Network Perturbation Aggregation for Graphon Estimation

Abstract

In recent years, various methods have been proposed to estimate the edge probability under the graphon model given a single observed network. Since the presence or absence of edges in the observed network is a stochastic realization from the underlying probability structure, estimating edge probabilities based on only a single observed realization is inherently limited and leads to estimates with high variance and high mean squared error (MSE). To address this issue, we propose the **Net**work **P**erturbation **ag**gregat**ing** (Net-Paging) method. The key idea is to construct multiple perturbed networks that preserve key graphon properties, mimicking multiple replications. We then obtain graphon estimates from each perturbed network and average these estimates to obtain the final estimate. We theoretically show that the more perturbed samples used in our algorithm, the smaller the MSE, with a linear dependency. Extensive simulation experiments and real data analysis show that Net-Paging effectively reduces the variance and MSE compared to existing methods.

Keywords: Graphon estimation, network perturbation, variance reduction, aggregation
Mathematics Subject Classification (2020): 62XXX

1 Introduction

Network data has provided unprecedented opportunities across diverse domains. For example, knowledge graphs play a key role in improving the semantic understanding and reasoning abilities of large language models (Hou et al., 2024; Pan et al., 2024). To better understand networks, various statistical random graph models have been to model their generative mechanisms. Most random graph models take a parametric approach, imposing strict assumptions on structures, including the Erdős–Rényi (ER) model (Gilbert, 1959; Erdős and Rényi, 1959), the stochastic block model (SBM) (Holland et al., 1983) and SBM variants (Karrer and Newman, 2011; Latouche et al., 2011; Cai and Li, 2015), the exponential random graph model (ERGM) (Lusher et al., 2012), and latent position model (Hoff et al., 2002). However, as the complexity of a network increases, it becomes increasingly more challenging to fit the data using a particular parametric model.

To address this challenge, the graphon model (Lovász and Szegedy, 2006), a non-parametric solution, has been proposed. In particular, the graphon model assumes the existence of an edge between nodes i and j follows a Bernoulli distribution with mean p_{ij}, which is determined by a measurable symmetric function f, i.e., $p_{ij} = f(\xi_i, \xi_j)$. The symmetric function f is called graphon. The sequence ξ_i are random variables sampled from a uniform distribution on $[0, 1]$. By avoiding imposing a specific form for f, the graphon model enjoys the advantage of flexibility

and generalizability. Many parametric models can be viewed as graphon's special cases - ER with the constant function and SBMs with step functions (Eldridge et al., 2016).

Graphon estimation aims to estimate the connecting probability p_{ij} under the graphon framework. Accurately estimating these probabilities facilitates many interesting downstream applications, such as link prediction (Zhou et al., 2022; Sultana et al., 2025), causal inference under network interference (Li and Wager, 2022), assessing vulnerabilities in the smart grid (Atat et al., 2023), and hypothesis testing for networks (Sischka and Kauermann, 2025). Graphon also acts as a critical input to GNN-based pipelines (Han et al., 2022b; Herbst and Jegelka, 2025; Sun et al., 2024; Wu et al., 2020). In the existing literature, many graphon estimation methods have been developed (Channarond et al., 2012; Airoldi et al., 2013; Chan and Airoldi, 2014; Chatterjee, 2015; Zhang et al., 2017; Qin et al., 2021).

Despite many successful applications of existing graphon estimation methods, most of them suffer from high variance issues, leading to an increased Mean Squared Error (MSE). This challenge is particularly evident in methods that require deterministic decisions. For example, neighborhood smoothing (NS) (Zhang et al., 2017) and iterative connection probability estimation (ICE) (Qin et al., 2021) both require determining whether a node is a neighbor or not. Such deterministic decisions are known to create instability (Bühlmann and Yu, 2002), thus increasing variance and MSE. Figure 1 illustrates this problem. By applying the NS method (Zhang et al., 2017) to three networks in Figure 1(a)-(c), which are generated from the same graphon model, we obtained three graphon estimates in Figure 1(d)-(f). The noticeable high variability among these graphon estimators underscores the issue of high variance.

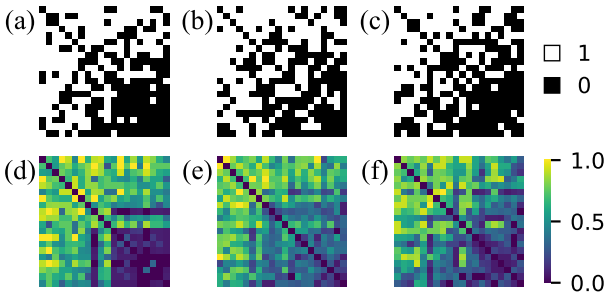

Figure 1: (a)-(c) Three adjacency matrices. (d)-(f) Heatmaps of the graphon estimators obtained using (a)-(c), respectively. High variability in (d)-(f) shows the high variance issue.

To address this fundamental challenge of variance reduction in graphon estimation, we draw inspiration from Bagging (**B**ootstrap **agg**regat**ing**) (Breiman, 1996), which is one of the most effective procedures for reducing variance and improving the accuracy of unstable estimators. The conventional bagging procedure, which involves generating resamples of the original dataset followed by aggregation, has demonstrated remarkable success in i.i.d. data settings (Bühlmann, 2012).

Several recent works have extended bootstrap-style resampling to network data, with goals that differ from ours. For example, Green and Shalizi (2022) propose empirical graphon and histogram bootstrap procedures for exchangeable random graphs, with consistency guarantees for smooth network functionals. The goal there is distributional inference: approximating the sampling distribution of motif densities to construct confidence intervals and hypothesis tests for functionals of the graphon. Our goal is different, we aim to reduce the variance of the graphon estimator itself by aggregating across resamples, which requires explicit bias correction and a different theoretical analysis.

A separate line of work develops other forms of network perturbation or resampling (Adamic et al., 2001; Levin and Levina, 2021; Wu et al., 2022; Li et al., 2020; Chen et al., 2019), focused on preserving specific network properties such as the degree distribution (Adamic et al., 2001) or the community structure (Wu et al., 2022); these methods lack theoretical guarantees for preserving the graphon structure. Other approaches employ low-rank matrix completion (Li et al., 2020), which imposes restrictive rank assumptions and incurs $O(n^3)$ cost from singular value decomposition; operate on subgraph collections (Fan et al., 2025; Bar-Shalom et al., 2024); or generate replicates that preserve hierarchical community structure from a statistical-mechanics perspective (Fushing et al., 2014). These methods are valuable for network resampling, but they do not target the statistical objective of this paper: constructing perturbed networks that accurately reflect the underlying graphon structure, so that graphon estimates from the perturbed networks can be aggregated to reduce MSE. Our contribution is therefore not bootstrap-style resampling for networks in general, but a perturbation-aggregation framework specifically designed for variance reduction in graphon estimation.

Our contributions. To address the aforementioned challenges, we propose the **Net**work **P**erturbation **ag**gregat**ing** (Net-Paging) method. The workflow of Net-Paging, as illustrated in Figure 2, consists of three key components: network perturbation, estimation with bias correction, and aggregation. First, we introduce controlled randomness to specific edge values (for selected node pairs) to generate multiple perturbed versions of the original network. For each perturbed network, we obtain an initial graphon estimate using existing estimation methods. To ensure estimation accuracy, we rigorously derive the distribution difference between perturbed and original networks, and develop a bias correction formula. We prove that a debiased graphon estimator of a perturbed network achieves the same convergence rate as the estimator based on the original network. The final step involves aggregating multiple debiased estimates through averaging, which effectively reduces variance and enhances stability.

We theoretically establish that increasing the number of perturbations leads to reduced MSE for graphon estimation. Extensive experiments on both synthetic and real-world networks demonstrate that Net-Paging substantially reduces variance and MSE, outperforming existing methods. From a practical view, the generation of multiple perturbed networks is highly efficient and can be easily parallelized, ensuring scalability for large-scale network data. Our framework is also highly flexible, compatible with any existing graphon estimation method.

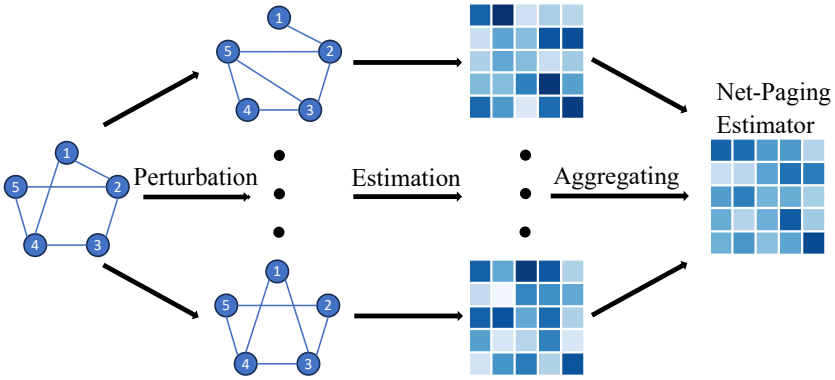

Figure 2: Workflow of the Net-Paging for graphon estimation. We first generate multiple perturbed networks, estimate the graphon for each, and then aggregate the estimates by averaging.

2 Preliminaries

Let $A = (a_{ij})_{1 \le i,j \le n}$ denote the adjacency matrix of a graph, where n is the number of nodes, $a_{ij} = 1$ if nodes i and j are connected by an edge; otherwise $a_{ij} = 0$. We only consider a network with no self-loops, so all diagonal entries of A are 0. We adopt the convention $a_{ii} = 0$ for simplicity, however the Net-Paging framework itself does not require zero diagonal entries and would apply directly to networks with self-loops if a corresponding base estimator were available. In this paper, we focus on undirected networks (A is symmetric).

2.1 Graphon model

To model the randomness in a network, we assume that the $\{a_{ij} : 1 \le i < j \le n\}$ follow independent Bernoulli distributions, i.e.,

$$a_{ij} \overset{\text{ind}}{\sim} \text{Ber}(p_{ij}), \qquad 1 \le i < j \le n, \tag{1}$$

where p_{ij} is the edge probability between i and j, and is not estimable without any assumptions, as there is only one observation for each parameter. To make p_{ij} estimable, we assume that

$$p_{ij} = f(\xi_i, \xi_j), \qquad 1 \le i < j \le n, \tag{2}$$

where $f : [0,1] \times [0,1] \to [0,1]$ is referred to as the graphon function, and the function is a bounded symmetric function. Here, $\xi_i \sim \text{Uniform}(0,1)$ represents latent positions associated with nodes i, independently and identically distributed from a uniform distribution on the interval $[0,1]$, $1 \le i \le n$. The combined model described by (1) and (2) is commonly referred to as the graphon model (Chan and Airoldi, 2014).

2.2 Graphon estimation

Graphon estimation in the literature usually refers to the estimation of the probability matrix $P = (p_{ij}) \in (0,1)^{n \times n}$ instead of the estimation of f due to the identifiability issue, as we will show below. As pointed out by Diaconis and Janson (2007) and Zhang et al. (2017), f is neither unique nor identifiable as f and ξ_i's are confounded with each other unless a strong assumption is imposed. In practice, since the main purpose of estimating f is to estimate P, the identifiability of f may not matter if P itself can be estimated, researchers now focus on estimating P directly. In recent years, various methodologies have been developed to estimate P within the graphon framework. A popular line of research employs local smoothing techniques by borrowing the idea of kernel smoothing.

Two representative methods are neighborhood smoothing (NS) (Zhang et al., 2017) and the sort-and-smooth procedure (SAS) (Chan and Airoldi, 2014). For each node i, both methods construct a "neighborhood" $\hat{\mathcal{N}}_i \subseteq \{1, \dots, n\} \setminus \{i\}$ from the observed adjacency matrix A. Throughout this paper, neighborhood does not refer to the adjacent nodes of node i. Rather, it denotes a data-adaptive set of nodes selected for smoothing according to a method-specific similarity criterion computed from A. Once such neighborhoods are selected, the next step is to estimate p_{ij} by averaging observed adjacency entries over neighborhoods associated with nodes

i and j, namely

$$\hat{p}_{ij} = \text{Avg}\left\{a_{i'j'} : (i', j') \in \hat{\mathcal{N}}_i \times \hat{\mathcal{N}}_j\right\}. \tag{3}$$

This formulation captures the common local-averaging idea; in practice, specific estimators may use slightly different computationally efficient variants of this averaging step.

In terms of neighborhood selection, both NS and SAS make hard, data-adaptive decisions from the observed adjacency matrix A: each candidate node is either included in $\hat{\mathcal{N}}_i$ or excluded according to whether a method-specific selection quantity falls within a cutoff. NS (Zhang et al., 2017) uses a common-neighbor dissimilarity $\tilde{d}^{\text{NS}}(i, i'; A)$ as the selection quantity. For a target node i and a candidate node $i' \neq i$, this dissimilarity compares, across reference nodes k, the number of common neighbors shared by (i, k) with those shared by (i', k). It is small when i and i' have similar common-neighbor profiles, and $\hat{\mathcal{N}}_i^{\text{NS}}$ is formed by selecting candidates whose dissimilarities fall below the h_{NS}-quantile, where $h_{\text{NS}} \in (0, 1)$ is a bandwidth hyperparameter. SAS (Chan and Airoldi, 2014) instead uses empirical degree as the selection quantity: nodes are sorted by $\hat{d}_i = \sum_{j \neq i} a_{ij}$, partitioned into consecutive blocks of size h_{SAS}, and $\hat{\mathcal{N}}_i^{\text{SAS}}$ is defined as the block containing node i.

High variance issue. The hard-selection nature of NS and SAS makes their empirical neighborhoods sensitive to sampling noise in A. Since A is a stochastic realization from P in (2), the corresponding selection quantities are noisy estimates of their population counterparts. Candidates whose selection quantities lie close to the relevant cutoff are therefore unstable: small perturbations of A can move them across the cutoff, changing whether they are included in $\hat{\mathcal{N}}_i$. Such boundary switches can propagate directly to the neighborhood average and increase the variability of the resulting graphon estimator.

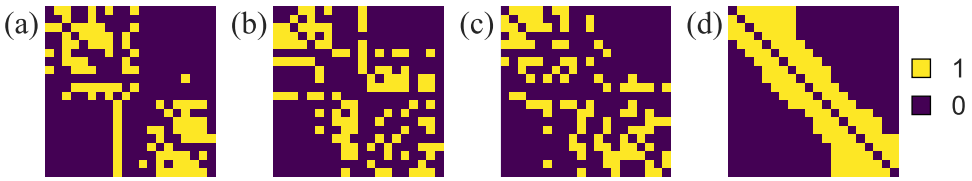

Figure 3: (a)-(c) Empirical neighborhood matrix of three network examples in Figure 1. (d) Oracle neighborhood matrix. The ijth entry is one if node j is selected as the neighbor of node i, and zero otherwise.

To illustrate this issue, we applied the NS method (Zhang et al., 2017) to three different network realizations (as shown in Figure 1) that are generated from the same graphon probability matrix. Figures 3(a)-(c) show the empirical neighborhood selection results obtained from these networks. In each matrix, the (i, j)th entry is one if node i is selected as the neighbor of node j, and zero otherwise.

For comparison, Figure 3(d) shows the oracle neighborhood matrix, which is calculated using the true probability matrix P instead of the observed adjacency matrix A. As we can see, the empirical neighborhood matrices exhibit significant variability across different network realizations, each deviating from the oracle one. This variability propagates to graphon estimation, leading to unstable and unreliable results, as also demonstrated in Figure 1. Importantly, such high-variance issues are not unique to the NS method. They are also present in many other methods (Chan and Airoldi, 2014; Chatterjee, 2015; Qin et al., 2021). The high-variance issue

underscores the need for strategies that can enhance estimator stability and reliability.

2.3 Bagging for i.i.d. data

In the context of independent and identically distributed (i.i.d.) data, one of the most effective techniques for reducing variance is bootstrap aggregating, commonly known as bagging. Consider an estimation task based on n i.i.d. data $\mathcal{L} = \{x_1, x_2, \ldots, x_n\}$. The estimator is given by $\hat{\theta}(\mathcal{L})$. In this setting, bagging consists of the following three steps: (1) Generate B bootstrap samples $\mathcal{L}_1^*, \mathcal{L}_2^*, \ldots, \mathcal{L}_B^*$, each of size n, by randomly drawing from the original dataset $\mathcal{L}$ with replacement (Efron, 1992); (2) For each bootstrap sample $\mathcal{L}_b^*$, we compute the estimate $\hat{\theta}_b^* = \hat{\theta}(\mathcal{L}_b^*)$, where $b = 1, \ldots, B$; (3) Aggregate these estimates by averaging: $\hat{\theta}_{\text{bag}} := \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}_b^*$.

In the original bagging paper (Breiman, 1996), Breiman points out that there can be a drastic variance reduction if the original estimator $\hat{\theta}$ is "unstable". Bühlmann and Yu (2002) then shows that the bagging method is particularly effective in addressing problems involving deterministic decisions. By aggregating estimators built from different bootstrap samples, bagging effectively smooths out decision boundaries. Nevertheless, applying bagging to network data poses significant challenges due to the complex dependency structures inherent in networks. Naively resampling nodes or edges can disrupt the inherent topological and structural properties. To address this challenge, we are motivated to develop Net-Paging, a new network perturbation method that can maintain the network's structural integrity while allowing for variance reduction, as we will show in the following section.

3 Net-Paging Method

In what follows, we present the detailed procedure of Net-Paging.

Generating perturbed network. We generate a perturbed network by randomly perturbing the edge values of a random subset $\mathcal{S}$ of all possible node pairs $\{(i,j) \mid 1 \le i < j \le n\}$, where each pair is included independently with probability ρ. The hyperparameter ρ controls the proportion of edges to be perturbed, allowing us to adjust the level of variability introduced. For each node pair in $(i,j) \in \mathcal{S}$, we mask the original edge value a_{ij} by replacing it with a new binary variable a_{ij}^* generated from a Bernoulli distribution with parameter $\text{Ber}(d)$, where $d \in (0,1)$ is another hyperparameter. For node pairs not in $\mathcal{S}$, we retain its original value. In this way, we obtain a perturbed network with adjacency matrix $A^* = (a_{ij}^*)$, where

$$a_{ij}^* = \begin{cases} a_{ij}, & \text{if } (i,j) \notin \mathcal{S}, \\ a_{ij}^*, \text{ where } a_{ij}^* \overset{\text{i.i.d.}}{\sim} \text{Ber}(d), & \text{else.} \end{cases} \tag{4}$$

Distribution of perturbed network. A key question in the development of Net-Paging is how the distribution of the perturbed adjacency matrix A^* relates to that of the original adjacency matrix A.

By employing the law of total probability, we can derive that

$$\begin{aligned}\mathbb{P}(a_{ij}^* = 1) &= \mathbb{P}\left(a_{ij}^* = 1 \mid (i,j) \in \mathcal{S}\right) \mathbb{P}\left((i,j) \in \mathcal{S}\right) \\ &\quad + \mathbb{P}\left(a_{ij}^* = 1 \mid (i,j) \notin \mathcal{S}\right) \mathbb{P}\left((i,j) \notin \mathcal{S}\right) \\ &= \rho d + (1-\rho) p_{ij}.\end{aligned}$$

Therefore, the distribution of a_{ij}^* is given by

$$a_{ij}^* \sim \text{Ber}\left(\rho d + (1-\rho) p_{ij}\right), \tag{5}$$

demonstrating that there is a distribution shift between A^* and A. The perturbation process modifies the probability of an edge being present, introducing a bias that depends on the hyper-parameters ρ (the proportion of perturbed edges) and d (the probability of assigning an edge in perturbed pairs). Importantly, because both ρ and d are known, there exists a one-to-one linear mapping between the perturbed and original probabilities: $p_{ij} = \frac{1}{1-\rho}\left(\mathbb{P}(a_{ij}^* = 1) - \rho d\right)$.

