Low-Complexity and Consistent Graphon Estimation From Multiple Networks

Roland B. Sogan
LPSM, Sorbonne Université, France

Tabea Rebafka
MIA-PS, AgroParisTech, France

Abstract

Recovering the random graph model from an observed collection of networks is known to present significant challenges in the setting, where the networks do not share a common node set and have different sizes. More specifically, the goal is the estimation of the graphon function that parametrizes the nonparametric exchangeable random graph model. Existing methods typically suffer from either limited accuracy or high computational complexity. We introduce a new histogram-based estimator with low algorithmic complexity that achieves high accuracy by jointly aligning the nodes of all graphs, in contrast to most conventional methods that order nodes graph by graph. Consistency results of the proposed graphon estimator are established. A numerical study shows that the proposed estimator outperforms existing methods in terms of accuracy, especially when the dataset comprises only small and variable-size networks. Moreover, the computing time of the new method is considerably shorter than that of other consistent methodologies. Additionally, when applied to a graph neural network classification task, the proposed estimator enables more effective data augmentation, yielding improved performance across diverse real-world datasets.

1 INTRODUCTION

Machine learning research on network-structured data has traditionally focused on the analysis of a single graph (Goldenberg et al., 2010; Kolaczyk, 2009). Increasingly though, modern applications demand methods that can handle collections of networks sharing structural properties, such as in neuroscience, ecology, sociology, and bioinformatics (Fornito et al., 2013; Weber-Zendrera et al., 2019; Poisot et al., 2016). It is therefore more important than ever to develop new statistical approaches that are computationally tractable and capable of jointly leveraging multiple networks to recover the common generative model. Such tools promise to improve performance in any setting where graphs arise under similar structural assumptions.

Graphons provide a powerful nonparametric model for representing complex network structures. Formally, a graphon is a bivariate symmetric measurable function, interpretable both as the limit of dense graph sequences (Lovász and Szegedy, 2006) and as the generative mechanism of exchangeable random graphs (Diaconis and Janson, 2007; Bickel and Chen, 2009). They have recently garnered increased attention in statistics and machine learning (Eldridge et al., 2016; Ruiz et al., 2020, 2023), being used for inferring network topology, comparing graphs, clustering, link prediction, and beyond. Estimating a graphon is challenging in itself and its inherent non-identifiability adds another layer of difficulty. Indeed, distinct graphons can induce the same distribution of observed networks, so one can only recover an equivalence class of functions rather than a unique truth (Borgs et al., 2015; Sischka and Kauermann, 2022). While these issues are challenging for a single network, they are significantly amplified when multiple networks with disjoint node sets of variable size must be analyzed jointly, where alignment and aggregation of information become delicate. Thus, joint graphon estimation from multiple networks is particularly demanding.

There is a long history of graphon inference methods in the literature. Most of them, however, focus exclusively on the single-network setting (Keshavan et al., 2010; Chatterjee, 2015; Airoldi et al., 2013; Chan and Airoldi, 2014; Yang et al., 2014; Wolfe and Olhede, 2013). When it comes to multiple networks, a straightforward strategy is to estimate a graphon sep-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

arately on each graph and then average these individual estimates. However, this approach has major drawbacks: first, we may encounter difficulties related to the lack of identifiability; second, graphon estimators computed on a small network are very biased and have a strong impact on the mean graphon estimator; third, the heterogeneity in network sizes is overlooked, since all graphs are treated equally regardless of scale. Advances such as Smoothed Gromov-Wasserstein Barycenters (GWB) (Xu et al., 2021) align graphs via Gromov–Wasserstein by representing graphons as step functions on a fixed grid, but remain computationally intensive for large graph collections and enforce a rigid resolution. Neural network–based approaches (Xia et al., 2023; Azizpour et al., 2025) introduce more flexibility, yet their training complexity makes them impractical for collections of large or numerous networks. Consequently, the problem of joint graphon estimation from a collection of graphs remains a significant open challenge, particularly demanding methods that are both statistically sound and computationally efficient.

Main contributions Building on these observations, we introduce the **Joint Graph Sorting** (JGS) estimator, a novel framework for graphon inference from collections of unaligned graphs with heterogeneous sizes. JGS extends the classical histogram estimator (Chan and Airoldi, 2014) while maintaining computational efficiency, and overcomes key limitations of existing multi-network approaches. Instead of estimating a graphon separately for each graph, JGS performs a joint alignment of all nodes across networks to construct a single global histogram estimate. This unified approach leads to a more efficient use of information. We establish the consistency of the estimator under mild regularity conditions and validate its practical advantages through extensive numerical experiments. Our main contributions are:

1. We introduce a deterministic, non-iterative joint sorting algorithm that simultaneously infers latent positions and produces a unified graphon estimate from the entire collection of networks.

2. We establish finite-sample guarantees for the latent position estimates and prove consistency for the associated graphon estimator.

3. Through extensive numerical experiments, we demonstrate that JGS consistently outperforms state-of-the-art baselines, particularly for collections of small graphs with heterogeneous sizes. The implementation of JGS and the scripts used to reproduce the experiments are publicly available at

4. We show that JGS executes significantly faster than other consistent methods, making it suitable for large-scale multi-network applications.

Related Works Classical graphon estimators such as Stochastic Blockmodel Approximation (SBA) (Airoldi et al., 2013), Sorting and Smoothing (SAS) (Chan and Airoldi, 2014), and Universal Singular Value Thresholding (USVT) (Chatterjee, 2015) are well established for single networks. The SBA method approximates the graphon by a block (step-function) model, partitioning nodes into communities and estimating connection probabilities between blocks. SAS first sorts nodes by empirical degree and then fits a histogram smoothed by total variation minimization. USVT uses singular value decomposition and applies a universal threshold to reconstruct a low-rank approximation of the adjacency matrix. While effective in the single-graph case, these methods are not designed to jointly align or leverage information across multiple graphs that lack node correspondence.

For collections of unaligned networks, several approaches have been developed. Xu et al. (2021) propose Smoothed Gromov-Wasserstein Barycenters (GWB), which aligns histogram-approximated graphons via the Gromov–Wasserstein distance and obtains the estimate as their barycenter. This strategy accommodates heterogeneous network sizes but is computationally intensive and enforces a common discrete resolution across all graphs. Neural network-based methods offer greater flexibility: Xia et al. (2023) introduce Implicit Graphon Neural Representation (IGNR), an implicit neural representation that models the graphon continuously and leverages Gromov–Wasserstein alignment. Azizpour et al. (2025) propose Scalable Implicit Graphon Learning (SIGL), combining implicit neural representations (Sitzmann et al., 2020) with graph neural networks (Wu et al., 2021) to parametrize the graphon and learn node ordering.

Other approaches model network collections using stochastic block models (SBMs) (Nowicki and Snijders, 2001; Celisse et al., 2012; Abbe, 2018). Chabert-Liddell et al. (2024) extend SBMs to multiple networks by enforcing a shared connectivity structure estimated via variational EM with penalized likelihood for model selection. Rebafka (2024) proposes hierarchical agglomerative clustering based on SBM mixtures, merging networks progressively under integrated classification likelihood while addressing label-switching. Although these methods aim to uncover global latent structures, their computational cost scales poorly with large network collections or large graph sizes.

2 BACKGROUND

Graphon A *graphon* is a symmetric measurable function $W : [0,\,1]^2 \to [0,\,1]$ encoding edge probabilities. To generate a random graph $G = (V, E)$ with n nodes from graphon W, first latent positions are drawn independently as

$$U_1, \ldots, U_n \overset{\text{i.i.d.}}{\sim} \text{Uniform}\,[0,\,1]\,. \tag{1}$$

Then, conditionally on $U_{1:n} = (U_1, \ldots, U_n)$, the adjacency matrix A is generated as

$$\begin{aligned} A_{i,j} \mid U_{1:n} &\sim \text{Bernoulli}\big(W(U_i, U_j)\big) && \text{for } i < j, \\ A_{j,i} &= A_{i,j} && \text{for } i < j, \\ A_{i,i} &= 0 && \text{for all } i. \end{aligned} \tag{2}$$

We denote by $\mathbb{P}_W$ the distribution of the associated random graph. The model is called the *exchangeable random graph model*, a flexible nonparametric framework that subsumes many familiar network models. In particular, the Erdős–Rényi model corresponds to a constant graphon $W(u, v) = p$, while stochastic block models arise from piecewise-constant graphons (Matias and Robin, 2014). A key difficulty is that the mapping $W \mapsto \mathbb{P}_W$ is not injective: different graphons may generate the same distribution of random graphs. This non-identifiability stems from measure-preserving relabelings of the latent space, as explained below.

Graphon identifiability Formally, for any measure–preserving map on the unit interval, the rearranged graphon $W^{\varphi}(u, v) = W(\varphi(u), \varphi(v))$ induces the same distribution of exchangeable graphs as W (Lovász and Szegedy, 2006; Diaconis and Janson, 2007). Thus, graphons are identifiable only up to such transformations φ.

This non-identifiability is closely related to the fact that graph nodes do not possess a natural ordering. Indeed, permuting the labels of the nodes of a graph simply corresponds to permuting the rows and columns of its adjacency matrix, without changing the underlying network structure. In the graphon framework, measure–preserving transformations play the continuous analogue of such permutations: applying φ rearranges the latent positions U_i while leaving the distribution of the generated graphs unchanged.

This phenomenon is similar to the label-switching problem in classical finite mixture models, where there is no natural ordering of the mixture components.

In the literature, a common condition used to obtain an identifiable representation of a graphon is based on the *normalized degree function*, defined by

$$g(u) = \int_0^1 W(u, v)\, dv, \qquad 0 \le u \le 1. \tag{3}$$

Note that $ng(u)$ corresponds to the expected degree of a node with latent position u in a graph with n nodes. A sufficient condition ensuring identifiability of the graphon is the strict monotonicity of the normalized degree function (Bickel and Chen, 2009). This condition selects a canonical representative within the equivalence class of graphons that generate the same distribution of exchangeable graphs.

Condition 2.1 (Strict monotonicity of degrees)**.** *There exists a measure–preserving transformation φ^* such that the normalized degree function g^* associated with W^{φ^*} is strictly increasing on the unit interval.*

Under this condition, the transformation φ^* is unique almost everywhere, which implies that the corresponding graphon W^{φ^*} is uniquely defined. We emphasize that this assumption is restrictive: as shown for instance in Janson (2020), not every graphon admits a measure–preserving transformation for which the degree function becomes strictly increasing. Nevertheless, this condition is widely adopted in the literature since it provides a convenient canonical representation that enables ordering the latent positions of nodes. In our framework, this ordering plays a key role for jointly aligning nodes across networks.

3 JOINT GRAPH SORTING ESTIMATOR

We observe a collection of M networks

$$\mathcal{G} = (G^{(1)}, \ldots, G^{(M)}),$$

where the m-th graph $G^{(m)} = (V^{(m)}, E^{(m)})$ has node set $V^{(m)}$ of size $n^{(m)} = |V^{(m)}|$, edge set $E^{(m)}$ and adjacency matrix $A^{(m)} \in \{0,1\}^{n^{(m)} \times n^{(m)}}$. Every network $G^{(m)}$ is sampled independently from the same underlying graphon W, with latent positions $U_{i,(m)}$ for $i = 1, \ldots, n^{(m)}$. In other words, node sets $V^{(m)}$ are distinct and may have different sizes. Our goal is to estimate W from the observed collection $\mathcal{G}$ by leveraging information across all networks. This section addresses this problem by introducing the new *Joint Graph Sorting (JGS) estimator* and establishing its asymptotic properties.

3.1 Estimation

A natural baseline in the multiple network setting is to apply, to each network, the SAS histogram estimator introduced by Chan and Airoldi (2014), and then average the resulting estimates, as implemented for instance in Xu et al. (2021). A brief description of these single-network graphon estimation methods is provided in the "Related Works" part bellow. However,

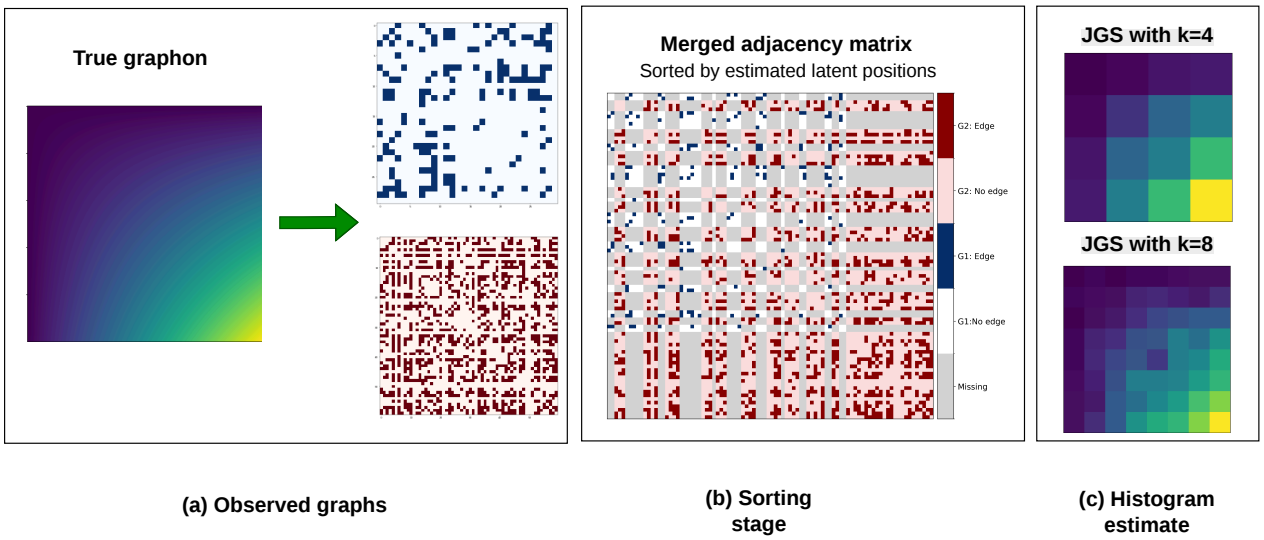

Figure 1: Illustration of the Joint Graph Sorting (JGS) estimator. (a) Networks are generated from a graphon. (b) Nodes are jointly sorted across networks using normalized degrees, yielding a merged adjacency matrix with missing entries (grey). Blue/white denote edges/non-edges in the first graph, and red/pink in the second. (c) JGS graphon estimates with different resolutions k, where colors represent estimated edge probabilities.

this approach overlooks the fact that information can be pooled more efficiently across networks by aligning all nodes simultaneously, rather than one graph at a time.

