

Supplementary Material for the paper “A Few Moments Please: Scalable Graphon Learning via Moment Matching”

Contents

A Detailed Related Work	1
B Proof of Theorem 1	2
B.1 Supporting Lemmas	2
B.2 Proof of Theorem 1	4
C Unbiasedness of Monte Carlo Estimator for an INR-Based Graphon Moment Estimator	5
C.1 Unbiasedness of the Estimator	5
D Time Complexity of MOMENTNET	6
E Proof of Proposition 1	7
F Methods Details	8
F.1 Latent Variable Invariance of MomentNet	8
F.2 MomentMixup Pseudocode	10
G List of Graphons	10
H Selected Motifs	11
I Centrality Measures	11
I.1 Graphon 1: The (xy) Model	12
I.2 Graphon 2: The $(e^{-(x^{0.7}+y^{0.7})})$ Model	13
J Extra Scalability Evaluations	14
K MomentMixup Evaluation Details	15
L Social impacts	15
A Detailed Related Work	

Graphon Estimation Graphon estimation aims to recover the underlying generative structure of observed networks. Classical approaches include methods based on histogram estimators by partitioning nodes according to degree or other structural properties [4, 13, 5], and fitting stochastic block models (SBMs) or their variants, which can be viewed as piecewise constant graphon estimators [1, 8]. Universal singular value thresholding (USVT) [6] offers a non-parametric approach for estimating graphons from a single adjacency matrix, particularly effective for low-rank structures. However, many of these methods face challenges in terms of computational cost for large graphs, achieving resolution-free approximation, or may rely on specific structural assumptions (e.g., piecewise constant for SBMs).

More recently, scalable graphon estimation techniques have gained prominence. For example, some works aim at minimizing distances between graph representations but often involve computationally expensive metrics like the GW distance [15, 21, 20], which can be a bottleneck for large networks. The advent of INRs has opened new avenues for continuous, resolution-free graphon estimation. For instance, IGNR (Implicit Graphon Neural Representation) [19] proposed to directly model graphons using neural networks, enabling the representation of graphons up to arbitrary resolutions and efficient generation of arbitrary-sized graphs. IGNR also addresses unaligned input graphs of different sizes by incorporating the Gromov-Wasserstein distance in its learning framework, often within an auto-encoder setup for graphon learning. Subsequently, SIGL (Scalable Implicit Graphon Learning) [3] further advanced INR-based graphon estimation by combining INRs with Graph Neural Networks (GNNs). In SIGL, GNNs are utilized to improve graph alignment by determining appropriate node orderings, aiming to enhance scalability and learn a continuous graphon at arbitrary resolutions, with theoretical results supporting the consistency of its estimator. While these INR-based techniques offer significant advantages in terms of resolution-free representation and handling unaligned data, they still implicitly involve latent variable modeling or rely on GW-like objectives for alignment. Our proposed method builds upon the representational power of INRs but distinguishes itself by directly recovering the graphon via moment matching. This avoids the need for latent variables, complex metric computations like GW, and provides a theoretically grounded estimation framework that naturally handles multiple observed graphs by matching aggregated empirical moments.

Data Augmentation for Graph Classification Data augmentation is crucial for improving the generalization of GNNs and other graph learning models, especially when labeled data is scarce. Mixup [23], which creates synthetic examples by linearly interpolating pairs of samples and their labels, has shown remarkable success in various domains. Its adaptation to graph data has been explored through several avenues, addressing challenges such as varying node counts, lack of alignment, and the non-Euclidean nature of graphs. For instance, Wang et al. [18] proposed interpolating hidden states of GNNs. Particularly relevant to our work are G-Mixup [9] and GraphMAD Navarro and Segarra [14], which recognize the difficulties of direct graph interpolation and propose to augment graphs for graph classification by operating in the space of graphons. GraphMAD Navarro and Segarra [14] projects graphs into the latent space of graphons and implements nonlinear mixup strategies like convex clustering. G-Mixup [9] first estimates a graphon for each class of graphs from the training data. Then, instead of directly manipulating discrete graph structures, G-Mixup interpolates these estimated graphons of different classes in their continuous, Euclidean representation to obtain mixed graphons. Synthetic graphs for augmentation are subsequently generated by sampling from these mixed graphons. This technique has also been adopted as an augmentation strategy in the evaluation pipelines of some graphon estimation studies for downstream tasks [3].

B Proof of Theorem 1

B.1 Supporting Lemmas

We rely on the following established and derived results. Lemma 1 is an original contribution of this work, while Lemma 2 is Theorem 3.7 (b) in Borgs et al. [4] and it is included here for completeness.

Lemma 1 (Concentration of Empirical Motifs). *Let F be a simple graph with $k = |\mathcal{V}_F|$ vertices. For $P \geq 1$ graphs G_1, \dots, G_P , each sampled independently from $G_n(W^*)$, and for any error tolerance $\epsilon_s > 0$, the probability that the empirical motif density $\bar{t}(F, W^*) = \frac{1}{P} \sum_{p=1}^P t(F, G_p)$ deviates from the true motif density $t(F, W^*)$ is bounded as*

$$\mathbb{P}[|\bar{t}(F, W^*) - t(F, W^*)| \geq \epsilon_s] \leq 2 \exp \left(-\frac{Pn}{4k^2} \left(\epsilon_s - \frac{k(k-1)}{2n} \right)^2 \right), \quad (8)$$

for $\epsilon_s > \frac{k(k-1)}{2n}$.

Proof. Let $X_p = t(F, G_p)$ for $p = 1, \dots, P$. The graphs G_p are independent samples from $G_n(W^*)$, so the random variables X_p are independent and identically distributed.

We leverage concentration properties of $t(F, G_n(W^*))$ in Borgs et al. [4, Lemma 4.4], stating that $t(F, G_n(W^*))$ is concentrated around $t(F, W^*)$ with probability

$$\mathbb{P}[|t(F, G_n(W^*)) - t(F, W^*)| > \delta] \leq 2 \exp(-n\delta^2/(4k^2)). \quad (9)$$

81 This implies that the variable $Z = t(F, G_n(W^*)) - t(F, W^*)$ behaves like a sub-Gaussian random
 82 variable [17]¹. Comparing the exponent $-\frac{n\delta^2}{4k^2}$ from (9) with the sub-Gaussian tail exponent $-\frac{\delta^2}{2\sigma^2}$,
 83 we see that $t(F, G_n(W^*)) - t(F, W^*)$ is sub-Gaussian with parameter $\sigma_Z^2 = \frac{2k^2}{n}$.

84 The variables we are averaging are $X_p = t(F, G_p)$ with $G_p \sim G_n(W^*)$. Let $\mu_n = \mathbb{E}[X_p] =$
 85 $\mathbb{E}[t(F, G_n(W^*))]$. The centered variables fulfill $X_p - \mu_n = (t(F, G_p) - t(F, W^*)) - (\mathbb{E}[t(F, G_p)] -$
 86 $t(F, W^*))$. Subtracting a constant (the bias $\mathbb{E}[t(F, G_p)] - t(F, W^*)$) from a sub-Gaussian variable
 87 preserves its sub-Gaussian property with the same parameter. Thus, $X_p - \mu_n$ are independent,
 88 zero-mean, and σ^2 -sub-Gaussian with $\sigma^2 = \sigma_Z^2 = \frac{2k^2}{n}$.