Bias correction. The presence of this bias necessitates an explicit correction step in our estimation procedure. To address this, we introduce a debiased estimation framework, which proceeds as follows: We first apply a specific graphon estimation to A^* to get a graphon estimate $\hat{P}(A^*)$, of which the (i,j)th entry is an estimate of $\mathbb{P}(a_{ij}^* = 1)$. Given the established relationship between p_{ij} and $\mathbb{P}(a_{ij}^* = 1)$, we obtain the debiased estimate using

$$\tilde{P}(A^*) = \frac{1}{1-\rho}(\hat{P}(A^*) - \rho d). \tag{6}$$

The debiased estimator $\tilde{P}(A^*)$ in (6) is not guaranteed to lie in $[0,1]$ in finite samples. To ensure that $\tilde{P}(A^*)$ remains a valid probability matrix, we clip any out-of-range entries to $[0,1]$ after debiasing. In practice, the proportion of entries requiring clipping is small. Figure 8 reports the proportion of entries affected by this truncation. Across the simulation settings, the proportion of clipped entries is generally small and decreases rapidly as the network size increases. For most graphons, even at the smallest network size, only a small fraction of entries falls outside $[0,1]$ after debiasing; for moderate and large network sizes, the clipping proportion is close to zero.

Aggregation. The final step of the Net-Paging involves aggregating the debiased estimators obtained from multiple perturbed networks. In particular, we repeat the aforementioned perturbation process B times to generate B perturbed networks $A_1^*, \ldots, A_B^*$. For each perturbed network, we estimate the graphon using a specific estimation method and subsequently apply the bias correction formula in (6) to obtain debiased estimates $\tilde{P}(A_1^*), \ldots, \tilde{P}(A_B^*)$. We then aggregate these estimates by averaging:

$$\hat{P}_{\text{pag}} = \frac{1}{B}\sum_{b=1}^{B} \tilde{P}(A_b^*) = \frac{1}{1-\rho}\left(\frac{1}{B}\sum_{b=1}^{B} \hat{P}(A_b^*) - \rho d\right). \tag{7}$$

The Net-Paging procedure is summarized in Algorithm 1.

Algorithm 1 Net-Paging

Input: Adjacency matrix A, masking rate ρ, perturbation number B, base graphon estimation method $\hat{P}$, and d.
Output: Probability matrix estimation $\hat{P}_{\text{pag}}$.
for $b = 1$ **to** B **parallel do**
(1) Randomly select a subset $\mathcal{S}$ from the upper triangle indices of A with proportion ρ.
(2) Generate the perturbed adjacency matrix A_b^* using eq. (4).
(3) Apply the base graphon estimation method to A_b^* to obtain $\hat{P}(A_b^*)$.
end for
Calculate $\hat{P}_{\text{pag}}$ using all the $\hat{P}(A_b^*)$'s by eq. (7)

4 Theoretical Analysis

4.1 Theoretical Property of the Perturbed Network

In this subsection, we investigate how well the perturbed network A^* preserves the key properties of the latent graphon function that generates the observed network A. Specifically, we derive the estimation error rate (Gao et al., 2016; Gao and Ma, 2021) for the graphon estimator obtained from a perturbed network A^*. If this estimator exhibits similar convergence rate as one obtained directly from the original network, it would suggest that A^* successfully emulates the process of random sampling networks from the underlying graphon structure. To build theoretical properties of A^*, we need to impose the following assumptions.

Assumption 1. *The graphon function class $\mathcal{F}$ is closed with respect to affine transformation, i.e., $f \in \mathcal{F} \Rightarrow (1-\rho)f + \rho d \in \mathcal{F}$ for $\forall \rho, d \in [0,1]$.*

Assumption 2. *The estimation error of $\hat{P}(A)$ is bounded by the convergence rate $C_1(n)$ given by the following equation.*

$$\max_{f \in \mathcal{F}} \mathbb{P}\left(n^{-2}\|\hat{P}(A) - P\|_F^2 \geq C_1(n)\right) \leq D_1(n),$$

where $\lim_{n\to\infty} C_1(n) = 0, \lim_{n\to\infty} D_1(n) = 0$.

Assumption 1 requires that applying the transformation $(1-\rho)f + \rho d$ to $f \in \mathcal{F}$ results in another function within $\mathcal{F}$, indicating that both P and $P^* := (1-\rho)P + \rho d$ belong to the same graphon function class. Many standard function classes in statistical learning and functional analysis, such as Lipschitz functions, are naturally closed under affine transformations. Assumption 2 requires convergence behavior of the graphon estimation method that we employ. Specifically, it states that the squared Frobenius norm of the difference between the estimated graphon $\hat{P}(A)$ and the true graphon P, when normalized by n^2, converges to zero at a rate of $C_1(n)$. This convergence holds for the worst-case function within the class $\mathcal{F}$ with probability at least $1 - D_1(n)$, where both $C_1(n)$ and $D_1(n)$ approach zero as the network size n increases. The specific form of $C_1(n)$ and $D_1(n)$ is not predetermined, as it varies depending on the chosen estimation method. For instance, in the case of the neighborhood smoothing method (Zhang et al., 2017), the function class $\mathcal{F}$ is the set of "piecewise bi-Lipschitz" functions, and $C_1(n) = C_1'(\log n/n)^{1/2}$, $D_1(n) = n^{-D_1'}$, where C_1' and D_1' are two positive constants.

Theorem 1 establishes the convergence properties of our debiased estimator $\tilde{P}(A^*)$. Under Assumptions 1 and 2, the expected squared Frobenius norm of the difference between $\tilde{P}(A^*)$ and the true graphon P, when normalized by n^2, is bounded by $\frac{C_1(n)}{(1-\rho)^2} + D_1(n)$. This result demonstrates that the debiased estimator computed from the perturbed network A^* achieves an asymptotic convergence rate that is comparable to the original estimator, differing only by a constant factor $\frac{1}{(1-\rho)^2}$. This ensures that the perturbation and subsequent debiasing procedure do not degrade the fundamental statistical properties of the estimation process.

Theorem 1. *Let A^* be the adjacency matrix of the perturbed network, and $\tilde{P}(A^*) = \frac{1}{1-\rho}\left(\hat{P}(A^*) - \rho d\right)$ be the debiased estimator applied on A^*. If the graphon function class satisfies Assumption 1 and the original estimator $\hat{P}$ satisfies Assumption 2, then the estimation $\tilde{P}(A^*)$ achieves an asymptotic convergence rate that is comparable to the original estimator, differing only by a constant factor $\frac{1}{(1-\rho)^2}$,*

$$\max_{f \in \mathcal{F}} \mathbb{P}(n^{-2}\|\tilde{P}(A^*) - P\|_F^2 \geq \frac{C_1(n)}{(1-\rho)^2}) \leq D_1(n),$$

where $C_1(n)$ and $D_1(n)$ are the same as that in Assumption 2.

4.2 Number of Perturbation Replications: The More The Better

From eq. (7), we know that $\hat{P}_{\text{pag}}$ depends on A_b^*, $b = 1, \ldots, B$. The randomness of A_b^* arises from three distinct sources. (1) Inherent randomness in the observed network A. (2) Random selection of a subset of node pairs designated for masking. Let $\mathcal{R}_b^{(1)}$ denote the randomness associated with the selection process in the bth replication. (3) Random masking of selected edges: For each selected node pair, the original edge value is replaced with a new value sampled from a Bernoulli distribution with parameter d. Let $\mathcal{R}_b^{(2)}$ denote the randomness introduced by this Bernoulli sampling in the bth replication. Since A_b^* is a function of $A, \mathcal{R}_b^{(1)}, \mathcal{R}_b^{(2)}$, we rewrite $\hat{P}_{\text{pag}}$ defined in eq. (7) as $\hat{P}_{\text{pag}}\left(A, \{\mathcal{R}_b^{(1)}\}_{b=1}^B, \{\mathcal{R}_b^{(2)}\}_{b=1}^B\right)$, to explicitly represent the dependence of $\hat{P}_{\text{pag}}$ on these sources of randomness. For simplicity of notation, we define $\mathcal{R} = \left(\{\mathcal{R}_b^{(1)}\}_{b=1}^B, \{\mathcal{R}_b^{(2)}\}_{b=1}^B\right)$. The MSE of the Net-Paging estimator is then defined as

$$\text{MSE}(\hat{P}_{\text{pag}}) = \mathbb{E}_{A,\mathcal{R}}\left[||\hat{P}_{\text{pag}}(A, \mathcal{R}) - P||_F^2\right], \tag{8}$$

where the expectation $\mathbb{E}_{A,\mathcal{R}}$ is taken over the randomness in A as well as the randomness introduced through $\mathcal{R}$. The following theorem derives the relationship between MSE and B.

Theorem 2. *The MSE of $\hat{P}_{pag}$ can be decomposed as follows,*

$$MSE(\hat{P}_{pag}) = E_1 + V_1 + \frac{1}{B}V_2.$$

where

$$E_1 = \left\|\mathbb{E}_A\left[\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}(A, \mathcal{R}^{(1)}, \mathcal{R}^{(2)})\right]\right] - P\right\|_F^2,$$

$$V_1 = \sum_{i,j} Var_A\left(\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}_{ij}(A, \mathcal{R}^{(1)}, \mathcal{R}^{(2)})\right]\right),$$

$$V_2 = \sum_{i,j} \mathbb{E}_A\left[Var_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left(\tilde{P}_{ij}(A, \mathcal{R}^{(1)}, \mathcal{R}^{(2)})\right)\right],$$

and $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$ denotes the randomness introduced by the selection process and Bernoulli sampling in one perturbation, respectively.

Theorem 2 provides a decomposition of the MSE of the estimator $\hat{P}_{\text{pag}}$ into three distinct components: (1) E_1 represents the squared bias between the debiased estimator $\tilde{P}(A, \mathcal{R}^{(1)}, \mathcal{R}^{(2)})$ and the true probability matrix P, with respect to the randomness introduced by $A, \mathcal{R}^{(1)}, \mathcal{R}^{(2)}$. (2) V_1 is the variance of the estimator's conditional expectation. (3) V_2 is the expectation of the estimator's conditional variance. Note that E_1, V_1, and V_2 are positive and do not depend on B, thus MSE is a linear function of $1/B$. The larger the B, the lower the MSE.

4.3 Why Net-Paging Improves Neighborhood Smoothing

The simulation study in Section 6 shows that Net-Paging can substantially reduce the estimation error of neighborhood smoothing (Zhang et al., 2017). This subsection explains the mechanism behind this empirical phenomenon. We do not aim to derive a new convergence rate or to claim that perturbation aggregation uniformly improves every estimator. Instead, we ask a more specific question: among the various sources of error in the NS estimator, which one does Net-Paging reduce, and through what mechanism?

The NS estimator contains a discontinuous, data-adaptive step: small changes in the observed adjacency matrix A can move a candidate node across the neighborhood cutoff, change the averaging set, and produce instability in the final estimator. The strategy of this section is to isolate the part of the NS error that arises from this discontinuous step and show how Net-Paging acts on it.

For any symmetric matrix $M \in \mathbb{R}^{n\times n}$, let $\widehat{\mathcal{N}}_i(M) \subseteq \{1, \ldots, n\} \setminus \{i\}$ denote the neighborhood selected by the NS rule applied to M. For NS,

$$\widehat{\mathcal{N}}_i(M) \;=\; \{i' \neq i : \tilde{d}^{\text{NS}}(i, i'; M) \leq c_i(M)\},$$

where $\tilde{d}^{\text{NS}}(\cdot, \cdot; M)$ is the distance measure in NS method evaluated at M, and $c_i(M)$ is the h_{NS}-quantile of $\{\tilde{d}^{\text{NS}}(i, i'; M) : i' \neq i\}$. Thus, $\widehat{\mathcal{N}}_i(A)$ is the empirical neighborhood used by NS, whereas $\widehat{\mathcal{N}}_i(P)$ is the population-level neighborhood obtained by applying the same NS selection rule to the probability matrix P.

Define the row-normalized weight matrix

$$W(M) = (w_{ik}(M))_{1\leq i,k\leq n}, \qquad w_{ik}(M) = \frac{\mathbf{1}\{k \in \widehat{\mathcal{N}}_i(M)\}}{|\widehat{\mathcal{N}}_i(M)|}, \tag{9}$$

under which the NS estimator in (3) admits the matrix form $\widehat{P}(M) = W(M)MW(M)^\top$.

Three-component error decomposition for NS. To separate the sources of error in NS, we introduce two intermediate quantities:

$$\widehat{P}^{PA} := W(P)\, A\, W(P)^\top, \qquad \widehat{P}^{PP} := W(P)\, P\, W(P)^\top.$$

The first estimator uses the population-level neighborhoods to average the observed edges. The second replaces both the neighborhoods and the edge values by their population-level counterparts. Neither quantity is computable from the observed network; they are introduced only for analysis.

Writing $\widehat{P}(A) - P$ as the sum of differences along the chain $\hat{P}(A) \to \hat{P}^{PA} \to \hat{P}^{PP} \to P$ yields

$$\widehat{P}(A) - P = (\widehat{P}(A) - \widehat{P}^{PA}) + (\widehat{P}^{PA} - \widehat{P}^{PP}) + (\widehat{P}^{PP} - P) = \mathcal{E}^{\text{sel}}(A) + \mathcal{E}^{\text{noise}}(A) + \mathcal{E}^{\text{bias}}. \quad (10)$$

This decomposition separates three sources of error. The first term, $\mathcal{E}^{\text{sel}}(A) = W(A)AW(A)^\top - W(P)AW(A)^\top$ is the error caused by using the empirical selector $W(A)$ instead of the population-level selector $W(P)$. The second term, $\mathcal{E}^{\text{noise}}(A) = W(P)(A - P)W(P)^\top$, is the remaining Bernoulli edge noise after the selector has been fixed at its population-level value. The third term, $\mathcal{E}^{\text{bias}} = W(P)PW(P)^\top - P$, is the deterministic approximation error from smoothing P over the population-level neighborhoods. Among these three terms, only $\mathcal{E}^{\text{sel}}(A)$ involves the random discontinuous map $A \mapsto W(A)$.

Three-component error decomposition for Net-Paging on NS. The previous decomposition applies to the original NS estimator. We next apply the same idea to the perturbed and debiased estimator used by Net-Paging. Recall that A_b^* denotes the bth perturbed network and $P^* = \mathbb{E}(A_b^* \mid P) = (1-\rho)P + \rho d \mathbf{1}_n \mathbf{1}_n^\top$ denotes its population counterpart. Applying the decomposition (10) to each $\widehat{P}(A_b^*)$ with P replaced by P^* in the bias and noise components:

$$\widehat{P}(A_b^*) - P^* \;=\; \mathcal{E}^{\text{sel},*}(A_b^*) + \mathcal{E}^{\text{noise},*}(A_b^*) + \mathcal{E}^{\text{bias},*},$$

where

$$\mathcal{E}^{\text{sel},*}(A_b^*) = W(A_b^*)A_b^*W(A_b^*)^\top - W(P^*)A_b^*W(P^*)^\top,$$

$$\mathcal{E}^{\text{noise},*}(A_b^*) = W(P^*)(A_b^* - P^*)W(P^*)^\top,$$

$$\mathcal{E}^{\text{bias},*} = W(P^*)P^*W(P^*)^\top - P^*.$$

Since the Net-Paging estimator is $\widehat{P}_{\text{pag}}(A) = \frac{1}{B(1-\rho)} \sum_{b=1}^{B} (\widehat{P}(A_b^*) - \rho d \, \mathbf{1}_n \mathbf{1}_n^\top)$, we have

$$\widehat{P}_{\text{pag}}(A) - P \;=\; \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \mathcal{E}^{\text{sel},*}(A_b^*) \;+\; \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \mathcal{E}^{\text{noise},*}(A_b^*) \;+\; \mathcal{E}^{\text{bias}}_{\text{pag}}, \quad (11)$$

where $\mathcal{E}^{\text{bias}}_{\text{pag}} := (1-\rho)^{-1}\mathcal{E}^{\text{bias},*}$. The derivation is given in Supplement Section A.4.3.

The improvement of Net-Paging for NS is attributed primarily to the selection component, rather than to the bias or noise components. The reason is that the other two components do not involve the empirical selector $W(A)$ or the perturbed empirical selectors $W(A_b^*)$. The bias component $\mathcal{E}^{\text{bias}}_{\text{pag}}$ is a population-level smoothing error determined by P^* and is therefore unaffected by averaging over data-adaptive selections. The noise component $\frac{1}{B(1-\rho)} \sum_{b=1}^{B} W(P^*)(A_b^* - P^*)W(P^*)^\top$ applies fixed population-level weights $W(P^*)$, to centered Bernoulli noise; it depends on the Bernoulli noise but not on the discontinuous selection map $A \mapsto W(A)$. Hence,

neither component captures the instability of NS neighborhood selection. The only component that directly involves the hard empirical selector is the selection component.

Neighborhood selection gain–cost decomposition in Net-Paging. The selection component $\mathcal{E}^{\mathrm{sel},*}(A_b^*)$ is driven by the deviation of the empirical weight matrix $W(A_b^*)$ from its population counterpart $W(P^*)$, since the same perturbed adjacency matrix A_b^* appears in both terms.

To study this selection effect directly, we work with the binary inclusion matrix

$$H(M) = (H_{ii'}(M))_{1\le i,i'\le n}, \qquad H_{ii'}(M) = \mathbf{1}\{i' \in \widehat{\mathcal{N}}_i(M)\},$$

with $H_{ii}(M) = 0$. The matrix $H(M)$ records the binary neighborhood membership decisions before row normalization. Thus, $H(A)$ is the original empirical selector using observed A, $H(P)$ is the population-level selector, and $H(A_b^*)$ is the selector obtained from the bth perturbed network A_b^*.

Net-Paging replaces a single hard selector by an average over perturbed selectors. Define the perturbation-averaged selector

$$\bar{H}_B := \frac{1}{B}\sum_{b=1}^{B} H(A_b^*).$$

Although each $H(A_b^*)$ is still a hard selector, their average is a soft selector: $\bar{H}_{B,ii'}$ estimates how often candidate i' is selected for node i under perturbation.

The selector-level question is whether this softened selector is closer to the population-level selector $H(P)$ than the original empirical selector $H(A)$. We therefore define the selector-level improvement

$$I_B(A) := \|H(A) - H(P)\|_F^2 - \mathbb{E}_{\mathcal{R}}\{\|\bar{H}_B - H(P)\|_F^2 \mid A\},$$

where $\mathbb{E}_{\mathcal{R}}\{\cdot \mid A\}$ averages over the perturbation randomness for fixed A. Thus, $I_B(A) > 0$ means that perturbation aggregation improves the selector, while $I_B(A) < 0$ means that it harms the selector.

Flip probability. To analyze $I_B(A)$, we introduce the perturbation flip probability. For a fixed observed network A, define

$$q_{ii'}(A) := \mathbb{P}_{\mathcal{R}}\{H_{ii'}(A^*) \neq H_{ii'}(A) \mid A\}.$$

This quantity measures how likely the perturbation is to change the neighborhood membership decision for the pair (i, i'): small $q_{ii'}(A)$ means the decision is stable under perturbation, large $q_{ii'}(A)$ means it is unstable. The flip probability is therefore a direct measure of the local instability of NS's hard selection rule near A.

Agreement and disagreement sets. Whether a flip is helpful depends on whether the original empirical selector was correct relative to the population-level selector. We therefore partition the pairs (i, i') according to whether the empirical selector $H(A)$ agrees with the population-level selector $H(P)$:

$$C(A) := \{(i,i') : i \neq i',\ H_{ii'}(A) = H_{ii'}(P)\}, \qquad D(A) := \{(i,i') : i \neq i',\ H_{ii'}(A) \neq H_{ii'}(P)\}.$$

A flip on a pair in $C(A)$ introduces selector error; a flip on a pair in $D(A)$ corrects it.

