A Few Moments Please: Scalable Graphon Learning via Moment Matching

Reza Ramezanpour, Victor M. Tenorio, Antonio G. Marques,
Ashutosh Sabharwal, Santiago Segarra

RICE UNIVERSITY

Universidad Rey Juan Carlos

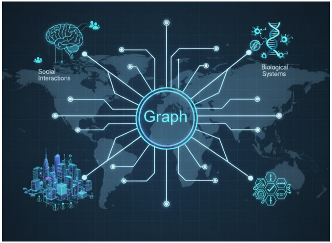
Networks are Everywhere

Networks are the fundamental structure for relational data:

- Social interactions
- Wireless Communication Channels

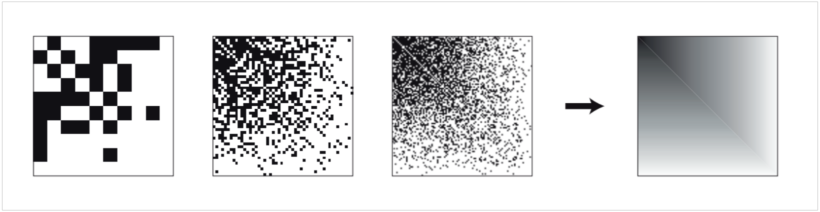
Our Goal: Understand the underlying generative principles of large-scale networks for tasks like:

- Community Detection
- Graph Classification



Graphons (Graph Limits) have emerged as a powerful framework.

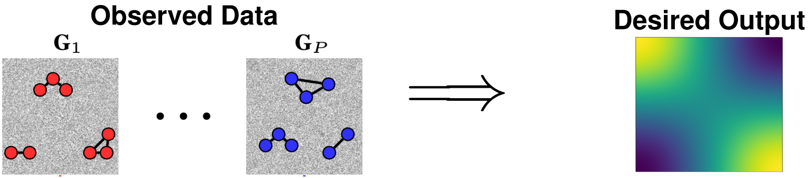
- A graphon $W : [0,1]^2 \to [0,1]$ is a continuous, generative model for graphs.
- $W(x,y)$ = probability of an edge between latent node positions x and y.
- It provides a **resolution-free** representation for large networks.



The Core Problem: Graphon Estimation

Problem

Given one or more observed graphs $\mathcal{G} = \{G_1, \ldots, G_P\}$, assumed to be sampled from an unknown true graphon W^*, how do we ***accurately*** *and* ***efficiently*** *recover an estimate $\hat{W}$?*



The Bottleneck: Why is this Hard?

RICE

Existing graphon estimation methods face significant challenges:

1. **Scalability:** Many methods (e.g., SAS, USVT) are limited by the sample graph's resolution.
2. **Complex Alignment:** Modern INR-based methods (e.g., IGNR) require estimating **latent node variables**.
3. **Costly Optimization:** They often rely on minimizing the **Gromov-Wasserstein (GW) distance**, which has combinatorial complexity.

The Gap: We need an estimator that is **scalable**, **direct** (no latent variables), and **efficient** (no GW distance).

Our Core Idea: Match Moments, Not Nodes

RICE

Key Insight

A graphon is uniquely determined by its "moments", the densities of all its subgraphs (motifs).

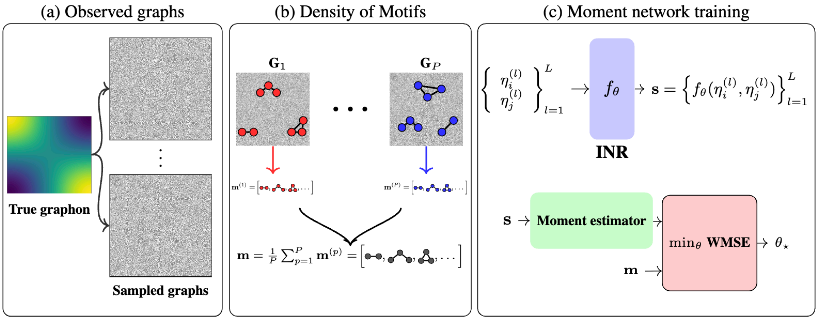
Our Proposal: Instead of aligning nodes, let's align *moments*.