89 The average of P independent σ^2 -sub-Gaussian random variables is (σ^2/P) -sub-Gaussian [17]. Let
 90 $\bar{Y} = \frac{1}{P} \sum_{p=1}^P (X_p - \mu_n) = \bar{t}(F, W^*) - \mu_n$. Then \bar{Y} is $\left(\frac{2k^2}{nP}\right)$ -sub-Gaussian. The tail bound for \bar{Y}
 91 is

$$\mathbb{P}[|\bar{Y}| \geq \delta] \leq 2 \exp\left(-\frac{\delta^2}{2 \cdot \frac{2k^2}{nP}}\right) = 2 \exp\left(-\frac{\delta^2 nP}{4k^2}\right). \quad (10)$$

92 Substituting $\bar{Y} = \bar{t}(F, W^*) - \mu_n$, we get the concentration bound for the empirical mean around the
 93 expected mean:

$$\mathbb{P}[|\bar{t}(F, W^*) - \mu_n| \geq \delta] \leq 2 \exp\left(-\frac{\delta^2 nP}{4k^2}\right). \quad (11)$$

94 We are interested in the deviation of $\bar{t}(F, W^*)$ from the true motif density $t(F, W^*)$. We use the
 95 triangle inequality to relate this deviation to the deviation from the mean μ_n

$$|\bar{t}(F, W^*) - t(F, W^*)| \leq |\bar{t}(F, W^*) - \mu_n| + |\mu_n - t(F, W^*)|. \quad (12)$$

96 Let $B_n = |\mu_n - t(F, W^*)|$ be the bias of the empirical estimate. It is known from the theory of
 97 graph limits (e.g., related to Borgs et al. [4, Lemma 4.3]) that this bias is bounded by $B_n \leq \frac{k(k-1)}{2n}$.
 98 If the deviation from the true density is at least ϵ_s , i.e., $|\bar{t}(F, W^*) - t(F, W^*)| \geq \epsilon_s$, then it must be
 99 that $|\bar{t}(F, W^*) - \mu_n| \geq \epsilon_s - B_n$. This implication requires $\epsilon_s > B_n$ for the bound to be meaningful.
 100 Thus, for $\epsilon_s > B_n$

$$\mathbb{P}[|\bar{t}(F, W^*) - t(F, W^*)| \geq \epsilon_s] \leq \mathbb{P}[|\bar{t}(F, W^*) - \mu_n| \geq \epsilon_s - B_n]. \quad (13)$$

101 Using the inequality (11) with $\delta = \epsilon_s - B_n$

$$\mathbb{P}[|\bar{t}(F, W^*) - t(F, W^*)| \geq \epsilon_s] \leq 2 \exp\left(-\frac{(\epsilon_s - B_n)^2 nP}{4k^2}\right). \quad (14)$$

102 Introducing the upper bound for the bias, $B_n \leq \frac{k(k-1)}{2n}$

$$\mathbb{P}[|\bar{t}(F, W^*) - t(F, W^*)| \geq \epsilon_s] \leq 2 \exp\left(-\frac{\left(\epsilon_s - \frac{k(k-1)}{2n}\right)^2 nP}{4k^2}\right) \quad (15)$$

$$= 2 \exp\left(-\frac{Pn}{4k^2} \left(\epsilon_s - \frac{k(k-1)}{2n}\right)^2\right). \quad (16)$$

103 This bound is valid when $\epsilon_s > \frac{k(k-1)}{2n}$, as required by the lemma statement. \square

104 **Lemma 2** (Motif Proximity Implies Cut Distance Proximity (Borgs et al. [4], Theorem 3.7 (b))).
 105 For any integer $k \geq 1$, if the motif distance between two graphons W_1 and W_2 fulfills $|t(F, W_1) -$
 106 $t(F, W_2)| < \delta_M = 3^{-k^2}$ for every simple graph $F \in \mathcal{F}_k$, then the cut distance between W_1 and W_2
 107 is upper bounded by

$$d_{cut}(W_1, W_2) \leq \eta = \frac{22C}{\sqrt{\log_2 k}}, \quad (17)$$

108 where $C = \max\{1, \|W_1\|_\infty, \|W_2\|_\infty\}$.

¹A random variable Y is σ^2 -sub-Gaussian if $\mathbb{E}[e^{\lambda Y}] \leq e^{\lambda^2 \sigma^2 / 2}$ for all $\lambda \in \mathbb{R}$, which implies the tail bound
 $\mathbb{P}[|Y| \geq \delta] \leq 2e^{-\delta^2 / (2\sigma^2)}$.

Assumption 1 (Neural Network Approximation Capability). *Given a sufficiently expressive neural network architecture, it can be trained to find parameters θ such that for any set of empirical motif densities $\{\bar{t}(F, W^*)\}_{F \in \mathcal{F}_k}$ computed from data, and any desired approximation error $\epsilon_a > 0$, the neural network's MC motif estimates $\hat{t}_\theta(F, \hat{W}_\theta)$ satisfy*

$$|\hat{t}_\theta(F, \hat{W}_\theta) - \bar{t}(F, W^*)| < \epsilon_a, \quad (18)$$

for all $F \in \mathcal{F}_k$.

This assumption is fundamentally supported by the Universal Approximation Theorem (UAT) [7, 10, 12]. The UAT posits that a neural network with sufficient capacity (e.g., an adequate number of neurons in one or more hidden layers and appropriate non-linear activation functions) can approximate any continuous function to an arbitrary degree of accuracy on a compact domain. In our context, the INR f_θ models the graphon $W : [0, 1]^2 \rightarrow [0, 1]$. The motif density $t(F, W)$ (as defined in Equation 1) is a continuous functional of W , meaning small changes in W lead to small changes in $t(F, W)$. Consequently, if the INR f_θ can approximate any continuous graphon function, it can learn a specific f_θ such that the motif densities of the graphon estimated by the INR $t(F, f_\theta)$ are arbitrarily close to some target values. Given that our estimated motif densities $\hat{t}_\theta(F, \hat{W}_\theta)$ are Monte Carlo approximations of $t(F, f_\theta)$, they too can approach these target values (the empirical densities $\bar{t}(F, W^*)$) as the approximation of the underlying function by f_θ improves as the number of Monte Carlo samples L increases. The assumption thus relies on the INR's capacity to learn a suitable graphon function f_θ and the optimization process's ability to find the parameters θ that make the resulting motif estimates $\hat{t}_\theta(F, \hat{W}_\theta)$ match the empirical observations $\bar{t}(F, W^*)$.

B.2 Proof of Theorem 1

Proof of Theorem 1. Our goal is to bound the cut distance $d_{\text{cut}}(\hat{W}_\theta, W^*)$ by η , which is achieved if we can show that $|\hat{t}_\theta(F, \hat{W}_\theta) - t(F, W^*)| < \delta_M$ for all simple graphs F with $|\mathcal{V}_F| = k$ and where the values of both η and δ_M are provided in Lemma 2.

Consider any graph $F \in \mathcal{F}_k$. Using the triangle inequality, we can bound the difference between the neural network's motif estimate and the true graphon motif

$$|\hat{t}_\theta(F, \hat{W}_\theta) - t(F, W^*)| \leq |\hat{t}_\theta(F, \hat{W}_\theta) - \bar{t}(F, W^*)| + |\bar{t}(F, W^*) - t(F, W^*)|. \quad (19)$$

By Assumption 1 on the neural network's training performance, we guarantee

$$|\hat{t}_\theta(F, \hat{W}_\theta) - \bar{t}(F, W^*)| < \epsilon_a = \frac{\delta_M}{2}, \quad (20)$$

for every $F \in \mathcal{F}_k$.

Now we need to bound the second term in the right-hand side of (19), the deviation of the empirical motif density from the true motif density $|\bar{t}(F, W^*) - t(F, W^*)|$. We use Lemma 1 with the sampling error tolerance set to $\epsilon_s = \frac{\delta_M}{2}$. For this lemma to apply, we require $\epsilon_s > \frac{k(k-1)}{2n}$, which is equivalent to $\frac{\delta_M}{2} > \frac{k(k-1)}{2n}$, or $n > \frac{k(k-1)}{\delta_M}$. This condition is enforced in the theorem statement.