The following proposition gives the exact finite-B tradeoff in terms of $q_{ii'}(A)$, $C(A)$, and $D(A)$. The proof is given in Supplement A.4.4. On agreement pairs, a flip produces cost $q_{ii'}^2$ plus a finite-B Monte Carlo variance term $B^{-1}q_{ii'}(1-q_{ii'})$. On disagreement pairs, a flip produces a first-order gain $2q_{ii'} - q_{ii'}^2$ minus the same Monte Carlo penalty.

Proposition 1 (Selector-level improvement decomposition)**.** *For any fixed observed network A and any $B \geq 1$, we have $I_B(A) = \mathrm{Gain}(A;B) - \mathrm{Cost}(A;B)$, where*

$$\mathrm{Gain}(A;B) := \sum_{(i,i')\in D(A)} \left\{ 2q_{ii'} - q_{ii'}^2 - \frac{1}{B} q_{ii'}(1-q_{ii'}) \right\},$$

and

$$\mathrm{Cost}(A;B) := \sum_{(i,i')\in C(A)} \left\{ q_{ii'}^2 + \frac{1}{B} q_{ii'}(1-q_{ii'}) \right\}.$$

Does higher $I_B(A)$ indicate larger MSE improvement? $I_B(A)$ captures improvement at the level of the binary selector, while the actual quantity of interest is the matrix-level MSE. We therefore empirically examine whether selector-level improvement is reflected in estimator-level MSE reduction. For each simulated network, under the graphon simulation settings in Section 6, we compute the empirical flip frequency $\widehat{q}_{ii'}(A) := B^{-1}\sum_{b=1}^{B} \mathbf{1}\{H_{ii'}(A_b^*) \neq H_{ii'}(A)\}$. Using these empirical flip frequencies, we calculate the selector-level improvement $\widehat{I}_B(A)$. We then compare $\widehat{I}_B(A)$ with the estimator-level MSE reduction $\Delta_{\mathrm{MSE}} := n^{-2}\|\widehat{P}(A) - P\|_F^2 - n^{-2}\|\widehat{P}_{\mathrm{pag}}(A) - P\|_F^2$, where $\Delta_{\mathrm{MSE}} > 0$ indicates that Net-Paging improves estimation accuracy relative to the original estimator.

Supplementary Figure 9 summarizes this comparison. Two patterns are apparent. First, in all simulation replicates, both $\widehat{I}_B(A)$ and Δ_{MSE} are positive, indicating that Net-Paging improves both the empirical selector and the final matrix estimator in these settings. Second, $\widehat{I}_B(A)$ is positively associated with Δ_{MSE}: replicates with larger selector-level improvement tend to exhibit larger reductions in matrix-level MSE. These results support the proposed mechanism that the finite-sample gain of Net-Paging is driven by stabilization of the empirical neighborhood selector.

Where does large flip probabilities occur? The previous analysis suggests that Net-Paging improves estimation by stabilizing the empirical neighborhood selector. We next examine where the perturbation-induced flips occur and whether they fall on the disagreement set $D(A)$ where flips correct selector error.

For NS, flip probability is closely related to how far a pair is from the selection cutoff. Define the empirical margin $m_{ii'}(A) := \widehat{d}^{\mathrm{NS}}(i,i';A) - c_i(A)$, where $\widehat{d}^{\mathrm{NS}}(i,i';A)$ is the NS distance between nodes i and i' computed from the observed adjacency matrix A, and $c_i(A)$ is the row-specific selection cutoff. Then $H_{ii'}(A) = \mathbf{1}\{m_{ii'}(A) \leq 0\}$. Thus, $m_{ii'}(A)$ measures the signed distance of the pair (i,i') from the selection boundary. Pairs with large $|m_{ii'}(A)|$ are far from the cutoff and are expected to be stable under perturbation. In contrast, pairs with $m_{ii'}(A)$ close to zero lie near the selection boundary and are therefore more likely to change their selection status.

Flips concentrate near the boundary. We verify this margin-based intuition empirically under the simulation settings in Section 6. For each simulation replicate, we compute the empirical margin $m_{ii'}(A)$ and the perturbation flip frequency $\widehat{q}_{ii'}(A)$. Supplementary Figure 10 plots the average flip frequency after binning directed pairs by $|m_{ii'}(A)|$. The flip frequency is highest near zero and decreases monotonically as $|m_{ii'}(A)|$ grows, confirming that perturbation acts primarily on near-boundary decisions.

Selection errors also concentrate near the boundary. We next examine whether the near-boundary region is also where the empirical selector is most prone to error. For each directed pair (i, i'), we compare the empirical selection decision $H_{ii'}(A)$ with the population-level decision $H_{ii'}(P)$ and record whether $H_{ii'}(A) \neq H_{ii'}(P)$. Supplementary Figure 11 plots the resulting empirical error rate after binning directed pairs by $|m_{ii'}(A)|$. The error rate is highest near the selection cutoff and decreases as $|m_{ii'}(A)|$ increases. Thus, pairs with small margins are disproportionately represented in $D(A)$, the set of pairs for which the empirical selector disagrees with the population-level selector.

Combining these two patterns: pairs with high flip frequency are also pairs with high misclassification rate, both concentrated near the selection boundary. Thus, the perturbation step mainly influences near-boundary decisions where the empirical selector is both unstable and more likely to be incorrect.

5 Hyperparameter Selection

Net-Paging involves three hyperparameters: the number of perturbation replications $B \in \mathbb{N}$, the masking rate $\rho \in (0, 1)$, and the Bernoulli replacement parameter $d \in [0, 1]$. These parameters play different roles.

For notational simplicity, we write $\tilde{P}(A^*)$ in place of $\tilde{P}(A, \mathcal{R}^{(1)}, \mathcal{R}^{(2)})$ and let $\bar{P}_{\rho,d}(A) := \mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}[\tilde{P}(A^*) \mid A]$ denote the conditional expectation of a single debiased perturbed estimator. The decomposition in Theorem 2 can then be written as

$$\mathrm{MSE}(\hat{P}_{\mathrm{pag}}) = \underbrace{\|\mathbb{E}_A \bar{P}_{\rho,d} - P\|_F^2}_{E_1} + \underbrace{\sum_{i,j} \mathrm{Var}_A(\bar{P}_{\rho,d,ij})}_{V_1} + \frac{1}{B} \underbrace{\sum_{i,j} \mathbb{E}_A[\mathrm{Var}_{\mathcal{R}}(\tilde{P}_{ij} \mid A)]}_{V_2}.$$

5.1 Choice of B

B controls a variance–computation tradeoff. For fixed ρ and d, increasing B reduces only the Monte Carlo term V_2/B in the MSE decomposition; the bias term E_1 and the conditional-mean variance term V_1 are unaffected. The computational cost, by contrast, grows linearly as $O(B\, T_{\mathrm{base}}(n))$, where $T_{\mathrm{base}}(n)$ is the cost of one run of the base estimator.

Figure 12 reports the sensitivity of RMSE to B across the eight graphon settings in Section 6. As expected, the RMSE decreases as B increases, but the incremental gain from further increasing B becomes smaller. Since the B perturbation replicates are conditionally independent given A, they can be computed in parallel. Therefore, when computational resources permit, we recommend using a relatively large value of B, such as $B = 500$.

5.2 Choice of d

Impact of d on E_1. We start from the squared-bias term $E_1 = \|\mathbb{E}_A[\bar{P}_{\rho,d}(A)] - P\|_F^2$. By the definition of the debiased perturbed estimator and the tower property, $\mathbb{E}_A[\bar{P}_{\rho,d}(A)] = \{\mathbb{E}[\hat{P}(A^*)] - \rho d\,\mathbf{1}\mathbf{1}^\top\}/(1-\rho)$. Since $P^* := \mathbb{E}[A^*] = (1-\rho)P + \rho d\,\mathbf{1}\mathbf{1}^\top$, we have $\rho d\,\mathbf{1}\mathbf{1}^\top = P^* - (1-\rho)P$. Substituting this identity into $\mathbb{E}_A[\bar{P}_{\rho,d}(A)] - P$ gives

$$\mathbb{E}_A[\bar{P}_{\rho,d}(A)] - P = \frac{\mathbb{E}[\hat{P}(A^*)] - P^*}{1-\rho}, \qquad E_1 = \frac{\|\mathbb{E}[\hat{P}(A^*)] - P^*\|_F^2}{(1-\rho)^2}. \tag{12}$$

Thus, d affects E_1 through the expected deviation of the base estimator from the perturbed target P^*. However, this identity does not lead to a practical optimization rule. For nonlinear base estimators such as NS, $\mathbb{E}[\hat{P}(A^*)]$ has no closed form, since it depends on how the data-adaptive neighborhood selector and the subsequent smoothing step respond to the perturbed network distribution. Hence, an exact optimal choice of d would require an estimator-specific analysis that is not available in closed form.

We therefore adopt the surrogate principle of minimizing $\|P^* - P\|_F^2$, motivated by the following heuristic: base estimators in the graphon literature are designed under structural assumptions on the target (e.g., piecewise Lipschitz smoothness for NS). When P^* remains close to P, the perturbed target is more likely to satisfy these structural assumptions to a similar degree as P, keeping the bias at the perturbed target, $\|\mathbb{E}\{\hat{P}(A^*)\} - P^*\|_F^2$, comparable to the bias at the original target, $\|\mathbb{E}\{\hat{P}(A)\} - P\|_F^2$, which is fixed (independent of d) and represents the irreducible bias of the base estimator. We do not claim that minimizing $\|P^* - P\|_F^2$ exactly minimizes $\|\mathbb{E}\{\hat{P}(A^*)\} - P^*\|_F^2$, only that it is a tractable surrogate.

Since $P^* - P = \rho(d\,\mathbf{1}\mathbf{1}^\top - P)$, we have $\|P^* - P\|_F^2 = \rho^2 \sum_{i,j}(d - p_{ij})^2$, with unique minimizer over $d \in [0,1]$ given by the average edge probability $d^* = \binom{n}{2}^{-1} \sum_{i<j} p_{ij}$. Since this is unknown, we use the plug-in estimator $\hat{\bar{p}} = \binom{n}{2}^{-1} \sum_{i<j} a_{ij}$, the observed network density.

Impact of d on V_1 and V_2. The dependence of V_1 on d runs through the full conditional law of $A^* \mid A$, not only its conditional mean, and is therefore estimator-specific. For linear base estimators, the constant offset ρd cancels in Var_A and V_1 is d-free. For the non-linear base estimators considered in this paper (e.g., NS), no universal closed-form rule is available; we therefore rely on empirical sensitivity analysis. The variance term V_2 enters MSE only through V_2/B and can be made negligible by choosing B moderately large, so the impact of d on V_2/B can be negligible as long as B is large enough.

Figure 13 reports a sensitivity analysis over $d \in \{0, 0.2, \ldots, 0.8, 1.0\}$. The variations in RMSE are at the 10^{-3} scale. The empirical sensitivity to d is practically negligible. In practice, we therefore recommend setting d to a value around the network density. This choice is motivated by the aforementioned surrogate analysis of minimizing $\|P^* - P\|_F^2$, and is supported by the empirical evidence above, which shows that performance is robust to small deviations from this default.

5.3 Choice of ρ

Impact of ρ on E_1. The Equation (12) in the d analysis shows that the masking rate ρ enters E_1 through two channels. (1) The explicit prefactor $(1-\rho)^{-2}$ arises from the debiasing step in (6) and amplifies any residual bias between $\hat{P}(A^*)$ and P^*. (2) In addition, the bias at the perturbed target, $\|\mathbb{E}\{\hat{P}(A^*)\} - P^*\|_F^2$, depends on how far P^* lies from P: $\|P^* - P\|_F^2 = \rho^2 \|d\,\mathbf{1}\mathbf{1}^\top - P\|_F^2$ grows quadratically in ρ, and under the surrogate heuristic introduced in the d analysis, larger $\|P^* - P\|_F^2$ is associated with larger bias at the perturbed target. Both channels push toward smaller ρ, and from E_1 alone the optimum is $\rho = 0$.

Impact of ρ on V_1 and V_2. The dependence of $V_1 = \sum_{i,j} \mathrm{Var}_A(\bar{P}_{\rho,d,ij}(A))$ on ρ involves competing forces. On one hand, the debiasing step in (6) contributes a multiplicative factor of $(1-\rho)^{-2}$ to each entrywise variance, which inflates V_1 as ρ grows. On the other hand, the per-edge coupling between A^* and A satisfies $\mathrm{Cov}(a^*_{ij}, a_{ij}) = (1-\rho)\, p_{ij}(1-p_{ij})$, so larger ρ weakens this coupling. For unstable nonlinear estimators such as NS, weaker coupling can smooth threshold-based selection decisions across perturbations and thereby reduce the variance contribution from the empirical neighborhood selector. This is the variance-reduction mechanism that motivates Net-Paging in the first place. The magnitude of this effect is estimator-specific, and the covariance identity alone does not imply a closed-form optimizer in ρ. The effect of ρ on V_1 is therefore a trade-off between bias inflation and selector-stabilization, and the balance depends on the specific base estimator. The variance term V_2 enters MSE only through V_2/B and can be made negligible by choosing B moderately large, so it does not serve as the primary criterion for selecting ρ.

Because E_1 favors $\rho = 0$ while the variance-reduction mechanism requires $\rho > 0$, we determine the ρ empirically. Supplementary Figure 14 reports the RMSE of Net-Paging + NS across eight graphons for masking rates $\rho \in \{0.01, 0.05, 0.10, \ldots, 0.70\}$. While Net-Paging + NS uniformly improves over the NS baseline, the optimal ρ is graphon-dependent: Graphons 1–4 exhibit a broad U-shape minimized around $\rho \in [0.1, 0.4]$, Graphons 5, 7, and 8 degrade as ρ grows, and only Graphon 6 favors larger values. We therefore recommend $\rho = 0.1$ as a robust default, since it lies near the minimum of the U-shaped curves while avoiding the substantial degradation observed at larger ρ on the monotonically increasing graphons.

6 Simulation Studies

In this section, we evaluate the empirical performance of our method on extensive synthetic data. All experiments are implemented on a machine with a 48-core CPU and 184 GB of RAM.

6.1 Simulation Setup

We generate networks using eight different graphon functions that are widely used in the literature (Zhang et al., 2017; Qin et al., 2021). The analytic forms are listed in the supplementary material. Figure 4 visualizes the corresponding heatmap of a 100×100 probability matrix P generated by each graphon function. High values in P are highlighted in yellow, while low values are colored in dark blue. Under each graphon setting, we generate networks of different sizes to

investigate the asymptotic performance, i.e., we vary the number of nodes $n \in \{50, 150, \ldots, 550\}$. As previously discussed, we set d as the network density, and set $\rho = 0.1$.

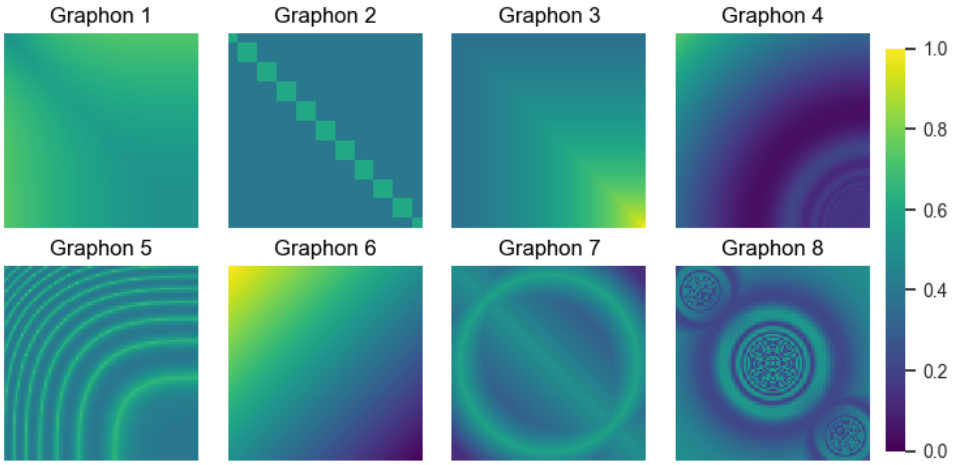

Figure 4: Heatmaps of eight 100×100 graphon probability matrices, where warm color represents high value, and cool color represents low value.

We compare Net-Paging with its unbagged counterparts, including NS (Zhang et al., 2017) and iterative connecting probability estimation (ICE) (Qin et al., 2021). When implementing NS and ICE, we use their default hyperparameter settings. When implementing Net-Paging, we set the number of perturbations B to the same value as the number of nodes, balancing the tradeoff between accuracy and computational time. Regarding the settings of d and ρ, we set d equal to the network density and set $\rho = 0.1$. The rationale for these choices is detailed in Section 3. All the experiments are based on 100 replications by independently generating the adjacency matrices.

6.2 Simulation Results

RMSE reduction. We begin by comparing the root mean squared error (RMSE) of the base estimators with their corresponding Net-Paging versions. Specifically, given a true probability matrix P, we generate K independent adjacency matrices $A^{(k)}$, $k = 1, \ldots, K$. For each $A^{(k)}$, we then apply a graphon estimation method to obtain the estimated probability matrix $\hat{P}^{(k)}$. The RMSE across the K replications is defined as

$$\text{RMSE}(\hat{P}) := \left(\frac{1}{K} \sum_{k=1}^{K} \frac{1}{n^2} ||\hat{P}^{(k)} - P||_F^2 \right)^{1/2}.$$

In this paper, we set $K = 100$. It is crucial to notice that K is different from B used within the Net-Paging method. The hyperparameter B refers to the number of perturbed networks used in Net-Paging, within a single replication to improve estimation. In contrast, K is the number of independent replications of the entire estimation process, each using a newly generated adjacency matrix. This repeated sampling allows us to approximate the expected estimation error across multiple independent realizations of the network.

Figure 5 depicts the RMSE of different methods across various network sizes, ranging from

50 to 550 nodes, comparing four methods: Net-Paging + NS (red solid line), NS (red dashed line), Net-Paging + ICE (blue solid line), and ICE (blue dashed line). From Figure 5, we have these observations. (1) Decreasing RMSE with larger networks: The RMSE for all methods consistently decreases as the number of nodes increases, indicating better performance in larger networks. (2) Effectiveness of Net-Paging: The application of Net-Paging reduces the RMSE for both the NS and ICE methods. This is evident from the lower values of the solid lines (which represent the Net-Paging estimators) compared to the corresponding original estimators (depicted by the dashed lines). (3) Greater impact in smaller networks: The benefit of Net-Paging is more pronounced in smaller networks. (4) Superiority of ICE combined with Net-Paging: The combination of ICE with Net-Paging consistently achieves the lowest RMSE across all scenarios, as indicated by the lowest solid blue lines in each graph. This suggests that ICE, which may have a higher variance, benefits the most from Net-Paging's variance reduction capability.