1. **Observe:** Compute empirical subgraph counts ($\mathbf{m}$) from the input graph(s) G. (Fast, parallelizable)
2. **Model:** Represent the graphon $\hat{W}_\theta$ with an Implicit Neural Rep (INR).
3. **Train:** Optimize the INR θ to *directly match* the observed moments.

$$\theta^* = \arg\min_\theta \text{Loss}(\, \hat{\mathbf{m}}(\, \hat{W}_\theta), \mathbf{m}\,)$$

Our Core Idea: Match Moments, Not Nodes

RICE



MomentNet Stage 1 - Empirical Moments

RICE

Step 1: Computing the density of motifs

- Consider set of motifs $\mathcal{F} := \{\mathcal{G}_{F_1}, \mathcal{G}_{F_2}, \ldots, \mathcal{G}_{F_{|\mathcal{F}|}}\}$
- Count occurrences of motifs (using ORCA)
- Calculate empirical motif density vector $\mathbf{m}^{(p)}$ for each graph $\mathcal{G}_p$
- Overall empirical moment vector $\mathbf{m} \in \mathbb{R}^{|\mathcal{F}|}$

$$\mathbf{m} = \frac{1}{P} \sum_{p=1}^{P} \mathbf{m}^{(p)}$$

Efficiency Gain

- Input graphs are discarded after this stage
- Reduces computational overhead significantly

MomentNet Stage 1 - Empirical Moments

RICE

Step 1: Computing the density of motifs

- Consider set of motifs $\mathcal{F} := \{\mathcal{G}_{F_1}, \mathcal{G}_{F_2}, \ldots, \mathcal{G}_{F_{|\mathcal{F}|}}\}$
- Count occurrences of motifs (using ORCA)
- Calculate empirical motif density vector $\mathbf{m}^{(p)}$ for each graph $\mathcal{G}_p$
- Overall empirical moment vector $\mathbf{m} \in \mathbb{R}^{|\mathcal{F}|}$

$$\mathbf{m} = \frac{1}{P} \sum_{p=1}^{P} \mathbf{m}^{(p)}$$

Efficiency Gain

- Input graphs are discarded after this stage
- Reduces computational overhead significantly

MomentNet Stage 2 - Training

RICE

Step 2: Training the moment network ($W_\theta = f_\theta$)

Moment estimator

- Computes estimated induced motif density $t'(\mathcal{G}_F, W_\theta)$
- Approximation via Monte Carlo with L samples

$$\hat{t}'(\mathcal{G}_F, f_\theta) = \frac{1}{L}\sum_{l=1}^{L}\left[\prod_{(i,j)\in\mathcal{E}_F} f_\theta(\eta_i^{(l)}, \eta_j^{(l)}) \prod_{(i,j)\notin\mathcal{E}_F} (1 - f_\theta(\eta_i^{(l)}, \eta_j^{(l)}))\right]$$

- The estimator $\hat{t}'(\mathcal{G}_F, W_\theta)$ is differentiable w.r.t. θ

Loss Function: Weighted Mean Squared Error

$$L(\theta) = \sum_{i=1}^{|\mathcal{F}|} w_i\,(m_i - \hat{m}_i(\theta))^2$$

- Minimizes discrepancy between $\mathbf{m}$ and estimated moments $\hat{\mathbf{m}}(\theta)$
- Inverse weighting $w_i = 1/m_i$ balances motif frequency

- Assumptions

  $\Rightarrow$ INR approximates empirical motif densities (Universal Approx.)