We introduce the *Joint Graph Sorting (JGS)* procedure, which consists in first ordering all nodes in the collection according to their normalized empirical degree and then constructing a single histogram estimator of the underlying graphon. The global sorting improves the estimation of the latent positions, yielding a more accurate graphon estimate.

Formally, *normalized empirical degrees* are defined by

$$\widehat{d}^{\,\mathrm{norm}}_{i,(m)} = \frac{1}{n^{(m)}-1}\sum_{j=1}^{n^{(m)}} A^{(m)}_{i,j}, \quad i \in [\![n^{(m)}]\!], m \in [\![M]\!]. \tag{4}$$

Nodes of the union of all node sets $\cup_{m\in[\![M]\!]} V^{(m)}$ are ordered by their normalized empirical degree $d^{\mathrm{norm}}_{i,(m)}$ in increasing order. Let $\widehat{\sigma}^{\mathrm{JGS}}$ be the permutation associated with this sorting such that $(\widehat{\sigma}^{\mathrm{JGS}})^{-1}(i,m)$ provides, for node i of the m-th network, the node's rank among the $N = \sum_{m\in[\![M]\!]} n^{(m)}$ nodes in the collection. Then a natural estimate of the latent position $U_{i,(m)}$ is given by

$$\widehat{U}^{\mathrm{JGS}}_{i,(m)} = \frac{(\widehat{\sigma}^{\mathrm{JGS}})^{-1}(i,m)}{N} - \frac{1}{2N}. \tag{5}$$

Note that the estimates $\widehat{U}^{\mathrm{JGS}}_{i,(m)}$ are equally spaced on the interval unit having a finer resolution than estimates computed on a single network. Specifically, the spacing is $1/N$ instead of $1/n^{(m)}$, which makes it possible to obtain more accurate estimates. Now let $A^{\hat{\sigma}^{\mathrm{JGS}}}$ be the $N\times N$ adjacency matrix representing the union of all observed networks with sorted nodes. Notably, many entries of $A^{\hat{\sigma}^{\mathrm{JGS}}}$ are *missing*, because edges exist only among nodes of the same original network.

To estimate graphon W, the square unit is partitioned into k^2 squares, each with a side-length $1/k$ and the observed edge frequencies on these squares are computed using the large sorted adjacency matrix $A^{\hat{\sigma}^{\mathrm{JGS}}}$ by

$$\widehat{H}^{\mathrm{JGS}}_{s,t} = \frac{\sum_{i=a_s}^{b_s}\sum_{j=a_t}^{b_t} A^{\widehat{\sigma}^{\mathrm{JGS}}}_{i,j}\mathbb{1}\left\{A^{\widehat{\sigma}^{\mathrm{JGS}}}_{i,j}\in\{0,1\}\right\}}{\max\left(1,\sum_{i=a_s}^{b_s}\sum_{j=a_t}^{b_t}\mathbb{1}\left\{A^{\widehat{\sigma}^{\mathrm{JGS}}}_{i,j}\in\{0,1\}\right\}\right)}. \tag{6}$$

with $a_s = \lfloor N(s-1)/k\rfloor + 1$, $b_s = \lfloor Ns/k\rfloor$, for $s,t \in [\![k]\!]$. The indicator functions in (6) help dealing with missing values in $A^{\hat{\sigma}^{\mathrm{JGS}}}$. The final graphon estimate of W is given by

$$\widehat{W}^{\mathrm{JGS}}(u,v) = \widehat{H}^{\mathrm{JGS}}_{s,t}, \qquad \text{if } (u,v)\in I_s\times I_t. \tag{7}$$

Figure 1 (b) illustrates this histogram estimator: after joint sorting, the merged matrix $A^{\hat{\sigma}^{\mathrm{JGS}}}$ has large grey regions corresponding to missing inter–graph edges; only the colored within–graph fields contribute to the edge frequencies by block $\widehat{H}^{\mathrm{JGS}}_{s,t}$.

Optional smoothing step. To obtain a more regular graphon estimator and satisfy possible smoothness

assumptions on the underlying graphon, an optional smoothing step may be applied to the histogram estimate $\widehat{W}^{\mathrm{JGS}}$. This reduces artefacts and discontinuities while preserving important global structures. Among the possible methods are those based on total variation minimization (e.g. as in SAS (Chan and Airoldi, 2014)) or variational image-processing techniques (Chambolle, 2004). The choice of the smoothing method depends on the desired properties such as fidelity to the data, edge preservation, and the tradeoff between bias and variance.

Selection of the number of blocks k. Choosing the number of blocks k is delicate yet crucial for the performance of the graphon estimator. Based on heuristic arguments, we propose a selection rule for the choice of k. First, to balance bias and variance, when the graphon W is continuous, the optimal choice of k is expected to verify

$$k \asymp S^{\frac{1}{4}}, \tag{8}$$

where $S = \sum_{m=1}^{M} (n^{(m)})^2$ is the number of observed edges and non edges in the graph collection $\mathcal{G}$ (without counting the missing values). Second, in the multiple-network setting one must guard against empty histogram blocks; a sufficient condition to avoid empty blocks is

$$\frac{N}{k} \geq c\,(M + \log N), \tag{9}$$

where c is a positive constant. Combining these arguments leads to the practical choice

$$k \asymp \min\left\{ S^{1/4},\ \frac{N}{c\,(M+\log N)} \right\}. \tag{10}$$

In experiments, $c = 2$ is found to be a convenient value. More details are provided in the supplementary material. This selection rule for k makes the graphon estimator $\widehat{W}^{\mathrm{JGS}}$ highly accurate.

Complexity. First, concerning the efficient implementation of the graphon estimator $\widehat{W}^{\mathrm{JGS}}$, note that for memory reasons it is not advisable to create the large matrix $A^{\hat{\sigma}^{\mathrm{JGS}}}$. Instead, the computation of the edge frequencies $\widehat{H}^{\mathrm{JGS}}_{s,t}$ is implemented more efficiently as

$$\widehat{H}^{\mathrm{JGS}}_{s,t} = \frac{\sum_{m\in[\![M]\!]} \sum_{i\in\widehat{J}^{(m)}_s} \sum_{j\in\widehat{J}^{(m)}_t} A^{(m)}_{i,j}}{\max\Big(1, \sum_{m\in[\![M]\!]} |\widehat{J}^{(m)}_s|\,|\widehat{J}^{(m)}_t|\Big)}, \quad s,t \in [\![k]\!].$$

where $\widehat{J}^{(m)}_s$ is the set of node indices i of network m such that $\widehat{U}^{\mathrm{JGS}}_{i,(m)} \in I_s$.

We analyze the complexity of JGS step by step. The computation of the normalized empirical degrees requires accessing all observed entries of the adjacency matrices and therefore has complexity $\mathcal{O}(S)$. Note that each observed edge contributes to the degrees of its two incident vertices.

The estimation of latent positions requires $\mathcal{O}(N \log N)$ operations due to the sorting step, plus an additional $\mathcal{O}(N)$ operations to assign the estimated positions.

For the histogram estimation step, the algorithm processes each of the S observed adjacency entries only once. For each entry, it uses the estimated latent positions to determine its corresponding block (s,t) in constant time and then updates the relevant sums in the numerator and denominator of $\widehat{H}^{\mathrm{JGS}}_{s,t}$. Therefore, the complexity of building the histogram is $\mathcal{O}(S)$.

As a result, the overall computational complexity of JGS is $\mathcal{O}(N \log N + S)$ ..

3.2 Theoretical results

This section studies the asymptotic properties of the JGS graphon estimator $\widehat{W}^{\mathrm{JGS}}$. The consistency of the estimator relies on the accurate recovery of the latent positions $\{U^{(m)}_i, i \in [\![n^{(m)}]\!], m \in [\![M]\!]\}$ for all nodes in the network collection. The following proposition states that, under suitable regularity assumptions on the normalized degree function g, the latent position estimators $\widehat{U}^{\mathrm{JGS}}_{i,(m)}$ are close to their true values.

Proposition 3.1 (Consistency of latent positions' estimators)**.** *Let the normalized degree function $g(u) = \int_0^1 W(u,v)\,dv$ be strictly increasing. Assume that there exists a constant $L_1 > 0$ such that*

$$L_1|x-y| \leq |g(x) - g(y)|, \qquad \forall 0 \leq x, y \leq 1.$$

Then, with probability at least $1-\delta$, for all $m \in [\![M]\!]$ and for all $i \in [\![n^{(m)}]\!]$, the following holds

$$\left|\widehat{U}^{\mathrm{JGS}}_{i,(m)} - U_{i,(m)}\right| \leq \eta^{(m)}_i,$$

where

$$\eta^{(m)}_i = \frac{2}{L_1}\left(\varepsilon^{(m)}_i + \frac{1}{N}\sum_{(j,\ell)\neq(i,m)} \varepsilon^{(\ell)}_j\right) + \varepsilon^{(N)}, \tag{11}$$

$\varepsilon^{(\ell)}_j = \sqrt{\frac{\log(2/\delta)}{2(\,n^{(\ell)}-1)}}$ *and* $\varepsilon^{(N)} = \sqrt{\frac{\log(2/\delta)}{2N}}$.

The key mechanism behind Proposition 3.1 is the concentration of the empirical normalized degrees $\widehat{d}^{\mathrm{norm}}_{i,(m)}$, which are unbiased estimators of the theoretical degrees $g(U_{i,(m)})$. The strict monotonicity of g, together

with the lower Lipschitz condition, allows us to translate degree convergence into latent position convergence. Hence, the consistency of the JGS latent position estimators $\widehat{U}^{\mathrm{JGS}}_{i,(m)}$ is entirely driven by the concentration of the degrees. A direct consequence of the proposition is the uniform consistency of the latent position estimators across two asymptotic regimes. We consider monotone sequences of network collections, that is, sequences in which either the networks grow by adding new nodes (i.e., $n^{(m)}$ increases for at least one m), or the collection grows by adding new graphs (i.e., M increases).

Corollary 3.2. *Under the assumptions of Proposition 3.1,*

(i) ***A graph size diverges.*** *If there is a network $m^\star$ such that $n^{(m^\star)} \to \infty$, the estimator of the latent positions of the nodes in the $m^\star$-th graph converges uniformly to the true positions. That is,*

$$\max_{i\in[\![n^{(m^\star)}]\!]} \left|\widehat{U}^{\mathrm{JGS}}_{i,(m^\star)} - U_{i,(m^\star)}\right| \xrightarrow{\mathbb{P}} 0.$$

(ii) ***All graph sizes diverge.*** *Let the number of graphs M be fixed. If $n^{(m)} \to \infty$ for every $m \in [\![M]\!]$, the latent positions are estimated uniformly consistently over the entire collection of networks. That is*

$$\max_{m\in[\![M]\!]} \max_{i\in[\![n^{(m)}]\!]} \left|\widehat{U}^{\mathrm{JGS}}_{i,(m)} - U_{i,(m)}\right| \xrightarrow{\mathbb{P}} 0,$$

Notably, these results do not cover the case where the number of graphs M tends to infinity while the graph sizes $n^{(m)}$ remain bounded. Indeed, our simulations suggest that the estimator is then not consistent, because the normalized degree estimators $\widehat{d}^{\mathrm{norm}}_{i,(m)}$ are no longer consistent. In summary, the consistency of the JGS graphon estimator $\widehat{W}^{\mathrm{JGS}}$ is obtained when the number of graphs M is fixed.

Theorem 3.3. *Let the number of graphs $M \geq 1$ be fixed. Assume:*

(i) The graphon W satisfies the strict monotonicity of degrees Condition 2.1 and is L–Lipschitz on the unit square. Its normalized degree function g is strictly increasing and satisfies the lower Lipschitz condition with constant L_1.

(ii) The number of bins $k = k(N)$ satisfies the no empty blocks *condition*

$$k = o\Big(\tfrac{N}{M+\log N}\Big). \tag{12}$$

Then, as $n_{\max} = \max_{m\in[\![M]\!]} n^{(m)} \to \infty$ and $k/n_{\max} \to 0$, the mean integrated squared error of the JGS graphon estimator tends to zero:

$$\mathrm{MISE}\Big(\widehat{W}^{\mathrm{JGS}}, W\Big) = \mathbb{E}\Big[\big\|\widehat{W}^{\mathrm{JGS}} - W\big\|^2_{L^2}\Big] \longrightarrow 0.$$

4 NUMERICAL EXPERIMENTS

In this section, the performance of JGS is assessed in several numerical experiments. First, the JGS graphon estimator is compared with the state of the art on synthetic data. Then, the behavior of the estimator and of the latent position estimators is examined in two asymptotic regimes. Finally, we illustrate the gain of using JGS in a graph neural network classification task on real-world datasets.

4.1 Comparison of JGS with the state of the art

In this study *JGS* refers to JGS histogram estimator $\widehat{W}^{\mathrm{JGS}}$ of the graphon, where the number of blocks k is chosen according to (10), and *JGS-smooth* to JGS combined with the total-variation–based smoothing algorithm of Chambolle (2004). Both methods are compared to the following existing methods: the stochastic block approximation *SBA* (Airoldi et al., 2013), sorting-and-smoothing *SAS* (Chan and Airoldi, 2014), universal singular value thresholding *USVT* (Chatterjee, 2015), smoothed Gromov–Wasserstein barycenters *SGWB* (Xu et al., 2021), and scalable implicit graphon learning *SIGL* (Azizpour et al., 2025). For SBA, SAS, USVT, and SGWB we adopt the multiple-graph implementations provided by Xu et al. (2021), which combine single-graph estimates via pooling across networks. For SIGL, we use the authors' publicly available code (Azizpour et al., 2025). Details on the algorithms are provides in the supplementary material.

Data are generated from the same thirteen graphons as those considered in Azizpour et al. (2025), see Table 1. The non-monotone graphons 10–13 do not verify the identifiability Condition 2.1 and are used to investigate the estimators' robustness. For each experiment, we simulate a collection of $M = 200$ networks, with network sizes $n^{(m)}$ randomly chosen between 10 and 100. To evaluate the estimation accuracy the mean integrated squared error (MISE) is computed for all methods based on 20 independent trials.

Compared to the state of the art, JGS consistently achieves lower MISE for all graphons except for graphon 9, see Table 1. Unlike the other graphons, graphon 9 produces very sparse networks and SIGL performs better than JGS in this case. This may indicate that JGS is not the optimal approach for very sparse networks, while still having the second best behavior among all methods. Concerning the smoothed version, JGS-smooth, a further reduction of the estimation error is observed. Although JGS relies on the strict monotonicity condition, its performance on the non-monotone graphons (10–13) is particularly noteworthy. These cases are clearly challenging for all

Table 1: MISE ($\times 10^{-3}$) for different graphon estimators on collections of networks with varying size.