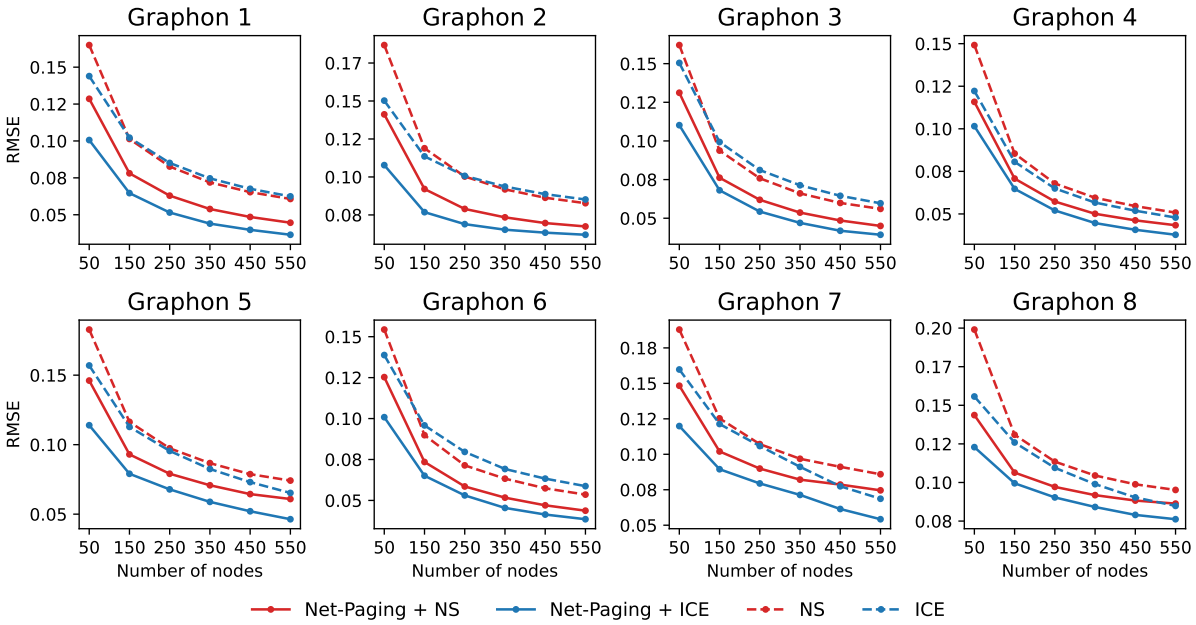

Figure 5: The line chart of RMSE, as the number of nodes increases from 50 to 550. Net-Paging+NS method (red solid line) is the Net-Paging version of the NS method (red dashed line). Net-Paging+ICE method (blue solid line) is the Net-Paging version of the ICE method (blue dashed line).

Variance reduction. To further investigate the source of this improvement, we now analyze the two critical components of MSE: variance and bias. We first compare the variance of the base estimators and their Net-Paging version. The variance of the estimator is computed as

$$\mathrm{Var}(\hat{P}) := \frac{1}{K-1}\sum_{k=1}^{K}\frac{1}{n^2}||\hat{P}^{(k)} - \bar{P}||_F^2,$$

where $\bar{P} = \frac{1}{K}\sum_{k=1}^{K}\hat{P}^{(k)}$ is the sample mean across K replications. Figure 6 presents the variance comparison across eight different graphon settings. From Figure 6, we have similar observations as in the RMSE analysis. Across all graphons, the variance associated with the Net-Paging method (solid lines) is consistently lower than that of the corresponding base estimators (dashed

lines). This suggests that the Net-Paging method provides more reliable estimates with reduced variability. In addition, the variance for all methods decreases as the number of nodes increases. In smaller networks, the variance reduction offered by Net-Paging is more pronounced. In larger networks, the variance of both the base estimators and the Net-Paging versions tends to stabilize, leading to a smaller relative improvement from the Net-Paging method. These results further confirm that Net-Paging significantly improves the robustness of graphon estimation by mitigating the inherent variance in traditional methods.

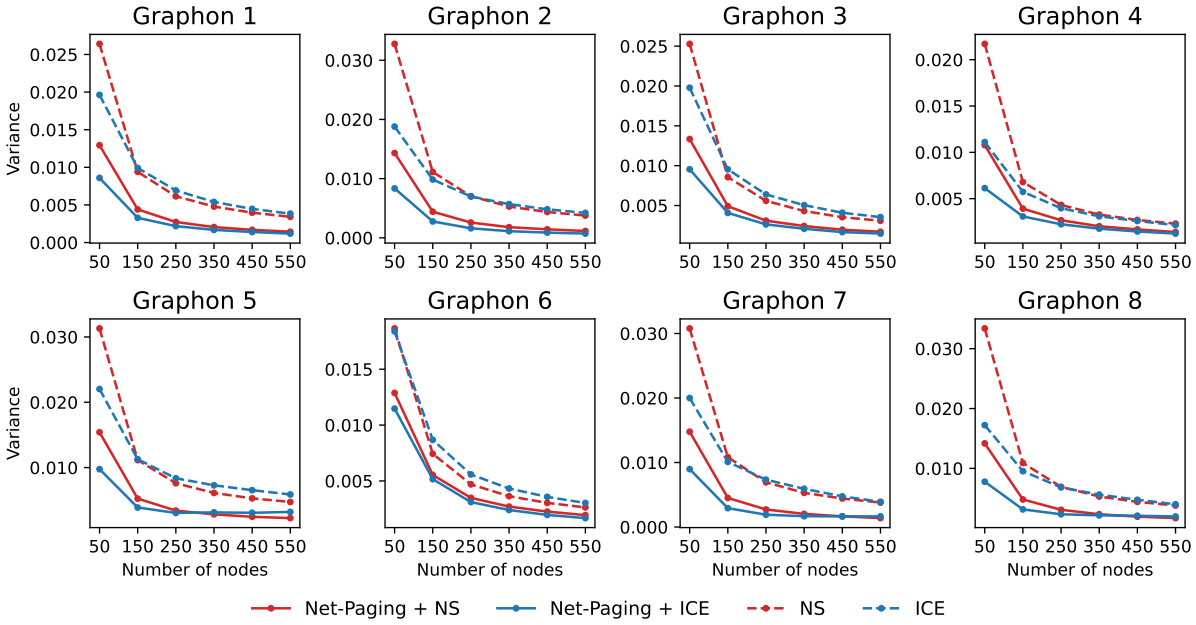

Figure 6: The line chart of variance, as the number of nodes increases from 50 to 550. Net-Paging+NS method (red solid line) is the Net-Paging version of the NS method (red dashed line). Net-Paging+ICE method (blue solid line) is the Net-Paging version of the ICE method (blue dashed line).

Bias performance. Figure 7 presents the squared bias performance for different graphon structures as the number of nodes increases. Across all graphons, the bias decreases as the network size grows, indicating improved estimator accuracy with more data. The Net-Paging variants exhibit bias levels comparable to their base counterparts, as shown by the similarity between the solid and dashed lines. This suggests that Net-Paging does not introduce additional bias while providing variance reduction. Therefore, the reduction in MSE observed with Net-Paging is primarily driven by the decrease in variance, rather than by bias reduction.

7 Real Data

For real networks, since the true probability matrix P is unknown, it is infeasible to calculate the MSE, or the sample variance because we only have one observed network. Thus the evaluation metrics used in the simulation studies are inapplicable in real-world applications. We thus follow the procedure in existing literature (Zhang et al., 2017; Qin et al., 2021) to assess the practical utility of Net-Paging in downstream analysis, i.e., link prediction (Zhang and Chen, 2018; Ma

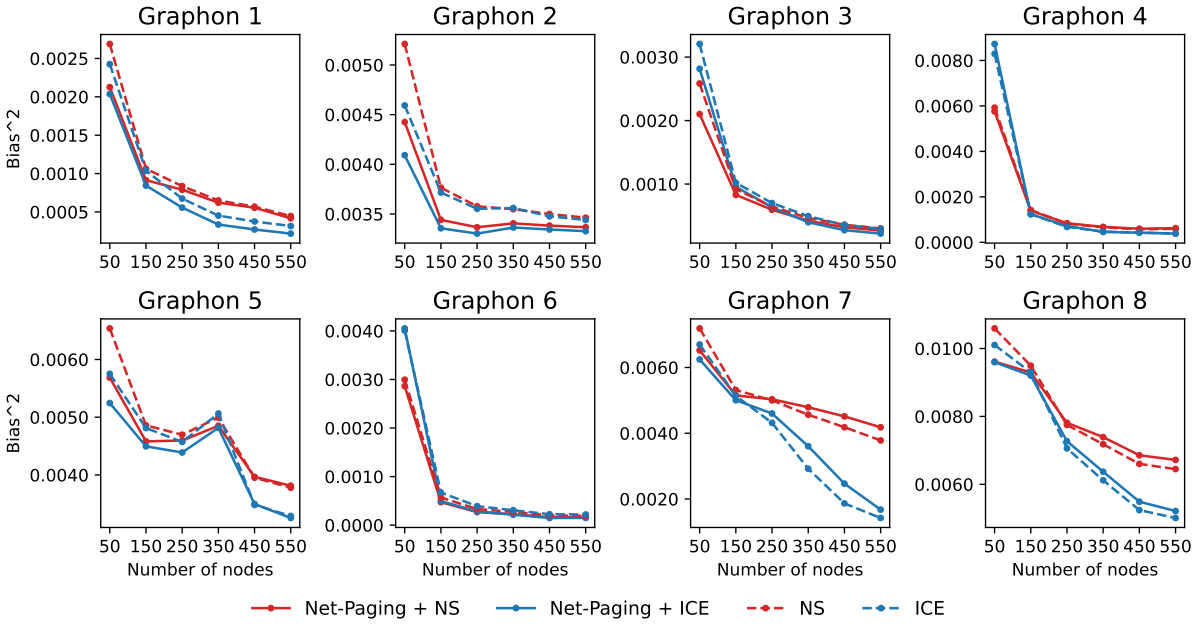

Figure 7: The line chart of squared bias, as the number of nodes increases from 50 to 550. Net-Paging+NS method (red solid line) is the Net-Paging version of the NS method (red dashed line). Net-Paging+ICE method (blue solid line) is the Net-Paging version of the ICE method (blue dashed line).

et al., 2024; Li et al., 2023). Additionally, we extend the evaluation to graph classification tasks (Sabanayagam et al., 2022; Duan et al., 2024), following the methodology proposed by (Han et al., 2022a), which utilizes graphon-based augmentation to enhance graph classification performance.

7.1 Link Prediction

To evaluate the utility of Net-Paging in link prediction for real networks, following Zhang et al. (2017); Qin et al. (2021), we employed a semi-supervised approach that simulates realistic scenarios where network data is incomplete. In this experimental setup, we assume access to a partially observed network and assess how well different methods can predict the missing connections.

Experimental Design. We begin with a complete adjacency matrix A representing the true network structure, which we treat as unobserved ground truth. To simulate realistic data collection scenarios where some connections may be missed or unrecorded, we create a partially observed version A_{miss} by randomly removing edges from the original network. This missingness process is modeled as $A_{\text{miss}} = M \odot A$, where $\odot$ denotes element-wise multiplication and $M = (M_{ij})$ is a binary mask matrix. Each element $M_{ij} \overset{i.i.d.}{\sim} \text{Ber}(1-p)$ and p is the missing rate. The estimation $\hat{P}(A_{\text{miss}})$ is based on the observed adjacency matrix A_{miss}.

Evaluation Metrics. We assess link prediction performance using standard binary classification metrics. For any threshold t, we define the true positive rate and false positive rate

as:

$$r_{\mathbf{TP}}(t) := \frac{\sum_{i,j} \mathbb{I}\left(\hat{P}_{ij}(A_{\text{miss}}) > t, A_{ij} = 1, M_{ij} = 0\right)}{\sum_{i,j} \mathbb{I}\left(A_{ij} = 1, M_{ij} = 0\right)},$$
$$r_{\mathbf{FP}}(t) := \frac{\sum_{i,j} \mathbb{I}\left(\hat{P}_{ij}(A_{\text{miss}}) > t, A_{ij} = 0, M_{ij} = 0\right)}{\sum_{i,j} \mathbb{I}\left(A_{ij} = 0, M_{ij} = 0\right)}.$$

With varying threshold $0 \leq t \leq 1$, we calculate the area under the receiver operating characteristic curve (AUC) as our evaluation metric.

Datasets and Baselines. We evaluated the performance on three widely used real networks. (1) Macaque cerebral cortex network (Harriger et al., 2012) with 242 nodes and 4090 edges, where each node is an anatomically defined macaque monkey brain region, and an edge describes neural projection between the regions. (2) Political blogs network (Adamic and Glance, 2005) with 1,222 nodes and 16,714 edges, where each node is a blog website, and an edge represents the hyperlink between the blogs. (3) Infectious SocioPatterns dynamic contact networks (Isella et al., 2011) with 410 nodes and 2,765 edges, where each node is a person, and an edge represents physical contact during an event. In addition to the graphon estimation methods (NS, ICE, USVT, SAS), we also include four link prediction methods called Jaccard coefficient (JC, (Liben-Nowell and Kleinberg, 2003)), resource allocation index (RAI, (Zhou et al., 2009)) and preferential attachment score (PAS, (Liben-Nowell and Kleinberg, 2003)) for comparison.

Results. Table 1 displays averaged AUC values over 100 repeated experiments, under various missing rates, p from 0.3 to 0.7. The key observations from the table include: (1) Superiority of Net-Paging-based methods: Across all three datasets and all missing rates, the combination of Net-Paging with NS consistently outperforms NS alone, and similarly, Net-Paging with ICE outperforms ICE alone. This indicates that the proposed Net-Paging framework substantially enhances the predictive power of baseline graphon estimators, even under increasing data sparsity. (2) Comparison with classical link prediction methods: Traditional link prediction heuristics (Jaccard coefficient, resource allocation index, preferential attachment score) consistently underperform relative to graphon-based estimators, especially as the missing rate increases. This further highlights the advantage of model-based approaches in leveraging latent network structure for link prediction. (3) Performance under high missing rates: Most methods show expected performance degradation as the missing rate increases from 0.3 to 0.7. However, the proposed method maintains relatively high performance even at challenging missing rates. For example, in the MACAQUE dataset at a missing rate of 0.7, Net-Paging improves the AUC from 0.744 to 0.838 for NS and from 0.819 to 0.862 for ICE. This demonstrates that NP-based methods are especially robust to information loss—a crucial property for real-world networks where missing data is common. Similarly, in the INFECT dataset at a missing rate of 0.7, the AUC increases from 0.755 to 0.833 for NS, and from 0.843 to 0.885 for ICE when incorporating Net-Paging. These results demonstrate that the proposed method offers significant resilience to information loss, a critical property for real-world networks where missing data is prevalent.

Table 1: AUC values calculated at different missing rates, averaged over 100 repeated experiments. To save the column space, we use NP as an abbreviation of Net-Paging.

Dataset	p	NP+NS	NP+ICE	NS	ICE	USVT	SAS	JC	RAI	PAS
MACAQUE	0.3	**0.919**	0.915	0.910	0.901	0.829	0.790	0.879	0.901	0.801
	0.4	**0.913**	0.910	0.900	0.895	0.774	0.789	0.873	0.897	0.801
	0.5	0.900	**0.902**	0.877	0.880	0.769	0.784	0.863	0.889	0.798
	0.6	0.880	**0.888**	0.831	0.860	0.751	0.779	0.847	0.875	0.795
	0.7	0.838	**0.862**	0.744	0.819	0.500	0.768	0.814	0.845	0.791
BLOG	0.3	**0.949**	0.947	0.931	0.936	0.926	0.905	0.872	0.894	0.908
	0.4	**0.945**	**0.945**	0.924	0.932	0.924	0.903	0.869	0.879	0.907
	0.5	0.939	**0.942**	0.912	0.925	0.867	0.901	0.863	0.867	0.906
	0.6	0.929	**0.936**	0.892	0.916	0.500	0.897	0.854	0.842	0.905
	0.7	0.910	**0.926**	0.852	0.900	0.500	0.891	0.836	0.817	0.902
INFECT	0.3	0.952	**0.954**	0.924	0.941	0.500	0.717	0.941	0.945	0.736
	0.4	0.937	**0.945**	0.902	0.932	0.500	0.710	0.932	0.936	0.732
	0.5	0.915	**0.934**	0.871	0.918	0.500	0.703	0.916	0.920	0.727
	0.6	0.885	**0.919**	0.828	0.893	0.500	0.687	0.892	0.894	0.722
	0.7	0.833	**0.885**	0.755	0.843	0.500	0.668	0.847	0.849	0.715

7.2 Graph Classification

Background. Graph classification is an important task in network analysis, where the objective is to assign a label to an entire graph. The importance of graph classification arises from its broad applicability across diverse scientific domains. For example, in cheminformatics, molecules are naturally represented as graphs where atoms correspond to nodes and chemical bonds to edges. Graph classification enables the prediction of molecular properties such as toxicity, solubility, and biological activity, significantly accelerating drug development pipelines (Gaudelet et al., 2021; Zhao et al., 2024). To improve the accuracy of the graph classification task, Han et al. (2022a) introduced G-Mixup, an innovative data augmentation framework that addresses the common challenge of limited training data by leveraging graphon to generate synthetic training examples. The key insight behind G-Mixup is that graphs from the same class often share similar underlying generating mechanisms, which can be captured by class-specific graphons. The method works by first estimating a graphon for each class using the available training graphs. G-Mixup then generates synthetic graphs by interpolating between these class-specific graphons, creating new training examples that preserve the essential structural characteristics of their respective classes while introducing controlled variations. Notably, the performance of G-Mixup is intrinsically linked to the quality of graphon estimation: more accurate graphon estimates yield synthetic graphs that better reflect the true characteristics of each class, thereby enhancing the effectiveness of data augmentation and, ultimately, the overall classification performance.

Experimental Design. To rigorously assess the effect of graphon estimation methods within the G-Mixup framework for graph classification, we begin by splitting each dataset into separate training (70%) and testing sets (30%), ensuring that all reported classification results are based on graphs that were not seen during either graphon estimation or the G-Mixup augmentation process. For each class in the training set, we estimate a class-specific graphon using various methods, including NS, ICE, USVT, SAS, and their Net-Paging-enhanced versions. This enables a direct evaluation of the impact of Net-Paging on graphon quality and downstream performance. The estimated class-specific graphons using graph attention networks (Veličković

et al., 2018) are then used as the basis for the G-Mixup procedure: synthetic graphs are generated by interpolating between graphons from the same class, thereby augmenting the original training set. A graph classification model is subsequently trained on this augmented dataset. We evaluate classification performance on the untouched testing set using standard metrics such as classification accuracy. All experiments are repeated over multiple random splits to ensure robustness, and results are reported as averages with standard deviations.

Datasets. We evaluate the classification performance of graphon-based G-Mixup augmentation on nine widely used benchmark datasets (Han et al., 2022a): IMDB-B, IMDB-M, PROTEINS, NCI1, D&D, FRANKENSTEIN, COLLAB, REDDIT-B, and REDDIT-M. These datasets encompass a diverse set of domains, including social networks (IMDB-B, IMDB-M, COLLAB, REDDIT-B, REDDIT-M) and bioinformatics (PROTEINS, NCI1, D&D, FRANKENSTEIN). The basic statistics of the datasets are summarized in Table 3. The number of graphs in each dataset ranges from 1,000 (IMDB-B) to 4,999 (REDDIT-M), with COLLAB containing 5,000 graphs, representing one of the largest collections. The classification tasks involve either two or three classes, with REDDIT-M posing a multi-class problem involving five classes. The average number of nodes per graph spans a wide range, from small graphs such as FRANKENSTEIN (16.90 nodes) and IMDB-B (19.77 nodes), to larger networks like REDDIT-M (508.52 nodes) and REDDIT-B (429.63 nodes). Overall, these datasets offer a comprehensive benchmark to evaluate the efficacy of graphon-based augmentation approaches under a wide range of real-world network scenarios.

Results. As presented in Table 2, integrating Net-Paging with baseline graphon estimators yields consistent or improved test accuracy across a broad range of benchmark datasets. The benefits of Net-Paging are particularly pronounced when combined with the NS and ICE estimators. For example, the addition of Net-Paging increases NS accuracy from 64.22% to 67.76% on the PROTEINS dataset and from 62.02% to 64.39% on NCI1. The combination with ICE also achieves the highest or near-highest performance on several datasets, including PROTEINS (68.57%), NCI1 (65.02%), and REDDIT-B (92.62%), while remaining competitive on COLLAB (80.36%) and FRANKENSTEIN (72.14%). These results demonstrate that Net-Paging effectively enhances graphon-based data augmentation.

Table 2: Test accuracy (%) on 8 benchmark datasets, comparing each method with and without Net-Paging (NP). Bold indicates improvement with NP.