For a specific graph $F \in \mathcal{F}_k$, the probability that the sampling error is large is bounded by Lemma 1

$$\mathbb{P} \left[|\bar{t}(F, W^*) - t(F, W^*)| \geq \frac{\delta_M}{2} \right] \leq 2 \exp \left(-\frac{Pn}{4k^2} \left(\frac{\delta_M}{2} - \frac{k(k-1)}{2n} \right)^2 \right). \quad (21)$$

Let $\mathbb{P}_{\text{fail}, F}$ denote this upper bound for a single graph $F \in \mathcal{F}_k$. However, we require the sampling error $|\bar{t}(F, W^*) - t(F, W^*)|$ to be less than $\frac{\delta_M}{2}$ for all graphs $F \in \mathcal{F}_k$ simultaneously. By the union bound, the probability that there exists at least one graph $F \in \mathcal{F}_k$ for which the sampling error is $\frac{\delta_M}{2}$ or more is at most the sum of the probabilities for each individual graph

$$\mathbb{P}[\exists F \in \mathcal{F}_k \text{ s.t. } |\bar{t}(F, W^*) - t(F, W^*)| \geq \frac{\delta_M}{2}] \leq \sum_{F \in \mathcal{F}_k} \mathbb{P}_{\text{fail}, F}. \quad (22)$$

Since $|\mathcal{V}_F| = k$ for all $F \in \mathcal{F}_k$, the bound $\mathbb{P}_{\text{fail}, F}$ is identical for all these graphs. The sum is thus $N_k \cdot \mathbb{P}_{\text{fail}, F}$, where we recall that $N_k = |\mathcal{F}_k|$. The condition (6) in the theorem is precisely set to

147 ensure that this total probability of failure is less than the desired confidence level ζ

$$N_k \cdot 2 \exp \left(-\frac{Pn}{4k^2} \left(\frac{\delta_M}{2} - \frac{k(k-1)}{2n} \right)^2 \right) < \zeta. \quad (23)$$

148 Therefore, with probability at least $1 - \zeta$ (over the random graph samples G_p), the event that
149 $|\bar{t}(F, W^*) - t(F, W^*)| < \frac{\delta_M}{2}$ holds for all $F \in \mathcal{F}_k$ occurs.

150 Conditioned on this high-probability event, and using the neural network approximation in Assump-
151 tion 1, we have for every $F \in \mathcal{F}_k$

$$|\hat{t}_\theta(F, \hat{W}_\theta) - t(F, W^*)| \leq |\hat{t}_\theta(F, \hat{W}_\theta) - \bar{t}(F, W^*)| + |\bar{t}(F, W^*) - t(F, W^*)| < \frac{\delta_M}{2} + \frac{\delta_M}{2} = \delta_M. \quad (24)$$

152 Since $|\hat{t}_\theta(F, \hat{W}_\theta) - t(F, W^*)| < \delta_M$ holds for all $F \in \mathcal{F}_k$, Lemma 2 implies that the cut distance
153 between the estimated graphon \hat{W}_θ and the true graphon W^* is less than η

$$d_{\text{cut}}(W_\theta, W) < \eta, \quad (25)$$

154 with probability at least $1 - \zeta$, concluding the proof. \square

155 C Unbiasedness of Monte Carlo Estimator for an INR-Based Graphon 156 Moment Estimator

157 Let $F = (\mathcal{V}_F, \mathcal{E}_F)$ be a graph, where \mathcal{V}_F is a set of $k = |\mathcal{V}_F|$ vertices and \mathcal{E}_F is the set of edges. Let
158 $f_\theta : [0, 1]^2 \rightarrow [0, 1]$ be an Implicit Neural Representation (INR) parameterized by θ , which models
159 the probability of an edge existing between two nodes based on their latent variables $\eta_i, \eta_j \in [0, 1]$,
160 and its estimated graphon is denoted by \hat{W}_θ .

161 The likelihood of observing the graph structure F given a specific set of latent variable assignments
162 $\boldsymbol{\eta} = \{\eta_v\}_{v \in \mathcal{V}_F}$ and the INR model f_θ is given by

$$P_\theta(\boldsymbol{\eta}; F, \hat{W}_\theta) = \prod_{(i,j) \in \mathcal{E}_F} \hat{W}_\theta(\eta_i, \eta_j) \prod_{(i,j) \notin \mathcal{E}_F} (1 - \hat{W}_\theta(\eta_i, \eta_j)). \quad (26)$$

163 The quantity $t'_\theta(F, \hat{W}_\theta)$ is defined as this likelihood integrated over all possible configurations of the
164 latent variables in the k -dimensional unit hypercube

$$t'_\theta(F, \hat{W}_\theta) = \int_{[0,1]^k} P_\theta(\boldsymbol{\eta}; F, \hat{W}_\theta) d\boldsymbol{\eta}, \quad (27)$$

165 where $d\boldsymbol{\eta} = \prod_{v \in \mathcal{V}_F} d\eta_v$.

166 The L -sample Monte Carlo estimator for $t'_\theta(F, \hat{W}_\theta)$ is given by

$$\hat{t}'_\theta(F, \hat{W}_\theta) = \frac{1}{L} \sum_{l=1}^L P(\boldsymbol{\eta}^{(l)}; F, \hat{W}_\theta). \quad (28)$$

167 For this estimation, each sample $\boldsymbol{\eta}^{(l)} = [\eta_{v_1}^{(l)}, \dots, \eta_{v_k}^{(l)}]$ is a vector where each component $\eta_v^{(l)}$ (for
168 $v \in \mathcal{V}_F$) is drawn independently from the uniform distribution $\mathcal{U}[0, 1]$.

169 C.1 Unbiasedness of the Estimator

170 **Theorem 1.** *The Monte Carlo estimator $\hat{t}'_\theta(F, \hat{W}_\theta)$ is an unbiased estimator of $t'_\theta(F, \hat{W}_\theta)$.*

171 *Proof.* To show that the Monte Carlo estimation $\hat{t}'_\theta(F, \hat{W}_\theta)$ is an unbiased estimator of $t'_\theta(F, \hat{W}_\theta)$,
172 we need to prove that $\mathbb{E}[\hat{t}'_\theta(F, \hat{W}_\theta)] = t'_\theta(F, \hat{W}_\theta)$.

173 The expectation of the estimator is:

$$\begin{aligned}\mathbb{E}[\hat{t}'_\theta(F, \hat{W}_\theta)] &= \mathbb{E}\left[\frac{1}{L} \sum_{l=1}^L P_\theta(\boldsymbol{\eta}^{(l)}; F, \hat{W}_\theta)\right] \\ &= \frac{1}{L} \sum_{l=1}^L \mathbb{E}[P_\theta(\boldsymbol{\eta}^{(l)}; F, \hat{W}_\theta)] \quad (\text{by linearity of the expectation}).\end{aligned}\quad (29)$$

174 Since each sample $\boldsymbol{\eta}^{(l)}$ is drawn independently from the same uniform distribution, therefore its pdf
175 is $p(\boldsymbol{\eta}) = 1$ on $[0, 1]^k$, the expectation $\mathbb{E}[P_\theta(\boldsymbol{\eta}^{(l)}; F, \hat{W}_\theta)]$ is the same for all l . Let this common
176 expectation be $\mathbb{E}[P_\theta(\boldsymbol{\eta}; F, \hat{W}_\theta)]$, whose value is

$$\begin{aligned}\mathbb{E}[P_\theta(\boldsymbol{\eta}; F, \hat{W}_\theta)] &= \int_{[0,1]^k} P_\theta(\boldsymbol{\eta}; F, \hat{W}_\theta) p(\boldsymbol{\eta}) d\boldsymbol{\eta} \\ &= \int_{[0,1]^k} P_\theta(\boldsymbol{\eta}; F, \hat{W}_\theta) \cdot 1 d\boldsymbol{\eta} \quad (\text{since } p(\boldsymbol{\eta}) = 1 \text{ on } [0, 1]^k) \\ &= t'_\theta(F, \hat{W}_\theta),\end{aligned}$$

177 according to (27). Substituting this back into (29)

$$\begin{aligned}\mathbb{E}[\hat{t}'_\theta(F, \hat{W}_\theta)] &= \frac{1}{L} \sum_{l=1}^L t'_\theta(F, \hat{W}_\theta) \\ &= \frac{1}{L} (L \cdot t'_\theta(F, \hat{W}_\theta)) \\ &= t'_\theta(F, \hat{W}_\theta).\end{aligned}$$