ID	SBA	SAS	USVT	SGWB	SIGL	JGS	JGS-smooth
Monotone graphons							
1	37.10 ± 0.4	69.64 ± 0.29	36.95 ± 0.4	7.63 ± 0.45	1.07 ± 0.34	$\mathbf{0.58 \pm 0.09}$	$\mathbf{0.43 \pm 0.09}$
2	43.32 ± 0.31	55.13 ± 0.32	43.37 ± 0.3	11.23 ± 0.33	1.1 ± 0.19	$\mathbf{0.82 \pm 0.07}$	$\mathbf{0.64 \pm 0.06}$
3	104.1 ± 0.6	123.6 ± 0.6	104 ± 0.6	12.42 ± 0.33	1.31 ± 0.19	$\mathbf{0.65 \pm 0.09}$	$\mathbf{0.46 \pm 0.08}$
4	89.66 ± 0.75	114.6 ± 0.8	89.34 ± 0.74	10.15 ± 0.74	1.32 ± 0.3	$\mathbf{0.73 \pm 0.11}$	$\mathbf{0.55 \pm 0.1}$
5	221.6 ± 0.5	204.7 ± 0.5	221.4 ± 0.5	14.09 ± 0.35	2.32 ± 0.39	$\mathbf{0.65 \pm 0.03}$	$\mathbf{0.53 \pm 0.03}$
6	193.7 ± 0.3	175.9 ± 0.4	193.7 ± 0.3	20.64 ± 0.22	2.42 ± 0.83	$\mathbf{1.92 \pm 0.07}$	$\mathbf{1.74 \pm 0.07}$
7	103.5 ± 0.4	110.6 ± 0.4	103.5 ± 0.4	21.38 ± 0.73	3.4 ± 0.96	$\mathbf{2.64 \pm 0.1}$	$\mathbf{2.51 \pm 0.09}$
8	83.94 ± 0.42	87.25 ± 0.49	84.02 ± 0.43	13.41 ± 0.42	1.44 ± 0.19	$\mathbf{1.11 \pm 0.19}$	$\mathbf{0.89 \pm 0.06}$
9	40.15 ± 0.15	39.19 ± 0.19	40.09 ± 0.16	14.16 ± 0.44	$\mathbf{1.06 \pm 0.23}$	2.34 ± 0.07	2.13 ± 0.07
Non-monotone graphons							
10	94.64 ± 0.14	96.3 ± 0.14	94.61 ± 0.14	76.08 ± 1.89	44.48 ± 0.63	$\mathbf{43.65 \pm 0.08}$	$\mathbf{43.46 \pm 0.07}$
11	217 ± 0.5	213.9 ± 0.6	217 ± 0.5	70.88 ± 2.64	54.42 ± 12.28	$\mathbf{43.77 \pm 0.08}$	$\mathbf{43.63 \pm 0.08}$
12	187 ± 2.1	225.7 ± 0.2	187.9 ± 2.3	78.77 ± 0.49	198.6 ± 65.6	$\mathbf{75.35 \pm 3.25}$	$\mathbf{74.47 \pm 3.34}$
13	246.5 ± 3.0	225.5 ± 0.2	253.3 ± 2.6	106.5 ± 35.1	197.2 ± 37.1	$\mathbf{81.6 \pm 10.3}$	$\mathbf{79.3 \pm 9.1}$

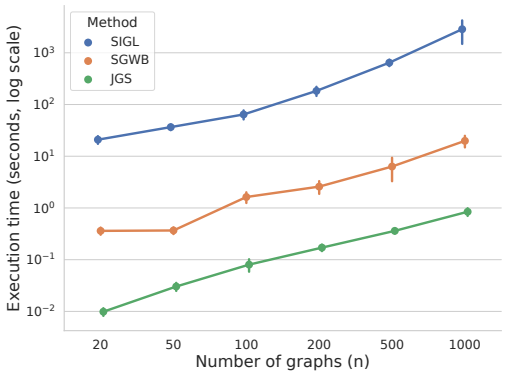

Figure 2: Time as a function of the graph size n

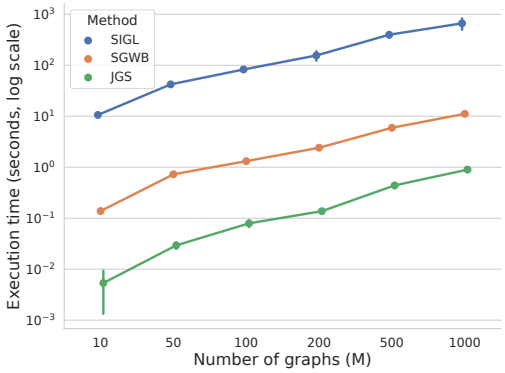

Figure 3: Time depending on the number of graphs M

methods, as reflected by the substantially larger errors compared to the monotone graphons.

Another advantage of JGS is its speed. As shown in Figures 2 and 3, its execution time remains short even when the number of graphs M or the network sizes $n^{(m)}$ are large. Compared to the alternative methods SIGL and SGWB, which are the most accurate approaches in the literature, the computing time of JGS is one or two orders of magnitude faster. All experiments were conducted on the same machine to ensure a fair comparison. Altogether, these results confirm that JGS is not only more effective at leveraging information from multiple networks than current methods, but also substantially faster in terms of execution time.

4.2 Scaling behavior of the JGS

The behaviour of the JGS estimator is studied under two distinct asymptotic regimes, each reflecting a different practical scenario for network collections.

In the first regime, commonly studied in the literature, the number of networks is fixed, here at $M = 30$, while the sizes $n^{(m)}$ of all networks increase, here from 30 to 1000 nodes. Data are generated from the graphon $W(u, v) = uv$. As shown in Figure 4, the MISE decreases steadily and approaches zero as the network size $n^{(m)}$ grows. This behavior aligns with the consistency result established in Theorem 3.3 and confirms that the JGS graphon estimator efficiently utilises larger networks to improve estimation accuracy.

In the second regime, which has received much less attention in the literature, all network sizes $n^{(m)}$ are fixed, here at $n^{(m)} = 30$, but the number of graphs M is steadily increased, here from 10 to 1000. In this setting, Figure 5 shows that although the MISE decreases as M grows, it converges to some non-zero value. This indicates that the JGS graphon estimator is not consistent in this regime, a limitation not specific to our method, but inherent to the fixed-graph-size asymptotic framework.

To better understand this behavior, we examine the estimation quality of the JGS latent positions $\hat{U}^{\text{JGS}}_{i,(m)}$. As shown in Figure 6, in the first regime, these estimates are consistent as the number of nodes increases. In contrast, Figure 7 reveals that in the second regime, the mean absolute error (MAE) of the latent posi-

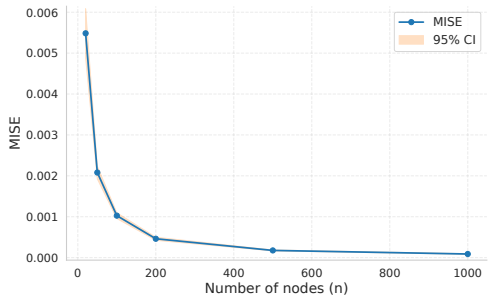

Figure 4: MISE as a function of the graph size n

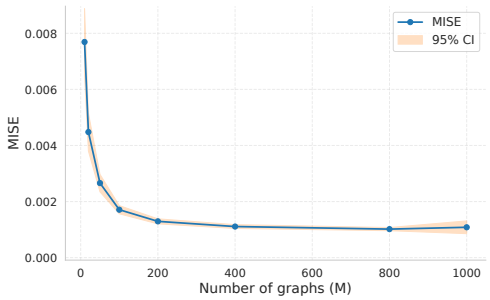

Figure 5: MISE depending on the number of graphs M

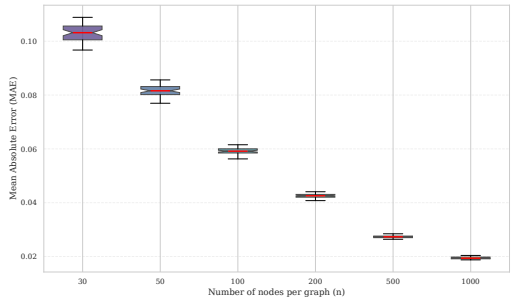

Figure 6: MAE of the latent positions' as the number of nodes $n^{(m)}$ increases

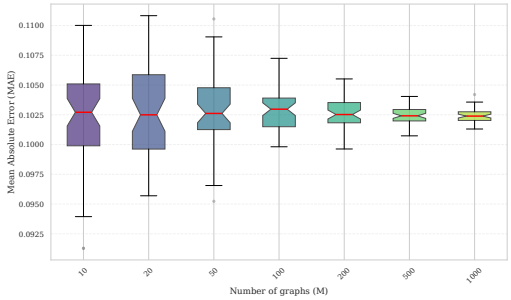

Figure 7: MAE of the latent positions' as the number of graphs M increases

Table 2: Comparison of the classification test accuracy (%) on IMDB-BINARY and IMDB-MULTI.

Method	IMDB-BINARY	IMDB-MULTI
Vanilla	73.0 ± 3.52	47.49 ± 1.27
G-Mixup / USVT	72.20 ± 2.57	48.99 ± 3.01
G-Mixup / SGWB	72.85 ± 2.50	49.67 ± 2.41
G-Mixup / SIGL	72.60 ± 3.38	48.63 ± 2.27
G-Mixup / JGS	$\mathbf{75.72 \pm 2.81}$	$\mathbf{50.23 \pm 2.15}$

tions does not converge to zero. This occurs because adding more graphs does not provide additional information about the latent positions of nodes in existing graphs. Indeed, in the first regime, the empirical normalized degrees converge to their theoretical counterpart. However, this does not happen in the second regime, where the estimated empirical degrees remain constant. Consequently, this regime is intrinsically more challenging than the classical setting, where graph sizes tend to infinity.

4.3 Application to a classification problem on real-world data

To assess the performance of JGS on real-world data, we integrate JGS into the G-Mixup framework (Han et al., 2022; Navarro and Segarra, 2023), a data augmentation approach for graph classification. G-Mixup uses a graphon estimate, which is evaluated on training data, to generate new synthetic graphs by interpolation. The aim is to enrich the training set to improve the model's generalization ability.

We conducted experiments on two widely used benchmark datasets: IMDB-BINARY and IMDB-MULTI (Morris et al., 2020; Yanardag and Vishwanathan, 2015). The IMDB-BINARY dataset consists of 1000 graphs representing actor relationships in movies, divided into two classes. IMDB-MULTI contains 1500 networks divided into three classes. The number of nodes of the graphs range from 7 to 89, which is suitable for the JGS approach.

For the classification task, the Graph Isomorphism Network (GIN) architecture (Xu et al., 2019) is applied following the training protocol described in (Han et al., 2022). Five variants are compared: a baseline model (vanilla) without data augmentation, and G-Mixup using the USVT, SGWB, SIGL, and JGS estimators. More details on the training parameters are provided in the supplement material.

As shown in Table 2, integrating our JGS method into G-Mixup leads to higher accuracy than the other variants on both IMDB-BINARY and IMDB-MULTI. This improvement is due to JGS's ability to efficiently leverage information from multiple graphs, making it well-suited for graphon inference in heterogeneous multi-network scenarios.

5 CONCLUSION

This work confirms that in the multiple network setting the averaging of graphon estimates, that were conceived for single networks, is not the optimal strategy. More consistent estimators are obtained by analyzing the entire collection of networks simultaneously as done by JGS. Furthermore, we demonstrated that such strategies do not have to be computationally intensive; our JGS approach achieves highly accurate results with negligible computing times compared to methods such as SIGL and SGWB.

The following open questions have arisen during the work. Firstly, JGS appears to perform less well on collections of very sparse networks. The question is how to improve JGS in this case, specifically how to obtain better estimates of the nodes' latent positions. Secondly, it has become clear that, in the asymptotic regime where the number of networks M tends to infinity while the network sizes $n^{(m)}$ are bounded for all networks, the latent position estimates are no longer consistent. This indicates that the JGS graphon estimator is not consistent in this regime. This limitation is related to the fact that the indicators used to estimate the latent positions, such as normalized degrees in our approach, cannot be consistently estimated when graph sizes remain bounded. Whether a consistent graphon estimator exists in this asymptotic framework remains an open question.

References

Abbe, E. (2018). Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86.

Airoldi, E. M., Costa, T. B., and Chan, S. H. (2013). Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, volume 26, pages 692–700.

Azizpour, A., Zilberstein, N., and Segarra, S. (2025). Scalable implicit graphon learning. In Li, Y., Mandt, S., Agrawal, S., and Khan, E., editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 3952–3960. PMLR.

Bickel, P. J. and Chen, A. (2009). A nonparametric view of network models and newman-girvan and other modularities. *Proc. Natl. Acad. Sci. USA*, 106(50):21068–21073.

Borgs, C., Chayes, J. T., and Smith, A. (2015). Private graphon estimation for sparse graphs. In *Advances in Neural Information Processing Systems*, pages 1369–1377, Montreal, Canada. Curran Associates, Inc.

Celisse, A., Daudin, J.-J., and Pierre, L. (2012). Consistency of maximum-likelihood and variational estimators in the Stochastic Block Model. *Electronical Journal of Statistics*, 6:1847–1899.

Chabert-Liddell, S.-C., Barbillon, P., and Donnet, S. (2024). Learning common structures in a collection of networks. An application to food webs. *The Annals of Applied Statistics*, 18(2):1213 – 1235.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97.

Chan, S. and Airoldi, E. (2014). A consistent histogram estimator for exchangeable graph models. In *International Conference on Machine Learning*, pages 208–216.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214.

Diaconis, P. and Janson, S. (2007). Graph limits and exchangeable random graphs. *Rendiconti di Matematica e delle sue Applicazioni, Series VII*, 28:33–61.

Eldridge, J., Belkin, M., and Wang, Y. (2016). Graphons, mergeons, and so on! In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Fornito, A., Zalesky, A., and Breakspear, M. (2013). Graph analysis of the human connectome: Promise, progress, and pitfalls. *NeuroImage*, 80:426–444.

Gao, C., Lu, Y., and Zhou, H. H. (2014). Rate-optimal graphon estimation. *Annals of Statistics*, 43:2624–2652.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010). A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233.

Han, X., Jiang, Z., Liu, N., and Hu, X. (2022). G-mixup: Graph data augmentation for graph classification. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8230–8248.

Janson, S. (2020). A graphon counter example. *Discrete Mathematics*, 343:111836.

Keshavan, R. H., Montanari, A., and Oh, S. (2010). Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Klopp, O., Tsybakov, A. B., and Verzelen, N. (2017). Oracle inequalities for network models and sparse graphon estimation. *The Annals of Statistics*, 45(1):316–354.