Dataset	USVT	USVT+NP	NS	NS+NP	SAS	SAS+NP	ICE	ICE+NP
IMDB-B	71.65 ± 1.83	71.05 ± 2.37	70.55 ± 4.17	**71.15 ± 4.32**	71.25 ± 2.88	**71.45 ± 3.34**	70.55 ± 3.39	**71.85 ± 3.61**
IMDB-M	48.23 ± 1.83	**49.00 ± 1.61**	48.60 ± 1.97	**49.37 ± 2.83**	47.93 ± 2.12	**48.93 ± 1.98**	48.43 ± 2.34	**49.17 ± 1.34**
PROTEINS	66.95 ± 3.03	**67.85 ± 3.75**	64.22 ± 2.10	**67.76 ± 2.27**	68.16 ± 2.62	67.71 ± 2.50	68.25 ± 1.70	**68.57 ± 2.16**
NCI1	62.38 ± 3.78	**65.38 ± 2.16**	62.02 ± 3.53	**64.39 ± 2.28**	63.49 ± 3.71	**64.36 ± 1.87**	63.66 ± 2.62	**65.02 ± 2.01**
D&D	65.17 ± 2.22	**65.42 ± 3.61**	65.00 ± 2.56	**66.02 ± 2.82**	65.17 ± 2.02	**65.30 ± 2.33**	66.23 ± 2.34	65.51 ± 2.64
FRANKENSTEIN	71.85 ± 1.25	71.54 ± 1.10	71.06 ± 1.04	**71.22 ± 0.81**	71.30 ± 0.95	**71.49 ± 1.21**	71.64 ± 1.49	**72.14 ± 1.69**
COLLAB	79.69 ± 1.48	**79.78 ± 1.20**	79.39 ± 1.54	**79.78 ± 1.72**	78.66 ± 2.20	**79.85 ± 1.14**	80.02 ± 1.73	**80.36 ± 1.53**
REDDIT-B	92.00 ± 0.97	91.97 ± 1.71	92.28 ± 1.38	91.43 ± 1.25	92.00 ± 1.02	**92.25 ± 1.32**	91.90 ± 1.43	**92.62 ± 1.31**
REDDIT-M	56.50 ± 1.20	**56.56 ± 1.89**	57.44 ± 1.09	56.72 ± 1.33	56.32 ± 2.16	**57.10 ± 1.58**	55.97 ± 1.04	**56.42 ± 1.65**

8 Discussion

This paper introduces Net-Paging, a network perturbation aggregation procedure that extends the classical bagging framework to network data for improved graphon estimation. Net-Paging

Table 3: Statistics of the datasets used in our experiments.

Property	COLLAB	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M	PROTEINS	D&D	NCI1	FRANKENSTEIN
#Graphs	5,000	1,000	1,500	2,000	4,999	1,113	1,178	4,110	4,337
#Classes	3	2	3	2	5	2	2	2	2
Avg. #Nodes	74.49	19.77	13.00	429.63	508.52	39.06	284.32	29.87	16.90
Avg. #Edges	2457.78	96.53	65.94	497.75	594.87	72.82	715.66	32.30	33.81

introduces controlled perturbations to the observed adjacency matrix while preserving key structural properties of the underlying graphon. By aggregating estimates obtained from multiple perturbed networks, Net-Paging effectively reduces variance and enhances estimation stability. Our theoretical analysis establishes two results: First, we prove that estimator from each perturbed network retains the same convergence rate as the original, preserving key property of the original network; Second, we demonstrate that the more perturbations used in the Net-Paging, the smaller the MSE is, with a linear dependency. Extensive experiments reveal the advantage of Net-Paging in reducing variance and MSE.

Limitations. Despite these advancements, several methodological challenges and open questions remain. First, the perturbation mechanism, while preserving expected graphon properties, operates at the individual edge level and may inadequately capture higher-order network structures such as community organization, clustering patterns, and motif structures that are crucial in many real-world networks. Second, our current aggregation approach employs simple averaging across perturbations, but this choice lacks theoretical optimization and may not achieve optimal bias-variance trade-offs. Third, the framework currently provides only point estimates without principled uncertainty quantification (Huang et al., 2023; Su et al., 2022), limiting its utility for statistical inference applications where confidence assessment is crucial.

Future Research Directions. These limitations naturally motivate several research directions that could substantially advance both the theoretical foundations and practical utility of perturbation-based graphon estimation.

Structure-Preserving Perturbation Mechanisms: Addressing the structural preservation challenge requires developing perturbation schemes that respect higher-order network properties while maintaining sufficient randomization for variance reduction. Potential approaches include community-aware perturbations that preserve block structure, motif-preserving perturbation strategies, or hierarchical perturbation schemes that operate at multiple structural levels. The theoretical challenge lies in characterizing which structural properties can be preserved while maintaining the variance reduction benefits of perturbation aggregation.

Optimal Aggregation: Addressing the aggregation limitation requires developing principled combination schemes tailored to the functional nature of graphon estimation. Alternative approaches such as weighted averaging based on perturbation-specific quality metrics, median-based robust aggregation methods, or adaptive combination schemes could potentially achieve superior performance (Gasparin and Ramdas, 2024). The theoretical challenge involves establishing frameworks for bias-variance optimization in functional estimation settings, potentially bridging ensemble learning theory with functional data analysis.

Perturbation-Based Confidence Intervals: The most significant extension addresses the uncertainty quantification limitation by developing rigorous statistical inference procedures. For

confidence intervals, the Net-Paging framework naturally motivates using the empirical distribution of graphon estimates from multiple perturbed networks to construct entrywise confidence intervals through empirical quantiles. However, the theoretical justification for this approach is non-trivial. The main difficulty arises from the complex dependence structure among perturbations, as well as the inherent dependence structure of network data itself. Rigorously proving that this empirical distribution accurately reflects the true sampling distribution of the graphon estimator, especially for complex non-Euclidean data like networks, is a significant challenge.

Perturbation-Based Hypothesis Testing: The framework also presents opportunities for developing perturbation-based hypothesis testing, such as determining whether two networks originate from the same graphon by comparing the sampling distributions of graphons using the perturbed networks. The key challenge in developing such testing lies in proposing an appropriate test statistic and deriving its asymptotic properties, while ensuring proper Type I error control through leveraging perturbation-based distributional approximations.

References

Lada A. Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, LinkKDD '05, pages 36–43, 2005.

Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. Search in power-law networks. *Physical Review E*, 64(4):046135, 2001.

Edo M Airoldi, Thiago B Costa, and Stanley H Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Rachad Atat, Muhammad Ismail, and Erchin Serpedin. Graphon-based synthetic power system model and its application in system risk analysis. In *2023 IEEE International Smart Cities Conference (ISC2)*, pages 1–6. IEEE, 2023.

Guy Bar-Shalom, Beatrice Bevilacqua, and Haggai Maron. Subgraphormer: Unifying subgraph gnns and graph transformers via graph products. *arXiv preprint arXiv:2402.08450*, 2024.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Peter Bühlmann. Bagging, Boosting and Ensemble Methods. In James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori, editors, *Handbook of Computational Statistics: Concepts and Methods*, pages 985–1022. Springer, Berlin, Heidelberg, 2012.

Peter Bühlmann and Bin Yu. Analyzing bagging. *The Annals of Statistics*, 30(4):927–961, 2002.

T. Tony Cai and Xiaodong Li. Robust and Computationally Feasible Community Detection in the Presence of Arbitrary Outlier Nodes. *The Annals of Statistics*, 43(3):1027–1059, 2015.

Stanley Chan and Edoardo Airoldi. A Consistent Histogram Estimator for Exchangeable Graph Models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 208–216, 2014.

Antoine Channarond, Jean-Jacques Daudin, and Stéphane Robin. Classification and estimation in the Stochastic Blockmodel based on the empirical degrees. *Electronic Journal of Statistics*, 6:2574–2601, 2012.

Sourav Chatterjee. Matrix estimation by Universal Singular Value Thresholding. *The Annals of Statistics*, 43(1), 2015.

Yuzhou Chen, Yulia R Gel, Vyacheslav Lyubchich, and Kusha Nezafati. Snowboot: Bootstrap methods for network inference. *arXiv preprint arXiv:1902.09029*, 2019.

Huimin Cheng, Yongkai Chen, Ping Ma, and Wenxuan Zhong. Graphon cross-validation: Assessing models on network data. In *The Fourteenth International Conference on Learning Representations*, 2026.

Persi Diaconis and Svante Janson. Graph limits and exchangeable random graphs, 2007. arXiv:0712.2749 [math].

Yutai Duan, Jie Liu, Shaowei Chen, Liyi Chen, and Jianhua Wu. G-prompt: Graphon-based prompt tuning for graph classification. *Information Processing & Management*, 61(3):103639, 2024.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. Springer, 1992.

Justin Eldridge, Mikhail Belkin, and Yusu Wang. Graphons, mergeons, and so on! *Advances in Neural Information Processing Systems*, 29, 2016.

Paul Erdős and Alfréd Rényi. On random graphs. i. *Publicationes Mathematicae*, 6:290–297, 1959.

Xinyuan Fan, Feiyan Ma, Chenlei Leng, and Weichi Wu. Low-rank approaches to graphon learning in networks, 2025.

Hsieh Fushing, Chen Chen, Shan-Yu Liu, and Patrice Koehl. Bootstrapping on undirected binary networks via statistical mechanics. *Journal of statistical physics*, 156:823–842, 2014.

Chao Gao and Zongming Ma. Minimax rates in network analysis. *Statistical Science*, 36(1): 16–33, 2021.

Chao Gao, Yu Lu, Zongming Ma, and Harrison H Zhou. Optimal Estimation and Completion of Matrices with Biclustering Structures. *Journal of Machine Learning Research*, 17(1):5602–5630, 2016.

Matteo Gasparin and Aaditya Ramdas. Conformal online model aggregation. *arXiv preprint arXiv:2403.15527*, 2024.

Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B R Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L Blundell, Michael M Bronstein, and Jake P Taylor-King. Utilizing graph machine learning

within drug discovery and development. *Briefings in Bioinformatics*, 22(6):bbab159, 05 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab159.

E. N. Gilbert. Random Graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

Alden Green and Cosma Rohilla Shalizi. Bootstrapping exchangeable random graphs. *Electronic Journal of Statistics*, 16(1):1058–1095, 2022.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pages 8230–8248. PMLR, 2022a.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-Mixup: Graph Data Augmentation for Graph Classification. In *Proceedings of the 39th International Conference on Machine Learning*, pages 8230–8248, 2022b.

Logan Harriger, Martijn P. van den Heuvel, and Olaf Sporns. Rich Club Organization of Macaque Cerebral Cortex and Its Role in Network Communication. *PLOS ONE*, 7(9):e46497, 2012.

Daniel Herbst and Stefanie Jegelka. Higher-order graphon neural networks: Approximation and cut distance. In *The Thirteenth International Conference on Learning Representations*, 2025.

Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–79, 2024.

Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems*, 36:26699–26721, 2023.

Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166–180, 2011.

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

Pierre Latouche, Etienne Birmelé, and Christophe Ambroise. Overlapping stochastic block models with application to the French political blogosphere. *The Annals of Applied Statistics*, 5(1):309–336, 2011.

Keith Levin and Elizaveta Levina. Bootstrapping Networks with Latent Space Structure, 2021. arXiv:1907.10821 [math, stat].

Juanhui Li, Harry Shomer, Haitao Mao, Shenglai Zeng, Yao Ma, Neil Shah, Jiliang Tang, and Dawei Yin. Evaluating graph neural networks for link prediction: Current pitfalls and new benchmarking. *Advances in Neural Information Processing Systems*, 36:3853–3866, 2023.

Shuangning Li and Stefan Wager. Random graph asymptotics for treatment effect estimation under network interference. *The Annals of Statistics*, 50(4):2334–2358, 2022.

Tianxi Li, Elizaveta Levina, and Ji Zhu. Network cross-validation by edge sampling. *Biometrika*, 107(2):257–276, 2020.

David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, CIKM '03, pages 556–559, 2003.

László Lovász and Balázs Szegedy. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

Dean Lusher, Johan Koskinen, and Garry Robins, editors. *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, Cambridge, 2012.

Li Ma, Haoyu Han, Juanhui Li, Harry Shomer, Hui Liu, Xiaofeng Gao, and Jiliang Tang. Mixture of link predictors on graphs. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 16043–16070. Curran Associates, Inc., 2024.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Yichen Qin, Linhan Yu, and Yang Li. Iterative Connecting Probability Estimation for Networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 1155–1166, 2021.

Mahalakshmi Sabanayagam, Leena Chennuru Vankadara, and Debarghya Ghoshdastidar. Graphon based clustering and testing of networks: Algorithms and theory. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=sTNHCrIKDQc.

Benjamin Sischka and Göran Kauermann. Nonparametric two-sample test for networks using joint graphon estimation. *Network Science*, 13:e6, 2025.

Yi Su, Jan Hannig, and Thomas CM Lee. Uncertainty quantification in graphon estimation using generalized fiducial inference. *IEEE Transactions on Signal and Information Processing over Networks*, 8:597–609, 2022.

Tangina Sultana, Md Delowar Hossain, Md Golam Morshed, and Young-Koo Lee. Enhancing link prediction in graph data augmentation through graphon mixup. *Neural Computing and Applications*, 37(8):6267–6282, 2025.

Yifei Sun, Qi Zhu, Yang Yang, Chunping Wang, Tianyu Fan, Jiajun Zhu, and Lei Chen. Fine-tuning graph neural networks by preserving graph generative patterns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9053–9061, 2024.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Shushan Wu, Huimin Cheng, Jiazhang Cai, Ping Ma, and Wenxuan Zhong. Subsampling in large graphs using ricci curvature. In *The Eleventh International Conference on Learning Representations*, 2022.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

Yuan Zhang, Elizaveta Levina, and Ji Zhu. Estimating network edge probabilities by neighbourhood smoothing. *Biometrika*, 104(4):771–783, 2017.

Bangyi Zhao, Weixia Xu, Jihong Guan, and Shuigeng Zhou. Molecular property prediction based on graph structure learning. *Bioinformatics*, 40(5):btae304, 05 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae304.

Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.

Yangze Zhou, Gitta Kutyniok, and Bruno Ribeiro. OOD Link Prediction Generalization Capabilities of Message-Passing GNNs in Larger Test Graphs. *Advances in Neural Information Processing Systems*, 35:20257–20272, 2022.

A Supplemental Material

A.1 The Graphons Used in Simulation Study

$$f_1(x,y) = \frac{1}{1+e^{-\max(x,y)^2+\min(x,y)^4}}$$

$$f_2(x,y) = 0.6\mathbb{I}_{\{\lfloor 10x\rfloor=\lfloor 10y\rfloor\}} + 0.4\mathbb{I}_{\{\lfloor 10x\rfloor\neq\lfloor 10y\rfloor\}}$$

$$f_3(x,y) = \exp\left\{-\max(x,y)^{3/4}\right\}$$

$$f_4(x,y) = \frac{x^2+y^2}{3}\cos\left(\frac{1}{x^2+y^2}\right) + 0.15$$

$$f_5(x,y) = \frac{1}{1.5+|\cos(-20(x^3+y^3))|}$$

$$f_6(x,y) = \frac{x+y}{2}$$

$$f_7(x,y) = 1/\left(1+\exp\left(10\min\left(\frac{|x-y|}{5}, \max\left(0.18-(x-\frac{1}{2})^2-(y-\frac{1}{2})^2, (x-\frac{1}{2})^2+(y-\frac{1}{2})^2-0.245\right)\right)\right)\right)$$

$$\begin{aligned} f_8(x,y) = \frac{1}{5}\min\Bigg(& e^{0.05-(x-\frac{1}{2})^2-(y-\frac{1}{2})^2}\cos\left(\frac{0.4}{(x-\frac{1}{2})^2+(y-\frac{1}{2})^2}\right), \\ & e^{0.0004-(x-\frac{1}{8})^2-(y-\frac{1}{8})^2}\cos\left(\frac{0.1}{(x-\frac{1}{8})^2+(y-\frac{1}{8})^2}\right), \\ & e^{0.0004-(x-\frac{7}{8})^2-(y-\frac{7}{8})^2}\cos\left(\frac{0.1}{(x-\frac{7}{8})^2+(y-\frac{7}{8})^2}\right)\Bigg) + 0.4 \end{aligned}$$

A.2 Algorithm Implementation and Data Source

We implement the Net-Bagging algorithm in python. The implementations of the base estimators and the link prediction algorithms can be found in the following repositories:

- NS: https://github.com/yzhanghf/NeighborhoodSmoothing.
- ICE: https://github.com/Siva-47/ICE.
- USVT: https://github.com/cran/graphon.
- SAS: https://github.com/airoldilab/SAS.
- JC, RAI, PAS & CCPA: https://github.com/networkx/networkx.
- $\mathcal{G}$-mixup https://github.com/ahxt/g-mixup

The Macaque Cerebral Cortex network can be found at https://neurodata.io/. The Political Blogs network is contained in the `R` package `gsbm`. The two networks can be directly downloaded at:

- Macaque Cerebral Cortex: https://s3.amazonaws.com/connectome-graphs/macaque/rhesus_brain_1.graphml
- Political Blogs: https://github.com/cran/gsbm/blob/master/data/blogosphere.RData

A.3 An Attempted Conditional Analysis of Variance Reduction of Net-Paging

In this section, we examine whether the variance reduction achieved by Net-Paging can be justified through a direct conditional analysis. The goal is not to present a complete estimator-specific optimality theorem for all base estimators. Rather, we present a natural proof strategy, identify where it is successful, and clarify why extending it to neighborhood smoothing is technically challenging.

A.3.1 Variance reduction for sensitive estimators

The basic idea is to view Net-Paging as a bagging procedure. If the individual perturbed estimators remain accurate and are sufficiently different from one another, then averaging them should reduce the final estimation error. This intuition is made precise by the following deterministic identity.

Lemma 1. *Let $\hat{P}_{\text{pag}} = \frac{1}{B}\sum_{b=1}^{B} \tilde{P}(A_b^*)$ be the aggregated estimator. For any matrix P,*

$$n^{-2}\|\hat{P}_{\text{pag}} - P\|_F^2 = \frac{n^{-2}}{B}\sum_{b=1}^{B}\|\tilde{P}(A_b^*) - P\|_F^2 - \frac{n^{-2}}{2B^2}\sum_{b_1,b_2=1}^{B}\|\tilde{P}(A_{b_1}^*) - \tilde{P}(A_{b_2}^*)\|_F^2. \tag{13}$$

Proof. Let $\bar{Q} = \hat{P}_{\text{pag}} = \frac{1}{B}\sum_b \tilde{P}(A_b^*)$. The left-hand side equals

$$\begin{aligned}
&\|\bar{Q} - P\|_F^2 + \frac{1}{2B^2}\sum_{b_1,b_2}\|\tilde{P}(A_{b_1}^*) - \tilde{P}(A_{b_2}^*)\|_F^2 \\
&= \|\bar{Q}\|_F^2 - 2\langle \bar{Q}, P\rangle_F + \|P\|_F^2 + \frac{1}{B}\sum_b \|\tilde{P}(A_b^*)\|_F^2 - \|\bar{Q}\|_F^2 \\
&= \frac{1}{B}\sum_b (\|\tilde{P}(A_b^*)\|_F^2 - 2\langle \tilde{P}(A_b^*), P\rangle_F + \|P\|_F^2) = \frac{1}{B}\sum_b \|\tilde{P}(A_b^*) - P\|_F^2,
\end{aligned}$$

where we used $\frac{1}{2B^2}\sum_{b_1,b_2}\|\tilde{P}(A_{b_1}^*) - \tilde{P}(A_{b_2}^*)\|_F^2 = \frac{1}{B}\sum_b \|\tilde{P}(A_b^*)\|_F^2 - \|\bar{Q}\|_F^2$. □

Lemma 1 shows that the error of the aggregated estimator is the average individual error minus a pairwise diversity term. Thus, a direct proof of variance reduction requires three steps:

1. showing that each individual perturbed estimator $\tilde{P}(A_b^*)$ remains accurate (Theorem 1);
2. showing that the perturbed networks $A_1^*, \ldots, A_B^*$ are mutually separated (Lemmas 2);
3. showing that separation at the network level propagates to separation at the estimator-output level (Assumption 3).