178 Thus, $\mathbb{E}[\hat{t}'_\theta(F, \hat{W}_\theta)] = t'_\theta(F, \hat{W}_\theta)$, which shows that the Monte Carlo estimator $\hat{t}'_\theta(F, \hat{W}_\theta)$ is an
179 unbiased estimator of $t'_\theta(F, \hat{W}_\theta)$. This means that, on average, the estimator will yield the true value
180 of the integral defined by f_θ and the graph structure F . \square

181 D Time Complexity of MOMENTNET

182 **Stage 1: parallel motif-density extraction.** For each graph $G_p = (\mathcal{V}_p, \mathcal{E}_p)$ let $n_p = |\mathcal{V}_p|$,
183 $e_p = |\mathcal{E}_p|$ and $d_p = \max_{v \in \mathcal{V}_p} \deg(v)$ be the number of nodes, number of edges, and maximum
184 degree of the graph G_p . ORCA [11] counts all 2–4-node graphlets in

$$T_{\text{ORCA}}(G_p) = O(e_p d_p + n_p d_p^3).$$

185 Because every graph can be processed independently, we dispatch the P graphs to M workers
186 ($M \leq P$). Hence the *wall-clock* preprocessing time is

$$T_{\text{stage 1}} = O\left(\left\lceil \frac{P}{M} \right\rceil \max_p (e_p d_p + n_p d_p^3)\right).$$

187 With one worker per graph ($M = P$) this shrinks to the single-graph cost that dominates (\max_p).

188 **Stage 2: training the Moment network.** Define:

- 189 • L : number of Monte-Carlo samples per epoch;
- 190 • N_e : number of training epochs;
- 191 • C_{INR} : cost of one forward/back-prop through the INR for a single edge probability;
- 192 • $|\theta|$: total number of trainable parameters.

193 Each motif instance F of size $|\mathcal{V}_F| \leq 4$ invokes the INR at most six times, a constant. One epoch
194 therefore costs

$$T_{\text{epoch}} = O(L C_{\text{INR}} + |\theta|), \quad T_{\text{stage 2}} = O(N_e (L C_{\text{INR}} + |\theta|)).$$

Overall wall-clock complexity.

$$T_{\text{MomentNet}} = O\left(\left\lceil \frac{P}{M} \right\rceil \max_p (e_p d_p + n_p d_p^3) + N_e (L C_{\text{INR}} + |\theta|)\right).$$

195 Comparison with SIGL in Sparse vs. Dense Regimes

196 SIGL [3] requires message-passing GNN training, histogram building and INR fitting; with N_e
197 epochs its wall-clock cost is $T_{\text{SIGL}} = O(P N_e n_T^2)$, where $n_T = \max_p n_p$.

198 • **Sparse regime** ($d_{\max} = O(1) \Rightarrow e_p = O(n_p)$):

199 – MOMENTNET: $T = O(\left\lceil \frac{P}{M} \right\rceil n_T + N_e (L C_{\text{INR}} + |\theta|))$;

200 – SIGL: $T = O(P N_e n_T^2)$.

201 Here MomentNet grows *linearly* in n_T (plus the network-training term), whereas SIGL is
202 quadratic. In practice we repeatedly observe MomentNet to be faster when graphs have
203 $e_p = O(n_p)$ even for very large n_p .

204 • **Dense regime** (Erdős–Rényi with $p_{\text{conn}} = 0.5$ implies $d_{\max} \approx n_T/2$ and $e_p = \Theta(n_T^2)$):

205 – MOMENTNET: $T = O(\left\lceil \frac{P}{M} \right\rceil n_T^4 + N_e (L C_{\text{INR}} + |\theta|))$;

206 – SIGL: $T = O(P N_e n_T^2)$.

207 Asymptotically, SIGL’s n_T^2 term is smaller than MomentNet’s n_T^4 . Yet empirical runs on
208 dense ER graphs with $p_{\text{conn}} = 0.5$ still show MomentNet to be faster once (i) Stage 1 is
209 fully parallelised and (ii) the constants behind GNN message passing and histogramming
210 dominate SIGL’s quadratic term. Thus, the theoretical advantage of SIGL in dense graphs
211 does not necessarily translate into shorter wall-clock times. Furthermore, MomentNet
212 utilizes a two-stage process. The initial stage involves computing motif counts from the
213 input graphs. Following this, the graphs are discarded. The second stage, which our
214 experiments show to be the dominant phase of our method, then trains an Implicit Neural
215 Representation (INR) using a vector of average moments derived from these counts. This
216 design provides a significant reason for our method’s improved speed, particularly in dense
217 scenarios. By isolating the computationally expensive motif counting to a preliminary step,
218 this cost is bypassed during the subsequent, dominant INR learning phase.

219 With graph-level parallelism, MOMENTNET is *provably linear* in the number of edges for sparse
220 networks and remains competitive on dense networks because its constant factors are smaller and its
221 training cost is independent of the graph size.

222 E Proof of Proposition 1

223 Let $W_1, W_2: [0, 1]^2 \rightarrow [0, 1]$ be two graphons and fix $\alpha \in (0, 1)$. Denote their convex combination
224 by

$$W_\alpha = \alpha W_1 + (1 - \alpha) W_2.$$

225 **Edge density (a linear functional).** For the single-edge motif F_e on vertices $\mathcal{V}_{F_e} = \{1, 2\}$ and
226 $\mathcal{E}_{F_e} = \{(1, 2)\}$, the induced density is

$$t'(F_e, W) = \int_{[0, 1]^2} W(\eta_1, \eta_2) d\eta_1 d\eta_2 = \mathbb{E}[W(\eta_1, \eta_2)].$$

227 Because the integrand is *linear* in W , we immediately have

$$t'(F_e, W_\alpha) = \alpha t'(F_e, W_1) + (1 - \alpha) t'(F_e, W_2),$$

228 so the edge density behaves affinely under convex combinations.

229 **The V-shape motif.** Let F be the V -shape (three-vertex path) on vertex set $\mathcal{V}_F = \{1, 2, 3\}$ and
 230 edge set $\mathcal{E}_F = \{(1, 2), (1, 3)\}$. Its induced density is

$$t'(F, W) = \int_{[0,1]^3} W(\eta_1, \eta_2) W(\eta_1, \eta_3) [1 - W(\eta_2, \eta_3)] d\eta_1 d\eta_2 d\eta_3. \quad (30)$$

231 Plugging W_α into (30)

$$\begin{aligned} t'(F, W_\alpha) = & \mathbb{E} \left[(\alpha W_1 + (1 - \alpha) W_2)_{12} (\alpha W_1 + (1 - \alpha) W_2)_{13} (1 - \alpha W_1 - (1 - \alpha) W_2)_{23} \right] \\ = & \alpha^3 \mathbb{E}[(W_1)_{12}(W_1)_{13}(1 - (W_1)_{23})] + (1 - \alpha)^3 \mathbb{E}[(W_2)_{12}(W_2)_{13}(1 - (W_2)_{23})] \\ & + \text{mixed terms}, \end{aligned} \quad (31)$$

232 where, to simplify notation, we used $(\cdot)_{ij}$ to denote that the graphon inside the parenthesis is evaluated
 233 on (η_i, η_j) and “mixed terms” contain products in which at least one factor comes from W_1 and
 234 another from W_2 . Because these mixed terms generally do not cancel, the right-hand side of (31)
 235 *does not* reduce to the affine combination

$$\alpha t'(F, W_1) + (1 - \alpha) t'(F, W_2), \quad (32)$$

236 except in degenerate cases (e.g. $W_1 = W_2$ or $\alpha \in \{0, 1\}$).