Kolaczyk, E. D. (2009). *Statistical analysis of network data: methods and models*. Springer Series in Statistics. Springer, New York, NY.

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

Matias, C. and Robin, S. (2014). Modeling heterogeneity in random graphs through latent space models: a selective review. *Esaim Proc. & Surveys*, 47:55–74.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. (2020). Tudataset: A collection of benchmark datasets for learning with graphs. In *Proceedings of the International Conference on Machine Learning, Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*.

Navarro, M. and Segarra, S. (2023). Graphmad: Graph mixup for data augmentation using data-driven convex clustering. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE.

Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstruc-

tures. *Journal of the American Statistical Association*, 96:1077–1087.

Poisot, T., Baiser, B., Dunne, J. A., Kéfi, S., Massol, F. c., Mouquet, N., Romanuk, T. N., Stouffer, D. B., Wood, S. A., and Gravel, D. (2016). mangal – making ecological network analysis simple. *Ecography*, 39(4):384–390.

Rebafka, T. (2024). Model-based clustering of multiple networks with a hierarchical algorithm. *Statistics and Computing*, 34:1–16.

Ruiz, L., Chamon, L., and Ribeiro, A. (2020). Graphon neural networks and the transferability of graph neural networks. *Advances in Neural Information Processing Systems*, 33:1702–1712.

Ruiz, L., Chamon, L. F., and Ribeiro, A. (2023). Transferability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 71:3474–3489.

Sischka, B. and Kauermann, G. (2022). Em-based smooth graphon estimation using mcmc and spline-based approaches. *Social Networks*, 68:279–295.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pages 7462–7473. Curran Associates, Inc.

Weber-Zendrera, A., Sokolovska, N., and Soula, H. (2019). Robust structure measures of metabolic networks that predict prokaryotic optimal growth temperature. *BMC Bioinformatics*, 20(499).

Wolfe, P. J. and Olhede, S. C. (2013). Nonparametric graphon estimation. *arXiv: Statistics Theory*.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Xia, X., Mishne, G., and Wang, Y. (2023). Implicit graphon neural representation. In Ruiz, F., Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 10619–10634. PMLR.

Xu, H., Luo, D., Carin, L., and Zha, H. (2021). Learning graphons via structured gromov-wasserstein barycenters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:10505–10513.

Xu, K., Jegelka, S., Hu, W., and Leskovec, J. (2019). How powerful are graph neural networks? In *7th International Conference on Learning Representations*. Conference dates: 2019-05-06 to 2019-05-09.

Yanardag, P. and Vishwanathan, S. V. N. (2015). Deep graph kernels. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Yang, J., Han, C., and Airoldi, E. M. (2014). Nonparametric estimation and testing of exchangeable graph models. In *International Conference on Artificial Intelligence and Statistics*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes
   (b) Complete proofs of all theoretical results. Yes
   (c) Clear explanations of any assumptions. Yes

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. Yes
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes
   (d) Information about consent from data providers/curators. Not Applicable
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Low-Complexity and Consistent Graphon Estimation From Multiple Networks: Supplementary Materials

A PROOFS OF THEORETICAL RESULTS

A.1 Notation summary

We first recall the main notations used in the proofs.

(i) Graphon. The true underlying graphon is a measurable symmetric function $W : [0,\ 1]^2 \to [0,\ 1]$, assumed to be L–Lipschitz, that is,

$$|W(u,v) - W(u',v')| \le L\big(|u-u'| + |v-v'|\big), \qquad \forall (u,v),(u',v') \in [0,1]^2.$$

(ii) Partition. Divide $[0,\ 1]$ into k equal subintervals

$$I_s = \Big[\frac{s-1}{k}, \frac{s}{k}\Big), \qquad s = 1, \dots, k.$$

For each graph $G^{(m)}$, let

$$J_s^{(m)} = \{i : U_{i,(m)} \in I_s\}, \qquad \widehat{J}_s^{(m)} = \{i : \widehat{U}^{\mathrm{JGS}}_{i,(m)} \in I_s\}.$$

These are the sets of nodes of network $G^{(m)}$ that fall into I_s according to the true latent positions $U_{i,(m)}$ and the estimated positions $\widehat{U}^{\mathrm{JGS}}_{i,(m)}$, respectively.

(iii) Dyads per block. The set of dyads in block (s,t) is

$$\Omega_{s,t} = \bigcup_{m \in [\![M]\!]} \Big\{(i,j;m) : i \in J_s^{(m)},\ j \in J_t^{(m)}\Big\},$$

with cardinal number $M_{s,t} = |\Omega_{s,t}|$. Similarly, with estimated positions,

$$\widehat{\Omega}_{s,t} = \bigcup_{m \in [\![M]\!]} \Big\{(i,j;m) : i \in \widehat{J}_s^{(m)},\ j \in \widehat{J}_t^{(m)}\Big\}, \qquad \widehat{M}_{s,t} = |\widehat{\Omega}_{s,t}|.$$

(iv) JGS graphon estimator. The JGS estimate defined on a $k \times k$ partition of $[0,1]^2$ is

$$\widehat{W}^{\mathrm{JGS}}(u,v) = \sum_{s,t \in [\![k]\!]^2} \widehat{H}^{\mathrm{JGS}}_{s,t}\, \mathbb{1}_{(u,v) \in I_s \times I_t},$$

with

$$\widehat{H}^{\mathrm{JGS}}_{s,t} = \frac{1}{\max(\widehat{M}_{s,t}, 1)} \sum_{m \in [\![M]\!]} \sum_{i \in \widehat{J}_s^{(m)}} \sum_{j \in \widehat{J}_t^{(m)}} A^{(m)}_{i,j}.$$

(v) Oracle histogram. The oracle version defined on a $k \times k$ partition of $[0,\ 1]^2$ using the true latent positions $U_{i,(m)}$ is

$$W^{\mathrm{oracle}}(u,v) = \sum_{s,t \in [\![k]\!]^2} H_{s,t}\, \mathbb{1}_{(u,v) \in I_s \times I_t},$$

where

$$H_{s,t} = \frac{1}{\max(M_{s,t}, 1)} \sum_{m \in [\![M]\!]} \sum_{i \in J_s^{(m)}} \sum_{j \in J_t^{(m)}} A_{i,j}^{(m)}.$$

(vi) Block averages of W. For the bias–variance analysis, define

$$\Phi_{s,t} = \mathbb{E}\left[H_{s,t} \,|\, \mathcal{U}\right] = \frac{1}{M_{s,t}} \sum_{m \in [\![M]\!]} \sum_{i \in J_s^{(m)}} \sum_{j \in J_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right),$$

and

$$\widehat{\Phi}_{s,t} = \mathbb{E}\left[\widehat{H}_{s,t}^{\mathrm{JGS}} \,\middle|\, \mathcal{U}\right].$$

Equivalently,

$$\widehat{\Phi}_{s,t} = \mathbb{E}\left[\frac{1}{\widehat{M}_{s,t}} \sum_{m \in [\![M]\!]} \sum_{i \in \widehat{J}_s^{(m)}} \sum_{j \in \widehat{J}_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right) \middle| \mathcal{U}\right].$$

Here

$$\mathcal{U} = \left\{U_{i,(m)} : m \in [\![M]\!],\ i \in [\![n^{(m)}]\!]\right\}$$

is the set of all latent positions in the collection of networks.

(vii) Discretized oracle graphon. Define the piecewise-constant version of W on a $k \times k$ partition of $[0,\ 1]^2$ as

$$W^{\mathrm{approx}}(u, v) = \sum_{s,t \in [\![k]\!]^2} \Phi_{s,t}\, \mathbb{1}_{(u,v) \in I_s \times I_t}.$$

(viii) Node ranks. For each node $(i, m) \in [\![n]\!]^{(m)} \times [\![M]\!]$, let

$$\widehat{r}_{i,(m)} = \left(\widehat{\sigma}^{\mathrm{JGS}}\right)^{-1}(i, m)$$

be its empirical rank among all $N = \sum_{m \in [\![M]\!]} n^{(m)}$ nodes. The *oracle rank* is

$$r_{i,(m)} = \left|\left\{(j, \ell) : g(U_{j,(\ell)}) \le g(U_{i,(m)})\right\}\right|,$$

where $g(u) = \int_0^1 W(u, v)\, dv$ is the normalized degree function.

A.2 Proof of Proposition 3.1

The consistency of the latent positions' estimators $\widehat{U}_{i,(m)}$ depends on the consistency of the correct ordering of the nodes, that is, on the exact estimation of the rank of a node in the collection of all nodes. This is formalized in the following lemma.

Lemma A.1 (Control of the rank discrepancy)**.** *Suppose that the normalized degree function*

$$g\colon\ [0,\ 1] \to \mathbb{R}$$

is strictly increasing and satisfies the lower Lipschitz condition. That is, there exists a constant $L_1 > 0$ such that, for all $x, y \in [0, 1]$,

$$L_1|x - y| \le |g(x) - g(y)|.$$

Then, for any fixed node $(i, m) \in [\![n]\!]^{(m)} \times [\![M]\!]$, with probability at least $1 - \delta$, the discrepancy between the empirical rank $\widehat{r}_{i,(m)}$ and the theoretical rank $r_{i,(m)}$ satisfies

$$\left|\frac{\widehat{r}_{i,(m)}}{N} - \frac{r_{i,(m)}}{N}\right| \le \eta_i^{(m)},$$

where $\eta_i^{(m)}$ is defined in the main paper in (11)*.*

Proof. Fix a node $(i,m) \in [\![n^{(m)}]\!] \times [\![M]\!]$. Recall that conditionally on the latent variables $U_{i,(m)}, U_{j,(m)}$, the variables $A_{i,j}^{(m)}$ are independent with Bernoulli distribution with parameter $W(U_{i,(m)}, U_{j,(m)})$. Therefore, conditionally on $U_{i,(m)}$, we have

$$\begin{aligned}\mathbb{E}\big[\widehat{d}_{i,(m)}^{\mathrm{norm}} \mid U_{i,(m)}\big] &= \frac{1}{n^{(m)}-1} \sum_{j \in [\![n^{(m)}]\!],\, j \neq i} \mathbb{E}[A_{i,j}^{(m)} \mid U_{i,(m)}] \\ &= \int_0^1 W(U_{i,(m)}, v)\, dv \\ &= g(U_{i,(m)}).\end{aligned}$$

Since the $A_{i,j}^{(m)} \in \{0,1\}$ are conditionally independent, Hoeffding's inequality yields, conditionally on $U_{i,(m)}$,

$$\mathbb{P}\left(\left|\widehat{d}_{i,(m)}^{\mathrm{norm}} - g(U_{i,(m)})\right| \geq \varepsilon_n^{(m)} \,\middle|\, U_{i,(m)}\right) \leq 2\exp\left(-2n^{(m)}(\varepsilon_n^{(m)})^2\right).$$

Setting

$$\varepsilon_n^{(m)} = \sqrt{\frac{\log(2/\delta)}{2n^{(m)}}},$$

we obtain

$$\mathbb{P}\left(\left|\widehat{d}_{i,(m)}^{\mathrm{norm}} - g(U_{i,(m)})\right| \geq \varepsilon_n^{(m)} \,\middle|\, U_{i,(m)}\right) \leq \delta.$$

We now define the following set of potentially misranked nodes

$$C_i^{(m)} := \left\{(j,\ell) \neq (i,m) : \left|g(U_{j,(\ell)}) - g(U_{i,(m)})\right| \leq \varepsilon_i^{(m)} + \varepsilon_j^{(\ell)}\right\}.$$

Only the nodes in $C_i^{(m)}$ can possibly alter the relative ordering between $\widehat{d}_{i,(m)}^{\mathrm{norm}}$ and $g(U_{i,(m)})$, since for all other pairs, the deviation is too large so that a possible reversal has negligible probability. Indeed, suppose that

$$g(U_{j,(\ell)}) < g(U_{i,(m)}) - \varepsilon_i^{(m)} - \varepsilon_j^{(\ell)}.$$

Then, by the concentration bounds

$$\widehat{d}_{j,(\ell)}^{\mathrm{norm}} \leq g(U_{j,(\ell)}) + \varepsilon_j^{(\ell)} < g(U_{i,(m)}) - \varepsilon_i^{(m)} \leq \widehat{d}_{i,(m)}^{\mathrm{norm}},$$

that is $\widehat{d}_{j,(\ell)}^{\mathrm{norm}} < \widehat{d}_{i,(m)}^{\mathrm{norm}}$. Similarly, if

$$g(U_{j,(\ell)}) > g(U_{i,(m)}) + \varepsilon_i^{(m)} + \varepsilon_j^{(\ell)},$$

then

$$\widehat{d}_{j,(\ell)}^{\mathrm{norm}} \geq g(U_{j,(\ell)}) - \varepsilon_j^{(\ell)} > g(U_{i,(m)}) + \varepsilon_i^{(m)} \geq \widehat{d}_{i,(m)}^{\mathrm{norm}},$$

that is $\widehat{d}_{j,(\ell)}^{\mathrm{norm}} > \widehat{d}_{i,(m)}^{\mathrm{norm}}$. Thus, only the nodes $(j,\ell) \in C_i^{(m)}$ may provoke a change in the empirical rank of node (i,m), while all others preserve the ordering. This justifies restricting the rank discrepancy analysis to the set $C_i^{(m)}$. We thus obtain

$$\left|\frac{\widehat{r}_{i,(m)}}{N} - \frac{r_{i,(m)}}{N}\right| \leq \frac{|C_i^{(m)}|}{N}.$$

Let us now control $\mathbb{E}[|C_i^{(m)}| \mid U_{i,(m)}]$. By definition,

$$\mathbb{E}\left[|C_i^{(m)}| \mid U_{i,(m)}\right] = \sum_{(j,\ell)\neq(i,m)} \mathbb{P}\left((j,\ell) \in C_i^{(m)} \,\middle|\, U_{i,(m)}\right).$$

Using the bi-Lipschitz property of g, we know that

$$\left|g(U_{j,(\ell)}) - g(U_{i,(m)})\right| \leq \varepsilon_i^{(m)} + \varepsilon_j^{(\ell)}.$$

Thus,

$$\left|U_{j,(\ell)} - U_{i,(m)}\right| \le \frac{\varepsilon_i^{(m)} + \varepsilon_j^{(\ell)}}{L_1}.$$

Since $U_{j,(\ell)} \sim \text{Uniform}[0,1]$ is independent of $U_{i,(m)}$, conditionally on $U_{i,(m)}$, we get

$$\mathbb{P}\left((j,\ell) \in C_i^{(m)} \,\middle|\, U_{i,(m)}\right) \le 1 \wedge \left(2 \cdot \frac{\varepsilon_i^{(m)} + \varepsilon_j^{(\ell)}}{L_1}\right).$$