The first step follows from Theorem 1, using a union bound over B perturbation replicates. The second step follows from the construction of the perturbed networks, as we will show in the following Lemmas 2. The third step is the difficult one: it requires the base estimator to respond sufficiently strongly to changes in the observed adjacency matrix.

Lemma 2 (Perturbed Network Diversity)**.** *Let A_1^* and A_2^* be two perturbed networks generated with disjoint masked sets $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ and masking rate ρ. For any $\gamma \in (0, 2d(1-d)]$,for*

$B = \lfloor 1/\rho \rfloor$ perturbed networks with pairwise disjoint masked sets,

$$\mathbb{P}\left\{ \min_{1 \le b_1 \ne b_2 \le B} \tfrac{1}{2n^2} \|A^*_{b_1} - A^*_{b_2}\|_F^2 > \gamma\rho \right\} \;>\; 1 - \rho^{-2} e^{-n^2\rho[2d(1-d)-\gamma]^2}.$$

Lemma 2 confirms that the perturbed networks are mutually well separated with high probability. To convert the network-level diversity established in Lemma 2 into estimator-level diversity, we require an assumption on how the base estimator $\hat{P}$ responds to changes in its input.

Assumption 3 ((α, C_3)-Sensitivity)**.** *There exist a constant $\alpha > 0$ and a positive sequence $C_3(n) > 0$ such that for any two adjacency matrices A_1, A_2,*

$$n^{-2}\|\hat{P}(A_1) - \hat{P}(A_2)\|_F^2 \;\ge\; C_3(n) \cdot \left(n^{-2}\|A_1 - A_2\|_F^2\right)^{\alpha}. \tag{14}$$

Assumption 3 requires that the base estimator is not too insensitive to changes in the observed network. In words, if two input networks differ substantially in normalized Frobenius norm, then their estimated graphons should also differ. The exponent α governs the power-law scaling of the lower bound: $\alpha = 1$ corresponds to a linear lower bound on the estimator response, $\alpha < 1$ to a sub-linear bound that is stronger for small input differences, and $\alpha > 1$ to a super-linear bound that is weaker for small input differences. The constant $C_3(n)$ controls the strength of the bound overall: larger $C_3(n)$ corresponds to stronger estimator sensitivity.

Discussion of Assumption 3. Whether the assumption holds with a useful constant depends on the base estimator. We examine two representative cases.

Block estimators. Consider an estimator that averages edges within each block of a known partition $z : [n] \to [K]$. Writing the estimator as $\hat{P}(A) = W(z)\, A\, W(z)^\top$ where $W(z)$ is the averaging matrix, we have

$$\|\hat{P}(A_1) - \hat{P}(A_2)\|_F^2 = \|W(z)(A_1 - A_2)W(z)^\top\|_F^2 \ge n^{-2}\|A_1 - A_2\|_F^2 \cdot \frac{\min_k n_k^2}{n^2},$$

by the Cauchy–Schwarz inequality, where $n_k = |z^{-1}(k)|$. When block sizes are balanced ($n_k \asymp n/K$), this gives Assumption 3 with $\alpha = 1$ and $C_3(n) = c/K^2$ for a constant $c > 0$.

Neighborhood smoothing. For NS, the situation is more complicated. The NS estimator averages edges within a neighborhood $\mathcal{N}_k(A)$ of size $|\mathcal{N}_k| \approx hn$ for bandwidth $h = C\sqrt{\log n/n}$. If the neighborhoods selected under A_1^* and A_2^* were fixed and identical, then the only difference between the two NS estimates would come from averaging different edge values over the same smoothing sets. In that hypothesized case, a direct calculation gives

$$\mathbb{E}\left[n^{-2}\|\hat{P}_{\mathrm{NS}}(A_1^*) - \hat{P}_{\mathrm{NS}}(A_2^*)\|_F^2\right] \;\approx\; \frac{d(1-d)\,\rho}{hn} \;=\; \frac{d(1-d)\,\rho}{C\sqrt{n\log n}}, \tag{15}$$

while $n^{-2}\|A_1^* - A_2^*\|_F^2 \approx 2d(1-d)\rho$. Hence Assumption 3 holds with $\alpha = 1$ and

$$C_3(n) = \frac{1}{2C\sqrt{n\log n}}. \tag{16}$$

We emphasize that this calculation does not characterize NS in full. Real perturbed networks induce different neighborhoods, and the membership flips create additional output variation that the fixed-neighborhood calculation ignores. The full $C_3(n)$ for NS may therefore be larger, allowing a stronger version of Assumption 3 than the simplified calculation alone establishes. We do not pursue this strengthening here, but its difficulty is itself informative: it shows that even the value of $C_3(n)$ for NS is non-trivial to pin down.

A.3.2 Resulting improvement bound

Theorem 3 shows that, with high probability, the squared error of the bagged estimator is bounded above by the single-replicate error bound $C_1(n)/(1-\rho)^2$ minus a strictly positive improvement term $\Delta(n, B, \rho) := (B-1)C_3(n)(2\gamma\rho)^\alpha/[2B(1-\rho)^2]$. Thus, when the base estimator has strong sensitivity, in the sense that $C_3(n)$ is bounded away from zero or decays slowly, the diversity induced by the perturbations translates into a non-negligible reduction in the error bound.

Theorem 3. *Suppose Assumptions 1, 2, and 3 hold. Let $A_1^*, \ldots, A_B^*$ be perturbed networks generated with pairwise disjoint masked sets, and $\hat{P}_{\text{pag}} = \frac{1}{B}\sum_b \tilde{P}(A_b^*)$ be the Net-Paging estimator. Then for any $\gamma \in (0,\, d(1-d)]$,*

$$\max_{f\in\mathcal{F}} \mathbb{P}\left(n^{-2}\|\hat{P}_{\text{pag}} - P\|_F^2 \;\geq\; \frac{C_1(n)}{(1-\rho)^2} - \frac{(B-1)\,C_3(n)\,(2\gamma\rho)^\alpha}{2B\,(1-\rho)^2} \right) \;\leq\; B\,D_1(n) + \rho^{-2} e^{-n^2\rho[2d(1-d)-\gamma]^2}, \tag{17}$$

where $C_1(n)$, $D_1(n)$, $C_3(n)$, and α are defined in Assumptions 2 and 3.

Proof. By Lemma 1,

$$n^{-2}\|\hat{P}_{\text{pag}} - P\|_F^2 = \frac{1}{B}\sum_b n^{-2}\|\tilde{P}(A_b^*) - P\|_F^2 - \frac{1}{2B^2(1-\rho)^2}\sum_{b_1,b_2} n^{-2}\|\hat{P}(A_{b_1}^*) - \hat{P}(A_{b_2}^*)\|_F^2. \tag{18}$$

By Theorem 1 and $B = \lfloor 1/\rho \rfloor$,

$$\max_{f\in\mathcal{F}} \mathbb{P}\left(\max_{1\leq b\leq B} n^{-2}\|\tilde{P}(A_b^*) - P\|_F^2 \geq \frac{C_1(n)}{(1-\rho)^2} \right) \;\leq\; B\,D_1(n).$$

$\frac{1}{B}\sum_b n^{-2}\|\tilde{P}(A_b^*) - P\|_F^2 \leq \frac{C_1(n)}{(1-\rho)^2}$. Given Assumption 3, we have $n^{-2}\|\hat{P}(A_{b_1}^*) - \hat{P}(A_{b_2}^*)\|_F^2 \geq C_3(n)(n^{-2}\|A_{b_1}^* - A_{b_2}^*\|_F^2)^\alpha$. By Lemma 2, the event $\mathcal{E} = \{\min_{b_1\neq b_2} \frac{1}{2n^2}\|A_{b_1}^* - A_{b_2}^*\|_F^2 > \gamma\rho\}$ satisfies $\mathbb{P}(\mathcal{E}) \geq 1 - \rho^{-2}e^{-n^2\rho[2d(1-d)-\gamma]^2}$. Since $\sum_{b_1,b_2}$ counts $B(B-1)$ ordered pairs with $b_1 \neq b_2$, one event $\mathcal{E}$,

$$\frac{1}{2B^2}\sum_{b_1,b_2} n^{-2}\|\hat{P}(A_{b_1}^*) - \hat{P}(A_{b_2}^*)\|_F^2 \;\geq\; \frac{B-1}{2B}\,C_3(n)\,(2\gamma\rho)^\alpha.$$

Substituting into (18) completes the proof. □

Block estimators. For block estimators with balanced blocks and fixed K, $C_3(n)$ is a positive constant. The improvement $\Delta(n, B, \rho)$ is therefore a constant fraction of the single-replicate error bound $C_1(n)/(1-\rho)^2$, and Theorem 3 delivers a non-vanishing constant-factor reduction.

Neighborhood smoothing. Consider the simplified scenario in which the NS neighborhoods are held fixed across two perturbed networks. In this case, we have $C_3(n) \asymp \frac{1}{\sqrt{n \log n}}$. Substituting this value and $h = C\sqrt{\log n/n}$ into the improvement term in Theorem 3 yields

$$\frac{\Delta(n, B, \rho)}{C_1(n)} \asymp \frac{1}{\log n},$$

up to constants depending on ρ and B, where $C_1(n) \asymp \sqrt{\log n/n}$ is the base NS error rate in Theorem 1. Thus, Theorem 3 gives a constant-improvement bound in finite samples, but it does not establish an improved asymptotic convergence rate for NS.

We emphasize that the fixed-neighborhood simplification is not realistic for actual NS: the dissimilarity-based selection rule depends on the data, so two perturbed networks generally induce different neighborhoods. The variation introduced by these neighborhood differences is the actual source of NS's empirical variance, and it is not captured by the fixed-neighborhood calculation. The bound above therefore underestimates the true sensitivity of NS, and we take its lower-order rate not as a characterization of NS itself but as evidence that Assumption 3 is the wrong instrument for analyzing NS. We discuss this structural mismatch in the next subsection.

In sum, this analysis clarifies the scope of our theoretical claim. Net-Paging should not be interpreted as a procedure that automatically improves the asymptotic rate of every graphon estimator. Under a sensitivity condition, the pairwise diversity induced by perturbation can yield a non-negligible finite-sample improvement term. For NS, however, the simplified fixed-neighborhood calculation yields a lower-order improvement relative to the baseline NS rate and therefore does not establish a rate improvement. The actual improvement mechanism for NS is instead tied to the data-adaptive neighborhood-selection step: perturbations can flip decisions near the selection cutoff, and aggregation can stabilize these boundary-sensitive decisions. A formal rate improvement for NS would require additional margin-type assumptions controlling the mass of near-boundary node pairs and the effect of perturbation on their selection status. We leave such an estimator-specific optimality analysis for future work.

A.4 Proofs

A.4.1 Proof of Theorem 1

Proof. By the definition of graphon, there is a graphon function f, such that $p_{ij} = f(u_i, u_j)$. According to the construction of A^*, the marginal distribution of A^* is $\mathrm{Ber}(P^*)$, where $p^*_{ij} = \rho d + (1-\rho)p_{ij} = \rho d + (1-\rho)f(u_i, u_j)$. Define $f^* := \rho d + (1-\rho)f$. Then A^* is a realization from the graphon model with f^*. By Assumption 1, $f^* \in \mathcal{F}$.

Based on Assumption 2, We have

$$\max_{f^* \in \mathcal{F}} \mathbb{P}\left(\frac{1}{n^2}||\hat{P}(A^*) - [(1-\rho)P + \rho d]||_F^2 \geq C_1(n)\right) \leq D_1(n).$$

Recall that $\tilde{P}(A^*) = \frac{1}{1-\rho}\left(\hat{P}(A^*) - \rho d\right)$. Therefore,

$$\max_{f \in \mathcal{F}} \mathbb{P}\left(\frac{1}{n^2}||\tilde{P}(A^*) - P||_F^2 \geq \frac{C_1(n)}{(1-\rho)^2}\right) \leq D_1(n),$$

which completes the proof. □

A.4.2 Proof of Theorem 2

Proof. Since the base estimator is bounded, by the Fubini theorem, the expectation of the bagging estimator can be calculated as

$$\begin{aligned}\mathbb{E}_{A,\mathcal{R}}\left[\hat{P}_{\text{bag}}(A,\mathcal{R})\right] &= \frac{1}{B}\sum_{b=1}^{B}\mathbb{E}_{A,\mathcal{R}}\left[\tilde{P}(A,\mathcal{R}_b^{(1)},\mathcal{R}_b^{(2)})\right] \\ &= \mathbb{E}_A\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})\right].\end{aligned}$$

Next, we calculate the second order moment of the bagging estimator. Let $\odot$ denote the Hadamard product of matrices, i.e. the element-wise product, and $A^{\odot 2}$ denote the element-wise square of any matrix A. In the following equations, when the operator $\mathbb{E}$ and Var are applied on matrices, they are by default applied on each element of the matrices.

$$\begin{aligned}&\mathbb{E}_{A,\mathcal{R}}\left[\hat{P}_{\text{bag}}(A,\mathcal{R})^{\odot 2}\right] \\ &= \frac{1}{B^2}\mathbb{E}_A\Bigg[\sum_{b=1}^{B}\mathbb{E}_{\mathcal{R}}\left[\tilde{P}(A,\mathcal{R}_b^{(1)},\mathcal{R}_b^{(2)})^{\odot 2}\right] \\ &\qquad + \sum_{b_1\neq b_2}\mathbb{E}_{\mathcal{R}}\left[\tilde{P}(A,\mathcal{R}_{b_1}^{(1)},\mathcal{R}_{b_1}^{(2)})\odot\tilde{P}(A,\mathcal{R}_{b_2}^{(1)},\mathcal{R}_{b_2}^{(2)})\right]\Bigg] \\ &= \mathbb{E}_A\Bigg[\frac{1}{B}\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})^{\odot 2}\right] \\ &\qquad + \frac{B-1}{B}\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})\right]^{\odot 2}\Bigg] \\ &= \mathbb{E}_A\left[\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})\right]^{\odot 2}\right] \\ &\qquad + \frac{1}{B}\mathbb{E}_A\left[\text{Var}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left(\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})\right)\right].\end{aligned}$$

The variance can be calculated as the expectation of square minus the square of expectation.

$$\begin{aligned}&\text{Var}_{A,\boldsymbol{R}}\left(\hat{P}_{\text{bag}}(A,\boldsymbol{R})\right) \\ &= \mathbb{E}_{A,\boldsymbol{R}}\left[\hat{P}_{\text{bag}}(A,\boldsymbol{R})^{\odot 2}\right] - \mathbb{E}_{A,\boldsymbol{R}}\left[\hat{P}_{\text{bag}}(A,\boldsymbol{R})\right]^{\odot 2} \\ &= \text{Var}_A\left(\mathbb{E}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left[\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})\right]\right) \\ &\qquad + \frac{1}{B}\mathbb{E}_A\left[\text{Var}_{\mathcal{R}^{(1)},\mathcal{R}^{(2)}}\left(\tilde{P}(A,\mathcal{R}^{(1)},\mathcal{R}^{(2)})\right)\right]\end{aligned}$$

The MSE is the sum of all the entries' MSE, which can be represented as the sum of all the squared bias and variance.

$$\text{MSE}(\hat{P}_{\text{bag}}) = \mathbb{E}_{A,\mathcal{R}}\left[||\hat{P}_{\text{bag}}(A,\mathcal{R}) - P||_F^2\right] = E_1 + V_1 + \frac{1}{B}V_2.$$

□

A.4.3 Derivation of Eq.(11)

Given,

$$\widehat{P}_{\text{pag}}(A) = \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \left(\widehat{P}(A_b^*) - \rho d \mathbf{1}_n \mathbf{1}_n^\top \right),$$

and $W(P^*)\mathbf{1}_n = \mathbf{1}_n$. Substituting the three-component decomposition $\widehat{P}(A_b^*) = \mathcal{E}_b^{\text{sel},*} + \mathcal{E}_b^{\text{noise},*} + W(P^*)P^*W(P^*)^\top$ into the estimator yields:

$$\begin{aligned}
\widehat{P}_{\text{pag}}(A) &= \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \left(\mathcal{E}_b^{\text{sel},*} + \mathcal{E}_b^{\text{noise},*} + W(P^*)P^*W(P^*)^\top - W(P^*)(\rho d \mathbf{1}_n \mathbf{1}_n^\top) W(P^*)^\top \right) \\
&= \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \mathcal{E}_b^{\text{sel},*} + \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \mathcal{E}_b^{\text{noise},*} + \frac{W(P^*)(P^* - \rho d \mathbf{1}_n \mathbf{1}_n^\top) W(P^*)^\top}{1-\rho}
\end{aligned}$$

Since $P = (1-\rho)^{-1}(P^* - \rho d \mathbf{1}_n \mathbf{1}_n^\top)$, the last term simplifies to $W(P^*)PW(P^*)^\top$. Subtracting P from both sides results in:

$$\widehat{P}_{\text{pag}}(A) - P = \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \mathcal{E}_b^{\text{sel},*} + \frac{1}{B(1-\rho)} \sum_{b=1}^{B} \mathcal{E}_b^{\text{noise},*} + \left(W(P^*)PW(P^*)^\top - P \right)$$

Defining $\mathcal{E}_{\text{pag}}^{\text{bias}} = W(P^*)PW(P^*)^\top - P$, and we have

$$\mathcal{E}_{\text{pag}}^{\text{bias}} = \frac{W(P^*)P^*W(P^*)^\top - \rho d \mathbf{1}_n \mathbf{1}_n^\top}{1-\rho} - \frac{P^* - \rho d \mathbf{1}_n \mathbf{1}_n^\top}{1-\rho} = \frac{W(P^*)P^*W(P^*)^\top - P^*}{1-\rho} = \frac{\mathcal{E}^{\text{bias},*}}{1-\rho}$$

which completes the derivation for Equation (11).

A.4.4 Proof of Proposition 1

Proof. **Step 1: Entrywise decomposition.** For any matrices $X, Y \in \mathbb{R}^{n \times n}$,

$$\|X - Y\|_F^2 = \sum_{i,i'} (X_{ii'} - Y_{ii'})^2.$$

By the convention $H_{ii}(M) = 0$ for all i and all matrices M, the diagonal entries of $\bar{H}_B$ and $H(P)$ vanish. Therefore

$$\|\bar{H}_B - H(P)\|_F^2 = \sum_{i \neq i'} (\bar{H}_{B,ii'} - H_{ii'}(P))^2. \tag{19}$$

Since the sum is finite, conditional expectation given A exchanges with the summation:

$$\mathbb{E}_{\mathcal{R}}\left[\|\bar{H}_B - H(P)\|_F^2 \mid A \right] = \sum_{i \neq i'} \mathbb{E}_{\mathcal{R}}\left[(\bar{H}_{B,ii'} - H_{ii'}(P))^2 \mid A \right]. \tag{20}$$

Step 2: Conditional law of $H_{ii'}(A_b^*)$ given A. Fix an off-diagonal pair (i, i') with $i \neq i'$. For notational simplicity, write

$$h_{ii'} := H_{ii'}(A), \qquad h_{ii'}^0 := H_{ii'}(P), \qquad q_{ii'} := q_{ii'}(A).$$

Recall that $q_{ii'}$ is the conditional probability that the perturbed selector changes relative to the original selector: $q_{ii'} = \mathbb{P}_{\mathcal{R}}\{H_{ii'}(A^*) \neq H_{ii'}(A) \mid A\}$. There are two cases. If $h_{ii'} = 1$, then the perturbed selector equals one precisely when it does not flip. Hence $\mathbb{P}_{\mathcal{R}}\{H_{ii'}(A_b^*) = 1 \mid A\} = 1 - q_{ii'}$. If $h_{ii'} = 0$, then the perturbed selector equals one precisely when it flips. Hence $\mathbb{P}_{\mathcal{R}}\{H_{ii'}(A_b^*) = 1 \mid A\} = q_{ii'}$.