237 **Concrete counter-example.** Take constant graphons $W_1(\eta_i, \eta_j) = p_1$ and $W_2(\eta_i, \eta_j) = p_2$ with
 238 p_1 and p_2 being constants satisfying $0 < p_1 \neq p_2 < 1$. Then $W_\alpha(\eta_i, \eta_j) = p_\alpha = \alpha p_1 + (1 - \alpha) p_2$,
 239 and

$$t'(F, W_i) = p_i^2(1 - p_i), \quad t'(F, W_\alpha) = p_\alpha^2(1 - p_\alpha),$$

240 for $i \in \{1, 2\}$. However,

$$p_\alpha^2(1 - p_\alpha) \neq \alpha p_1^2(1 - p_1) + (1 - \alpha) p_2^2(1 - p_2)$$

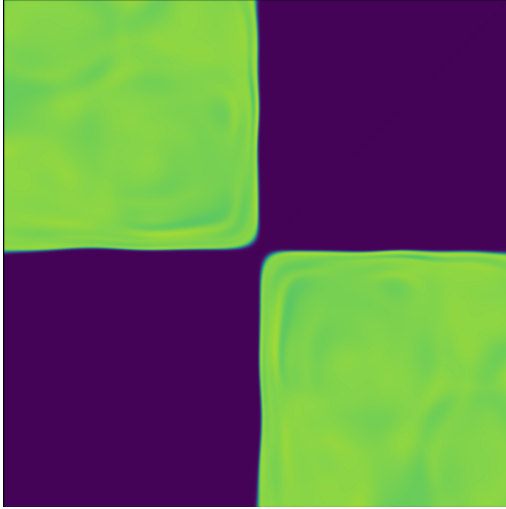
241 whenever $p_1 \neq p_2$ and $\alpha \in (0, 1)$, confirming that the V -shape moment is *not* affine in W .

242 **Conclusion.** Edge moments are linear in the graphon, but higher-order induced moments involve
 243 *non-linear* (polynomial) combinations of W . Consequently, a convex combination of graphons
 244 preserves edge moments but fails to preserve the remaining components of the motif-moment
 245 vector. \square

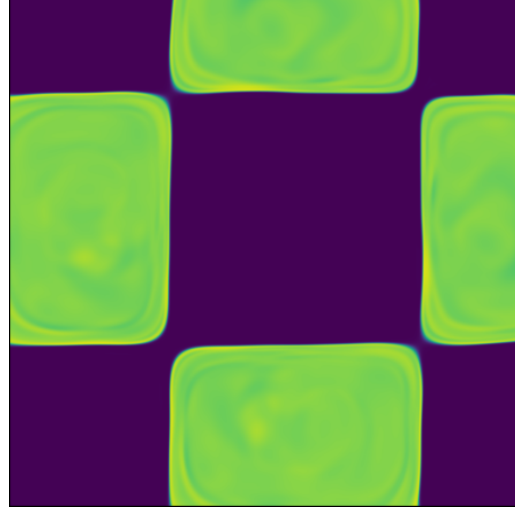
246 F Methods Details

247 F.1 Latent Variable Invariance of MomentNet

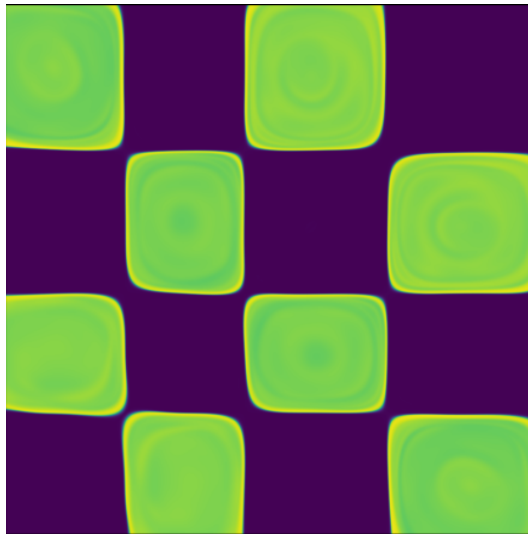
248 The graphon model and our proposed model to learn it exhibit invariance to the specific ordering
 249 or labeling of latent variables. This means that the estimated graphon is unchanged under measure
 250 preserving transformations [4]. In other words, if the underlying structure of a graphon is rearranged
 251 or relabeled, MomentNet can still accurately capture the essential underlying connectivity patterns.
 252 To illustrate this crucial property, we conduct an experiment using an SBM graphon, more precisely
 253 the one indexed by 12 in Table 2. For this experiment, we utilize the same dataset that was generated
 254 for the performance comparison of MomentNet discussed in Section (5). The learned graphons for
 255 three different realizations of this experiment are presented in Figure 3. It is evident that all three
 256 estimated graphons closely resemble the ground truth graphon, which is depicted in Figure 4. Also,
 257 the three estimated graphons reflect the same underlying structure, and all of them share a similar
 258 GW loss, which is a loss function invariant to measure preserving transformations. This essentially
 259 means that, no matter which of the three depicted graphons we sample graphs from, the underlying
 260 structure of all these graphs will be the same. This outcome strongly verifies that MomentNet’s
 261 primary mechanism involves matching the moments of the graph, without caring about the ordering
 262 of the latent variables. Consequently, and in contrast to other methodologies, its estimated graphon
 263 accurately reflects the ground truth structure, allowing for differences only up to a permutation of the
 264 latent variable locations.



(a) Estimated SBM graphon (Sample 1).



(b) Estimated SBM graphon (Sample 2).



(c) Estimated SBM graphon (Sample 3).

Figure 3: Three samples of estimated graphons derived from a SBM.

Algorithm 1 MomentMixup Augmentation

Input: α_{mix} : float, mixing coefficient ($0 \leq \alpha_{\text{mix}} \leq 1$).
 $\mathcal{G}_i, \mathcal{G}_j$: list of graphs, graph datasets for classes i and j .
 y_i, y_j : integer, label for classes i and j .
 N_{sample} : integer, number of graphs to sample from each class dataset to compute average moments.
 N_{nodes} : integer, number of nodes for each new graph.
 N_{graphs} : integer, number of augmented graphs to generate.

Output: \mathcal{G}_{aug} : list of graphs and labels, newly generated augmented graphs.

```

1:  $\mathbf{m}_i \leftarrow \frac{1}{N_{\text{sample}}} \sum_{G \in \mathcal{G}_i} \text{ComputeGraphMoments}(G)$   $\triangleright$  Compute average moment vector for class  $i$ 
2:  $\mathcal{S}_i \leftarrow$  Randomly select  $N_{\text{sample}}$  graphs from  $\mathcal{G}_i$ 
3:  $\mathbf{m}_i \leftarrow \frac{1}{N_{\text{sample}}} \sum_{G \in \mathcal{S}_i} \text{ComputeGraphMoments}(G)$ 
4:  $\mathbf{m}_j \leftarrow \frac{1}{N_{\text{sample}}} \sum_{G \in \mathcal{S}_j} \text{ComputeGraphMoments}(G)$   $\triangleright$  Compute average moment vector for class  $j$ 
5:  $\mathcal{S}_j \leftarrow$  Randomly select  $N_{\text{sample}}$  graphs from  $\mathcal{G}_j$ 
6:  $\mathbf{m}_j \leftarrow \frac{1}{N_{\text{sample}}} \sum_{G \in \mathcal{S}_j} \text{ComputeGraphMoments}(G)$ 
7:  $\mathbf{m}_{\text{target}} \leftarrow \alpha_{\text{mix}} \cdot \mathbf{m}_i + (1 - \alpha_{\text{mix}}) \cdot \mathbf{m}_j$   $\triangleright$  Compute target mixed moments
8:  $y_{\text{target}} \leftarrow \alpha_{\text{mix}} \cdot y_i + (1 - \alpha_{\text{mix}}) \cdot y_j$   $\triangleright$  Compute the label for the new samples
9:  $W_{\text{aug}} \leftarrow \text{MomentNet}(\mathbf{m}_{\text{target}})$   $\triangleright$  Trains MomentNet for  $\mathbf{m}_{\text{target}}$ 
10:  $\mathcal{G}_{\text{aug}} \leftarrow []$   $\triangleright$  Initialize list for augmented samples
11: for  $k \leftarrow 1$  to  $N_{\text{graphs}}$  do
12:    $G_{\text{new}} \leftarrow \text{SampleGraph}(W_{\text{aug}}, N_{\text{nodes}})$   $\triangleright$  Sample new graph
13:   Add  $(G_{\text{new}}, y_{\text{target}})$  to  $\mathcal{G}_{\text{aug}}$ 
14: end for
15: return  $\mathcal{G}_{\text{aug}}$ 