Therefore

$$\mathbb{E}\left[|C_i^{(m)}| \,|\, U_{i,(m)}\right] \le N \wedge \left\{\frac{2}{L_1}\left((N-1)\varepsilon_i^{(m)} + \sum_{(j,\ell)\neq(i,m)} \varepsilon_j^{(\ell)}\right)\right\}.$$

To get a high-probability bound on $|C_i^{(m)}|$, we use Hoeffding's inequality conditionally on $U_{i,(m)}$, since $|C_i^{(m)}|$ is a sum of independent indicators, for any $\eta > 0$,

$$\mathbb{P}\left(\left|\frac{|C_i^{(m)}|}{N} - \frac{\mathbb{E}[|C_i^{(m)}| \,|\, U_{i,(m)}]}{N}\right| \ge \eta \,\middle|\, U_{i,(m)}\right) \le 2\exp\left(-2N\eta^2\right).$$

By choosing $\eta = \sqrt{\frac{\log(2/\delta)}{2N}}$, we get

$$\mathbb{P}\left(\left|\frac{|C_i^{(m)}|}{N} - \frac{\mathbb{E}[|C_i^{(m)}| \,|\, U_{i,(m)}]}{N}\right| \ge \sqrt{\frac{\log(2/\delta)}{2N}} \,\middle|\, U_{i,(m)}\right) \le \delta.$$

Finally, with probability at least $1-\delta$, we obtain

$$\left|\frac{\widehat{r}_{i,(m)}}{N} - \frac{r_{i,(m)}}{N}\right| \le 1 \wedge \left\{\frac{2}{L_1}\left(\varepsilon_i^{(m)} + \frac{1}{N}\sum_{(j,\ell)\neq(i,m)} \varepsilon_j^{(\ell)}\right)\right\} + \sqrt{\frac{\log(2/\delta)}{2N}}.$$

□

Proof of Proposition 3.1. Fix a node $(i,m) \in [\![n^{(m)}]\!] \times [\![M]\!]$. The estimation error of $\widehat{U}_{i,(m)}^{\text{JGS}}$ is bounded by a sum of three terms

$$\left|\widehat{U}_{i,(m)}^{\text{JGS}} - U_{i,(m)}\right| \le \left|\widehat{U}_{i,(m)}^{\text{JGS}} - \frac{\widehat{r}_{i,(m)}}{N}\right| + \left|\frac{\widehat{r}_{i,(m)}}{N} - \frac{r_{i,(m)}}{N}\right| + \left|\frac{r_{i,(m)}}{N} - U_{i,(m)}\right|. \tag{13}$$

For the first term, by definition of the estimator $\widehat{U}_{i,(m)}^{\text{JGS}} = \frac{\widehat{r}_{i,(m)}}{N} - \frac{1}{2N}$, we immediately get

$$\left|\widehat{U}_{i,(m)}^{\text{JGS}} - \frac{\widehat{r}_{i,(m)}}{N}\right| = \frac{1}{2N}.$$

The second term in (13) is bounded by using Lemma A.1.

The third term $\left|\frac{r_{i,(m)}}{N} - U_{i,(m)}\right|$ corresponds to the error between the empirical and theoretical quantiles of $g(U_{i,(m)})$. Since the normalized degree function g is strictly increasing and $U \sim \text{Unif}\,[0,\ 1]$,

$$F_g(g(U_{i,(m)})) = \mathbb{P}(g(U) \le g(U_{i,(m)})) = \mathbb{P}(U \le U_{i,(m)}) = U_{i,(m)}.$$

On the other hand, we have

$$\widehat{F}_g(g(U_{i,(m)})) = \frac{r_{i,(m)}}{N},$$

which is the empirical CDF of $g(U)$, using all N nodes in the collection, evaluated at $g(U_{i,(m)})$. By the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality, for any $\delta \in (0,1)$, we have with probability at least $1-\delta$,

$$\sup_{x\in\mathbb{R}}\left|\widehat{F}_g(x) - F_g(x)\right| \le \sqrt{\frac{\log(2/\delta)}{2N}}.$$

In particular, this gives

$$\left|\frac{r_{i,(m)}}{N} - U_{i,(m)}\right| = \left|\widehat{F}_g(g(U_i^{(m)})) - F_g(g(U_{i,(m)}))\right| \leq \sqrt{\frac{\log(2/\delta)}{2N}}.$$

Combining the three terms, we obtain

$$\left|\widehat{U}^{\mathrm{JGS}}_{i,(m)} - U_{i,(m)}\right| \leq \frac{1}{2N} + 1 \wedge \left\{\frac{2}{L_1}\left(\varepsilon_i^{(m)} + \frac{1}{N}\sum_{(j,\ell)\neq(i,m)} \varepsilon_j^{(\ell)}\right)\right\} + 2\sqrt{\frac{\log(2/\delta)}{2N}}.$$

□

A.3 Proof of Theorem 3.3

The proof of the theorem is structured around three auxiliary lemmas: *(i)* the first lemma controls the discretization error between the true graphon W and its block approximation W^{approx}, *(ii)* the second lemma bounds the error between the oracle histogram W^{oracle} and the block approximation of the true graphon W^{approx}, and *(iii)* the third lemma establishes the stochastic error between the JGS estimator $\widehat{W}^{\mathrm{JGS}}$ and the oracle graphon W^{oracle}, showing how vertex misclassification propagates to the final error.

Lemma A.2 (Discretization error)**.** *Assume that the true graphon $W : [0,\,1]^2 \to [0,\,1]$ is L–Lipschitz. Let W^{approx} denote its blockwise discretization on a $k \times k$ partition of $[0,\,1]^2$. Then there exists a constant $C > 0$, independent of N, such that*

$$\mathbb{E}\left[\|W^{\mathrm{approx}} - W\|_{L^2}^2\right] \;\leq\; \frac{C\,L^2}{k^2}.$$

The proof of this lemma is standard in the literature on graphon models and histogram approximations (see Lemma 1 in Chan and Airoldi (2014)).

Lemma A.3 (Oracle vs. block approximation)**.** *Let W^{oracle} denote the oracle histogram estimator with blockwise discretization on a $k \times k$ partition of $[0,\,1]^2$, and let W^{approx} denote the block approximation of the true graphon. Then*

$$\mathbb{E}\left[\left\|W^{\mathrm{oracle}} - W^{\mathrm{approx}}\right\|_{L^2}^2\right] = \frac{1}{k^2}\sum_{s,t\in[\![k]\!]^2} \mathbb{E}\left[(H_{s,t} - \Phi_{s,t})^2\right] \;\leq\; \frac{1}{4k^2}\sum_{s,t\in[\![k]\!]^2} \mathbb{E}\Big[\frac{1}{M_{s,t}}\Big],$$

where $\Phi_{s,t}$ is the blockwise expectation of W on $I_s \times I_t$ and $M_{s,t}$ is the number of observed dyads in block (s,t).

Proof. Fix $(s,t) \in [\![k]\!]^2$. Conditional on the latent positions $\mathcal{U}$, we have

$$\Phi_{s,t} = \mathbb{E}\left[H_{s,t}\,|\,\mathcal{U}\right] = \frac{1}{M_{s,t}}\sum_{m\in[\![M]\!]}\sum_{i\in J_s^{(m)}}\sum_{j\in J_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right),$$

Therefore

$$\begin{aligned}\mathbb{E}\big[(H_{s,t} - \Phi_{s,t})^2\,\big|\,\mathcal{U}\big] &= \mathrm{Var}\left(\frac{1}{M_{s,t}}\sum_{m\in[\![M]\!]}\sum_{i\in J_s^{(m)}}\sum_{j\in J_t^{(m)}} A_{i,j}^{(m)} \mid U_{i,(m)}, U_{j,(m)}\,\Big|\,\mathcal{U}\right)\\ &\leq \frac{1}{M_{s,t}^2}\sum_{m\in[\![M]\!]}\sum_{i\in J_s^{(m)}}\sum_{j\in J_t^{(m)}} \mathrm{Var}\left(A_{i,j}^{(m)} \mid U_{i,(m)}, U_{j,(m)}\right).\end{aligned}$$

Since $A_{i,j}^{(m)}|\mathcal{U} \sim \mathrm{Bernoulli}(W\left(U_{i,(m)}, U_{j,(m)}\right)$,

$$\mathrm{Var}\left((A_{i,j}^{(m)} \mid U_{i,(m)}, U_{j,(m)}\right) = W\left(U_{i,(m)}, U_{j,(m)}\right)\left(1 - W\left(U_{i,(m)}, U_{j,(m)}\right)\right) \leq \frac{1}{4}.$$

Hence

$$\mathbb{E}\big[(H_{s,t} - \Phi_{s,t})^2\,\big|\,\mathcal{U}\big] \;\leq\; \frac{1}{M_{s,t}^2}\cdot M_{s,t}\cdot\frac{1}{4} = \frac{1}{4M_{s,t}}.$$

Taking the expectation over the latent positions gives

$$\mathbb{E}\left[(H_{s,t}-\Phi_{s,t})^2\right] \;\le\; \mathbb{E}\Big[\frac{1}{4M_{s,t}}\Big].$$

This yields

$$\mathbb{E}\left[\left\|W^{\text{oracle}}-W^{\text{approx}}\right\|_{L^2}^2\right]=\frac{1}{k^2}\sum_{s,t\in[\![k]\!]^2}\mathbb{E}\left[(H_{s,t}-\Phi_{s,t})^2\right] \;\le\; \frac{1}{4k^2}\sum_{s,t\in[\![k]\!]^2}\mathbb{E}\Big[\frac{1}{M_{s,t}}\Big].$$

□

Lemma A.4 (Stochastic error)**.** *Assume that for all vertices $(i,m)\in[\![n^{(m)}]\!]\times[\![M]\!]$ the latent position errors satisfy*

$$\left|\widehat{U}^{\text{JGS}}_{i,(m)}-U_{i,(m)}\right| \;\le\; \eta_i^{(m)},$$

where $\eta_i^{(m)}$ is defined in the main paper in (11)*. Then*

$$\mathbb{E}\left[\left\|\widehat{W}^{\text{JGS}}-W^{\text{oracle}}\right\|_{L^2}^2\right] \;\le\; \frac{1}{k^2}\sum_{s,t\in[\![k]\!]^2}\mathbb{E}\left[\frac{3}{4\widehat{M}_{s,t}}+\frac{3}{4M_{s,t}}+\frac{48}{M_{s,t}^2}\left(\sum_{m\in[\![M]\!]}\left(|J_t^{(m)}|\,|B_s^{(m)}|+|J_s^{(m)}|\,|B_t^{(m)}|\right)\right)^2\right],$$

where $J_r^{(m)}=\{\,i: U_{i,(m)}\in I_r\,\}$ and $M_{s,t}=|\Omega_{s,t}|$, and

$$C_s^{(m)}:=\left\{i\in[\![n^{(m)}]\!]:\operatorname{dist}(U_{i,(m)},\partial I_s)\le\eta_i^{(m)}\right\}.$$

Proof. We note that

$$\mathbb{E}\left[\left\|\widehat{W}^{\text{JGS}}-W^{\text{oracle}}\right\|_{L^2}^2\right]=\frac{1}{k^2}\sum_{s,t\in[\![k]\!]^2}\mathbb{E}\big[(\widehat{H}^{\text{JGS}}_{s,t}-H_{s,t})^2\big]. \tag{14}$$

By the elementary inequality $(a+b+c)^2\le 3(a^2+b^2+c^2)$, we obtain

$$\mathbb{E}\left[(\widehat{H}^{\text{JGS}}_{s,t}-H_{s,t})^2\right] \;\le\; 3\,\mathbb{E}\left[(\widehat{H}^{\text{JGS}}_{s,t}-\widehat{\Phi}_{s,t})^2\right]+3\,\mathbb{E}\left[(\widehat{\Phi}_{s,t}-\Phi_{s,t})^2\right]+3\,\mathbb{E}\left[(\Phi_{s,t}-H_{s,t})^2\right]. \tag{15}$$

For the first term, by the conditional independence of Bernoulli edges, we compute

$$\begin{aligned}
\mathbb{E}\left[(\widehat{H}^{\text{JGS}}_{s,t}-\widehat{\Phi}_{s,t})^2\right] &= \mathbb{E}\left[\operatorname{Var}\left(\widehat{H}^{\text{JGS}}_{s,t}\,|\,\mathcal{U}\right)\right]\\
&= \mathbb{E}\left[\frac{1}{\widehat{M}_{s,t}^2}\sum_{m\in[\![M]\!]}\sum_{i\in\widehat{J}_s^{(m)}}\sum_{j\in\widehat{J}_t^{(m)}}\operatorname{Var}\big(A_{i,j}^{(m)}\mid U_{i,(m)},U_{j,(m)}\big)\right]\\
&\le \mathbb{E}\left[\frac{1}{\widehat{M}_{s,t}^2}\sum_{m\in[\![M]\!]}\sum_{i\in\widehat{J}_s^{(m)}}\sum_{j\in\widehat{J}_t^{(m)}}\frac{1}{4}\right]\\
&= \mathbb{E}\left[\frac{1}{4\widehat{M}_{s,t}}\right],
\end{aligned} \tag{16}$$

where we used $\operatorname{Var}\left((A_{i,j}^{(m)}\mid U_{i,(m)},U_{j,(m)}\right)\le 1/4$. Similarly, for the third term, we obtain

$$\mathbb{E}\left[(H_{s,t}-\Phi_{s,t})^2\right]\le\mathbb{E}\left[\frac{1}{4M_{s,t}}\right]. \tag{17}$$

We now turn to the middle term. By definition,

$$\Phi_{s,t} = \frac{1}{M_{s,t}} \sum_{m\in[\![M]\!]} \sum_{i\in J_s^{(m)}} \sum_{j\in J_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right),$$

$$\widehat{\Phi}_{s,t} = \mathbb{E}\left[\frac{1}{\widehat{M}_{s,t}} \sum_{m\in[\![M]\!]} \sum_{i\in \widehat{J}_s^{(m)}} \sum_{j\in \widehat{J}_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right) \,\middle|\, \mathcal{U} \right].$$