Theorem: Cut Distance Probabilistic Bound

For $N > \frac{k(k-1)}{\delta_M}$, if $F_k \cdot 2\exp\left(-\frac{PN}{4k^2}\left(\frac{\delta_M}{2} - \frac{k(k-1)}{2N}\right)^2\right) < \zeta$ where $\zeta > 0$ is a confidence level and $\delta_M > 0$ is the motif deviation threshold. With prob. at least $1-\zeta$,

$$d_{\text{cut}}(\hat{W}_\theta, W^*) < \eta$$

- Result

  $\Rightarrow$ With probability $1-\zeta$ the cut distance is bounded

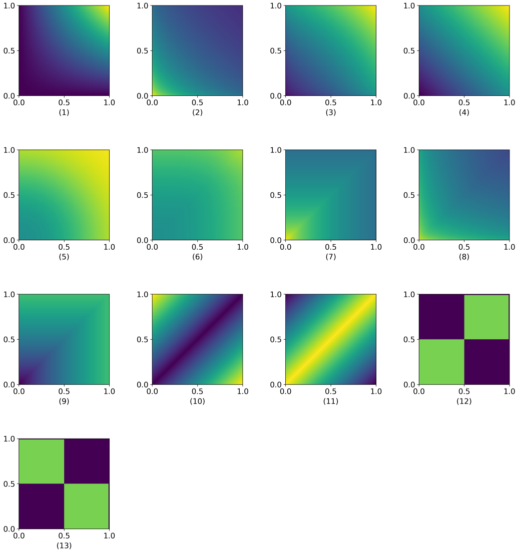
  $\Rightarrow$ Probability constraint decays exponentially with P, N

  $\Rightarrow$ Motif fidelity in data implies proximity to W^* in cut distance

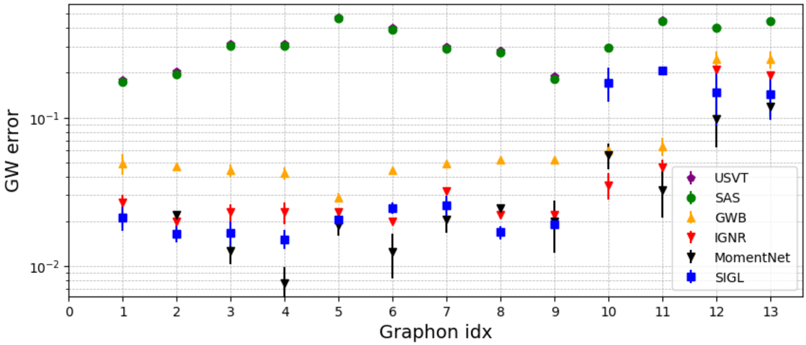
From each graphon, we then generate 10 distinct graphs of varying sizes, specifically containing $\{75, 100, \ldots, 300\}$ nodes respectively. The 13 graphons used are:

	$W(x,y)$
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	$\lvert x-y\rvert$
11	$1-\lvert x-y\rvert$
12	$0.8\mathbf{I}_2 \otimes \mathbb{1}_{[0,\frac{1}{2}]^2}$
13	$0.8(1-\mathbf{I}_2) \otimes \mathbb{1}_{[0,\frac{1}{2}]^2}$

Numerical Experiments - Synthetic Graphons I



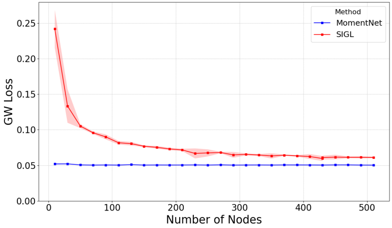
- Performance comparison
- Metric: GW loss (distance between estimated and true graphon)
- Results
  - ⇒ MomentNet outperforms SOTA in 9/13 graphons
  - ⇒ Shows robustness across different graphon types



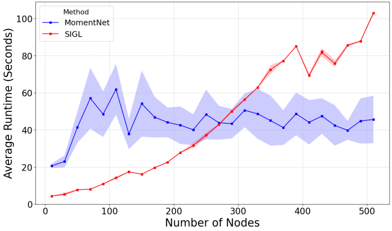
Numerical Experiments - Scalability

RICE

- Scalability for Graphon $W(\eta_i, \eta_j) = 0.5 + 0.1\cos(\pi\eta_i)\cos(\pi\eta_j)$
  - $\Rightarrow$ Comparison with SOTA baseline SIGL [Azizpour24]
- GW Loss vs. Node Count N (a)
  - $\Rightarrow$ MomentNet consistently low GW loss across all N
  - $\Rightarrow$ SIGL matches MomentNet for very large N
- Runtime vs. Node Count N (b)
  - $\Rightarrow$ MomentNet's runtime increases only modestly
  - $\Rightarrow$ SIGL's runtime still escalates sharply
- MomentNet: practical approach for large-scale networks