```

Table 2: Table of Graphons

	$W(x, y)$
1	xy
2	$e^{-(x^{0.7} + y^{0.7})}$
3	$\frac{1}{4}(x^2 + y^2 + \sqrt{x} + \sqrt{y})$
4	$\frac{1}{2}(x + y)$
5	$(1 + e^{(-2(x^2 + y^2))})^{-1}$
6	$(1 + e^{(-\max\{x, y\}^2 - \min\{x, y\}^4)})^{-1}$
7	$e^{(-\max\{x, y\}^{0.75})}$
8	$e^{(-\frac{1}{2}(\min\{x, y\} + \sqrt{x} + \sqrt{y}))}$
9	$\log(1 + \max\{x, y\})$
10	$ x - y $
11	$1 - x - y $
12	$0.8\mathbf{I}_2 \otimes \mathbf{1}_{[0, \frac{1}{2}]^2}$
13	$0.8(1 - \mathbf{I}_2) \otimes \mathbf{1}_{[0, \frac{1}{2}]^2}$

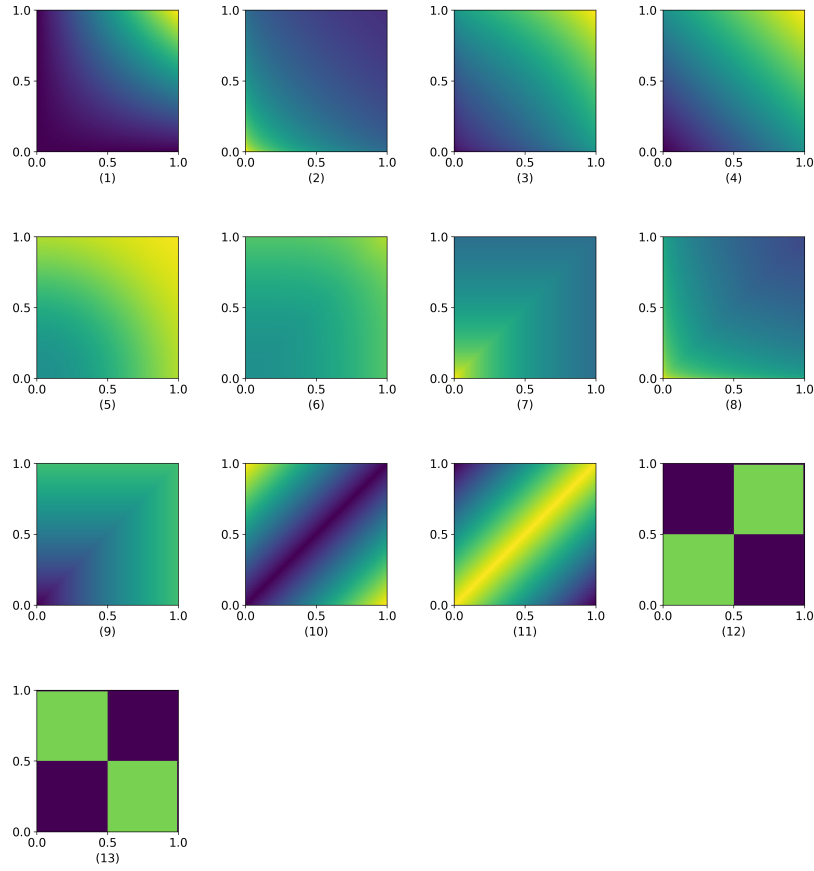


Figure 4: Representation of the graphons defined in Table 2.

268 H Selected Motifs

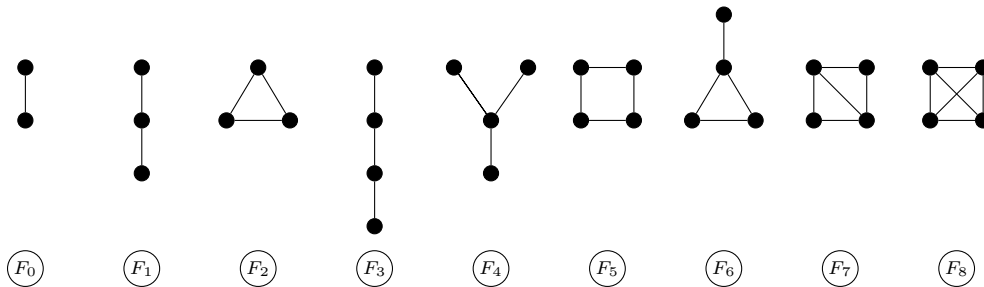


Figure 5: Motifs up to four nodes.

269 I Centrality Measures

270 In real-world graph statistical analysis, **centrality measures** are of significant interest to researchers.
 271 Building upon the work of Avella-Medina et al. [2], who demonstrated the computability of these
 272 measures on graphons, we use several centrality metrics to further evaluate the quality of the estimated
 273 graphons. Specifically, we employ:

- 274 • **Degree Centrality:** This measure quantifies the number of direct connections a node possesses.
- 275 – *High Value:* Indicates a node with many direct connections, often acting as a local hub with
- 276 numerous immediate interactions. Such a node is highly active in its local neighborhood.
- 277 – *Low Value:* Suggests a node with few direct connections, implying less immediate activity or
- 278 influence within its local vicinity.
- 279 • **Eigenvector Centrality:** This identifies influential nodes by considering that connections to other
- 280 highly-connected (and thus influential) nodes contribute more significantly to a node’s score. It
- 281 measures how well-connected a node is to other well-connected nodes.
- 282 – *High Value:* A node with high eigenvector centrality is connected to other nodes that are
- 283 themselves influential. This node is likely a key player within an influential cluster or a leader
- 284 among leaders.
- 285 – *Low Value:* A node with low eigenvector centrality is typically connected to less influential
- 286 nodes or has relatively few connections overall. Its influence is not strongly amplified by the
- 287 influence of its neighbors.
- 288 • **Katz Centrality:** This measure considers all paths in the graph, assigning exponentially more
- 289 weight to shorter paths while still accounting for longer ones. It uses an attenuation factor α , which
- 290 determines the weight given to longer paths: smaller values of α emphasize shorter paths, while
- 291 larger values give more importance to longer paths, up to a theoretical limit to ensure convergence.
- 292 – *High Value:* Indicates a node that is reachable by many other nodes through numerous paths,
- 293 with shorter paths contributing more. This node is generally well-connected throughout the
- 294 network, both directly and indirectly, and can efficiently disseminate or receive information.
- 295 – *Low Value:* Suggests a node that is not easily reachable by many other nodes or is primarily
- 296 connected via very long paths. Its overall influence or accessibility within the network is limited.
- 297 • **PageRank Centrality:** Originally developed for web pages, PageRank assesses a node’s importance
- 298 based on the number and quality of its incoming links. A link from an important node carries more
- 299 weight than a link from a less important one. It uses a damping factor β , representing the probability
- 300 that a random walker will follow a link to an adjacent node, while $(1 - \beta)$ is the probability they
- 301 will jump to a random node in the graph, ensuring that all nodes receive some rank and preventing
- 302 rank-sinking in disconnected components.
- 303 – *High Value:* A node with high PageRank centrality receives many “votes” (incoming connections)
- 304 from other important nodes. This indicates that significant entities within the network consider
- 305 this node to be important or authoritative.
- 306 – *Low Value:* A node with low PageRank centrality receives few incoming connections or is
- 307 primarily linked by less important nodes. It is not widely recognized as important by other
- 308 influential nodes in the network.