We add and subtract

$$\frac{1}{M_{s,t}} \sum_{m\in[\![M]\!]} \sum_{i\in \widehat{J}_s^{(m)}} \sum_{j\in \widehat{J}_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right)$$

inside the conditional expectation to obtain

$$\begin{aligned}
\widehat{\Phi}_{s,t} - \Phi_{s,t} &= \mathbb{E}\left[\Big(\frac{1}{\widehat{M}_{s,t}} - \frac{1}{M_{s,t}}\Big) \sum_{m\in[\![M]\!]} \sum_{i\in \widehat{J}_s^{(m)}} \sum_{j\in \widehat{J}_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right) \,\middle|\, \mathcal{U} \right] \\
&\quad + \frac{1}{M_{s,t}} \mathbb{E}\left[\sum_{m\in[\![M]\!]} \Big(\sum_{i\in \widehat{J}_s^{(m)}} \sum_{j\in \widehat{J}_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right) - \sum_{i\in J_s^{(m)}} \sum_{j\in J_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right) \Big) \,\middle|\, \mathcal{U} \right] \\
&= \mathbb{E}\left[\Big(\frac{1}{\widehat{M}_{s,t}} - \frac{1}{M_{s,t}}\Big) \sum_{m\in[\![M]\!]} \sum_{i\in \widehat{J}_s^{(m)}} \sum_{j\in \widehat{J}_t^{(m)}} W\left(U_{i,(m)}, U_{j,(m)}\right) \,\middle|\, \mathcal{U} \right] \\
&\quad + \frac{1}{M_{s,t}} \mathbb{E}\left[\sum_{(i,j;m)\in D_{s,t}} W\left(U_{i,(m)}, U_{j,(m)}\right) \,\middle|\, \mathcal{U} \right],
\end{aligned}$$

where $D_{s,t} = \Omega_{s,t} \triangle \widehat{\Omega}_{s,t} = \big(\widehat{\Omega}_{s,t} \setminus \Omega_{s,t}\big) \cup \big(\Omega_{s,t} \setminus \widehat{\Omega}_{s,t}\big)$ the symmetric difference of the index sets, with

$$\Omega_{s,t} = \bigcup_{m\in[\![M]\!]} \{(i,j;m) : i \in J_s^{(m)},\, j \in J_t^{(m)}\}, \quad \text{and} \quad \widehat{\Omega}_{s,t} = \bigcup_{m\in[\![M]\!]} \{(i,j;m) : i \in \widehat{J}_s^{(m)},\, j \in \widehat{J}_t^{(m)}\}.$$

Since $0 \le W \le 1$, we obtain the bound

$$\begin{aligned}
|\widehat{\Phi}_{s,t} - \Phi_{s,t}| &\le \mathbb{E}\left[\left| \frac{1}{\widehat{M}_{s,t}} - \frac{1}{M_{s,t}} \right| \sum_{m\in[\![M]\!]} \sum_{i\in \widehat{J}_s^{(m)}} \sum_{j\in \widehat{J}_t^{(m)}} 1 \,\middle|\, \mathcal{U} \right] \quad + \frac{1}{M_{s,t}} \mathbb{E}\left[\left| \sum_{(i,j;m)\in D_{s,t}} 1 \,\middle|\, \mathcal{U} \right| \right] \\
&\le \mathbb{E}\left[\left| \frac{1}{\widehat{M}_{s,t}} - \frac{1}{M_{s,t}} \right| \widehat{M}_{s,t} \,\middle|\, \mathcal{U} \right] + \frac{1}{M_{s,t}} \mathbb{E}\left[|D_{s,t}| \,|\, \mathcal{U} \right] \\
&= \frac{1}{M_{s,t}} \mathbb{E}\left[\left| \widehat{M}_{s,t} - M_{s,t} \right| \,\middle|\, \mathcal{U} \right] + \frac{1}{M_{s,t}} \mathbb{E}\left[|D_{s,t}| \,|\, \mathcal{U} \right].
\end{aligned}$$

Since $|M_{s,t} - \widehat{M}_{s,t}| \le |D_{s,t}|$, we deduce

$$|\widehat{\Phi}_{s,t} - \Phi_{s,t}| \;\le\; \frac{2}{M_{s,t}} \mathbb{E}\left[|D_{s,t}| \,|\, \mathcal{U} \right]. \tag{18}$$

Now, under the event $|\widehat{U}^{\mathrm{JGS}}_{i,(m)} - U_{i,(m)}| \le \eta_i^{(m)}$, only a node $i \in J_s^{(m)}$ can be misclassified if $U_{i,(m)}$ lies within $\eta_i^{(m)}$ of the boundary of I_s with high probability. Define

$$B_s^{(m)} := \Big\{ i \in [\![n^{(m)}]\!] : \mathrm{dist}(U_{i,(m)}, \partial I_s) \le \eta_i^{(m)} \Big\}.$$

Then every misclassified vertex of $J_s^{(m)}$ belongs to $B_s^{(m)}$. If a vertex i leaves $J_s^{(m)}$ due to misclassification, it can affect at most $|J_t^{(m)}|$ dyads in block (s,t). By symmetry, misclassifications in $J_t^{(m)}$ contribute at most $|J_s^{(m)}|\,|B_t^{(m)}|$ dyads in block (s,t). Therefore,

$$|D_{s,t}| \;\le\; \sum_{m\in[\![M]\!]} \Big(|J_t^{(m)}|\,|B_s^{(m)}| + |J_s^{(m)}|\,|B_t^{(m)}|\Big).$$

Plugging this into (18), we obtain

$$|\widehat{\Phi}_{s,t} - \Phi_{s,t}| \le \frac{2}{M_{s,t}} \sum_{m\in[\![M]\!]} \Big(|J_t^{(m)}|\,|B_s^{(m)}| + |J_s^{(m)}|\,|B_t^{(m)}|\Big),$$

and hence

$$\mathbb{E}\left[(\widehat{\Phi}_{s,t} - \Phi_{s,t})^2\right] \;\le\; \mathbb{E}\left[\left(\frac{2}{M_{s,t}} \sum_{m\in[\![M]\!]} \big(|J_t^{(m)}|\,|B_s^{(m)}| + |J_s^{(m)}|\,|B_t^{(m)}|\big)\right)^2\right]. \tag{19}$$

Combining the equations (14)–(19) concludes the proof. □

Proof of Theorem 3.3. We start from the decomposition of the MISE

$$\mathbb{E}\left[\big\|\widehat{W}^{\mathrm{JGS}} - W\big\|_{L^2}^2\right] \le 3\,\mathbb{E}\left[\big\|\widehat{W}^{\mathrm{JGS}} - W^{\mathrm{oracle}}\big\|_{L^2}^2\right] \quad + 3\,\mathbb{E}\left[\big\|W^{\mathrm{oracle}} - W^{\mathrm{approx}}\big\|_{L^2}^2\right] + 3\,\mathbb{E}\left[\big\|W^{\mathrm{approx}} - W\big\|_{L^2}^2\right].$$

By Lemma A.2, Lemma A.3, Lemma A.4, we have

$$\begin{aligned}\mathbb{E}\|\widehat{W}^{\mathrm{JGS}} - W\|_{L^2}^2 \le \frac{3C\,L^2}{k^2} + \frac{3}{4k^2} \sum_{s,t\in[\![k]\!]^2} \mathbb{E}\Big[\frac{1}{M_{s,t}}\Big] + \frac{1}{k^2} \sum_{s,t\in[\![k]\!]^2} \mathbb{E}\left[\frac{3}{4\widehat{M}_{s,t}} + \frac{3}{4M_{s,t}}\right] \\ + \mathbb{E}\left[\frac{48}{M_{s,t}^2}\left(\sum_{m\in[\![M]\!]} \big(|J_t^{(m)}|\,|B_s^{(m)}| + |J_s^{(m)}|\,|B_t^{(m)}|\big)\right)^2\right]\end{aligned}$$

For fixed M, when $n_{\max}\to\infty$, we have the following asymptotic equivalences

$$M_{s,t} \asymp \frac{S}{k^2}, \qquad \widehat{M}_{s,t} \asymp \frac{S}{k^2}, \quad \text{where} \quad S = \sum_{m\in[\![M]\!]} (n^{(m)})^2,$$

so that

$$\frac{1}{M_{s,t}}, \frac{1}{\widehat{M}_{s,t}} \lesssim \frac{k^2}{S}, \qquad \frac{1}{M_{s,t}^2} \lesssim \frac{k^4}{S^2}.$$

Moreover, since the number $|J_s^{(m)}|$ of nodes from network m in interval I_s follows a binomial distribution $|J_s^{(m)}| \sim \mathrm{Binomial}(n^{(m)}, 1/k)$ with mean $n^{(m)}/k$, Markov's inequality yields for any $\lambda > 0$,

$$\mathbb{P}\left(|J_s^{(m)}| \ge \lambda\,\frac{n^{(m)}}{k}\right) \le \frac{1}{\lambda}.$$

Hence, $|J_s^{(m)}| = O_{\mathbb{P}}(n^{(m)}/k)$. Moreover, for any t and s,

$$|J_t^{(m)}|\,|B_s^{(m)}| + |J_s^{(m)}|\,|B_t^{(m)}| \;\lesssim\; \frac{n_{\max}}{k}\,\big(|B_s^{(m)}| + |B_t^{(m)}|\big).$$

Combining the bounds gives

$$\mathbb{E}\left[\|\widehat{W}^{\mathrm{JGS}} - W\|_{L^2}^2\right] \lesssim \frac{L^2}{k^2} + \frac{k^2}{S} + \frac{k^4}{S^2}\,\frac{1}{k^2} \sum_{s,t\in[\![k]\!]^2} \frac{n_{\max}^2}{k^2}\,\mathbb{E}\left[\left(\sum_{m\in[\![M]\!]} (|B_s^{(m)}| + |B_t^{(m)}|)\right)^2\right].$$

Now,

$$\mathbb{E}\left[\left(\sum_{m\in[\![M]\!]}(|B_s^{(m)}|+|B_t^{(m)}|)\right)^2\right]=\sum_{m\in[\![M]\!]}\mathbb{E}\left[\left(|B_s^{(m)}|+|B_t^{(m)}|\right)^2\right]+\sum_{m\neq\ell}\mathbb{E}\left[|B_s^{(m)}|+|B_t^{(m)}|\right]\,\mathbb{E}\left[|B_s^{(\ell)}|+|B_t^{(\ell)}|\right].$$

Using $\mathrm{Var}(X)\le\mathbb{E}[X^2]$ and $|\mathrm{Cov}(X,Y)|\le\frac{1}{2}(\mathrm{Var}(X)+\mathrm{Var}(Y))$, we obtain

$$\mathbb{E}\left[\left(|B_s^{(m)}|+|B_t^{(m)}|\right)^2\right]\le 2(\mu_s^{(m)}+\mu_t^{(m)})+(\mu_s^{(m)}+\mu_t^{(m)})^2,$$

where

$$\mu_s^{(m)}=\mathbb{E}\left[|B_s^{(m)}|\right]=\sum_{i\in[\![n]\!]^{(m)}}\mathbb{P}\big(\mathrm{dist}(U_{i,(m)},\partial I_s)\le\eta_i^{(m)}\big)\;\le\;2\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}.$$

Hence

$$\begin{aligned}\mathbb{E}\left[\left(\sum_{m\in[\![M]\!]}(|B_s^{(m)}|+|B_t^{(m)}|)\right)^2\right]&\le 2\sum_{m\in[\![M]\!]}(\mu_s^{(m)}+\mu_t^{(m)})+\left(\sum_{m\in[\![M]\!]}(\mu_s^{(m)}+\mu_t^{(m)})\right)^2\\&\le 4\sum_{m\in[\![M]\!]}\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}+\left(2\sum_{m\in[\![M]\!]}\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}\right)^2.\end{aligned}$$

Putting everything together, we arrive at

$$\mathbb{E}\left[\|\widehat{W}^{\mathrm{JGS}}-W\|_{L^2}^2\right]\lesssim\frac{L^2}{k^2}+\frac{k^2}{S}+\frac{n_{\max}^2k^2}{S^2}\left\{4\sum_{m\in[\![M]\!]}\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}+\left(2\sum_{m\in[\![M]\!]}\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}\right)^2\right\}.$$

For the graphs of bounded size $n^{(m)}$ we have

$$\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}\;\le\;C,$$

so that

$$\sum_{m\in[\![M]\!]}\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}\;\le\;MC+o(1).$$

Similarly,

$$\left(\sum_{m\in[\![M]\!]}\sum_{i\in[\![n]\!]^{(m)}}\eta_i^{(m)}\right)^2\;\le\;(MC+o(1))^2.$$

Substituting into the MISE bound gives

$$\mathbb{E}\left[\|\widehat{W}^{\mathrm{JGS}}-W\|_{L^2}^2\right]\lesssim\frac{L^2}{k^2}+\frac{k^2}{S}+\frac{n_{\max}^2k^2}{S^2}\,(MC+o(1))^2.$$

Since $S=\sum_{m\in[\![M]\!]}(n^{(m)})^2\gtrsim n_{\max}^2$, each term on the right-hand side tends to zero as $n_{\max}\to\infty$ and $k/n_{\max}\to 0$. Therefore,

$$\mathrm{MISE}\Big(\widehat{W}^{\mathrm{JGS}},W\Big)\;\longrightarrow\;0.$$

□

A.4 Choice of the number of blocks k

For Lipschitz-continuous graphons, a standard bias–variance calculation for histogram estimators in the single graph setting gives

$$\mathrm{MISE}(\widehat{W}) \;=\; \mathcal{O}\!\left(\frac{k^2}{n^2}+\frac{1}{k^2}\right), \tag{20}$$

see, e.g., Gao et al. (2014, Lemma 2.1 and the discussion after Theorem 1.2); see also Klopp et al. (2017). Balancing the two terms yields the heuristic choice

$$k_{\mathrm{select}} \;\asymp\; n^{1/2},$$

which corresponds to a risk of order $1/n$.

From a minimax perspective, Gao et al. (2014) established that in the latent (non-aligned) setting, the optimal rate for estimating the probability matrix associated with a Lipschitz graphon under the mean-squared error is of order $\mathcal{O}((\log n)/n)$.

In the multiple-network setting, when the number of networks M is fixed, the same argument applies with n^2 replaced by the total number of observed dyads

$$S \;=\; \sum_{m\in[\![M]\!]} (n^{(m)})^2,$$

leading to

$$\mathrm{MISE}(\widehat{W}) \;=\; \mathcal{O}\!\left(\frac{k^2}{S}+\frac{1}{k^2}\right) \qquad \text{and} \qquad k_{\mathrm{select}} \;\asymp\; S^{1/4}. \tag{21}$$

Avoiding empty blocks Denote $M_{s,t}$ the number of observed dyads in block (s,t). To ensure all blocks are nonempty with high probability, that is $M_{s,t}>0$, we analyze the conditions on k.