(a)



(b)

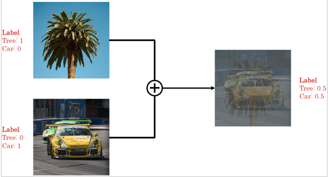
MomentMixup - Introduction

RICE

- Data augmentation: crucial for generalization in graph learning

- Mixup
  - ⇒ New samples by combination of existing
  - ⇒ Synthetic sample $(\tilde{x}, \tilde{y})$ as

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \; ; \; \tilde{y} = \lambda y_i + (1 - \lambda) y_j$$



- Mixup in graphs
  - ⇒ Graph structure non-Euclidean
  - ⇒ Mixup is challenging
    - Previous graph mixup methods
      - ⇒ G-Mixup [Han22] operate in graphon domain
      - ⇒ GraphMAD [Navarro22] operate in latent space
    - MomentMixup approach
      - ⇒ Mixup in moment space
      - ⇒ Uses MomentNet to obtain the mixed graphon W_{aug}

MomentMixup - Overview

RICE

- 1. Compute Class Moments
  ⇒ $\mathbf{m}_c$: average moment vector for each class c
- 2. Target Moment and Label
  ⇒ Target mixed moment $\mathbf{m}_{target} = \sum \alpha_c \mathbf{m}_c$
  ⇒ Target mixed label $y_{target} = \sum \alpha_c y_c$
- 3. Train MomentNet
  ⇒ Train MomentNet with $\mathbf{m}_{target} \rightarrow$ mixed graphon W_{aug}
- 4. Sample and Augment
  ⇒ Generate N_{graphs} new samples $\mathcal{G}_{new}$ from W_{aug}
  ⇒ Integrate $(\mathcal{G}_{new}, y_{target})$ into the training set

MomentMixup - Overview

RICE

- 1. Compute Class Moments
  - ⇒ $\mathbf{m}_c$: average moment vector for each class c
- 2. Target Moment and Label
  - ⇒ Target mixed moment $\mathbf{m}_{target} = \sum \alpha_c \mathbf{m}_c$
  - ⇒ Target mixed label $y_{target} = \sum \alpha_c y_c$
- 3. Train MomentNet
  - ⇒ Train MomentNet with $\mathbf{m}_{target} \rightarrow$ mixed graphon W_{aug}
- 4. Sample and Augment
  - ⇒ Generate N_{graphs} new samples $\mathcal{G}_{new}$ from W_{aug}
  - ⇒ Integrate $(\mathcal{G}_{new}, y_{target})$ into the training set

- 1. Compute Class Moments
  - ⇒ $\mathbf{m}_c$: average moment vector for each class c
- 2. Target Moment and Label
  - ⇒ Target mixed moment $\mathbf{m}_{target} = \sum \alpha_c \mathbf{m}_c$
  - ⇒ Target mixed label $y_{target} = \sum \alpha_c y_c$
- 3. Train MomentNet
  - ⇒ Train MomentNet with $\mathbf{m}_{target} \rightarrow$ mixed graphon W_{aug}
- 4. Sample and Augment
  - ⇒ Generate N_{graphs} new samples $\mathcal{G}_{new}$ from W_{aug}
  - ⇒ Integrate $(\mathcal{G}_{new}, y_{target})$ into the training set

MomentMixup - Overview

- 1. Compute Class Moments
  - ⇒ $\mathbf{m}_c$: average moment vector for each class c
- 2. Target Moment and Label
  - ⇒ Target mixed moment $\mathbf{m}_{target} = \sum \alpha_c \mathbf{m}_c$
  - ⇒ Target mixed label $y_{target} = \sum \alpha_c y_c$
- 3. Train MomentNet
  - ⇒ Train MomentNet with $\mathbf{m}_{target} \to$ mixed graphon W_{aug}
- 4. Sample and Augment
  - ⇒ Generate N_{graphs} new samples $\mathcal{G}_{new}$ from W_{aug}
  - ⇒ Integrate $(\mathcal{G}_{new}, y_{target})$ into the training set