309 The mathematical formulations for these graphon-based centrality measures are adopted directly from
 310 Avella-Medina et al. [2], corresponding to equations (7), (8), (9), and (10) in their paper, respectively.
 311 For a detailed analysis, we focus on graphons 1 and 2, as specified in Table 2. We compute both
 312 analytical and sample-based centrality measures, establishing these as baselines for comparison
 313 with our results. The analytical computations directly apply the aforementioned formulas from
 314 Avella-Medina et al. [2]. For the sample-based approach, we generate discrete graph instances by
 315 drawing samples from the ground truth graphon and subsequently compute the centrality measures
 316 within this discrete domain. Further details regarding each graphon are presented in the subsequent
 317 subsections.

318 I.1 Graphon 1: The (xy) Model

319 The analytical centrality measures formulas for this graphon are as follows:

- **Degree Centrality:**

$$C^d(x) = \frac{x}{2}$$

- **Eigenvector Centrality:**

$$C^e(x) = \sqrt{3}x$$

- **Katz Centrality:**

$$C_{\alpha}^k(x) = (6 - 2\alpha) + 3\alpha x$$

- **PageRank Centrality:**

$$C_{\beta}^{\text{pr}}(x) = (1 - \beta) + 2\beta x$$

These measures are for the given latent variable $x \in [0, 1]$, after computing its centrality vector, we normalize it before comparison with discrete graph centralities [2]. Since the ordering for these experiments is important, we create a new dataset of 20 graphs with 100 nodes each, preserving the latent variables for all the nodes. The experiment results are illustrated in Figure 6. Our results show that centrality measures from the MomentNet-predicted graphon (blue lines in the figure) are close to the analytical computations (ground truth, black dashed lines). Furthermore, these graphon-based centralities by MomentNet also provide a good approximation for centrality measures computed over discrete graph samples (red dots).

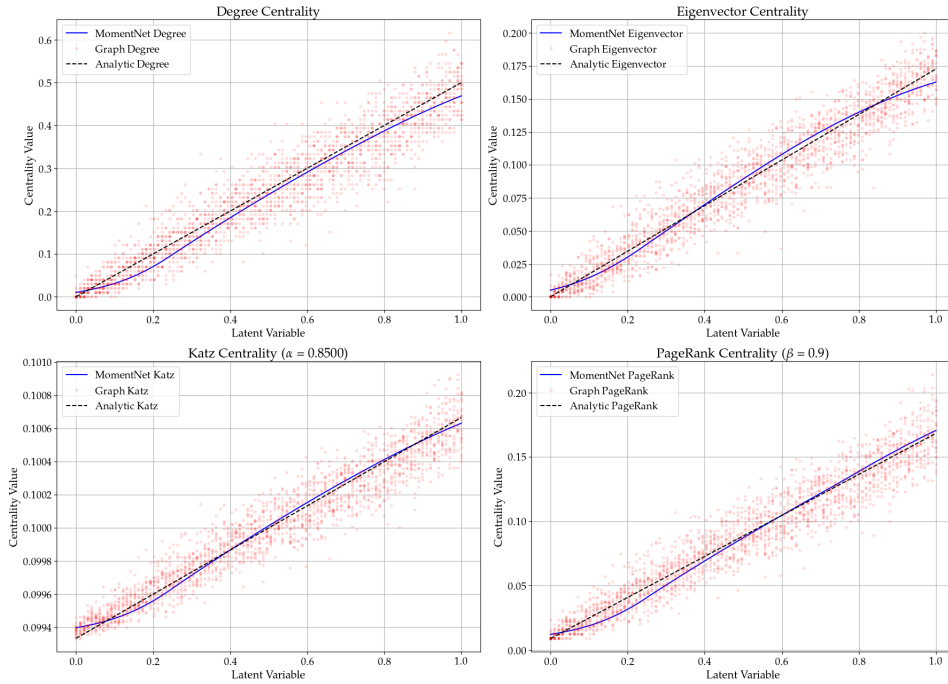


Figure 6: Centrality measures: MomentNet vs. analytic computation for the xy graphon.

328 I.2 Graphon 2: The $(e^{-(x^{0.7}+y^{0.7})})$ Model

329 To test the generalizability and consistent performance of our method across varying complexities, we
 330 replicated the experiment on a more complex graphon. The analytical centrality measures formulas
 331 for this graphon are as follows:

- **Degree Centrality:**

$$C^d(x) = 0.7492 e^{-x^{0.7}}$$

- **Eigenvector Centrality:**

$$C^e(x) = \frac{e^{-x^{0.7}}}{\sqrt{0.473}}$$

- **Katz Centrality:**

$$C_{\alpha}^k(x) = 1 + \frac{0.7492 \alpha e^{-x^{0.7}}}{1 - 0.473 \alpha}$$

- **PageRank Centrality:**

$$C_{\beta}^{\text{pr}}(x, \beta) = (1 - \beta) + \frac{\beta}{0.7492} e^{-x^{0.7}}$$

332 The experiment results are illustrated in Figure 7. Similar to the previous experiment, after computing
 333 the centrality measures on the graphon and analytically, we normalize them to compare them with
 334 the discrete graph measurement. As the plots show, similar to the previous graphon, our estimation is
 335 very close to the ground truth results obtained by analytical calculation.

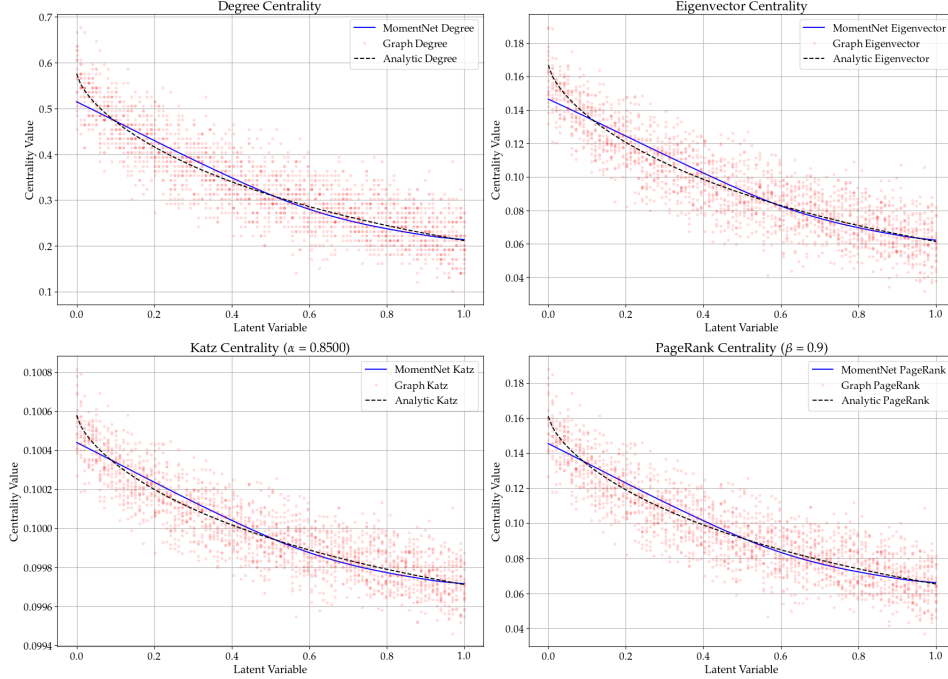


Figure 7: Centrality measures: MomentNet vs. analytic computation for the $e^{-(x^{0.7}+y^{0.7})}$ graphon.

336 J Extra Scalability Evaluations

337 We conducted an additional experiment to evaluate the scalability of SIGL and MomentNet. For
 338 this assessment, rather than focusing on SIGL’s known weaknesses in latent variable estimation, we
 339 selected graphon number 5 from Table 2, a model that both methods accurately estimate. We generate
 340 10 graphs for each node size $n \in \{10, 20, \dots, 810\}$.