Fix a block (s,t). For off-diagonal blocks ($s\neq t$), $M_{s,t}=0$ occurs if and only if, in every graph m, at least one of the classes s or t is empty. Define the event

$$E_m = \left\{|J_s^{(m)}|\geq 1\right\}\cap\left\{|J_t^{(m)}|\geq 1\right\}.$$

Then $\{M_{s,t}=0\}=\bigcap_{m\in[\![M]\!]} E_m^c$, and by independence across graphs,

$$\mathbb{P}(M_{s,t}=0)=\prod_{m\in[\![M]\!]}\mathbb{P}(E_m^c).$$

Now,

$$\mathbb{P}(E_m^c)\leq \mathbb{P}\left(|J_s^{(m)}|=0\right)+\mathbb{P}\left(|J_t^{(m)}|=0\right).$$

Since each node is assigned to class s with probability $1/k$,

$$\mathbb{P}\left(|J_s^{(m)}|=0\right)=(1-1/k)^{n^{(m)}}\leq e^{-n^{(m)}/k},$$

so $\mathbb{P}(E_m^c)\leq 2e^{-n^{(m)}/k}$. For diagonal blocks ($s=t$), the same bound holds since $\mathbb{P}(E_m^c)\leq e^{-n^{(m)}/k}\leq 2e^{-n^{(m)}/k}$. By combining these bounds, we obtain

$$\mathbb{P}(M_{s,t}=0)\leq 2^M\exp\left(-\frac{1}{k}\sum_{m\in[\![M]\!]}n^{(m)}\right)=\exp\left(-\frac{N}{k}+M\log 2\right),$$

where $N=\sum_{m\in[\![M]\!]}n^{(m)}$. A union bound over all k^2 blocks yields

$$\mathbb{P}\left(\exists(s,t):M_{s,t}=0\right)\leq k^2\exp\left(-\frac{N}{k}+M\log 2\right).$$

To ensure this probability vanishes asymptotically, we require

$$\frac{N}{k} - M\log 2 - 2\log k \longrightarrow \infty,$$

which is guaranteed by

$$\frac{N}{k} \;\geq\; c\,(M + \log k) \tag{22}$$

for some constant $c > 0$. Since $\log k \leq \log N$, a simpler sufficient condition is

$$k \;\leq\; \frac{N}{c\,(M + \log N)}. \tag{23}$$

Practical choice of k Combining the rate-optimal choice $k \asymp S^{1/4}$ with the empty-block constraint yields the practical rule

$$k \;\asymp\; \min\left\{ S^{1/4},\;\; \frac{N}{c\,(M + \log N)} \right\}. \tag{24}$$

Special cases

- Single graph of size n: $S = n^2$ gives $k \asymp \sqrt{n}$ (classical rate).
- M graphs of comparable size n: $S \asymp Mn^2$ gives $k \asymp M^{1/4}\sqrt{n}$, when (23) holds.
- One dominant graph: $S \asymp n_{\max}^2$ gives $k \asymp \sqrt{n_{\max}}$.

B ADDITIONAL EXPERIMENTS

B.1 Table of true graphons

The true graphons used in the simulation experiments are listed in Table 3.

Table 3: Synthetic graphons used in the simulation study (from Azizpour et al. (2025)).

ID	Graphon function $W(u, v)$
	Monotone graphons
1	$u\,v$
2	$\exp\big(-(u^{0.7} + v^{0.7})\big)$
3	$\frac{1}{4}\,(u^2 + v^2 + \sqrt{u} + \sqrt{v})$
4	$\frac{1}{2}(u + v)$
5	$\big[1 + \exp\big(-2(u^2 + v^2)\big)\big]^{-1}$
6	$\big[1 + \exp\big(-(\max(u, v)^2 + \min(u, v)^4)\big)\big]^{-1}$
7	$\exp\big(-\max(u, v)^{0.75}\big)$
8	$\exp\left[-\frac{1}{2}\big(\min(u, v) + \sqrt{u} + \sqrt{v}\big)\right]$
9	$\log\big(1 + \max(u, v)\big)$
	Non-monotone graphons
10	$\lvert u - v\rvert$
11	$1 - \lvert u - v\rvert$
12	$0.8\,\mathrm{I}_2 \otimes \mathbb{1}_{[0,1/2]^2}$
13	$0.8\,(1 - \mathrm{I}_2) \otimes \mathbb{1}_{[0,1/2]^2}$

To illustrate the type of networks generated in our simulation study, Figure 8 displays sample graphs drawn from several synthetic graphons listed in Table 3. Each network is generated using the sampling procedure described in the main text.

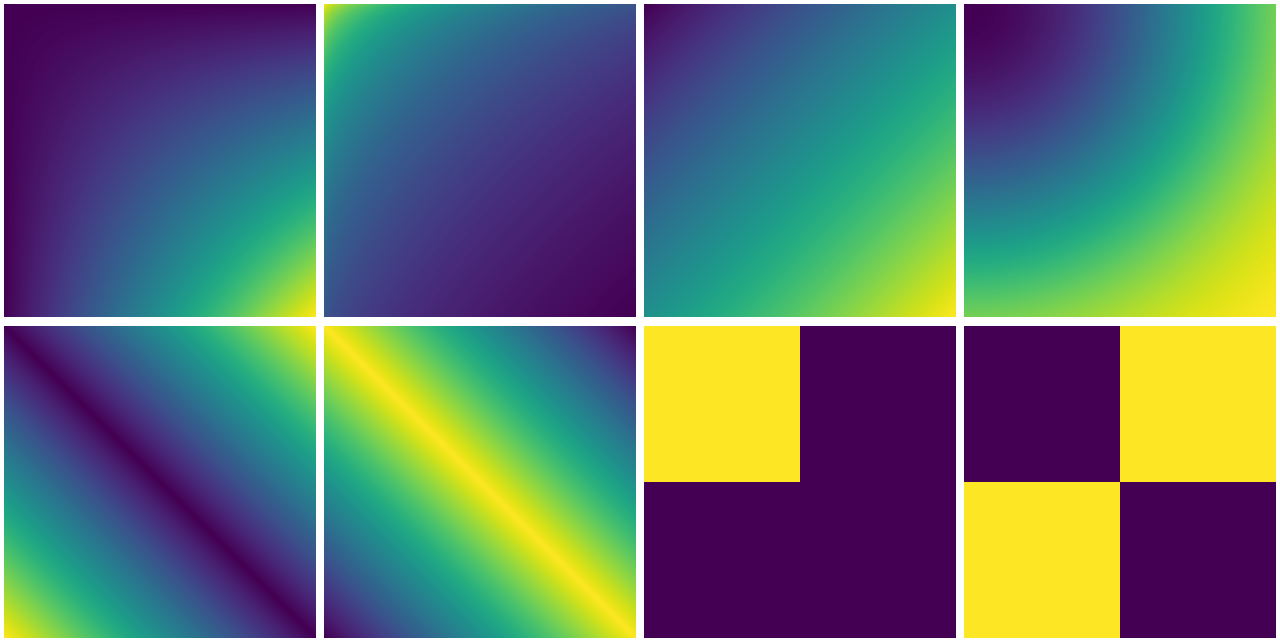

Figure 8: Examples of simulated graphs generated from the synthetic graphons used in the experiments. Each panel shows a graph sampled from a different graphon listed in Table 3.

B.2 Baseline methods and hyperparameter settings

Below we provide additional implementation details for the baseline methods used in the simulation study. Unless otherwise stated, the hyperparameters were chosen following the recommendations of the original papers or of recent empirical studies. This ensures a fair and reproducible comparison with the proposed JGS estimator.

All methods are applied to each graph separately when required by their original formulation. In the multi-network setting considered in this paper, the final graphon estimate is obtained either by pooling the graphs (when the method allows it) or by averaging the estimates across graphs. The largest graph size in the collection is denoted by $n_{\max}$.

- ▷ **SBA (Chan and Airoldi, 2014)** The Sorting-and-Binning Algorithm estimates latent node ordering from empirical degrees and then constructs a block histogram of the graphon. The method depends on a threshold parameter Δ controlling the binning step. We adopt $\Delta = 0.1$, following the setting used in Xu et al. (2021).
- ▷ **SAS (Airoldi et al., 2013)** The Sorting-And-Smoothing estimator first orders nodes according to empirical degrees and then applies a histogram smoothing step. The main hyperparameter is the histogram window size h. In our experiments, we use $h = \log(n_{\max})$, as recommended in the original SAS paper and adopted in subsequent studies such as Xu et al. (2021).
- ▷ **USVT (Chatterjee, 2015)** Universal Singular Value Thresholding estimates the probability matrix by truncating the singular value decomposition of the adjacency matrix. The key hyperparameter is the singular value threshold τ used to remove noisy components. We set $\tau = 0.2\sqrt{n_{\max}}$, following the calibration used in Azizpour et al. (2025).
- ▷ **SGWB (Xu et al., 2021)** The SGWB estimator relies on an optimal transport formulation to align graphs through Gromov–Wasserstein barycenters. The method involves several optimization parameters. We follow the settings recommended in the original work: proximal weight $\beta = 0.005$, number of outer iterations $L = 5$, number of Sinkhorn iterations $S = 10$, and smoothness regularization weight $\alpha = 0.0002$.
- ▷ **SIGL (Azizpour et al., 2025)** SIGL estimates the graphon using a neural architecture combining a graph neural network for latent position learning and an implicit neural representation (INR) for the graphon function. We use the hyperparameter configuration proposed in Azizpour et al. (2025): hidden layer

sizes inr_dim_hidden = [20, 20] and gnn_dim_hidden = [8, 8, 8], number of epochs n_epochs = 100, checkpoint interval epoch_show = 20, frequency parameter $w_0 = 10$, learning rate lr = 0.01, batch size batch_size = 1024, and resolution Res = 1000 used to sample the estimated graphon.

All methods are implemented using the same graph collections and evaluation protocol in order to ensure fair comparisons of both estimation accuracy and computational cost. All experiments were conducted on a personal workstation equipped with a 13th Gen Intel Core i5-1335U CPU running Linux.

B.3 JGS implementation

Algorithm 1 summarizes the implementation of the JGS estimator. In practice, the procedure only requires computing normalized degrees across all graphs, performing a global sorting step to obtain latent position estimates, and constructing a block histogram of the graphon using these estimated positions. The method naturally aggregates information from the entire collection of networks through the pooled ordering of nodes. Additional implementation details, including the automatic choice of k and optional post-processing steps, are provided in the publicly available code accompanying this paper: `https://github.com/RolandBonifaceSogan/JGS-graphon`..

Algorithm 1: Joint Graph Sorting (JGS) Estimation

Input: Adjacency matrices $\{A^{(1)}, \dots, A^{(M)}\}$ of sizes $n^{(1)}, \dots, n^{(M)}$; number of blocks k
Output: Estimated graphon matrix $\widehat{H}^{\mathrm{JGS}} \in [0,\,1]^{k\times k}$

Step 1: Latent Position Estimation
foreach $m \in [\![M]\!]$ **do**
  foreach $i \in \{1, \dots, n^{(m)}\}$ **do**
    Compute normalized degrees $\widehat{d}^{\mathrm{norm}}_{i,(m)} = \dfrac{1}{n^{(m)}} \sum_{j\in[\![n^{(m)}]\!]} A^{(m)}_{i,j}$;

Concatenate all degrees into a vector of length $N = \sum_{m\in[\![M]\!]} n^{(m)}$;
Sort this vector and denote by $\widehat{\sigma}^{\mathrm{JGS}}$ the associated permutation ;
foreach *node* (i, m) **do**
  Compute the latent position estimate $\widehat{U}^{\mathrm{JGS}}_{i,(m)} = \dfrac{(\widehat{\sigma}^{\mathrm{JGS}})^{-1}(i,m)}{N} - \dfrac{1}{2N}$;

Step 2: Histogram Estimation
Partition $[0,\,1]$ into k intervals $I_1, \dots, I_k$ of length $1/k$;
foreach $s, t \in [\![k]\!]$ **do**
  Let $\widehat{J}_s(m) = \left\{ i : \widehat{U}^{\mathrm{JGS}}_{i,(m)} \in I_s \right\}$ and $\widehat{J}^{(m)}_t = \left\{ i : \widehat{U}^{\mathrm{JGS}}_{i,(m)} \in I_t \right\}$;
  Compute $\widehat{H}^{\mathrm{JGS}}_{s,t} = \dfrac{\sum_{m\in[\![M]\!]} \sum_{i\in\widehat{J}^{(m)}_s} \sum_{j\in\widehat{J}^{(m)}_t} A^{(m)}_{i,j}}{1 \vee \sum_{m\in[\![M]\!]} |\widehat{J}^{(m)}_s||\widehat{J}^{(m)}_t|}, \quad s, t \in [\![k]\!]$

B.4 G-Mixup

The training parameters are kept identical for the three methods to ensure a fair comparison. Optimization is performed using the Adam algorithm (Kingma and Ba, 2014), with an initial learning rate of 0.01, which is halved every 100 epochs. The batch size is set to 128. Datasets are split into 70% for training, 10% for validation, and 20% for testing. The test accuracy reported in Table 2 corresponds to the accuracy obtained at the epoch achieving the best validation score, averaged over ten independent runs.

The Graph Isomorphism Network (GIN) architecture (Xu et al., 2019) is used as the base classifier. Following the protocol of Han et al. (2022), the same model architecture and training procedure are applied across all methods so that the only difference lies in the graph augmentation step. This ensures that the performance differences observed in Table 2 can be attributed to the quality of the graphon estimates used within the G-Mixup procedure.

In the G-Mixup framework, graphons are estimated exclusively from the training graphs and are then used

Table 4: Computational complexities of the considered methods in the multi-network setting.

Method	Complexity
SBA	$\mathcal{O}(M\,k\,n_{\max}\log n_{\max})$
SAS	$\mathcal{O}(M\,n_{\max}\log n_{\max} + k^2\log k^2)$
USVT	$\mathcal{O}(M\,n_{\max}^3)$
SGWB	$\mathcal{O}(L\,S\,M\,(S\,k + n_{\max}k^2))$
SIGL	$\mathcal{O}(M\,L\,n_{\max}^2)$
JGS	$\boldsymbol{\mathcal{O}(N\log N + S)}$

to generate additional synthetic graphs via interpolation. More precisely, given two estimated class-specific graphons W_1 and W_2, a mixed graphon is constructed as

$$W_\lambda = \lambda W_1 + (1-\lambda)W_2, \tag{25}$$

where λ is a mixing coefficient. Graphs are subsequently sampled from W_λ to enrich the training dataset.

In our experiments, we generate an additional 20% of training graphs using this interpolation mechanism. The mixing parameter λ is sampled uniformly from the interval $[0.1, 0.2]$, following the setup of Han et al. (2022). Restricting λ to this range produces synthetic graphs that remain close to the original class structures while still introducing meaningful variability in the training data.