Combining these two cases gives the compact expression

$$H_{ii'}(A_b^*) \mid A \sim \mathrm{Ber}(p_{ii'}), p_{ii'} = h_{ii'}(1 - q_{ii'}) + (1 - h_{ii'})q_{ii'} = h_{ii'} + (1 - 2h_{ii'})q_{ii'}.$$

When $h_{ii'} = 1$, $p_{ii'} = 1 - q_{ii'}$; when $h_{ii'} = 0$, $p_{ii'} = q_{ii'}$. Since $p_{ii'}$ is either $q_{ii'}$ or $1 - q_{ii'}$,

$$p_{ii'}(1 - p_{ii'}) = q_{ii'}(1 - q_{ii'}). \tag{21}$$

Step 3: Conditional mean and variance of $\bar{H}_{B,ii'}$. The perturbed networks $A_1^*, \ldots, A_B^*$ are conditionally i.i.d. given A by construction. Therefore $H_{ii'}(A_1^*), \ldots, H_{ii'}(A_B^*)$ are conditionally i.i.d. Bernoulli random variables with success probability $p_{ii'}$. Hence

$$\mathbb{E}_{\mathcal{R}}[\bar{H}_{B,ii'} \mid A] = p_{ii'} = h_{ii'} + (1 - 2h_{ii'})q_{ii'}, \tag{22}$$

and

$$\mathrm{Var}_{\mathcal{R}}(\bar{H}_{B,ii'} \mid A) = \frac{1}{B}p_{ii'}(1 - p_{ii'}) = \frac{1}{B}q_{ii'}(1 - q_{ii'}). \tag{23}$$

Step 4: Entrywise conditional MSE against $H_{ii'}(P)$. For any random variable X and constant c,

$$\mathbb{E}[(X - c)^2] = \mathrm{Var}(X) + \{\mathbb{E}(X) - c\}^2.$$

Applying this identity conditionally on A, with $X = \bar{H}_{B,ii'}$ and $c = h_{ii'}^0$, gives

$$\mathbb{E}_{\mathcal{R}}\Big[(\bar{H}_{B,ii'} - h_{ii'}^0)^2 \mid A\Big] = \frac{1}{B}q_{ii'}(1 - q_{ii'}) + \Big\{h_{ii'} - h_{ii'}^0 + (1 - 2h_{ii'})q_{ii'}\Big\}^2. \tag{24}$$

We now evaluate the squared-bias term separately on agreement and disagreement pairs.

If $(i, i') \in \mathcal{C}(A)$, then $h_{ii'} = h_{ii'}^0$. Therefore

$$h_{ii'} - h_{ii'}^0 + (1 - 2h_{ii'})q_{ii'} = (1 - 2h_{ii'})q_{ii'}.$$

Since $h_{ii'} \in \{0, 1\}$, $(1 - 2h_{ii'})^2 = 1$, and hence

$$\Big\{h_{ii'} - h_{ii'}^0 + (1 - 2h_{ii'})q_{ii'}\Big\}^2 = q_{ii'}^2.$$

Thus

$$\mathbb{E}_{\mathcal{R}}\Big[(\bar{H}_{B,ii'} - h_{ii'}^0)^2 \mid A\Big] = q_{ii'}^2 + \frac{1}{B}q_{ii'}(1 - q_{ii'}), \qquad (i, i') \in \mathcal{C}(A). \tag{25}$$

If $(i, i') \in \mathcal{D}(A)$, then $h_{ii'} \neq h_{ii'}^0$. Since both are binary, $h_{ii'}^0 = 1 - h_{ii'}$. Hence

$$h_{ii'} - h_{ii'}^0 + (1 - 2h_{ii'})q_{ii'} = h_{ii'} - (1 - h_{ii'}) + (1 - 2h_{ii'})q_{ii'} = (2h_{ii'} - 1)(1 - q_{ii'}).$$

Since $h_{ii'} \in \{0,1\}$, $(2h_{ii'}-1)^2 = 1$, and therefore

$$\left\{h_{ii'} - h^0_{ii'} + (1-2h_{ii'})q_{ii'}\right\}^2 = (1-q_{ii'})^2.$$

Thus

$$\mathbb{E}_{\mathcal{R}}\left[(\bar{H}_{B,ii'} - h^0_{ii'})^2 \mid A\right] = (1-q_{ii'})^2 + \frac{1}{B}q_{ii'}(1-q_{ii'}), \qquad (i,i') \in \mathcal{D}(A). \tag{26}$$

Step 5: Summation over agreement and disagreement pairs. Substituting (25) and (26) into (20), and using the partition $\mathcal{C}(A) \cup \mathcal{D}(A)$ of the off-diagonal pairs, gives

$$\begin{aligned}\mathbb{E}_{\mathcal{R}}\left[\|\bar{H}_B - H(P)\|_F^2 \mid A\right] = &\sum_{(i,i')\in\mathcal{C}(A)} \left\{q^2_{ii'} + \frac{1}{B}q_{ii'}(1-q_{ii'})\right\} \\ &+ \sum_{(i,i')\in\mathcal{D}(A)} \left\{(1-q_{ii'})^2 + \frac{1}{B}q_{ii'}(1-q_{ii'})\right\}.\end{aligned} \tag{27}$$

The squared error of the original selector is

$$\|H(A) - H(P)\|_F^2 = \sum_{i\neq i'}(h_{ii'} - h^0_{ii'})^2 = |\mathcal{D}(A)|.$$

Therefore,

$$\begin{aligned}&\mathbb{E}_{\mathcal{R}}\left[\|\bar{H}_B - H(P)\|_F^2 \mid A\right] - \|H(A) - H(P)\|_F^2 \\ &= \sum_{(i,i')\in\mathcal{C}(A)} \left\{q^2_{ii'} + \frac{1}{B}q_{ii'}(1-q_{ii'})\right\} \\ &\quad + \sum_{(i,i')\in\mathcal{D}(A)} \left\{(1-q_{ii'})^2 + \frac{1}{B}q_{ii'}(1-q_{ii'}) - 1\right\}.\end{aligned} \tag{28}$$

For each $(i,i') \in \mathcal{D}(A)$,

$$(1-q_{ii'})^2 + \frac{1}{B}q_{ii'}(1-q_{ii'}) - 1 = -\left\{2q_{ii'} - q^2_{ii'} - \frac{1}{B}q_{ii'}(1-q_{ii'})\right\}.$$

Substituting this into (28) yields

$$\mathbb{E}_{\mathcal{R}}\left[\|\bar{H}_B - H(P)\|_F^2 \mid A\right] - \|H(A) - H(P)\|_F^2 = \mathrm{Cost}(A;B) - \mathrm{Gain}(A;B),$$

where

$$\mathrm{Cost}(A;B) := \sum_{(i,i')\in\mathcal{C}(A)} \left\{q^2_{ii'} + \frac{1}{B}q_{ii'}(1-q_{ii'})\right\},$$

and

$$\mathrm{Gain}(A;B) := \sum_{(i,i')\in\mathcal{D}(A)} \left\{2q_{ii'} - q^2_{ii'} - \frac{1}{B}q_{ii'}(1-q_{ii'})\right\}.$$

This proves the proposition. □

A.5 Additional Simulation Results

A.5.1 Clipping proportions due to the debiasing step

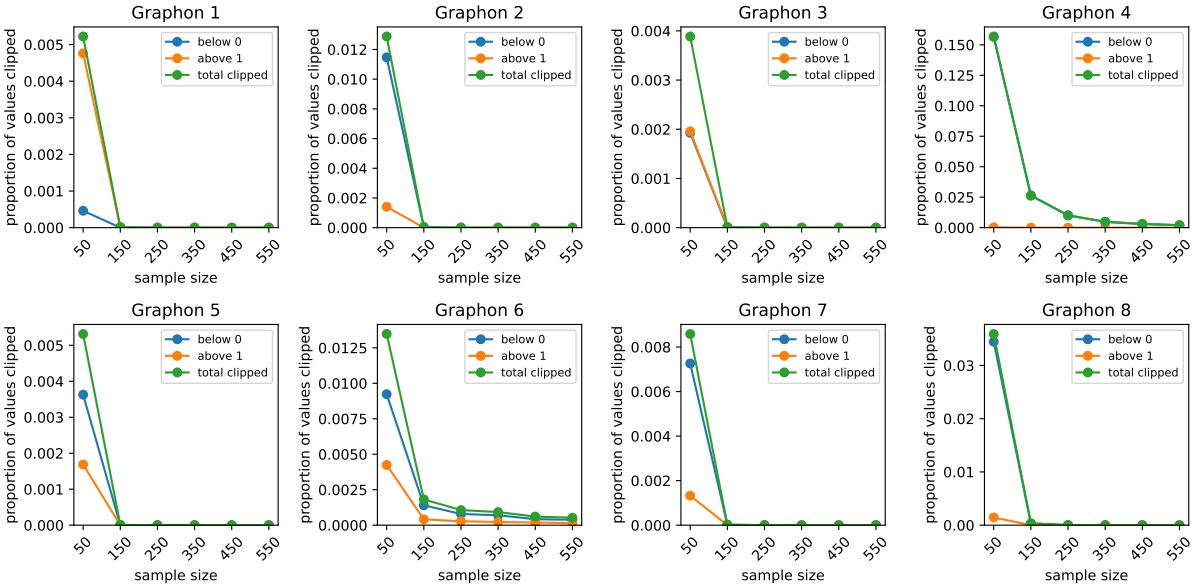

Figure 8: Clipping proportions across different sample sizes. The blue and orange lines denote the proportion of elements below 0 and above 1, respectively. The green line represents their sum, which is the total fraction of values clipped into $[0, 1]$.

A.5.2 Empirical results supporting why Net-Paging improves Neighborhood Smoothing

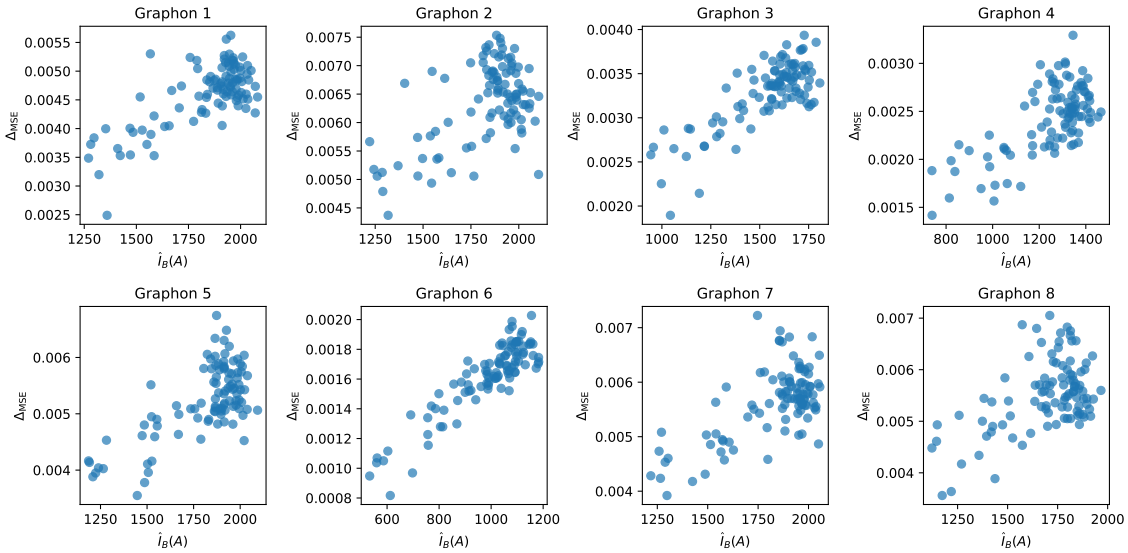

Figure 9: Selector-level improvement is associated with estimator-level MSE reduction. For each graphon setting, each point represents one simulation replicate. The x-axis is the empirical selector-level improvement $\widehat{I}_B(A)$, and the y-axis is the MSE reduction $\Delta_{\mathrm{MSE}} = n^{-2}\|\widehat{P}(A) - P\|_F^2 - n^{-2}\|\widehat{P}_{\mathrm{pag}}(A) - P\|_F^2$. Positive Δ_{MSE} indicates that Net-Paging improves estimation accuracy. The positive association supports the proposed mechanism that stabilizing the neighborhood selector contributes to the final MSE reduction.

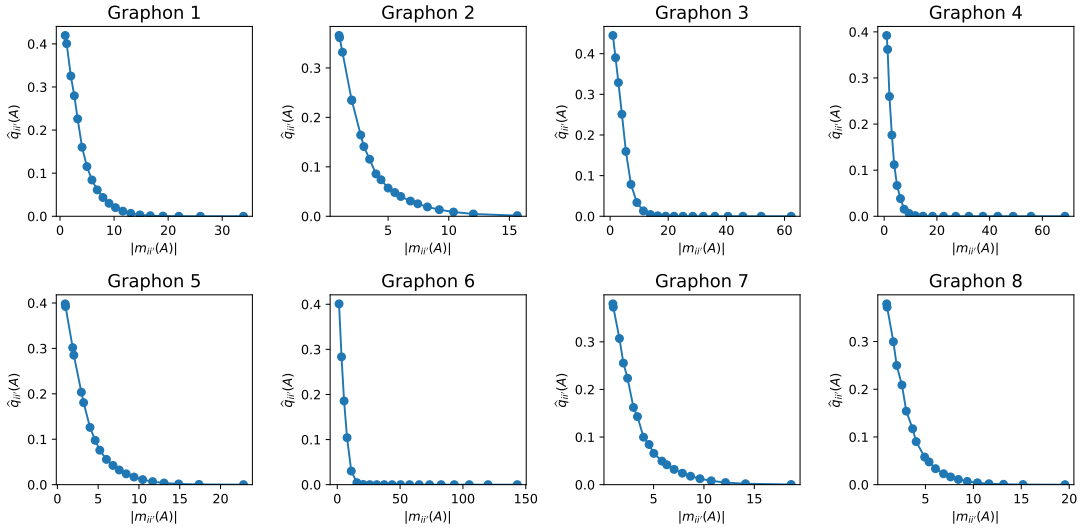

Figure 10: Perturbation flips concentrate near the neighborhood-selection boundary. For each graphon setting, the x-axis is the absolute empirical margin $|m_{ii'}(A)|$, measuring distance from the NS selection cutoff, and the y-axis is the empirical flip frequency $\widehat{q}_{ii'}(A)$ across perturbations. The flip frequency is highest near the cutoff and decreases as pairs move farther from the selection boundary.

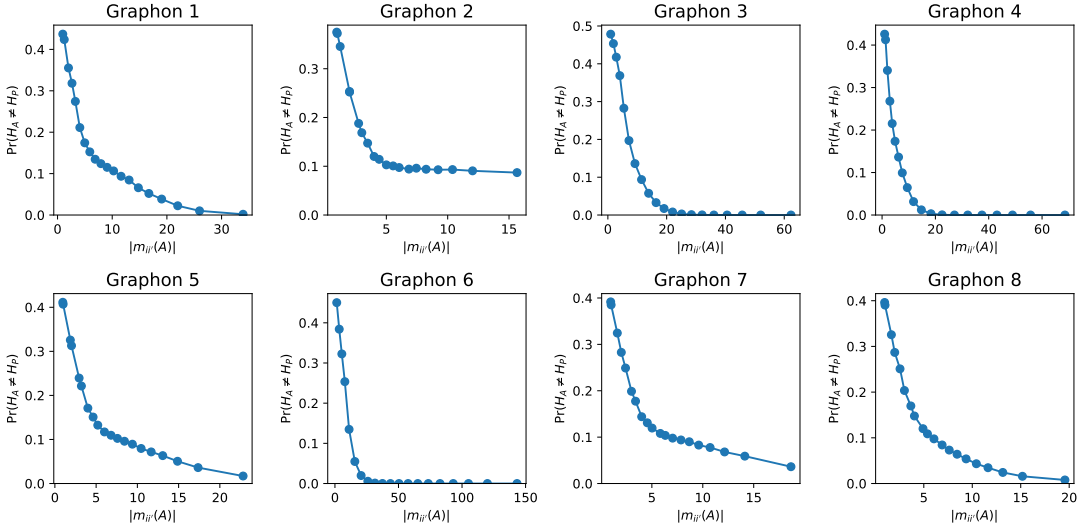

Figure 11: Selection error concentrate near the neighborhood-selection boundary. For each graphon setting, the y-axis is the empirical error rate $\Pr(H_A \neq H_P)$, the fraction of pairs whose selection on the observed adjacency A disagrees with the selection on the underlying probability matrix P. The error rate is highest near the cutoff and decreases as pairs move farther from the selection boundary, confirming that pairs with small margins are the ones most likely to be selected wrongly.

A.5.3 Sensitivity to hyperparameters

Figure 12, Figure 13 and Figure 14 demonstrate the sensitivity to hyperparameters B, d and ρ respectively.

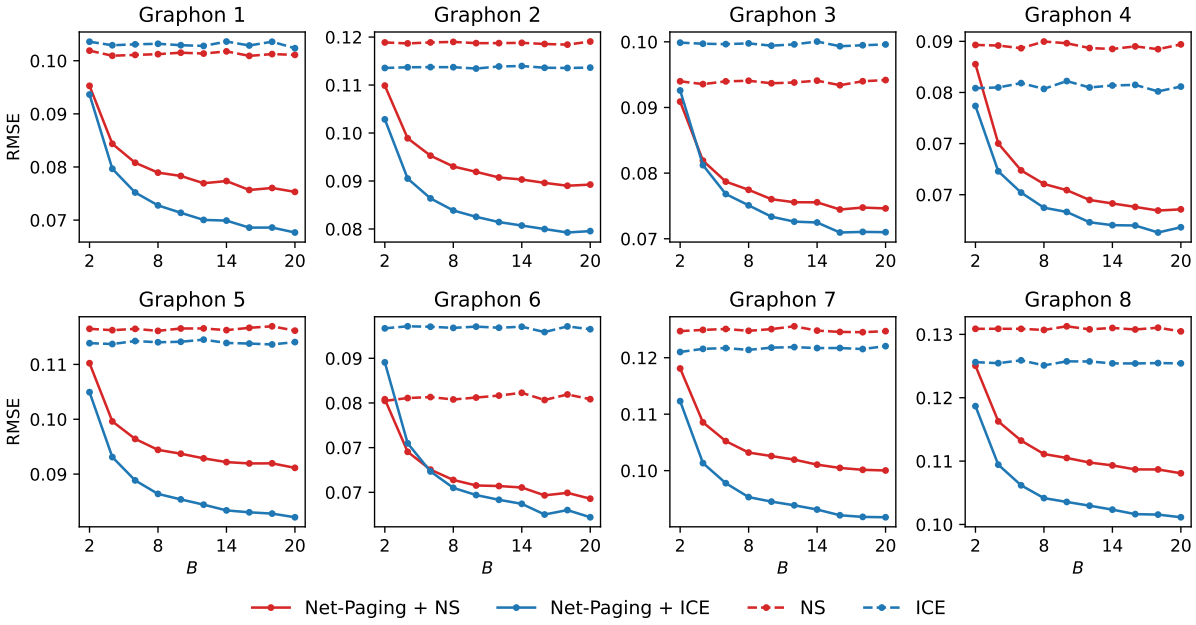

Figure 12: RMSE at different values of B. For Net-Paging, a larger B consistently yields a lower RMSE.