341 Figure 8 illustrates the scalability of MomentNet and SIGL in terms of both performance, measured
 342 by GW loss, and average runtime, as a function of the number of nodes. Subfigure (a) of Figure 8
 343 reveals that MomentNet (blue line) maintains a consistently low GW loss across the tested range
 344 of node sizes, indicating stable performance. In contrast, SIGL’s (red line) GW loss starts notably
 345 higher for smaller networks but decreases substantially as the number of nodes increases, eventually
 346 matching or even slightly outperforming MomentNet’s loss for larger networks.

347 However, subfigure (b) of Figure 8 highlights a significant difference in computational efficiency:
 348 MomentNet’s average runtime exhibits only a modest and gradual increase with the number of nodes.
 349 Conversely, SIGL’s runtime escalates sharply, demonstrating significantly poorer scalability.

350 Consequently, while SIGL might offer a marginal advantage in GW Loss for very large graphs,
 351 MomentNet’s vastly superior runtime scalability makes it a more practical and favorable approach,
 352 particularly for applications involving large-scale networks where computational resources and time
 353 are critical factors.

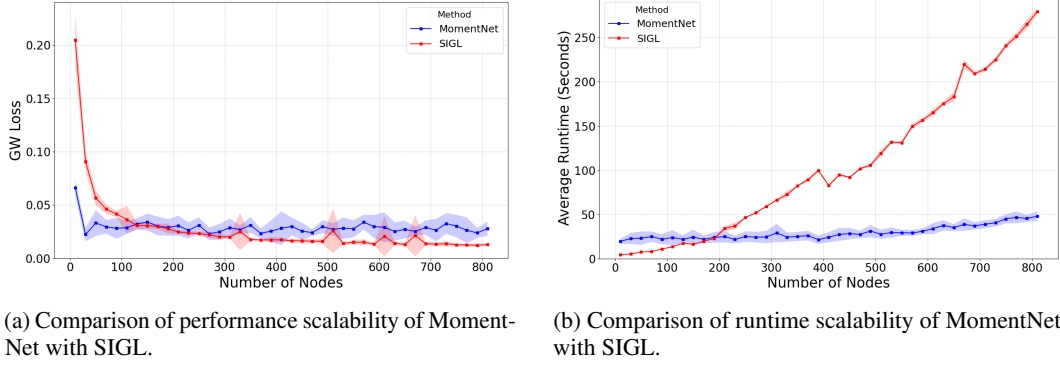


Figure 8: Scalability Comparison of MomentNet and SIGL

K MomentMixup Evaluation Details

Our experimental evaluation is conducted on four diverse benchmark datasets widely used in graph classification research. Table 3 provides a detailed overview of these datasets, outlining their specific characteristics and the nature of their respective classification tasks.

Table 3: Description of the benchmark datasets used for evaluation. Each dataset represents a different type of graph structure and classification task.

Dataset	Description	Classification Task	Citation
IMDB-B	Movie collaboration graphs; nodes represent actors/actresses, and an edge connects two actors/actresses if they appear in the same movie.	Binary genre classification.	[22]
IMDB-M	A multi-class version of IMDB-B, representing movie collaborations with similar graph construction.	Multi-class genre classification.	[22]
REDD-B	Social network graphs from Reddit; nodes represent users, and an edge indicates an interaction (e.g., one user commented on another’s post).	Binary community (subreddit) classification.	[22]
AIDS	Bioinformatics graphs representing molecules; nodes are atoms, and edges are covalent bonds between them.	Binary classification based on anti-HIV activity (active vs. inactive).	[16]

L Social impacts

The methods presented for graphon estimation via moment-matching INRs and data augmentation through MomentMixup, while offering powerful tools for understanding network structures and enhancing graph-based machine learning, are not without potential societal risks if deployed without careful consideration. For instance, in social network analysis, if the empirical moments used for graphon estimation are derived from graphs reflecting existing societal biases (e.g., in representation or connectivity), both the estimated graphons and synthetic graphs generated via MomentMixup could inadvertently perpetuate or even amplify these biases. This could lead to inequitable outcomes when models trained on such data are used for applications like resource allocation, recommendation systems, or public policy modeling. Similarly, in critical domains such as epidemiology or financial systems, inaccuracies in graphon estimation or the generation of unrepresentative augmented data could lead to flawed predictions, potentially resulting in misguided interventions or financial instability. While graphon estimation offers a level of abstraction, careful attention must also be paid to ensure that the process does not inadvertently leak sensitive information from the original graph data, especially when dealing with networks containing personal or confidential details. Therefore,

it is crucial for practitioners to be acutely aware of these potential pitfalls. This includes critically examining input data and chosen moments for biases, rigorously validating the fidelity and representativeness of estimated graphons and generated graphs, and thoughtfully considering the ethical implications of their application, particularly in domains with direct and significant societal impact.

References

- [1] Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. In *Journal of Machine Learning Research*, volume 9, pages 1981–2014.
- [2] Avella-Medina, M., Parise, F., Schaub, M. T., and Segarra, S. (2020). Centrality measures for graphons: Accounting for uncertainty in networks. *IEEE Transactions on Network Science and Engineering*, 7(1):520–537.
- [3] Azizpour, A., Zilberstein, N., and Segarra, S. (2025). Scalable implicit graphon learning. In *The 28th International Conference on Artificial Intelligence and Statistics*.
- [4] Borgs, C., Chayes, J., Lovász, L., Sós, V., and Vesztegombi, K. (2008). Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801–1851.
- [5] Chan, S. and Airoldi, E. (2014). A consistent histogram estimator for exchangeable graph models. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 208–216, Beijing, China. PMLR.
- [6] Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1).
- [7] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals, and systems*, 2(4):303–314.
- [8] Gao, C., Lu, Y., and Zhou, H. H. (2015). Rate-optimal graphon estimation. *The Annals of Statistics*, 43(6):2624–2652.
- [9] Han, X., Jiang, Z., Liu, N., and Hu, X. (2022). G-Mixup: Graph Data Augmentation for Graph Classification. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 8450–8465. PMLR.
- [10] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.
- [11] Hočevár, T. and Demšar, J. (2014). A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565.
- [12] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867.
- [13] Lovász, L. (2012). *Large networks and graph limits*, volume 60. American Mathematical Soc.
- [14] 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 (ICASSP)*, pages 1–5.
- [15] Peyré, G., Cuturi, M., and Solomon, J. (2016). Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR.
- [16] Riesen, K. and Bunke, H. (2008). Iam graph database repository for graph based pattern recognition and machine learning. In da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J. T., Georgiopoulos, M., Anagnostopoulos, G. C., and Loog, M., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, pages 287–297, Berlin, Heidelberg. Springer Berlin Heidelberg.

- 418 [17] Van Handel, R. (2014). Probability in high dimension. *Lecture Notes (Princeton University)*,
419 2(3):2–3.
- 420 [18] Wang, Y., Wang, W., Liang, Y., Cai, Y., and Hooi, B. (2021). Mixup for node and graph
421 classification. In *Proceedings of the Web Conference 2021*, WWW ’21, page 3663–3674, New
422 York, NY, USA. Association for Computing Machinery.
- 423 [19] Xia, X., Mishne, G., and Wang, Y. (2023). Implicit graphon neural representation. In *Proceed-*
424 *ings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of
425 *Proceedings of Machine Learning Research*, pages 10619–10634. PMLR.
- 426 [20] Xu, H., Luo, D., Carin, L., and Zha, H. (2021). Learning graphons via structured gromov-
427 wasserstein barycenters. *Proceedings of the AAAI Conference on Artificial Intelligence*,
428 35(12):10505–10513.
- 429 [21] Xu, H., Luo, D., Zha, H., and Duke, L. C. (2019). Gromov-Wasserstein learning for graph
430 matching and node embedding. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of*
431 *the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine*
432 *Learning Research*, pages 6932–6941. PMLR.
- 433 [22] Yanardag, P. and Vishwanathan, S. (2015). Deep graph kernels. In *Proceedings of the 21th*
434 *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’15,
435 page 1365–1374, New York, NY, USA. Association for Computing Machinery.
- 436 [23] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk
437 minimization. In *International Conference on Learning Representations (ICLR)*.