Finally, all graphon estimators are applied independently within each class of the training data, ensuring that the generated graphs preserve class-specific structural patterns. This setup allows us to evaluate the impact of improved graphon estimation on downstream graph classification performance within the G-Mixup framework.

B.5 Comparison of running times

To complement the results reported in the main paper regarding the computational efficiency of the proposed estimator, we provide here a more detailed comparison of execution times across all considered methods. In addition to the most accurate approaches (JGS, SIGL, and SGWB), we also include classical estimators such as SBA, SAS, and USVT. Although these methods generally yield lower estimation accuracy, they are often considered computationally inexpensive and therefore serve as useful baselines for evaluating the trade-off between statistical accuracy and computational cost.

Table 4 summarizes the asymptotic computational complexities of the different estimators in the multi-network setting, where $N = \sum_m n^{(m)}$ denotes the total number of nodes, M the number of graphs, k the number of histogram blocks, $S = \sum_{m\in[\![M]\!]}(n^{(m)})^2$ the total number of observed dyads, and L the number of iterations for iterative methods. As indicated by the table, the proposed JGS estimator has one of the lowest theoretical computational complexities.

To empirically evaluate the practical computational cost of the different approaches, we conducted additional simulation experiments using a fixed experimental setting with $M = 50$ networks and heterogeneous graph sizes $n^{(m)}$ around $n = 100$. For each method, both the estimation error (measured by the MISE) and the execution time were recorded over several repetitions.

Figure 9 reports the distribution of execution times across repetitions. As expected, iterative optimization-based methods such as SIGL and SGWB are substantially slower than the other approaches. In contrast, JGS remains computationally efficient and exhibits execution times comparable to the fastest baseline methods. This confirms that the joint sorting procedure can be implemented with very low computational overhead even when multiple graphs are processed simultaneously.

Figure 10 presents the corresponding distribution of estimation errors. These results highlight the substantial accuracy advantage of JGS over classical methods such as SBA and SAS. While USVT remains competitive in terms of estimation error, it does not provide a clear advantage in terms of computational efficiency when compared with JGS. Overall, these results illustrate that JGS achieves an attractive balance between statistical accuracy and computational efficiency.

Together, these results confirm that JGS provides a particularly favorable trade-off between estimation accuracy and computational cost, making it well suited for large collections of networks.

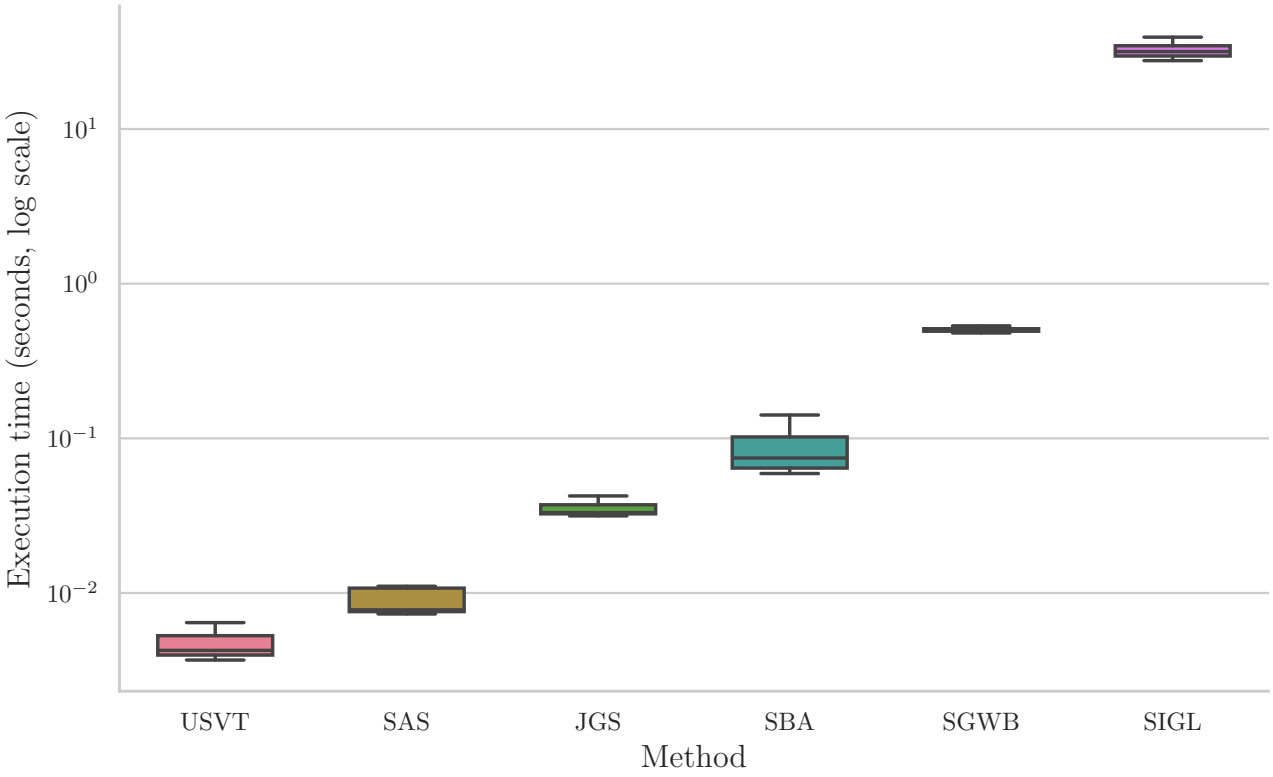

Figure 9: Distribution of execution times across repetitions for the different graphon estimation methods (log scale).

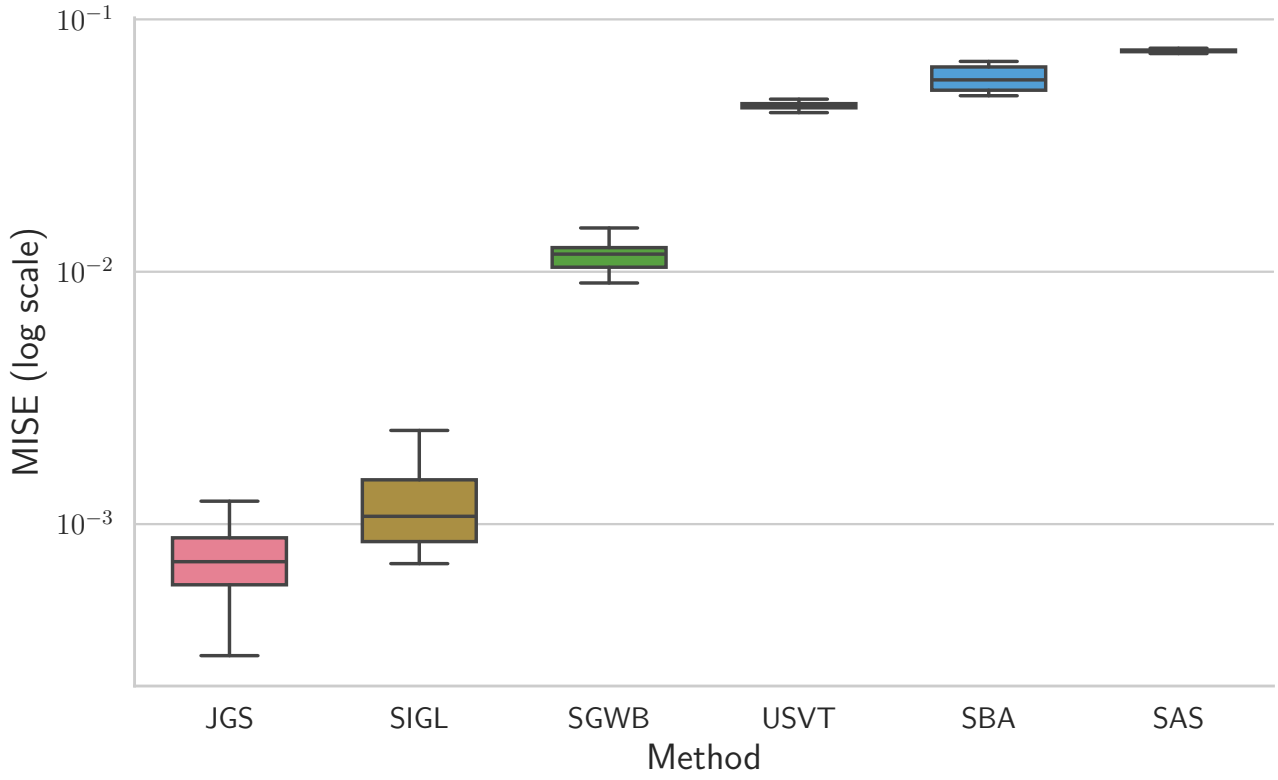

Figure 10: Distribution of estimation errors (MISE) across repetitions for the different graphon estimation methods (log scale).

B.6 Sensitivity of JGS to the choice of k

Figure 11 provides a detailed view of how the performance of JGS depends on the number of histogram blocks k. Several features can be observed.

First, for the identifiable graphons (IDs 1–9), the MISE curves display a clear bias–variance trade-off. When k is too small, the histogram resolution is too coarse and the resulting estimator suffers from a large approximation bias, which explains the relatively high errors on the left-hand side of the figure. As k increases, the histogram becomes more flexible and captures the structure of the graphon more accurately, leading to a rapid decrease of the MISE. Beyond a certain point, however, the number of observations per block becomes too small, which increases the estimation variance and causes the error to rise again. This explains the characteristic U-shaped profiles observed for most identifiable graphons.

Second, for the non-identifiable graphons (IDs 10–13), the behavior is markedly different. The corresponding curves are much flatter and remain at a substantially higher error level throughout the whole range of k. In these cases, refining the histogram partition does not lead to the same gain as in the identifiable setting, because the main difficulty is no longer the approximation of the graphon by a step function, but the lack of a canonical alignment of nodes across graphs. As a result, the benefits of increasing k are limited, and the minima are less pronounced.

Third, the figure shows that the theoretical guideline stated in (24),

$$k_{\text{theoretical}} = \min\left\{ S^{1/4},\ \frac{N}{2\,(M + \log N)} \right\},$$

provides a practically meaningful choice of the resolution parameter. In the present experiment, this rule gives $k_{\text{theoretical}} = 24$, represented by the vertical dashed line in Figure 11. For most identifiable graphons, this value falls very close to the empirical minimizer or inside a broad near-optimal region. This agreement is particularly encouraging because it shows that the theoretical rule captures well the balance between approximation error and estimation variance.

Finally, the broad flat regions around the minima indicate that the performance of JGS is relatively robust to moderate misspecification of k. In other words, choosing a value of k slightly smaller or larger than the empirical optimum often has only a minor impact on the MISE. This is an important practical advantage, since it means that the theoretical rule for selecting k can be used directly in applications without costly tuning procedures, while still yielding highly accurate graphon estimates.

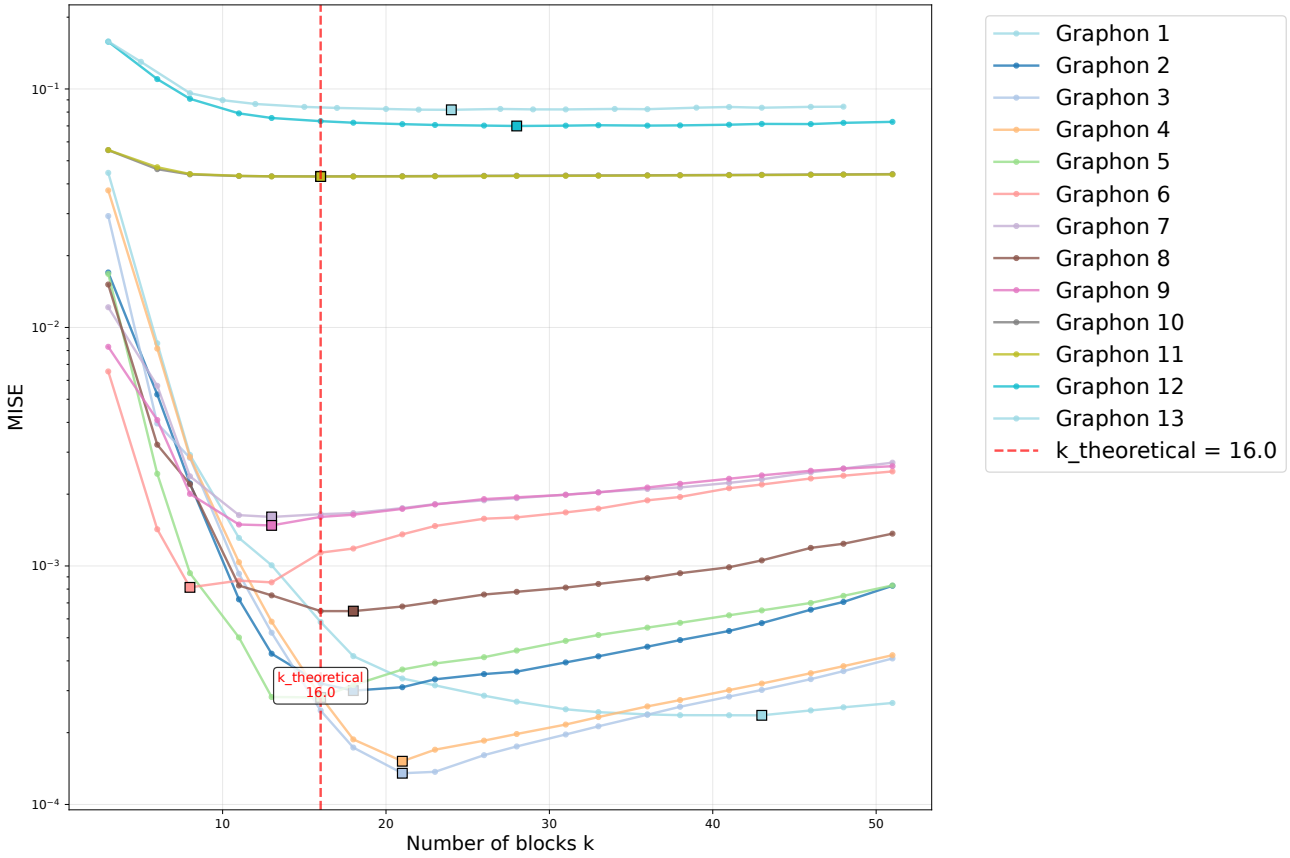

Figure 11: Sensitivity of JGS to the choice of k. The dashed vertical line indicates the theoretical choice $k_{\text{theoretical}} = 24$, while squares mark the empirical optima. For identifiable graphons, the curves exhibit a U-shaped behavior, whereas for non-identifiable graphons they remain flatter and at a higher error level.