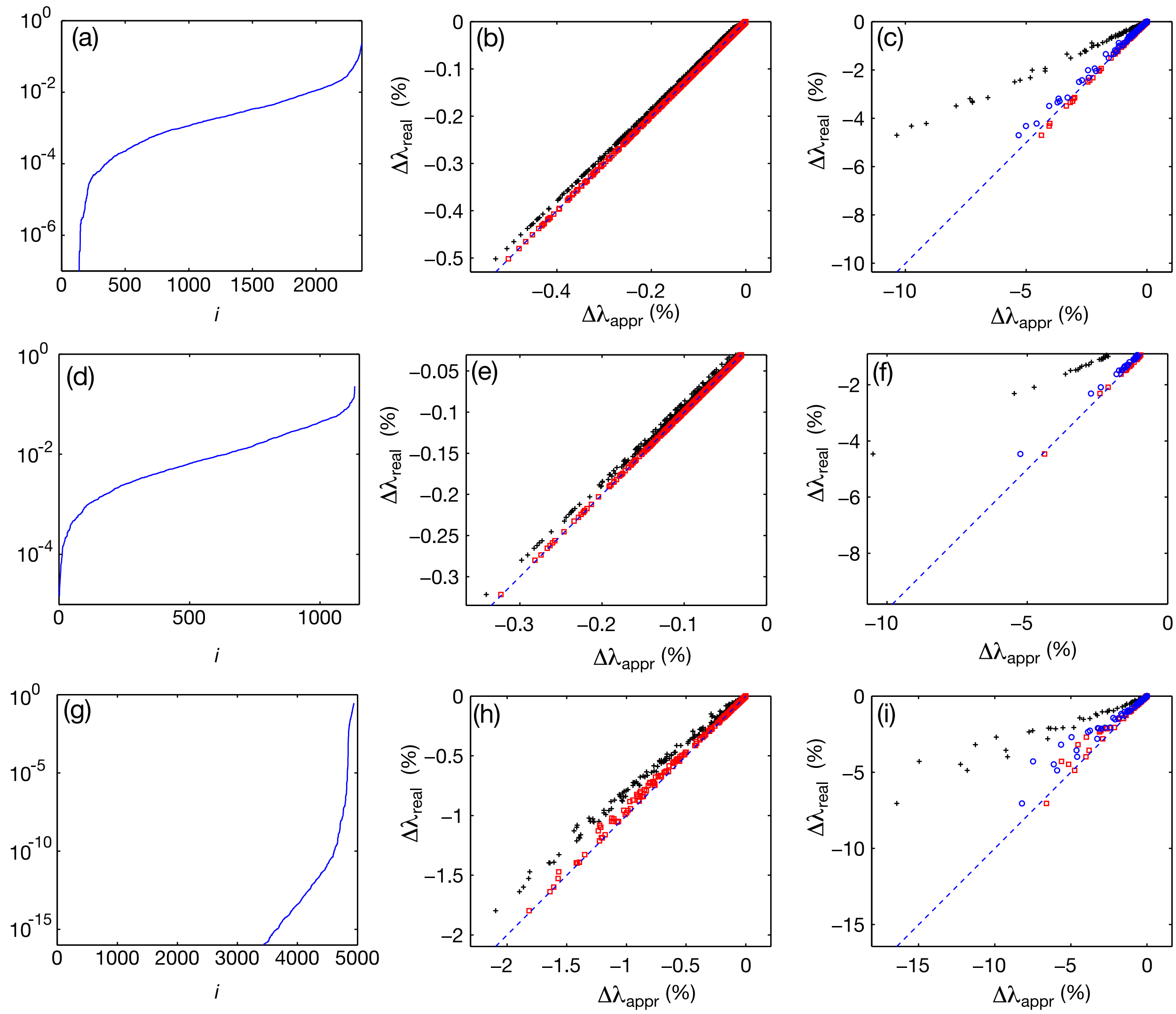

FIG. 2: (Color online) Results for the three examples of real-world networks $\mathcal{G}_2$ (top row), $\mathcal{G}_3$ (middle row) and $\mathcal{G}_4$ (bottom row) in Table I. The left column [(a), (d), and (g)] shows the eigenvector components sorted in the increasing order. The middle column [(b), (e), and (h)] shows the true SI vs approximated SI for the removal of edges, while the right column [(c), (f), and (i)] is for the removal of nodes. The meaning of the symbols is the same as in Fig. 1.

Assuming that the spectrum of A satisfies $|\lambda| > |\lambda_2| \geq \ldots \geq |\lambda_n|$, one can adopt the power method [14–16] to solve for the dominant eigenvalue λ and its corresponding left and right eigenvectors u and v. The starting point of this method is a normalized vector $v^{(0)}$, that in this case can be taken as

$$v^{(0)} = \frac{1}{\sqrt{n}}[1, 1, \cdots, 1]^T. \tag{29}$$

Then, for $t = 1, 2, \ldots$, until convergence, the following is iterated:

$$y^{(t)} = Av^{(t-1)},\ \lambda^{(t)} = ||y^{(t)}||_2,\ v^{(t)} = y^{(t)}/||y^{(t)}||_2. \tag{30}$$

The convergence of both the eigenvalue and eigenvector is geometric, with rate $O(|\lambda_2/\lambda|^t)$ [39]. This algorithm can be straightforwardly adjusted for the computation of the left eigenvector u, and the same convergence rates apply. Although other iterative schemes for the computation of the dominant eigenvalue and the corresponding eigenvector are available, the power method is used here since its iterations directly highlight the local information of a graph.

Indeed, the first iteration of the power method gives the degree, the number of connections that each node has, up to a normalization constant. The kth component of $u^{(1)}$ is proportional to the *in-degree* of node k, $d_k^{\text{in}} = \sum_{i=1}^{n} a_{ik}$, while the kth component of $v^{(1)}$ scales with the *out-degree* of the same node, $d_k^{\text{out}} = \sum_{i=1}^{n} a_{ki}$. If the graph is undirected and unweighted, the kth component of both $u^{(1)}$ and $v^{(1)}$ are proportional to the number of direct neighbors that node k has (counting itself, if a self-loop is in place).