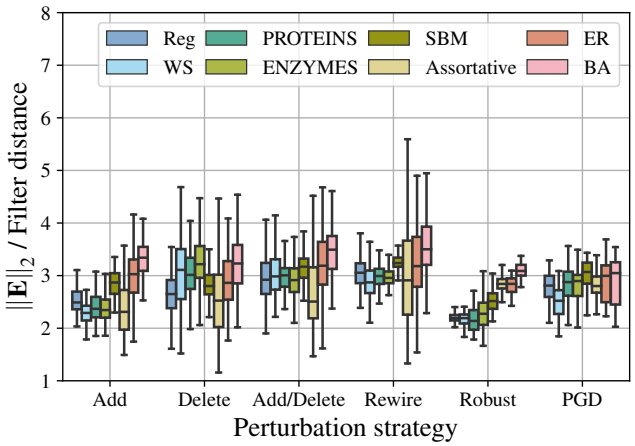


Figure S4. Looseness of the linear stability bound.

experiments we run which are valid (in the sense that the assumption on α_u is satisfied) for all combinations of random graph model and perturbation strategy. As noted in Section 5, for the assumption on α_u to be violated the degree of at least one node must at least double after perturbation. Since Delete only decreases the degree of nodes, and Rewire preserves the degree of all nodes, experiments with these perturbation strategies are always valid as expected. The graph models with high-variance degree distributions (ER, BA, Assortative) tend to have the assumption violated more often. This is possibly due to a large number of leaf nodes, to which if a single edge is added the assumption on α_u would be violated.

D.5. How tight is the bound $||\mathbf{E}||_2 \leq ||\mathbf{E}||_1$?

In Section 6.2 we consider the tightness of the inequality $||\mathbf{E}||_2 \leq ||\mathbf{E}||_1$ for various random graph models and perturbation strategies. For each strategy we plot the absolute value of $||\mathbf{E}||_2$ and $||\mathbf{E}||_1$ for all random graph models in Fig. S5. We plot just 5 repeats for clarity. In Fig. S6 we plot the looseness of the bound.

D.6. How tight are the bounds on $||\mathbf{E}||_1$ and $||\mathbf{E}||_2$?

In Section 6.3 we consider the looseness of Eq. 10. We can consider the looseness for each component seperately which is shown in Fig. S7. It can be seen that the bound on the third term is the loosest.

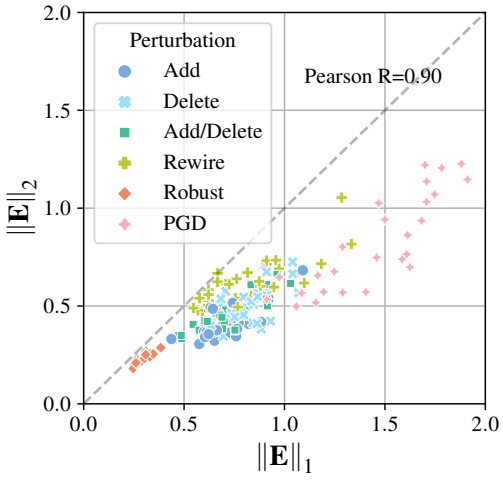


Figure S5. Comparison of $\|\mathbf{E}\|_1$ and $\|\mathbf{E}\|_2$. The dashed line represents $\|\mathbf{E}\|_2 = \|\mathbf{E}\|_1$.

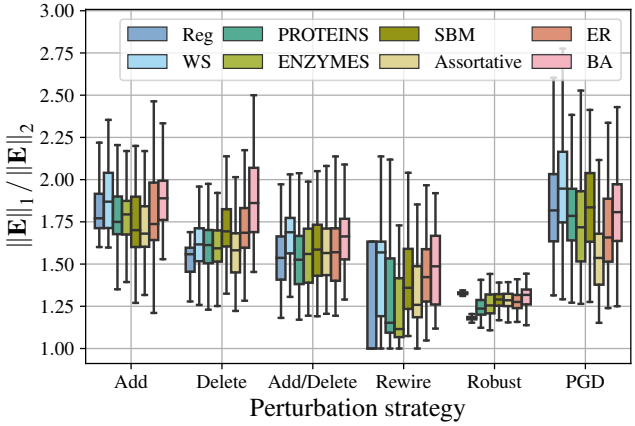


Figure S6. Looseness of the inequality $\|\mathbf{E}\|_1 \leq \|\mathbf{E}\|_2$ under different perturbation strategies and random graph models.

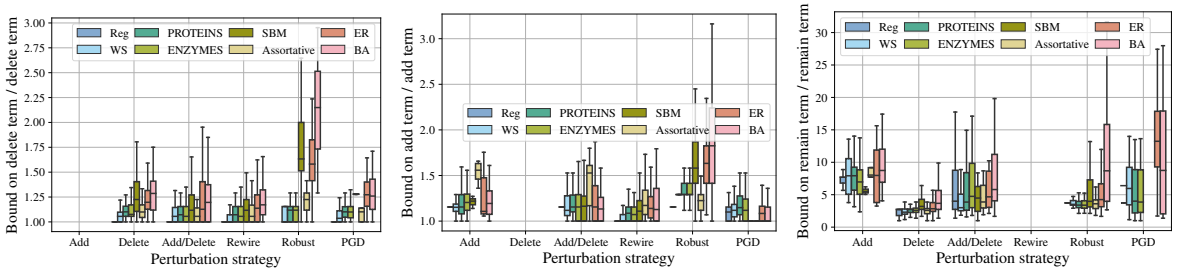


Figure S7. Looseness of the bound for the three terms in Eq. (6). The first term, as well as the bound for the first term, evaluate to zero for the Add strategy and thus looseness is undefined. The same applies to the second term for the Delete strategy, and the third term for the Rewire strategy.

Graph	Add	Delete	Add/Delete	Rewire	Robust	PGD
K-Reg	0.73	1.00	0.98	1.00	1.00	0.65
WS	0.85	1.00	0.95	1.00	1.00	0.69
PROTEINS_full	0.70	1.00	0.90	1.00	1.00	0.68
ENZYMES	0.80	1.00	0.92	1.00	1.00	0.80
SBM	0.03	1.00	0.23	1.00	1.00	0.02
Assortative	0.04	1.00	0.28	1.00	1.00	0.09
ER	0.07	1.00	0.33	1.00	1.00	0.14
BA	0.38	1.00	0.91	1.00	1.00	0.35

Table S2. Proportion of experiments which are valid ($\alpha_u \in [0, 1)$ for all nodes $u \in \mathcal{V}$) across all graph types and perturbation strategies.