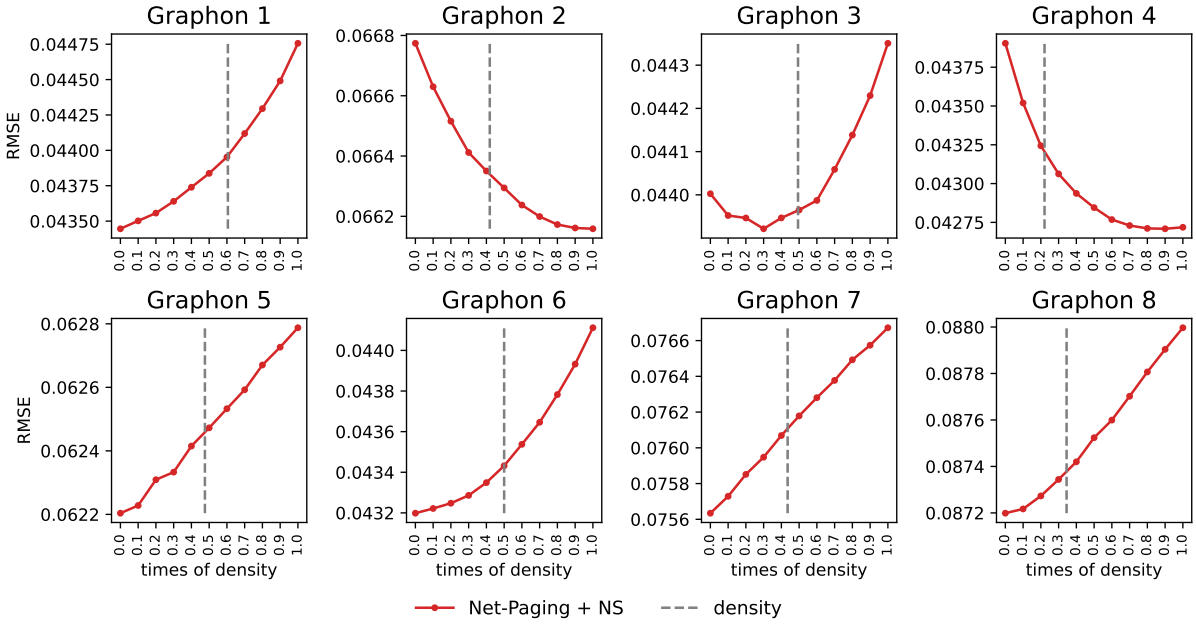

Figure 13: The variation of d has a negligible and pattern-less effect on RMSE. The absolute differences in RMSE are restricted to a marginal scale of 10^{-3}, which is practically insignificant. Furthermore, the absence of a universal trend—ranging from monotonically increasing (e.g., Graphon 1) to decreasing (e.g., Graphon 2) or U-shaped (e.g., Graphon 3)—confirms that these micro-fluctuations are merely data-specific rather than a systemic impact caused by d.

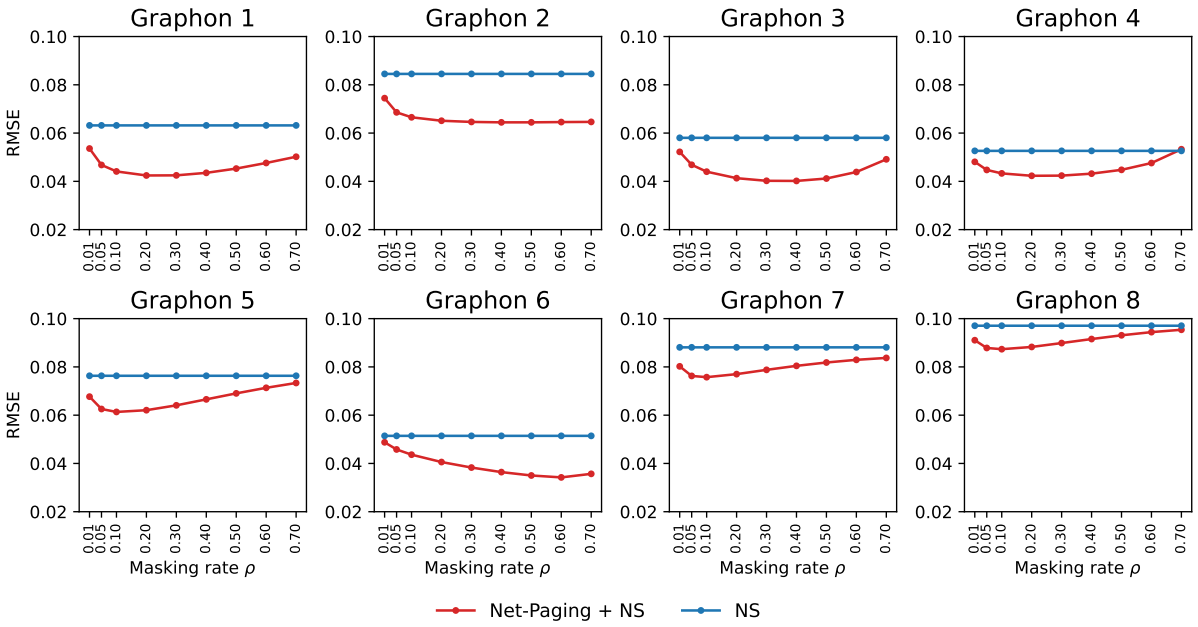

Figure 14: RMSE at different values of the masking rate ρ.

A.5.4 Other kernel versions of Neighborhood Smoothing

The original neighborhood smoothing (NS) estimator can be viewed as a kernel smoother with a box kernel. Specifically, NS assigns equal weight to nodes whose empirical dissimilarity from the target node falls below a bandwidth threshold, and zero weight to all other nodes. In this sense, the standard NS estimator corresponds to a box, uniform kernel. This hard threshold creates a discontinuous neighborhood-selection map: small perturbations of the observed network can cause nodes near the cutoff to enter or leave the smoothing neighborhood.

To examine whether the gain from Net-Paging is specific to this hard selection rule, we also consider kernelized versions of NS using Gaussian and Epanechnikov kernels. The Gaussian kernel assigns smoothly decaying positive weights to all nodes and therefore removes the hard inclusion/exclusion boundary. The Epanechnikov kernel is smoother than the box kernel but remains compactly supported, so it still contains a boundary beyond which weights are zero. Bandwidths are calibrated so that the effective neighborhood size is comparable to that used by the original NS estimator, namely the $\sqrt{\log(n)/n}$ quantile rule in the main text. All other simulation settings, including the graphons, sample sizes, and number of replications, are the same as in Section 6.

If Net-Paging improves NS mainly by stabilizing discontinuous or near-discontinuous neighborhood decisions, then the improvement should be largest for the box kernel, smaller but still visible for compactly supported kernels such as the Epanechnikov kernel, and weakest for fully smooth kernels such as the Gaussian kernel. Thus, these additional experiments assess whether the empirical gains are driven by selector instability.

Gaussian kernel. For each row i, let h_i be the box-kernel bandwidth used by NS, defined as the empirical q_n-quantile of the distances $\{d_{ij}\}$ with $q_n = \sqrt{\log(n)/n}$. The box kernel uniform

on $[0, h_i]$ has second moment $\frac{1}{h_i}\int_0^{h_i} u^2\,du = \frac{h_i^2}{3}$. We set the Gaussian-kernel bandwidth to $\sigma_i = h_i/\sqrt{3}$, matching the second moments of the Gaussian and box kernels for a fair comparison. The Gaussian-kernel weights are

$$w_{ij} \;\propto\; \exp\!\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right) \;=\; \exp\!\left(-\frac{3d_{ij}^2}{2h_i^2}\right),$$

which are strictly positive for every pair and decay smoothly in d_{ij}.

Epanechnikov kernel Using the same row-wise bandwidth h_i, the Epanechnikov-kernel weights are

$$w_{ij} \;\propto\; \left(1 - \frac{d_{ij}^2}{h_i^2}\right) \mathbf{1}\{d_{ij} < h_i\}.$$

where the normalizing constant $3/(4h_i)$ is absorbed by the subsequent row-sum normalization. The indicator introduces a hard cutoff at $d_{ij} = h_i$: pairs just inside the bandwidth contribute, pairs just outside contribute exactly zero, and a small perturbation can flip a pair across the boundary.

Figure 15 compares box, Gaussian, and Epanechnikov versions of NS, with and without Net-Paging. First, the results show a clear ordering in the benefit of Net-Paging. The improvement is largest for the box kernel, smaller for the Epanechnikov kernel, and nearly absent for the Gaussian kernel. This pattern is consistent with the proposed mechanism: Net-Paging is most useful when the base estimator contains an unstable neighborhood-selection step. The box kernel corresponds to the original hard-threshold NS estimator, where small perturbations can move nodes across the inclusion boundary. The Epanechnikov kernel is smoother but remains compactly supported, so it still retains a boundary effect. In contrast, the Gaussian kernel assigns smoothly decaying positive weights to all nodes, and small perturbations produce only gradual changes in the weights. Consequently, the base Gaussian-kernel estimator is already stable, and Net-Paging provides little additional improvement.

Second, the variance and bias decompositions further shows a bias-variance tradeoff. The Gaussian kernel generally has the smallest variance, reflecting the stabilizing effect of smooth weighting, but this variance reduction is often accompanied by larger squared bias. This pattern reflects the standard bias–variance tradeoff in nonparametric kernel regression: smoother kernels reduce variance through wider averaging but incur bias when the target varies on scales finer than the bandwidth. Net-Paging combined with the box kernel achieves competitive RMSE in many settings by reducing the box kernel's variance through perturbation aggregation while preserving its lower bias.

Per-kernel bandwidth tuning. The effective neighborhood size (ENS)-matching rule used above isolates the role of kernel shape by holding the effective neighborhood size fixed across kernels. To assess the robustness of the ordering reported in Figure 15, we re-run the experiment with bandwidths tuned per-kernel and per-graphon. Specifically, we replace the fixed quantile rule $q_n = \sqrt{\log(n)/n}$ with $q_n = c\sqrt{\log(n)/n}$ and, for each kernel and each graphon, tune the constant c by graphon cross-validation (Cheng et al., 2026). The same selected bandwidth is then used for both NS and Net-Paging+NS, so that the comparison within each kernel remains a clean Net-Paging-versus-no-Net-Paging contrast. All other simulation settings are unchanged. Figure 16 reports RMSE, variance, and squared bias under this per-kernel, per-graphon tuning. Three observations emerge.

First, the qualitative ordering of Net-Paging benefit across kernels is preserved, but the magnitudes shrink. Net-Paging continues to deliver the largest RMSE reduction for the box kernel, a smaller reduction for the Epanechnikov kernel, and essentially no improvement for the Gaussian kernel. The persistence of this ordering under per-kernel tuning indicates that the gain from Net-Paging is not an artifact of the ENS-matching choice. Per-kernel tuning narrows but does not close the gap, because bandwidth tuning alone cannot remove the inclusion-boundary instability that Net-Paging is designed to improve.

Second, per-graphon tuning compresses the variance while leaving the squared bias on a similar scale to Figure 15. The variance panels are roughly half the magnitude of those in Figure 15 across the three kernels, and the kernel ordering on variance (Gaussian $<$ Epanechnikov $<$ box) is preserved with smaller separation. The squared-bias panels remain on the same scale and continue to show the Gaussian kernel with the largest bias in most graphons, the same ordering as in Figure 15. This pattern reflects the same bias–variance tradeoff in nonparametric kernel regression as before: the Gaussian kernel's smooth weighting reduces variance through wider averaging but incurs bias when the target varies on scales finer than the bandwidth, and per-kernel bandwidth tuning narrows but does not eliminate this tradeoff.

Taken together, the comparison between Figures 15 and 16 addresses the concern that ENS-matching might be unfair to smoother kernels. Per-kernel, per-graphon tuning narrows the cross-kernel performance gap, as expected, but the ranking of Net-Paging benefit by kernel persist.

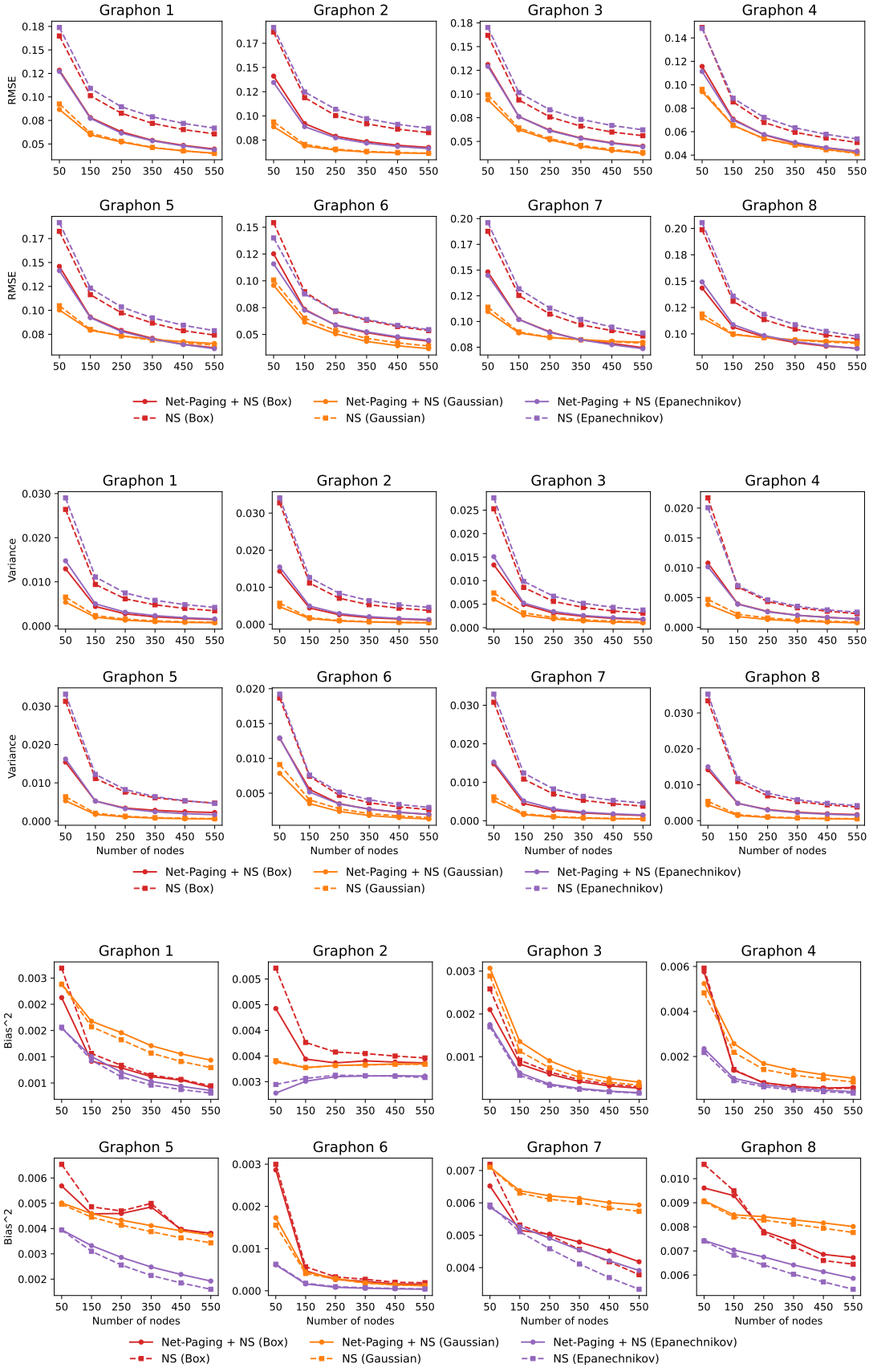

Figure 15: RMSE (top), variance (middle), and squared bias (bottom) for plain NS (dashed lines, square markers) and Net-Paging+NS (solid lines, circle markers), under the box kernel (red), Gaussian kernel (orange), and Epanechnikov kernel (purple), across eight graphon settings. The gap between dashed and solid lines of the same color measures the Net-Paging improvement for that kernel.

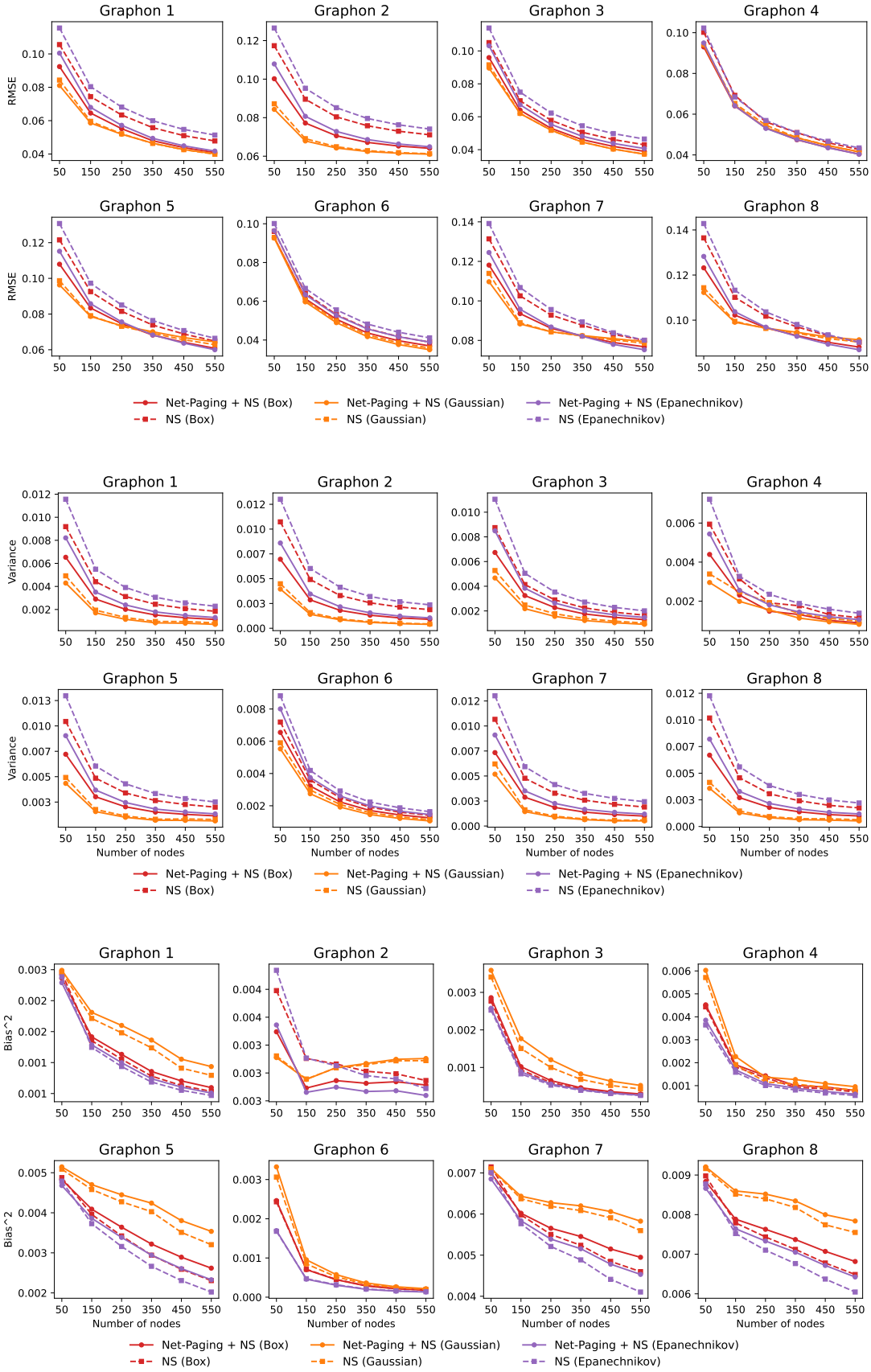

Figure 16: RMSE (top), variance (middle), and squared bias (bottom). The bandwidths are tuned per-kernel, and per-graphon