- Graph classification task
  - ⇒ Base classifier: Graph Isomorphism Network (GIN) [Xu18]
  - ⇒ Metric: Classification accuracy on the test set
- Results
  - ⇒ MomentMixup better performance over standard G-Mixup
  - ⇒ Shows a clear advantage on datasets with smaller graphs (AIDS)

Dataset	**IMDB-B**	**IMDB-M**	**REDD-B**	**AIDS**
#graphs	1000	1500	2000	2000
#classes	2	3	2	2
#avg.nodes	19.77	13.00	429.63	15.69
#avg.edges	96.53	65.94	497.75	16.2
		GIN		
No Augmentation	71.55±3.53	48.83±2.75	91.78±1.09	98±1.2
G-Mixup w/ USVT	71.94±3.00	50.46±1.49	91.32±1.51	97.8±0.9
G-Mixup w/ SIGL	73.95±2.64	50.70±1.41	**92.25**±1.41	97.3±1
MomentMixup	**74.30**±2.70	**50.95**±1.93	91.8 ± 1.2	**98.5**±0.6

Conclusion & Contributions

RICE

Contribution 1: MomentNet

- **A scalable graphon estimator** that matches moments (motif densities) using an INR.

  ⇒ **Novelty:** Bypasses latent variable estimation and costly GW distance optimization.

  ⇒ **Result:** Achieves SOTA accuracy (9/13 graphons) with vastly superior runtime.

Contribution 2: Theoretical Guarantee

- **Provides a cut distance bound** for the estimation error.

  ⇒ Links motif fidelity in the data to the final estimation accuracy.

  ⇒ Shows error probability decays **exponentially** with data size (P, N).

Contribution 3: MomentMixup

- **A new data augmentation technique** for graph classification.

  ⇒ **Novelty:** Performs mixup directly in the **moment space**, not the graphon space.

  ⇒ **Result:** Outperforms G-Mixup baselines on 3/4 datasets.

Conclusion & Contributions

RICE

Contribution 1: MomentNet

- **A scalable graphon estimator** that matches moments (motif densities) using an INR.

  $\Rightarrow$ **Novelty:** Bypasses latent variable estimation and costly GW distance optimization.

  $\Rightarrow$ **Result:** Achieves SOTA accuracy (9/13 graphons) with vastly superior runtime.

Contribution 2: Theoretical Guarantee

- **Provides a cut distance bound** for the estimation error.

  $\Rightarrow$ Links motif fidelity in the data to the final estimation accuracy.

  $\Rightarrow$ Shows error probability decays **exponentially** with data size (P, N).

Contribution 3: MomentMixup

- **A new data augmentation technique** for graph classification.

  $\Rightarrow$ **Novelty:** Performs mixup directly in the **moment space**, not the graphon space.

  $\Rightarrow$ **Result:** Outperforms G-Mixup baselines on 3/4 datasets.

Conclusion & Contributions

RICE

Contribution 1: MomentNet

- **A scalable graphon estimator** that matches moments (motif densities) using an INR.

  ⇒ **Novelty:** Bypasses latent variable estimation and costly GW distance optimization.

  ⇒ **Result:** Achieves SOTA accuracy (9/13 graphons) with vastly superior runtime.

Contribution 2: Theoretical Guarantee

- **Provides a cut distance bound** for the estimation error.

  ⇒ Links motif fidelity in the data to the final estimation accuracy.

  ⇒ Shows error probability decays **exponentially** with data size (P, N).

Contribution 3: MomentMixup

- **A new data augmentation technique** for graph classification.

  ⇒ **Novelty:** Performs mixup directly in the **moment space**, not the graphon space.

  ⇒ **Result:** Outperforms G-Mixup baselines on 3/4 datasets.

Thank You

Paper Link: