nature portfolio

Peer Review File

Coarse-graining network flow through statistical physics and machine learning

Corresponding Author: Mr Zhang Zhang

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
The manuscript proposes a new method for coarse-graining in complex networks that combines recent results based on the density matrix with machine learning algorithms. This approach utilizes the statistical physics model of the network as input and trains the networks to perform different 'blocks' and reduce the network accordingly. The manuscript does not propose any substantial novelty to the actual RG problem on complex networks. The only novelty they can justify is using the density matrix and machine learning algorithms, which can be useful for performing network reduction in some specific cases. However, it still needs to gain information on the physical meaning of the coarse-graining process. I think this article is more subtle to be published in other specific journals such as Scientific Reports or Communications Physics.

Major comments:

Pag. 3 When the authors introduce the actual RG problem on top of complex networks, they state that: "non of these methods is designed or guaranteed to preserve the flow of information among sytem's units", and "even the Laplacian Renormalization Group, which is explicitly based on diffusion dynamics does not aim to preserve the flow and, therefore, leads to compressions that poorly resemble the information dynamics in the original networks."

That is not true. Previous ways to perform the coarse-grain process on complex networks have exploited its intrinsic scale-invariant properties, as, for instance, in the pioneering works of Song et al. Still, these methods used an 'a priori' knowledge about the network unavailable in general disordered structures. The community has made much effort to develop further approaches, e.g., embedding complex networks in hyperbolic spaces to perform the analogous procedure to Kadanoff blocks in real space, as usually done in statistical physics. I insist that ALL these methods preserve the flow of information in the specific cases they considered. Instead, the Laplacian Renormalization Group tackles the problem of directly extending field theoretical approaches to heterogeneous structures. How is it possible to do that? By considering the analogous to the discrete Laplacian in heterogeneous networks: the Laplacian matrix. This allows to define both the real-space and k-space framework to exactly solve the RG coarse-grain problem in disordered structures. In fact, as the authors explicitly say, they use the same density matrix as in the Laplacian Renormalization Group, which ensures an isospectral transformation and, therefore, preserves the flow of information among units at different resolution scales depending on the network architecture.

Pag.4 The authors say: "For instance, in the BA network, hub nodes are coarsened together, while leaf nodes are coarsened together..." "We will present more experimental results in subsequent sections and show that our approach preserves the information flow even for large compressions, in synthetic and empirical systems."

What do they mean when they ensure they preserve the information flow even for large compressions? Of course, if they are using the information of the density matrix, they will preserve the network properties for Barabasi Albert networks, which are essentially scale-free structures. They are canceling out eigenvalues of the network Laplacian by only employing a 'black-box' algorithm that uses the information from the density matrix. Again, it is not a main novelty but a particular application of the Laplacian Renormalization Group framework to select particular supernodes. The choice of making Kadanoff blocks is not unique in homogeneous or heterogeneous structures. Still, in complex networks, there exists a myriad of combinations leading to a reduced version of the network: the authors are selecting a particular one.

Pag. 15 The authors say: "A definitive framework for network renormalization that reduces the size of large network datasets

while preserving their fundamental properties is still missing." This statement is incorrect and requires to be rephrased.

Minor comments:

I suggest to rewrite the title to mention the use of machine learning for coarse-graining analysis explicitly.

References: The authors have overlooked important references in the network RG problem. In my opinion, the manuscripts of (i) Gfeller, D. & De Los Rios Phys. Rev. Lett. 99, 038701 (2007), (ii) Goh, K., et al., Phys. Rev. Lett. 96, 018701 (2006), (iii) Radicchi, F., et al. Phys. Rev. Lett. 101, 148701 (2008). or (iv) Rozenfeld, H. et al. Phys. Rev. Lett. 104, 025701 (2010), and (v) Boguña, M., et al. Nat. Rev. Phys. 3.2 (2021): 114-135 must be appropriately cited.

Pag. 6. The authors define the network free and internal energy, but they no longer use these properties. This can be confusing for the readers. If the authors want to provide a full perspective on the canonical statistical physics description for complex networks, they would also cite the manuscript of Villegas et al. Phys. Rev. Res. 4.3 (2022): 033196.

Pag.17: "cross-entorpy" must be "cross-entropy"
Pag. 16: "synthetic network" must be " synthetic networks"
Pag.11: "50 node" would be "50-node"

Reviewer #2

(Remarks to the Author)
I read with interest the paper Network Information Dynamics Renormalization Group by Zhang et al.
I liked very much the approach outlined by the authors trying to figure out a simple method that might consider both the structural properties of the complex networks and the dynamic processes that are acting on them.
I think the paper might be suitable for your journal once the authors clarify some critical points.

If I understood clearly the Network Information Dynamics Renormalization Group introduced here, this is still based on the decimation part of RG, but the coarsening is not driven by the lattice/network structure but instead is driven by the role played in dynamics (hubs with hubs and leaves with leaves).

My first question is how to ensure network connectedness with such a decimation procedure.

Furthermore, I assume that the scaling part of RG must also reflect this particular choice of decimation procedure. It would be nice (I acknowledge that it could be challenging and take a lot of time to make a formal comparison) if the authors could comment on this point when describing the modification induced in the partition function,

Finally, could the authors comment if the system is considered at equilibrium and why? This is the only case in which a proper partition function can be defined,

The work is original, and claims can be supported with a revision. The methodology is sound, and details are provided. I

Reviewer #3

(Remarks to the Author)
The authors use a graph neural network approach to introduce a renormalization scheme that preserves global information dynamics in complex networks. Information dynamics is modeled as a diffusion process ruled by the Laplacian of the network. A multilayer perceptron is utilized to minimize the mean absolute error between the normalized partition function of the compressed network and the original one. In this manner, macroscopic magnitudes derived from the partition function, such as the Von Neumann entropy and the free energy, are expected to be preserved in the renormalization flow.

While the topic is certainly relevant, and the idea of preserving the partition function is appealing, I have major concerns. The main contribution of the paper appears to be portrayed as a mere technical improvement. In other words, the novelty and significance of the contribution is not clearly conveyed, and many claims lack supporting evidence. Additionally, the manuscript contains inaccuracies. I would not recommend accepting the manuscript in its present form. A new version may be reconsidered once the ideas introduced in the manuscript are thoroughly addressed, their significance and novelty clarified, and the inaccuracies corrected. Other aspects, such as the tendency to overstate, also need improvement.

The research in the manuscript is very close to the Laplacian renormalization method recently introduced in Ref. 48. It is fundamental to clearly and convincingly explain how the introduced method differs from Laplacian renormalization and advances the results obtained in that reference.

It is also fundamental to expand the evidence provided to support the claims. There are many unsupported claims in the manuscript. For instance:

* Abstract: "Our work offers a low-complexity renormalization method breaking the size barrier for meaningful compressions of extremely large networks". I did not see extremely large networks, say of 100000 nodes, addressed in this work. The synthetic networks under analysis have only 256 nodes, and the empirical networks have a maximum of 2500 nodes.

Results of the analysis of large networks, say of about 100000 nodes, should be reported.

* An analysis of a more extensive set of real networks is required. Given the authors' claim about the speed of computation and the already collected dataset of 1100 empirical networks, extending the analysis beyond the two analyzed real networks should not be overly challenging. Statistics regarding the performance of the method across various networks should be provided. Additionally, is there any observable difference depending on the network domain?

No details are provided on how the alternative renormalization methods were implemented. A "See Supplementary Materials for details" was included in the main text, but I could not find the information there.

More details about training are fundamental to understand the computational cost of the GNN, but are not provided. In particular, what does it mean "with sufficient training" in section Neural Network Architecture?

Also in reference to the training process, the authors measure the deviation of the partition function of the compressed layer from the original layer by uniformly selecting ten points from the logarithmic interval of time before the partition function converges. These points compose a vector representing the partition function for minimizing the mean absolute error between the two. More details should be provided regarding the evolution of the partition function. Is it always smooth? Are the ten selected points truly representative of the curve?

The section on interpretability does not open the black box of the neural network used. It essentially serves as a sensitivity analysis of the contribution of structural features to the model's output, as acknowledged by the authors. From this analysis, the authors suggest that nodes with a similar "ecological niche" are grouped together. This claim should be substantiated further.

There are inaccurate statements in the manuscript. For instance:

* "This non-local effect has no structural counterpart in other methods, as in the classical network renormalization methods, super-nodes always consist of connected subnetworks." This is simply wrong. There are network renormalization methods that coarsen groups of nodes even without the need that those nodes are connected.

* The authors assert that geometric methods are order N^3, while in the relevant papers it is stated that the method is N^2.

The investigation leaves many relevant questions unanswered:

* What is the behavior of the network topology in the flow? Are the degree distribution and the clustering spectrum preserved? What about degree-degree correlations? Which are the effects of having a homogeneous/heterogeneous connectivity, high/low levels of clustering, etc.?

* In relation to the point above, results in Fig. 2 suggest that the method works much better for trees. This point needs to be checked more carefully.

Less critical but still important, the terminology is occasionally ambiguous and confusing, and there is some language misuse. For instance:

*"upon reaching a certain compression size that plays the role of a transition point, further compressions lead to significantly higher costs", and "transition point" is used in several other places. In our field, a transition has a very specific meaning. I could not see any transition point in the reported results.

* The niches the authors talk about in the paper are not ecological in any form.

* Second paragraph of the introduction "to search for proper compressions of networks...that preserve the flow.", and the same idea again in the third paragraph. Meaningless sentence. What is preserved is not the flow, but some magnitude in the flow.

Also, be attentive to inflated self-citations. Specifically, in the first paragraph of the introduction, there are citations on multiplexity and interdependence, a topic not directly relevant to the topic of the manuscript. These citations are used to introduce seven references, of which four are self-citations.

There are typos in the text, for instance "open the doors" in the first paragraph of the introduction, etc..

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

I have appreciated the meticulous and detailed response of the referees. Still, as my first referee report explained, the paper does not propose any substantial novelty to the actual RG problem on complex networks. Indeed, in the present form, the manuscript can add some confusion to the global picture of the problem. Again, after addressing a major revision I can recommend it to be published in other specific journals such as Scientific Reports or Communications Physics. In what follows, the authors can find my comments on the revised version of the manuscript, which has not essentially changed from its original form.

I suggest the authors do a major revision to different new parts of the manuscript.

1) "LRG focuses on the specific heat at the specific scale of phase transition to define their information cores, we focus on replicating the full partition function profile at all scales of propagation." "LRG compresses network size by discarding the fast modes of diffusion processes (corresponding to eigenvalues greater than τ1), while our method tries to preserve the complete distribution of eigenvalues at any scale".

I want to clarify about the nature and the spirit of the LRG. The authors make some statements that are not true both in the rebuttal letter and across the manuscript. They must be softened.

The LRG maps the Fourier space information contained in the eigenvalues of the Laplacian onto the density matrix the authors use here. In fact, tightly coupling spatial and temporal scales, it uses specific times to perform different network reductions. But, still, I insist it is a general theory does not depend on specific scales but that allows to analyze of different network mesoscopic scales.

Instead, I want to emphasize here what the RG is. The renormalization group (RG) is the fundamental tool in statistical physics and statistical field theory to study
and describe the physical behavior of systems showing a multi-scale and/or scale-invariant behavior as, for instance, it happens in the proximity of a critical point of a second-order phase transition at and out of equilibrium. It comprises three fundamental ingredients:

i) A coarse-graining procedure: it can be performed either (a) in real space, through the derivation of mesoscopic collective variables (i.e., block variables) from a local resummation of the microscopic ones inside identical mesoscopic cells of size Λ tiling the whole space, or (b) in the conjugate Fourier space by rewriting the model through the Fourier modes of the microscopic variables and explicitly integrating all the modes with wavelength smaller than an arbitrary mesoscopic value Λ. In both cases, the new lower cut-off of the model is given, respectively by Λ. For instance, in the Ising model, we can either define à la Kadanoff mesoscopic magnetic moments summing the microscopic spins inside identical cells, or à la Wilson, through the Hubbard-Stratonovich transformation, we can rewrite the model in terms of a continuous spatial field whose Gaussian approximation is diagonal in Fourier space.

ii) Reformulation of the model (i.e., redefinition of the interaction constants of the system) in terms of the block variables in real space and of the variable modes with wavelength larger than the new lower cut-off.

iii) Spatial rescaling, so that the new lower cut-off becomes the new unit length.

To extend this approach to models defined on heterogeneous graphs, the fundamental step to overcome to formulate a correct RG approach was the first one. Indeed, given a homogeneous model (i.e., with interaction constants identical for all spatial points as the Ising model with nearest-neighbor interactions, for instance), the definition of coarse-graining in identical mesoscopic cells or integrating small wavelength modes all over the space is immediate both in homogeneous spaces (e.g., Rd) or homogeneous lattices or trees due to the continuous or discrete translational invariance of the space itself. This fundamental property is completely lost in heterogeneous networks. Consequently, both the definition of real space block variables and the resummation procedure of small wavelengths of the model in Fourier space may appear completely arbitrary or even meaningless. The LRG scheme for connected undirected graphs tackles this point as a natural extension of the coarse-graining in homogeneous spaces. First, I want to notice that the Laplacian operator in such spaces can be strictly
related to the progressive coarse-graining procedure in the RG: it is an operator with the spectrum of eigenvalues given by eigenvalues –k^2 , with k Fourier wave-vector, i.e., the inverse of the wavelength, which has no characteristic scale. The corresponding eigenvectors are exactly given by the Fourier basis of plane waves e^ikx , which are also eigenfunctions of the translation group.

I think that the authors can easily see that a network reduction automatically implies a rescaling function of the eigenvalues distribution function, which needs to be properly rescaled if the network maintains its intrinsic properties, but that CANNOT remains constant when changing the system scale.

The basic concepts I have explained here can be easily followed in the classical books of Kardar, M. Statistical Physics of Fields (Cambridge Univ. Press, 2007). or Amit, D. J. and Martin-Mayor, V. Field Theory, the Renormalization Group, and Critical Phenomena 3rd edn (World Scientific, 2005).

2) "LRG performs well on heterogeneous networks, especially on BA networks and scale-free trees. Therefore, LRG successfully delivers what it promises— preserving the flow of information in those cases"

It is not true. BA or random trees are specific examples of some networks that do not change their intrinsic properties when a

network rescaling is performed. The LRG works for every network, leading to scale-dependent properties in such networks presenting characteristic scales.

3) Figure 5. Barabasi network compression.

I have some specific comments about the figure the authors have presented in the rebuttal letter. In particular, with BA networks fixing the parameter m=1, we know these networks are trees. The final network does not maintain the intrinsic properties of the original one. I would like to see, for instance, some plots showing the diameter for a growing process and a reduction process of those networks.

4) How we theoretically differ from and operationally outperform LRG

I have some significant comments regarding this specific new paragraph. The authors ensure that: "We base our method on the fact that the partition function contains the full information of the macroscopic descriptors of the flow, including network entropy and free energy— as demonstrated in the previous sections. Therefore, by replicating the full partition function profile at all propagation scales (all values of τ), we replicate the macroscopic features like the diversity of diffusion pathways and the flow speed in reaching distant parts of the network." This is right what the LRG does. So, this paragraph must be removed from the paper or completely rewritten.

It is important to note that the LRG is not limited to heterogeneous networks like Barabasi-Albert (BA) and scale-free trees, as the authors suggest. In fact, the LRG is designed to work with any general network or topology, including those found in real-world scenarios.

"Instead, our method preserves the complete distribution of eigenvalues at all scales simultaneously by maintaining the full behavior of the partition function." As I have explained before, general RG procedures MUST rescale the distribution of eigenvalues.

Minor comments:

I still think the authors do not use at all the free energy nor it is useful for the specific statements they treat here. I recommend to move it to a methods section of the manuscript or tu Supplementary Information.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)
I am glad to see that the authors had a positive attitude against the comments outlined in my report. My opinion is that the author successfully answered to criticisms raised and modified the paper accordingly.
I am in favour of publication of paper in the present form

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)
The authors have addressed most of my comments, and I thank them for the thorough answers. The work has been significantly improved. However, many of these improvements are not reflected in the main text, and there are still some issues that need to be addressed.

In the experiments designed to answer some of my previous comments, the authors found:
"Regardless of the network's size or the field it belongs to, its transitivity (the fraction of all possible triangles present in the network) has a major impact on compression efficiency. As shown in the right subplot of Figure 8, we plotted the transitivity against the error in the partition function before and after compression for all networks and observed a positive correlation (correlation coefficient of 0.676): a network with lower transitivity tends to be more easily compressed."
This effect is important, and I urge the authors to describe this result and the corresponding figure in the main text, along with the explanations given in the rebuttal:
"as the clustering coefficient decreases and long-range connections increase, global information diffusion becomes easier. Therefore, reducing the clustering coefficient makes large-scale information diffusion easier to complete. This is reflected in the partition function, meaning it approaches zero faster at large scales", and
"This means that the model performs better on networks with lower transitivity. This is likely because high transitivity increases the complexity of information diffusion dynamics. Since the transitivity of a tree is 0, this explains why the compression effect on trees differs from that on other synthetic networks."

My question about the behavior of network topology in the flow remains unresolved. Instead of directly addressing the question, the authors demonstrated the impact of different network topological properties on the partition function in Figure

12 of the rebuttal. While this is interesting, it does not answer my original question. The explanation provided on page 63 of the rebuttal states:
"it's important to note that although many topological properties can influence the partition function, our renormalization method can preserve the partition function with high accuracy. This does not mean that our renormalization method will necessarily preserve these structural properties. This is fundamentally because our model produces a graph with weights and self-loops. The addition of weights and self-loops allows the model to maintain the information flow without using some of the structural features (for example, by controlling the amount of information that stays at the node itself through weighted self-loops). Therefore, we did not directly compare the structural properties of networks before and after compression."
This explanation is important and relevant for clarifying the limitations of the proposed methodology and should be included in the main text.

Another significant point clarified in the rebuttal letter but not fully addressed in the manuscript concerns the complexity of the method. In the abstract, the authors state,
"Our work offers a low-complexity renormalization method",
which is not entirely accurate. As clarified in the rebuttal:
"the most time-consuming part is calculating the partition function of the original network to serve as supervisory information, which has a time complexity of O(N3). Therefore, calculating the partition function for extremely large networks is almost impossible, even if the machine learning renormalization model can be trained on smaller network datasets."
These clarifications make the section "One Renormalization Model for Many Real-world Networks" specially relevant. In this section, the authors assert:
"The results demonstrate highly effective learning that can successfully compress the test set, generating macro-networks that preserve the flow of information".
This statement should be accompanied by quantitative analyses.

Minor:

My previous criticism regarding the use of the term "transition point" applies to its substitution with "critical size".

On page 9 of the manuscript, the placement of punctuation marks in the sentence
"other models that often perform at O(N3)--including box-covering, Laplacian RG, or O(N2)-- for instance, geometric RG" is incorrect. It should be revised as:
"other models that often perform at O(N3)--including box-covering and Laplacian RG--, or O(N2), for instance, geometric RG."

(Remarks on code availability)
The code is available at the address provided by the authors.

Version 2:

Reviewer comments:

Reviewer #3

(Remarks to the Author)
I recommend the publication of the paper, provided the authors address some final remarks:

The new title, "Reducing the Dimension of Network-Driven Processes Using Generalized Thermodynamics", is still not very appropriate. Dimension is not a representative word for this text, neither is dimensionality. How are dimension or dimensionality defined in this paper? I understand that this terminology is used as in computer science, but network compression or coarse-graining could be more suitable instead reducing network dimension. In addition, not all network-driven processes are tractable with the proposed method, which focuses on diffusion. And the way thermodynamics is generalized here is quite specific, using machine learning. I suggest that the authors find a more appropriate title.

In page 4, the text "Note that many network renormalization methods coarsen sub-networks into supernodes in a (compresssed) macro network. In contrast, our machine learning model has automatically learned a novel renormalization strategy: it coarsens nodes with similar local structures into a supernode." is not clear. The idea of coarsening nodes (independently of whether they are connected or not) with similar local structures into a supernode is not novel. The existing methods coarsen nodes with similar local structures into supernodes. At this level, the present work belongs to the same category and there is nothing novel in this respect. The methods differ in the precise way they define what is a similar local structure. The present work introduces an alternative way to define what is a similar local structure.

The following text in page 5: "In those frameworks, for instance, a scale-free system is desired to keep its main properties through coarse-graining, while systems with characteristic scales are expected to show varying behavior through reduction. In contrast, our approach is not concerned with revealing the existence of scales in the system. Rather, it tends to find a coarse-graining that preserves the macroscopic flow properties regardless of the network type and topological scales. However, we compare the results of our work against these widely used methods, including Box-Covering, Geometric Renormalization, and Laplacian Renormalization, to show that they can not fulfill our aim (See Supplementary Materials for details)." is confusing and unclear. For instance, "a scale-free system is desired to keep its main properties through coarse-graining", which properties and under which renormalization scheme? I do not think that the results presented in this

manuscript support the claim that the mentioned methods cannot preserve the macroscopic flow properties of networks.

I found some typos, for instance "Experimental results in subsequent sections and show" in page 4.

(Remarks on code availability)

Letter for the Referees:

Detailed Response

Referee #1 (Remarks to the Author):

We thank the referee very much for the insightful comments and questions regarding our manuscript. The referee's feedback has significantly contributed to enhancing the quality of our paper. We appreciate the importance of the raised issues and have addressed each one of them in full detail. To facilitate a quick overview of our responses, we have summarized 3 key points as follows:

First, we would like to thank the referee for highlighting a significant issue in our work: the lack of a clear explanation about the similarity and difference between our method and the Laplacian Renormalization Group(LRG). To address this, we have added a new section in the main text where we discuss how our method differs from LRG. In short, both the theoretical foundations and the performance significantly differ between the two methods, as they are designed to solve different problems.

We show that while both methods use density matrices, they are using them in radically different ways. We lay out the theoretical foundations of both and highlight the difference. Then we use a number of network examples to demonstrate how our method outperforms LRG significantly (often by orders of magnitudes) in replicating the macroscopic information flow features, that are the target of our study.

Also, as the reviewer pointed out, the model's workings need to be discussed further. In summary, we find that our model tends to compress nodes with similar local structures into supernodes. This approach is unique and innovative, and it preserves the function of local structures in information diffusion while reducing the number of redundant local structures. We also discuss how our method's ability to make weighted compressions through optimized grouping strategies allows for precise control of information flow in

the macro network.

We hope this summary has captured the key points of our response. Once again, we thank the referee for the valuable comments. For more specific details and one-by-one response to the referee's comments, please refer to the content in the sections below. We look forward to the referee's further guidance and comments.

The manuscript proposes a new method for coarse-graining in complex networks that combines recent results based on the density matrix with machine learning algorithms. This approach utilizes the statistical physics model of the network as input and trains the networks to perform different 'blocks' and reduce the network accordingly. The manuscript does not propose any substantial novelty to the actual RG problem on complex networks. The only novelty they can justify is using the density matrix and machine learning algorithms, which can be useful for performing network reduction in some specific cases. However, it still needs to gain information on the physical meaning of the coarse-graining process. I think this article is more subtle to be published in other specific journals such as Scientific Reports or Communications Physics.

Major comments:

Pag. 3 When the authors introduce the actual RG problem on top of complex networks, they state that: "non of these methods is designed or guaranteed to preserve the flow of information among sytem's units", and "even the Laplacian Renormalization Group, which is explicitly based on diffusion dynamics does not aim to preserve the flow and, therefore, leads to compressions that poorly resemble the information dynamics in the original networks."

That is not true. Previous ways to perform the coarse-grain process on complex networks have exploited its intrinsic scale-invariant properties, as, for instance, in the pioneering works of Song et al. Still, these methods used an 'a priori' knowledge about the

network unavailable in general disordered structures. The community has made much effort to develop further approaches, e.g., embedding complex networks in hyperbolic spaces to perform the analogous procedure to Kadanoff blocks in real space, as usually done in statistical physics. I insist that ALL these methods preserve the flow of information in the specific cases they considered. Instead, the Laplacian Renormalization Group tackles the problem of directly extending field theoretical approaches to heterogeneous structures. How is it possible to do that? By considering the analogous to the discrete Laplacian in heterogeneous networks: the Laplacian matrix. This allows to define both the real-space and k-space framework to exactly solve the RG coarse-grain problem in disordered structures. In fact, as the authors explicitly say, they use the same density matrix as in the Laplacian Renormalization Group, which ensures an isospectral transformation and, therefore, preserves the flow of information among units at different resolution scales depending on the network architecture.

We thank the reviewer for the careful and brilliant comments that has certainly imrpoved our manuscript.

We fully acknowledge the large and strong literature on RG. Furthermore, we agree that these methods preserve the flow, at least to some extent and in some cases. Also, we confirm that the Laplacian Renormalization Group (LRG) has successfully pushed the boundaries of RG to include structural heterogeneity. In spite of that, in the following, we better explain how our approach is fundamentally different from the existing approaches and how it leads to outperforming them in terms of flow preservation.

Before that, we would like to mention that while "using network density matrices" for RG is what we have in common with LRG, we dramatically differ in "how we use them," and that is why our results are radically different.

Also, we would like to emphasize here that our difference is not limited to the use of

machine learning. In fact, it is much more pronounced in the theoretical foundations and conceptualization of the problem: while LRG focuses on the specific heat at the specific scale of phase transition to define their information cores, we focus on replicating the full partition function profile at all scales of propagation, based on the fact that the partition function contains the full information of the macroscopic descriptors of the flow. Both LRG and our method can renormalize networks at any scale. However, LRG compresses network size by discarding the fast modes of diffusion processes (corresponding to eigenvalues greater than $\frac{1}{\tau}$), while our method tries to preserve the complete distribution of eigenvalues at any scale, thereby maintaining the full behavior of the partition function (see Fig. 3 for a comparison of the two methods in terms of their ability to preserve eigenvalues).

As the referee rightly noted, the spectrum coupled with the temporal scale of signal propagation, describes the flow. More technically, metrics built on top of the spectrum, like the network Von Neumann entropy and free energy, encode macroscopic properties like how diverse the flow pathways are and how fast the flow is [1]. Fortunately, and similarly to the statistical physics of particles, the network partition function contains full information of such metrics— i.e., they can be derived from the full partition function profile. That is why compressions preserving the partition function profile, simultaneously preserve the macroscopic descriptors of information flow across scales— the full derivation is in the text.

Again, we fully agree with the referee that LRG performs well on heterogeneous networks, especially on BA networks and scale-free trees. Therefore, LRG successfully delivers what it promises— preserving the flow of information in those cases. However, the topology of real-world networks is not limited to what the BA model can generate. The two decades of network science has revealed that real-world networks also exhibit important features such as segregation while integration and order while disorder— i.e.,

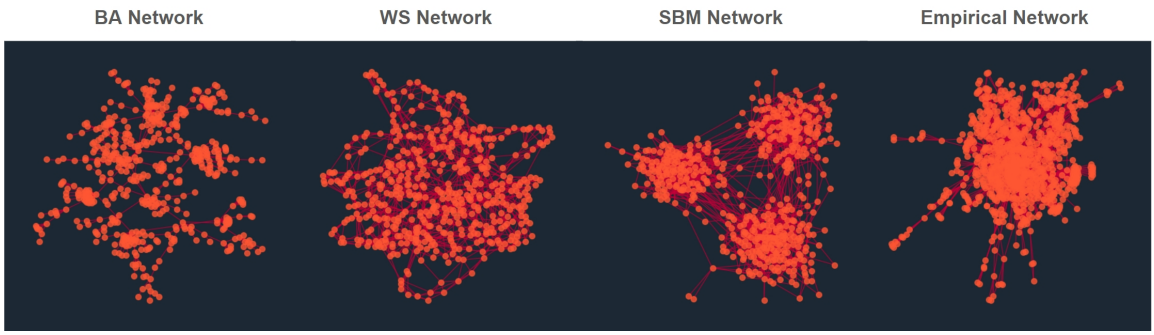

Figure 1: A schematic diagram showing three typical synthetic networks and one empirical network in biology.

modules, motifs, etc. That is why the RG methods must eventually go beyond the current model-specific limiting assumptions.

As we show here, our method clearly outperforms LRG across synthetic and real-world networks. Remarkably, even in the case of the BA model, our method's richer expressive capacity, like allowing for weighted compressions, leads to better flow preservation than LRG. As examples, please see our analysis of a BA (N=500, m=1), Watts-Strogatz (N=500, k=3, p=0.1), an SBM (N=500, mixing parameter μ=0.067), and a real biological network data: a visualization of the networks in the Fig. 1, and the comparitive results in terms of partition functions Fig. 2 and Laplacian spectrum Fig. 3.

In Fig. 2, we show the partition functions of the networks compressed by both methods up to different compression sizes. The subplots show the difference in the partition functions before and after compression (measured by Mean Absolute Error). Specifically, we compressed the size of each network to 50%, 25%, 12.5%, and 6.25% of its original size. We can see that LRG works well with the BA network. However, our method clearly outperforms it **by an order of magnitude** across all datasets, demonstrating its ability to preserve information flow.

In Fig. 3, we display the distribution of eigenvalues of the Laplacian matrix for the

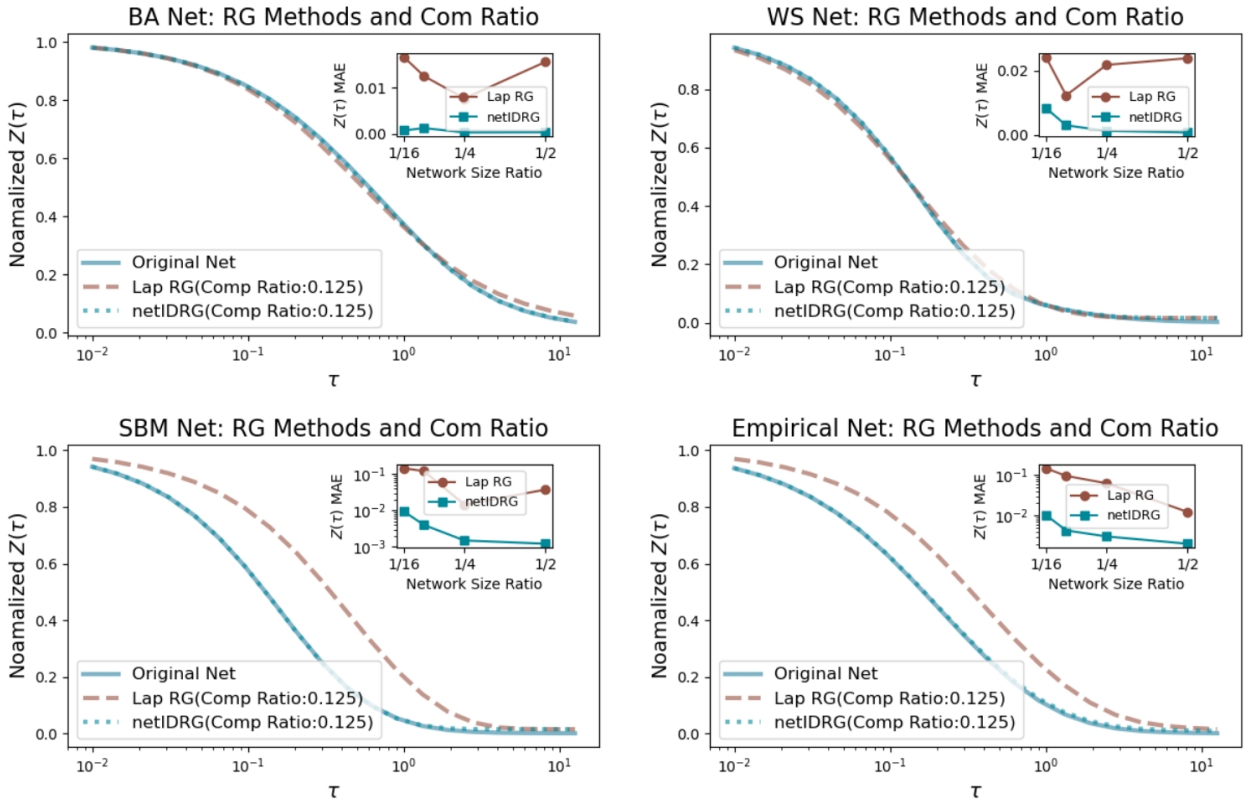

Figure 2: Comparison for netIDRG method and the Laplacian RG method across different networks: On each subplot, we plotted the partition function curve of the original network and the curves when the network is compressed to 12.5% of its original size by two methods. In the insets of each subplot, we show the mean absolute error of the normalized partition function of the original network after being compressed to 50%, 25%, 12.5%, and 6.25% by different methods, measured across a vector of values from 15 uniformly selected points in logarithmic space from 10^{-2} to 10^{1}.

original network and the networks compressed by the netIDRG and LRG methods. Since the network's partition function is defined by $Z(\tau) = Tr(e^{-\tau L}) = \sum_{l=1}^{N} e^{-\tau\lambda_l}$, where λ_l denotes the Lth smallest eigenvalue of the laplacian matrix, all the eigenvalues are non-negative, $\lambda_l \geq 0$, and at least one of the eigenvalues is $\lambda_1 = 0$, the smaller eigenvalues will determine the network's diffusion behavior. For some networks, we only show the first 80% of the sorted eigenvalues to highlight the differences in distributions. The figure shows that our method can more accurately maintain the consistency of the eigenvalue distribution while compressing the network size, thereby ensuring the consistency of information flow behavior. This indicates that our method can preserve a wide range of dynamics.

Additionally, LRG and netIDRG fundamentally differ in their principles for selecting supernodes. LRG selects a subnetwork of the original network to form a supernode based on the results of the diffusion process, and this strategy is discrete, meaning each node can only be part of one supernode. In contrast, netIDRG selects nodes with similar local structures to form a supernode, and this strategy is continuous; for example, node i can enter supernode A with an 80% proportion and supernode B with a 20% proportion. To demonstrate this difference, in Fig 4, we chose a typical BA network with 15 nodes and $m = 1$, and used LRG and netIDRG to renormalize it, compressing it into a macroscopic network with two supernodes (A and B). The left subplot shows LRG's grouping strategy for nodes, where supernodes correspond to two subnetworks formed by adjacent nodes. The right subplot shows netIDRG's strategy, where nodes with similar local structures are given similar grouping results that are continuous, thus finely controlling the structure of the macroscopic network.

Moreover, an intuitive display of the structure of the compressed network can also reflect the differences between the netIDRG method and the Laplacian RG method, as shown in Fig. 5: The Laplacian RG method compresses the original network into a

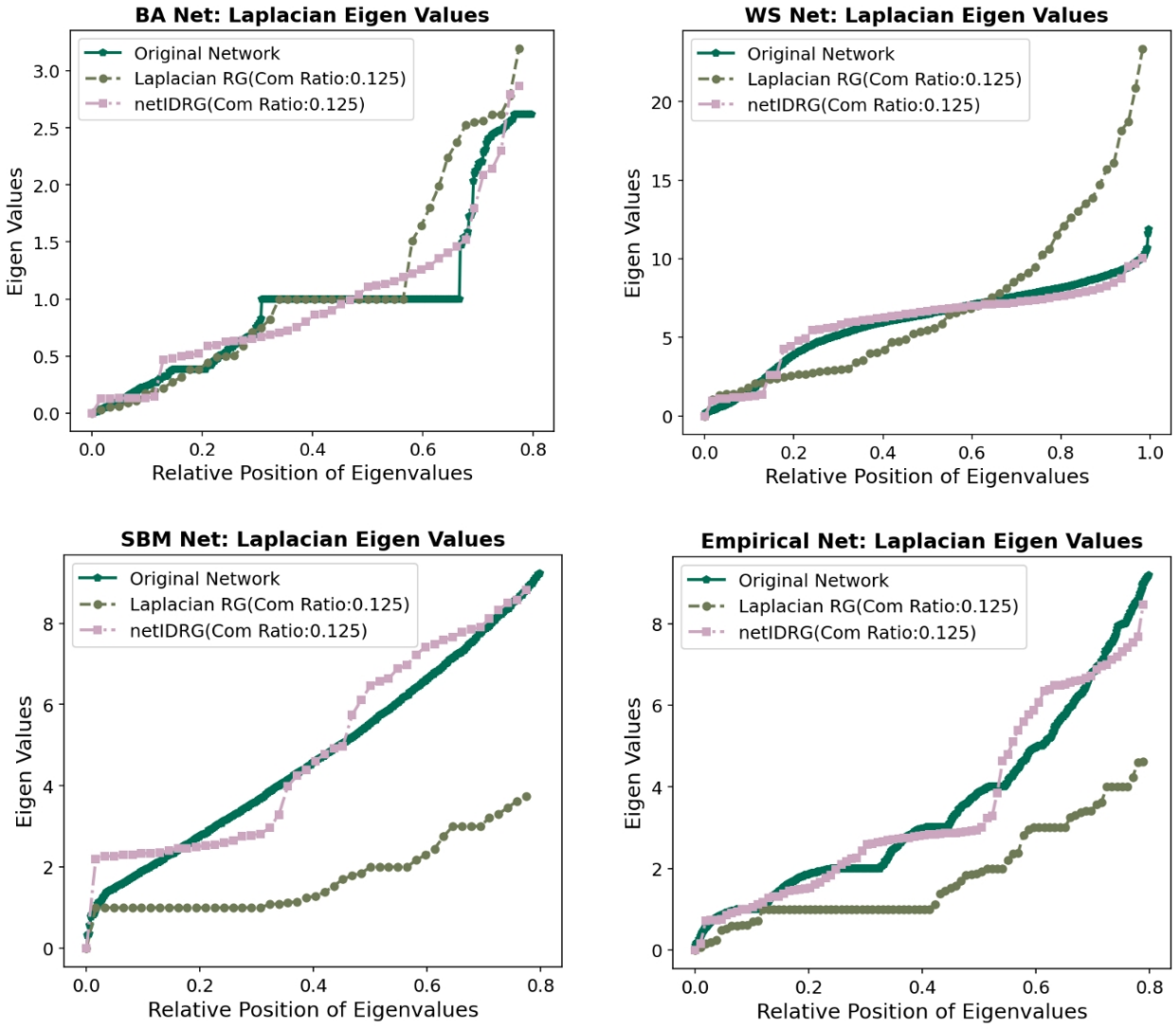

Figure 3: **Distribution of Laplacian eigenvalues:** We present the distribution of the Laplacian matrix's eigenvalues of networks, sorted from smallest to largest, on three synthetic networks and one empirical network. This includes the original network, networks compressed by the Laplacian RG method, and networks compressed by the netIDRG method. Since the partition function is primarily determined by the smaller eigenvalues, we only display the smallest 80% of the eigenvalues for some networks.

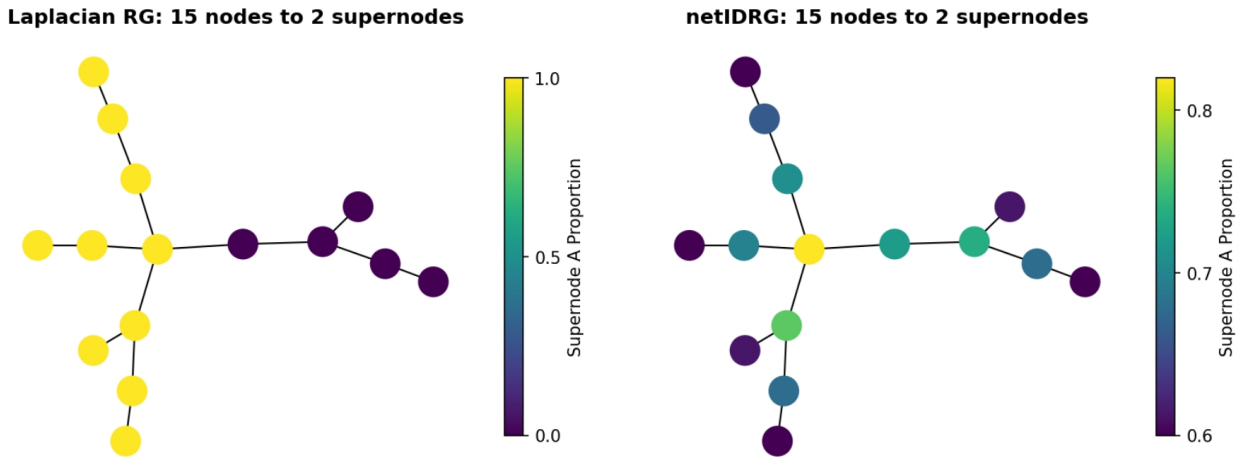

Figure 4: **Differences in supernode composition between methods Laplacian RG and netIDRG:** In this figure, we take a BA network with 15 nodes and renormalize it to two nodes using methods Laplacian RG and netIDRG: supernode A and supernode B. In the left subplot, Laplacian RG groups connected subnetworks into Supernodes. In the right subplot, each node belongs to Supernode A at a certain proportion.

smaller unweighted graph without self-loops. In contrast, the netIDRG method produces a weighted graph that may include self-loops. A weighted network allows for more precise adjustments in the behavior of information flow, thereby enhancing the ability to preserve the information flow of the original network.

To resolve the ambiguities of our previous submission, we add a section to our text comparing our approach and LRG, laying out the provided arguments and attaching the supplementary analyses to the supplementary materials.

We are confident that our theoretical and numerical arguments answer the concerns raised by the reviewer, as it demonstrates that our approach is theoretically and fundamentally different from LRG and that enables flow preservations that work across network models, outperforming LRG by an order of magnitude.

Pag.4 The authors say: "For instance, in the BA network, hub nodes are coarsened together, while leaf nodes are coarsened together..." "We will present more experimental results in subsequent sections and show that our approach preserves the information flow

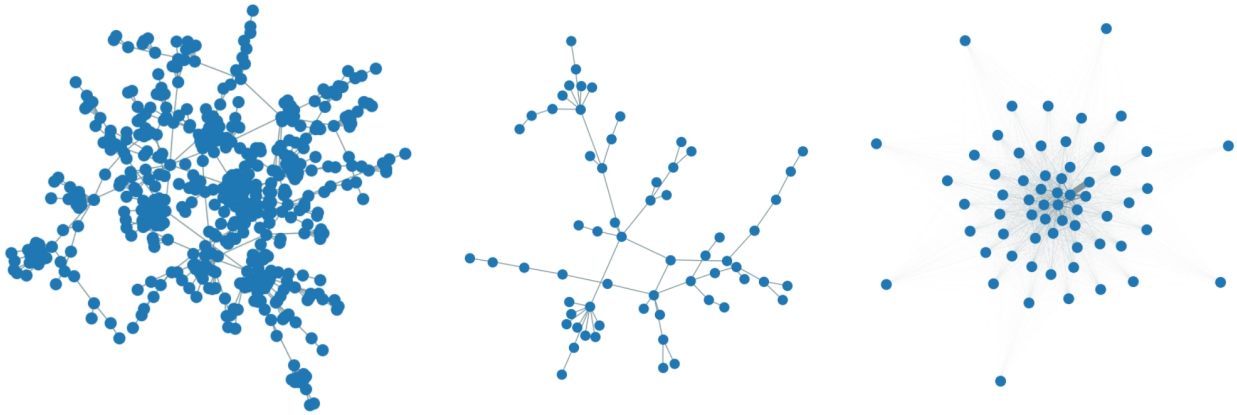

Figure 5: **Visualization of network structures before and after compression:** In the left subplot, we display a BA network with N = 500 and m = 1. In the middle subplot, we show the network structure obtained using the Laplacian RG method, while the network structure obtained with the netIDRG method is presented in the right subplot.

even for large compressions, in synthetic and empirical systems."

What do they mean when they ensure they preserve the information flow even for large compressions? Of course, if they are using the information of the density matrix, they will preserve the network properties for Barabasi Albert networks, which are essentially scale-free structures. They are canceling out eigenvalues of the network Laplacian by only employing a 'black-box' algorithm that uses the information from the density matrix. Again, it is not a main novelty but a particular application of the Laplacian Renormalization Group framework to select particular supernodes. The choice of making Kadanoff blocks is not unique in homogeneous or heterogeneous structures. Still, in complex networks, there exists a myriad of combinations leading to a reduced version of the network: the authors are selecting a particular one.

We thank the reviewer for this comment. It certainly helped us identify ambiguities and improve the representation of our results and method in the manuscript. This part requires an extensive answer. For the referee's convenience, we break it into sections.

How we theoretically differ from and operationally outperform LRG: We would

like to start by highlighting that, as the reviewer certainly knows, scale-freeness is not the only feature of real-world networks, since real-world networks also exhibit important features such as segregation while integration and order while disorder— i.e., modules, motifs, etc. Of course, these features strongly affect the flow of information. So, it is crucial to go beyond the assumptions of previous methods to break the limitations of the state-of-the-art RG.

To provide a summary of how our work differs from others, we attache Table 1. For example, as we show in the table, LRG is suitable for heterogeneous networks, while the Geometric RG model is designed for networks that can be embedded into hyperbolic space S1/H2. As we establish in our work, netIRG has a broader range of applicability and is based on optimizing a different network function.

Table 1: RG methods comparison

Network RG	Works best with	Preserve
Laplacian RG	Heterogeneous with scale-free structures	slow modes of diffusion(corresponding to small eigenvalues)
Geometric RG	Networks that can be embedded in S1/H2 space	Structural Features
netIDRG	**Every network type tested**	Full partition function profile **(flow)**

We would also like to mention that the network density matrices and their spectrum are not limited to merely encoding the hetero/homogeneity of the degree distribution. Instead, as the referee knows, they are rich objects capable of incorporating myriads of topological properties coupled with dynamic processes, giving compact descriptions of network flow. Just to be sure, for a detailed account of how and when network density matrices are reducible to degree distributions, one can check one of our previous works [2].

This generality makes network density matrices powerful tools to characterize the transport properties of complex systems at the macroscopic level. How fast can the quan-

tities of interest flow? How diverse are the carrying pathways? What is the probability that a unit of the flow returns to its initial location? Statistical physics measures, like network entropy and free energy, have been widely used to provide a quantitative proxy answering these questions [1, 3, 4, 5]. Essentially, all these macroscopic measures are functions of the spectrum and the propagation scale. Yet it's worth noting that these functions are non-linear, and their properties can't be simply reduced to their input variables— e.g., the degree distribution of the graph or even the eigenvalue distribution on its own.

To assess whether a compression preserves the flow properties, one can compare it against the macroscopic features of the original flow captured by entropy and free energy. Yet, as we show in the text, a shortcut exists: if a compression replicates the full partition function curve (across τ), it also replicates all the metrics. This is the cornerstone of our approach.

Nevertheless, here, we show how our approach outperforms LRG by an order of magnitude in preserving the flow encoded by the partition functions. It also outperforms LRG when it comes to recovering the distribution of eigenvalues(See Fig. 3). Furthermore, we show that the supernodes built by the two methods significantly differ(see Fig. 4).

How topological features other than heterogeneity are important for the network flow: To demonstrate further how structural properties other than heterogeneity are important for the flow, we refer to Fig. 6. We can see from the subplot a that network density has a significant impact on the partition function. By increasing network density (the probability of edge creation in the Erdos Renyi graph), the network's capability for information propagation is greatly enhanced, which is why we observe a noticeable acceleration in the speed at which the partition function approaches its minimum value.

Similarly, for the small-world effect in the subplot b, the probability of rewiring each edge in the WS network affects the macroscopic measures of flow, especially at larger

propagation scales. This is because increasing the rewiring probability in WS network can reduce the average shortest path length of the network, thereby speeding up the global spread of information. In subplot c, we generate networks using the SBM model and altered the network's μ to observe the effect of inter-community connections on information diffusion.

In the SBM model, $\mu = \frac{k_{out}}{k_{in}+k_{out}}$, in which k_{out} denotes the number of connections between communities and k_{in} denotes the number of connections inside the communities. By removing intra-community edges and adding inter-community edges as the mixing parameter grows, the decline of the partition function slows down at the beginning of the curve. Conversely, the tail of the partition function declines faster because the increase in inter-community edges accelerates the macroscopic flow.

In subplot d, we measure the impact of degree-degree correlation on the network's partition function, which can be quantified by the assortativity coefficient r. When the assortativity coefficient is higher, the initial decline of the partition function is faster. This is because the distribution of node degrees is relatively uniform, with each node having a sufficient number of neighbors, facilitating local information diffusion. When the assortativity coefficient is lower, the decline in the tail of the partition function is faster. This is because a few high-degree nodes connect to many low-degree nodes, acting as bridges and increasing the speed of global information diffusion.

Finally, in subplot e, we studied the impact of homogeneity/heterogeneity on the partition function. In a homogeneous network, different nodes have similar local structures(i.e. regular networks). In a heterogeneous network, the distribution of node degrees is broad, with certain nodes (referred to as central or hub nodes) having many more connections than others, leading to a diversity in the network's structure and connection patterns. The BA network is a typical example of a heterogeneous network. The homogeneity/heterogeneity of a network can be measured by various indicators, and

here we chose the Gini coefficient of degree distribution to measure the heterogeneity. In economics, the Gini coefficient is used to measure income inequality; here, we use it to measure the unevenness of the network's degree distribution. The Gini coefficient ranges from 0 to 1, where 0 represents complete equality (total homogeneity) and 1 represents maximum inequality (total heterogeneity). In this experiment, we started with a regular network and rewired it with preferential attachment mechanism. After enough rewirings, this homogeneous regular network transformed into a heterogeneous scale-free network. The partition function decreases quickly at smaller τ but slows down at larger τ. Conversely, heterogeneous networks show the opposite pattern, with a slower decline at smaller τ and a faster decrease at larger τ. This difference can be understood from the dynamics of information diffusion. when τ is small, representing local information spread, homogeneous networks, where each node has the same number of neighbors, facilitate easy local information diffusion. In contrast, in heterogeneous networks, many nodes have only a few neighbors, making local information spread more challenging. When τ become larger, the network come to the global diffusion phase, the heterogeneity of networks, due to the presence of hub nodes, accelerates global information communication, whereas homogeneous networks lack such channels for global communication, hence slowing down at this stage.

Overall, from the analysis presented in Fig. 6, we conclude that in addition to heterogeneity, other topological properties are determinants of the flow.

How the machine-learning implementation is not completely a black box: We would like to emphasize again that our novelty, in comparison to the existing RG methods, is in the conceptualization of the problem and theoretical grounds. However, we acknowledge that machine learning is a crucial part of the puzzle, being our means to reach the theoretical end.

Deep learning is often seen as a complete black box. Especially recent large language

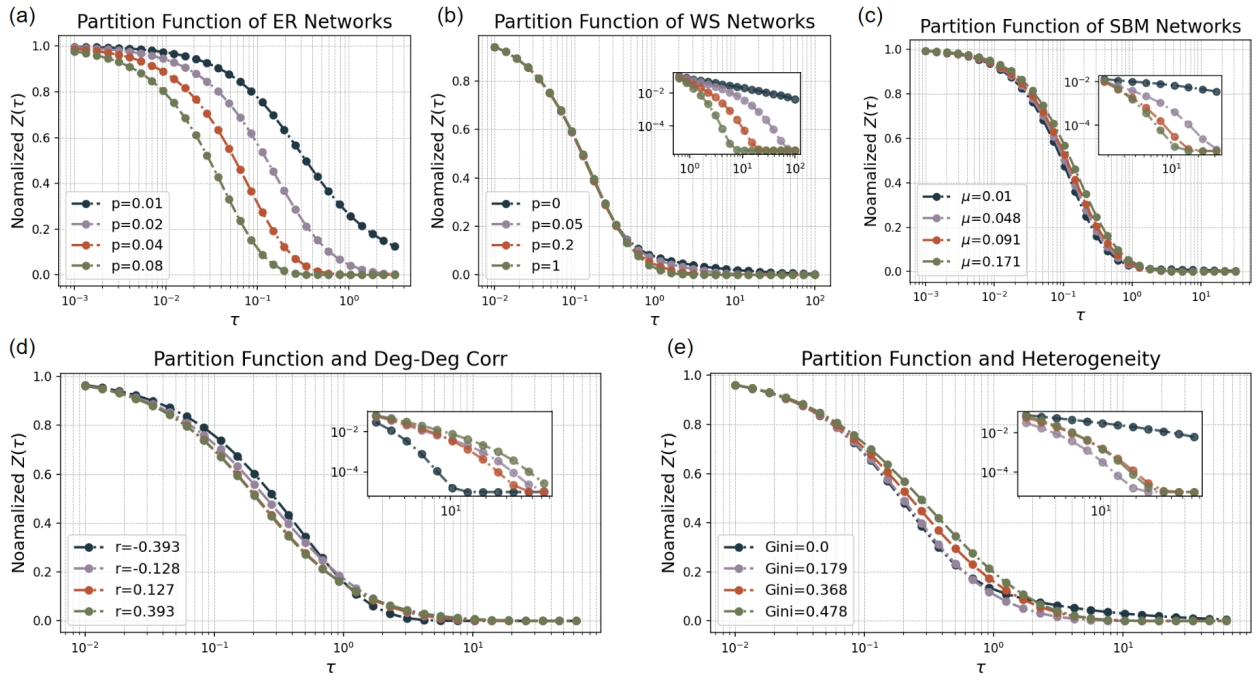

Figure 6: **Network topology and information flow:** In this figure, we show how the network's partition function responds to changes in structural properties. In the subplot a, we set different network densities by adjusting the probability of edge creation for an Erdos Renyi graph with 400 nodes. In subplot b, we observe the effect of the small-world phenomenon on the partition function by adjusting the probability of rewiring each edge in a WS graph (with 300 nodes, each having 6 neighbors). In subplot c, we examine the impact of links between community structures on the partition function by setting the mixing parameter μ for a stochastic block network (with 3 communities, each containing 50 nodes). Here, $\mu = \frac{k_{out}}{k_{in}+k_{out}}$, where k_{in} denotes the number of connections within a community, and k_{out} denotes the number of connections between communities. In subplot d, we observe the impact of the network's degree-degree correlation on the partition function, which can be measured by the assortativity coefficient r. With constant network density, we increase the network's assortativity by adding edges between nodes of similar degrees and removing edges between nodes of differing degrees. We reduce assortativity by removing edges between nodes of similar degrees and adding edges between nodes of differing degrees. In subplot e, we gradually rewire the edges of a regular network with 100 nodes, each with 4 neighbors, using the principle of preferential attachment to transition it into a scale-free network. During this process, homogeneity decreases while heterogeneity increases, and we use the Gini coefficient of the degree distribution to measure the level of heterogeneity.

models, researchers mainly focus on improving model performance by increasing the amount of parameters and data, rather than understanding how the model works. Of course, it is clear that deep learning is an efficient optimization algorithm based on gradient descent. In our case, the model is a graph neural network that groups the nodes, then aggregates them and forms the supernodes. However, the black-boxness lies in how the grouping works. We use the example of compressing a simple network, as shown in Figure 7, to illustrate how our model derives a grouping strategy.

The essence of graph neural networks is to train a learnable function that takes local structures as input and outputs groupings. Thus, our model has a natural property: nodes with similar local structures will likely be grouped together, turning out to be actually beneficial for preserving the flow. This can be intuitively understood through the analysis of several typical topological properties.

High Clustering Coefficient Nodes: Nodes with a high clustering coefficient often have neighbors that are interconnected, acting to trap information locally during the diffusion of information in the network. Grouping nodes with a high clustering coefficient together, their numerous interconnections will become high-weight self-loops on the supernodes, similarly playing a role in trapping information locally, as shown in Fig. 7, subplot a.

Hubs: Hubs act as bridges for global information diffusion within the network. By clustering hub nodes together, the edges between hub nodes and other nodes transform into connections between supernodes in the compressed network, still ensuring the smooth passage of global information. Suppose we group a hub node and its neighbors into one; then, their interconnections become self-loops of that supernode, acting as a barrier to information transmission, as shown in Fig. 7, subplot b.

Community membership: Networks with distinct community structures experience slow global information diffusion (due to few inter-community edges). Grouping nodes of

the same community together transforms the numerous intra-community edges into self-loops on the supernode, trapping information locally. Meanwhile, the few inter-community edges become low-weight edges between supernodes, similarly serving to slow down the speed of global information diffusion as shown in Fig. 7, subplot c. From these three examples, we can understand why 'grouping nodes with similar local structural features together' is reasonable. This approach preserves the functional role of the structure in information diffusion while reducing structural redundancy

However, based on the principle that "nodes with similar local structures are grouped together," the graph neural network still needs to optimize the specific grouping strategy. This is mainly because, on one hand, the number of groups is often far less than the number of types of nodes with different local structures (imagine a network with 3 different types of nodes, whereas we need to compress it into a macro network with 2 nodes). On the other hand, our model produces continuous grouping results, for example, a certain type of node might enter group A with an 80% ratio and group B with a 20% ratio. Each different ratio represents a new grouping strategy, and we need to find the most effective one from infinitely many continuous grouping strategies, which is an optimization problem, hence requiring the aid of a machine learning model. Subplots $d - f$ of Fig. 7 show discrete grouping strategies and the learned continuous grouping strategy for compressing a network with 5 types of nodes into a macroscopic network with 4 supernodes.

In subplot d of Figure 7, we know that since there are 5 types of nodes with different local structural features. When we want to compress them into 4 groups, if each node clearly belongs to a certain group and every group must have at least one node, we will produce C_5^2 different grouping strategies. We call these strategies "hard grouping." However, if we allow a type of node to belong to multiple groups at the same time (for example, belonging to group A with a proportion of 0.3 and to group B with a proportion of 0.7), then there are infinitely many grouping strategies. We call these strategies "soft

grouping." Each grouping strategy will produce a macroscopic network and generate a new partition function. In subplot d of Figure 7, we show the macroscopic networks and corresponding partition functions produced by two hard grouping strategies. In subplots e and f, we respectively show the grouping matrix of the soft grouping strategy learned by our model and the macroscopic network structure. In subplot g, we present the differences between the partition functions of all reduced networks(including the networks generated by all hard grouping strategies and our learned soft grouping strategy) and the original network's partition function. We find that the soft grouping strategy, due to its stronger expressive power, can produce a macroscopic network with a more similar partition function. This type of grouping strategy is not manually designed but optimized through the gradient descent method. By taking gradients of the GNN's internal parameters, our model automatically discovered the grouping strategy shown in subplot e.

In summary, although the specific details of our neural network are not completely visible, we theoretically argue and numerically validate it works based on the basic idea that "nodes with similar local structures enter the same group."

Conclusion. Overall, given the considerable difference in the theoretical grounds and implementation strategy discussed here and in the previous section, we are confident that our approach can not be classified as a special case of any of the previous approaches we know of. While LRG uses the advanced statistical physics of networks, it uses it in a clearly distinct way. We thank the referee once again for helping us clarify the novelty of our model better.

Pag. 15 The authors say: "A definitive framework for network renormalization that reduces the size of large network datasets while preserving their fundamental properties is still missing." This statement is incorrect and requires to be rephrased.

We thank the reviewers for the comment. We agree with the reviewers' viewpoint

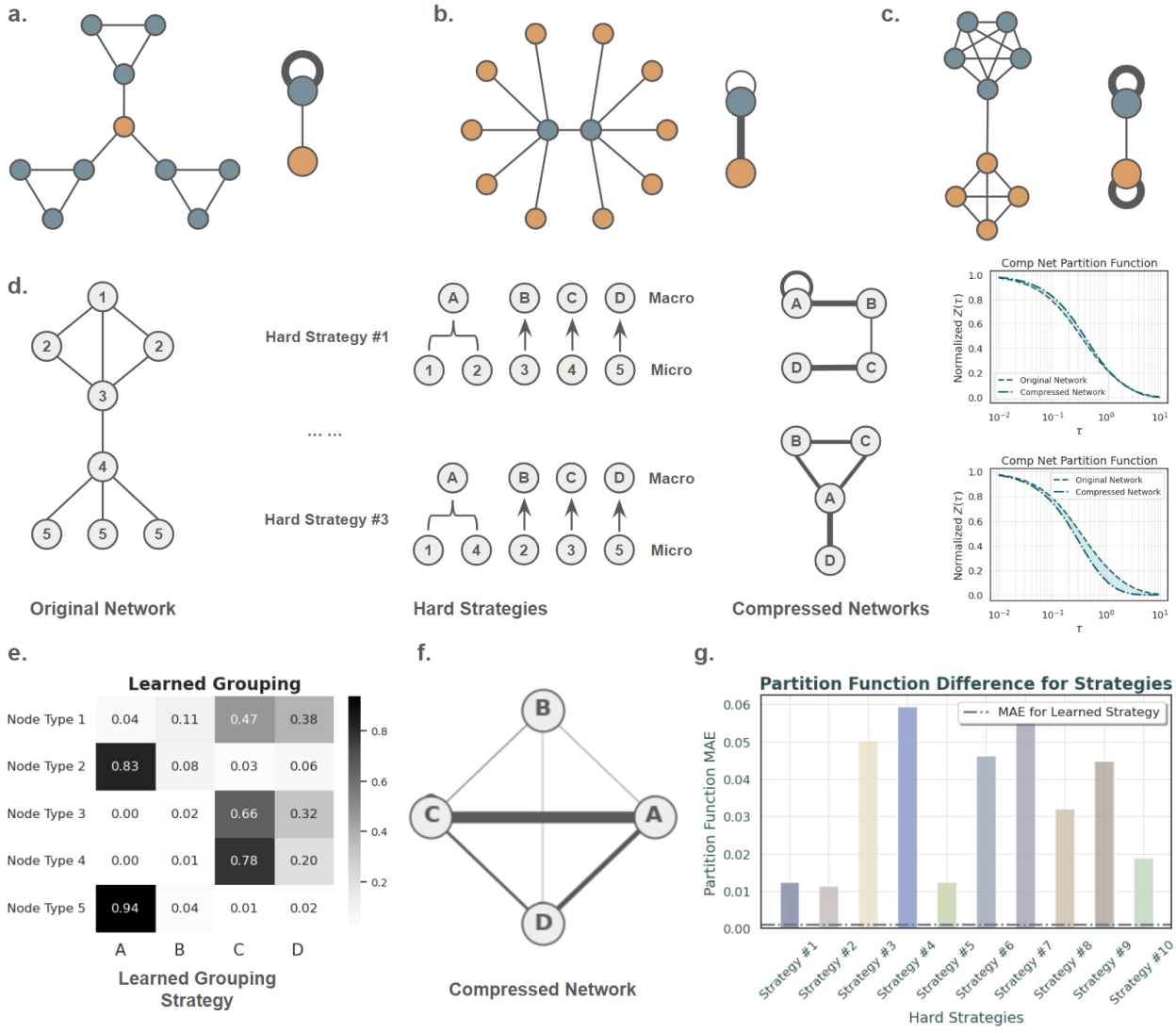

Figure 7: **Grouping strategies and the structure of the macroscopic network:** Subplots $a-c$: Compressing nodes with similar local structures into a supernode reduces structural redundancy and preserves the role of local structures in information diffusion. Nodes with high clustering coefficients and community structures tend to trap information locally, while hub nodes act as global bridges. Subplot d: we show the macroscopic network after compression using different hard(discrete) grouping strategies for a toy network with 8 nodes and 5 types of different local structures, and the changes in the corresponding partition function. Subplot e: the soft(continuous) grouping strategy learned by our model. In subplot f, we present the macroscopic network structure given by the grouping strategy from subplot e. Subplot g: comparison of the normalized partition function error before and after compression under different grouping strategies.

that all works on complex network renormalization try to preserve certain important properties of the networks. However, our method is the first to preserve all macroscopic properties of information flow simultaneously across scales. Additionally, our method does not rely on specific assumptions about the topological structure. Our experiments demonstrate that it works well under various topological features (such as heterogeneity, community structure, small-world properties, etc.). Therefore, to more appropriately express this, we have revised the text to say, 'a method that can accurately preserve the full properties of the network information flow across different structural properties is still missing.'.

I suggest to rewrite the title to mention the use of machine learning for coarse-graining analysis explicitly.

We thank the reviewer for this comment. We acknowledge that employing machine learning is an important part of our work. Therefore, we follow the referee's comment and change the title of the paper to "Network Information Dynamics Renormalization Group Combines Statistical Physics and Machine Learning."

References: The authors have overlooked important references in the network RG problem. In my opinion, the manuscripts of (i) Gfeller, D. & De Los Rios Phys. Rev. Lett. 99, 038701 (2007), (ii) Goh, K., et al., Phys. Rev. Lett. 96, 018701 (2006), (iii) Radicchi, F., et al. Phys. Rev. Lett. 101, 148701 (2008). or (iv) Rozenfeld, H. et al. Phys. Rev. Lett. 104, 025701 (2010), and (v) Boguña, M., et al. Nat. Rev. Phys. 3.2 (2021): 114-135 must be appropriately cited.

we thank the reviewer for this suggestion. The literature has helped us better outline the development of the field of network renormalization. These papers have been appropriately cited in the third paragraph of the main text.

Pag. 6. The authors define the network free and internal energy, but they no longer

use these properties. This can be confusing for the readers. If the authors want to provide a full perspective on the canonical statistical physics description for complex networks, they would also cite the manuscript of Villegas et al. Phys. Rev. Res. 4.3 (2022): 033196.

We thank the reviewer for this suggestion. Indeed, we fully agree that free energy hasn't been directly used in our calculations. However, we think keeping the derivation of free energy from partition function is crucial for our work. This is because our theoretical basis is on the fact that all macroscopic measures of information flow— like diversity of pathways and dynamical trapping quantified in terms of entropy and energy— can be derived from the partition function. In other words, partition function is a reach mathematical object containing all relevant macroscopic information. Therefore, an RG that replicates the partition function preserves those features of information flow.

Therefore, while we fully agree with the referee that we don't directly used free energy, we think keeping the derivation showing that free energy can be derived from the partition function profile is an important part of our narrative.

We also thank the reviewer for the comment about the citations. We have cited the papers that originally introduced the network density matrix formalism in its current form [6], generalized it [4], and worked out the canonical statistical description for complex networks [3]. However, we would certainly add the reference mentioned by the reviewer to include a study of specific heat and information cores complementing our work.

Pag.17: ”cross-entorpy” must be ”cross-entropy” Pag. 16: ”synthetic network” must be ” synthetic networks” Pag.11: ”50 node” would be ”50-node”

We thank the reviewer for the meticulous detection of our errors. We have made one-to-one corrections in the main text.

end of response to referee #1

Referee #2 (Remarks to the Author):

First and foremost, we sincerely appreciate the referee's interest in and encouraging view of our work. Also, we thank the referee for the insightful comments that certainly improved the representation and depth of our manuscript. To facilitate a quick overview of our responses to each point raised, we have prepared this summary at the beginning of our letter. The main 3 response are list below:

First, we addressed the reviewer's concerns regarding network connectivity. A detail in our method explains how the connectivity of the macroscopic network is preserved; each edge of the original network becomes either a part of a self-loop or a weighted edge in the macroscopic network through compression. Additionally, connectivity significantly influences the behavior of information diffusion. Thus, our method, functioning as an optimization algorithm, continuously refines the partitioning matrix to ensure appropriate levels of connectivity.

Also, our previous manuscript lacked a formal description of the decimation process, its properties, and how it preserves the partition function. Here, we describe how our model operates from the perspectives of input and output of machine learning models, the network compression process, and optimization objectives. Additionally, we explain the rationale behind the decimation strategy optimized by the machine.

Finally, we better explain how our systems are out of equilibrium and why maximum entropy principles are not required. This is clarified by analyzing the properties of information flow in the network when it is subjected to external disturbances.

We hope that this summary provides a clear overview of our responses. Once again, we thank the referee for the valuable comments, which have significantly improved the quality of our manuscript and addressed the key issues of the previous version. Please note that detailed responses to each of the referee's points can be found in the subsequent

sections of this letter. We look forward to the referee's further guidance and feedback.

I read with interest the paper Network Information Dynamics Renormalization Group by Zhang et al. I liked very much the approach outlined by the authors trying to figure out a simple method that might consider both the structural properties of the complex networks and the dynamic processes that are acting on them. I think the paper might be suitable for your journal once the authors clarify some critical points.

We thank the reviewer for the interest in and encouraging view of our work. Also, we thank the referee for the insightful comments that certainly improved the representation and depth of our manuscript.

If I understood clearly the Network Information Dynamics Renormalization Group introduced here, this is still based on the decimation part of RG, but the coarsening is not driven by the lattice/network structure but instead is driven by the role played in dynamics (hubs with hubs and leaves with leaves).

My first question is how to ensure network connectedness with such a decimation procedure.

We thank the reviewer for the insightful comment. We acknowledge that we lacked a clear explanation of the coarse-graining in the previous version of the manuscript. Following the referee's comment, we add here and also to the revised version of our work.

Firstly, we confirm the referee's suggestion that our model does not work with a predefined structural merging rule. We also agree that our method works with the functional roles nodes play in the flow.

In fact, our model determines the grouping of nodes based on their local structure—including degree, clustering coefficient, etc. The model determines how to combine these features to reach an optimal grouping that preserves the flow. We then study the model

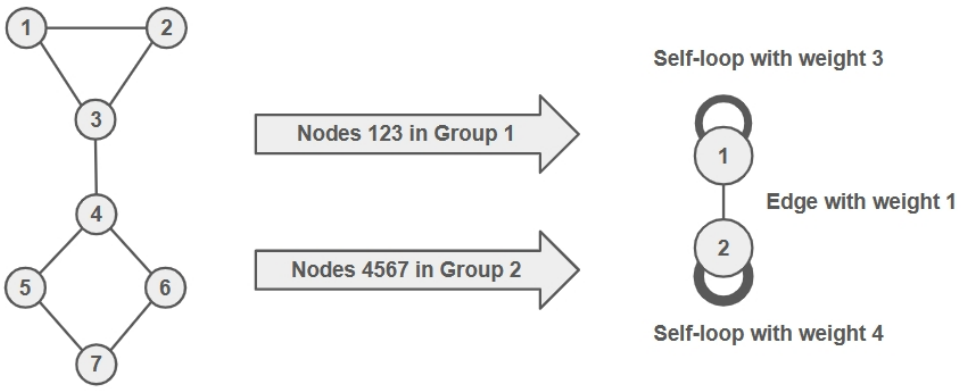

Figure 1: A schematic diagram shows how our method compresses the network given the grouping information.

and show that after learning, the model often groups nodes based on structural similarity.

Regarding connectedness, we provide a schematic of the compression in Fig. 1. In our approach, edges within the same group are compressed into weighted self-loops of the supernode, while edges between different groups are compressed into weighted edges between supernodes. Every edge in the original network will be preserved in the form of self-loops or connections between supernodes. Therefore, if the original network is connected, the compressed network will necessarily be connected as well.

Of course, the weighted nature of our compression can make the connectedness problem more intricate. An increase in the number of edges between the supernodes will enhance the compression's connectivity, while an increase in the self-loops retains information within the supernodes. Therefore, the coarse-graining strategy determines the degree of connectivity in the network. Here, our approach provides a reasonable grouping strategy: to preserve the flow, our model has to find a suitable grouping strategy among many that preserves connectivity to some extent— please see Fig. 2.

We thank the reviewer once again for this comment and change and extend the manuscript accordingly to explain the decimation procedure better.

Figure 2: A schematic diagram shows how our model ensures reasonable connectedness by rewarding and punishing different strategies. In this schematic, Strategy 1 receives a reward during the optimization process because it generates a macroscopic network with a partition function more similar to that of the original network through reasonable grouping. Conversely, Strategy 2 receives a penalty. This is specifically reflected by adjusting the parameters related to the grouping strategy within the graph network through gradient descent.

Furthermore, I assume that the scaling part of RG must also reflect this particular choice of decimation procedure. It would be nice (I acknowledge that it could be challenging and take a lot of time to make a formal comparison) if the authors could comment on this point when describing the modification induced in the partition function,

We appreciate the reviewer's attention to our decimation procedure. Indeed, we need to systematically explain how our method performs the decimation.

Initially, our goal is to compress the network structure while preserving the dynamic behavoir of information diffusion, which can be described by the network's partition function. The partition function corresponds to a clear physical meaning: under the influence of diffusion dynamics, it represents the amount of information retained at the original nodes over time.

Our basic approach to network reduction involves grouping existing nodes using graph network methods, where many original nodes in a group are reduced to a sin-

gle supernode. Edges within the original network's groups become weighted self-loops on the supernodes, and edges between different groups become weighted edges between supernodes. The grouping strategy is the core of our decimation procedure. To achieve a reasonable grouping strategy, we use the similarity of the normalized partition functions of the networks before and after compression as an optimization goal to adjust the parameters of the graph neural network, thus obtaining a sensible grouping strategy.

The experimental results show that our method performs well on various synthetic and real networks: our method successfully compresses the network size while preserving the partition function and maintaining the distribution of eigenvalues of the original network's Laplacian matrix. This indicates that our method can preserve a wide range of dynamics after performing the decimation.

Furthermore, the analysis of the model's interpretability shows that the grouping strategy provided by our model differs from existing renormalization methods, which often compress a local subnetwork into a single supernode. Instead, our method chooses to compress nodes with similar local structures into a supernode, regardless of whether these nodes are interconnected. We find this to be a sensible strategy. For example, nodes with high clustering coefficients are often closely connected to their neighbors, acting to trap information locally. Compressing them into a supernode turns their many interconnections into self-loops on the supernode, similarly trapping information locally. Hub nodes act as global bridges for information diffusion; compressing hub nodes into a supernode transforms their connections to other nodes into inter-supernode connections, similarly facilitating global information spread. Overall, our method preserves the functional role of the structure in information diffusion while reducing structural redundancy. Additionally, as a machine learning approach, it continuously optimizes the specific grouping strategy based on the principle that similar nodes enter similar groups. Each adjustment reflects structural changes in the weighted macro-network, allowing for

fine control of the macro-network's information flow to remain consistent with the original network. This demonstrates that although our method is based on machine learning, it is not a complete black box, but rather it learns an intuitively understandable grouping strategy, thereby efficiently performing the decimation procedure.

Finally, could the authors comment if the system is considered at equilibrium and why? This is the only case in which a proper partition function can be defined,

We thank the reviewer for this brilliant comment. Here, we take this chance to explain our way of doing statistical physics of information dynamics better.

As the referee mentioned, complex systems are open and far from equilibrium, and we lack an adequate theoretical framework to explain and predict their states and transitions. One step towards their characterization is taken by the development of network density matrices.

Let G be the structure of a complex system, modeled as a network of N nodes and $|E|$ connections, respectively. The links between network units are often encoded into an adjacency matrix $\mathbf{A}$, where $A_{ij} = 1$ if nodes i and j are connected, and $A_{ij} = 0$ otherwise. In the following we will generally refer to information exchange to characterize any type of signal that can be used for unit-unit communication from short- to long-range scales.

A variety of dynamical processes has been used to model the flow of information between the nodes. Yet, one of the simplest and most versatile dynamical processes that can be used to proxy the flow of information through the networks is diffusion governed by the graph Laplacian matrix. Of course, the diffusion equation and its solution can be written as:

$$\partial_\tau \psi_\tau^t = -\mathbf{L}\psi_\tau^t,$$
$$\psi_\tau^t = e^{-\tau\mathbf{L}}\psi_0^t, \tag{1}$$

where ψ_τ, similar to its transpose ψ_τ^t, is the concentration vector, describing the distribution of a quantity over nodes and $e^{-\tau\mathbf{L}}$ gives the time-evolution operator. For instance, let $v_i = (0, 0, ..., 1, 0, ..., 0)$ be a canonical vector in which all elements are zero except the i-th one, indicating node i. In this case $v_j \left(e^{-\tau\mathbf{L}}\right) v_i^t$ proxies the flow from node i to j.

Note that here, τ determines the propagation scale. If τ is small, signals travel mostly between neighboring nodes, whereas large values of τ enable long-range communications. More technically, one can expand the time-evolution operator as

$$e^{-\tau\mathbf{L}} = \sum_{n=0}^{\infty} \frac{(-\tau)^n}{n!}\mathbf{L}^n. \tag{2}$$

For instance, the neighbors' communication term $-\tau\mathbf{L}$ activates where $\tau \gg \epsilon$, where ϵ is considered to be an extremely small non-negative number. This is because $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is a function of the adjacency matrix encoding the neighbors' connections. Whereas the second neighbors start to communicate through pathways of length two given by $\frac{\tau^2}{2}\mathbf{L}^2$, where $\frac{\tau^2}{2} \gg \epsilon$. Similarly, this is because $\mathbf{L}^2 = (\mathbf{D} - \mathbf{A})^2$ includes the second power of the adjacency matrix counting the pathways of length two. A generalization is valid for longer pathways activating as τ takes larger values.

Another important mathematical object is the statistical propagator, which encodes the system's response to stochastic perturbations. Assume each of the N nodes can be perturbed with probability $1/N$. If node i is perturbed, the flow is given by $e^{-\tau\mathbf{L}}v_i^t$ and its outer product with itself, gives the local propagation $e^{-\tau\mathbf{L}}v_i^t v_i e^{-\tau\mathbf{L}^t}$ encoding the cor-

relations between any pairs of nodes in receiving signals from node i as off-diagonal and distribution of signal energy among the nodes as diagonal elements. The statistical propagator is a superposition of the local propagators:

$$\mathcal{U}_{2\tau} = \frac{1}{N}\sum_{i=1}^{N} e^{-\tau\mathbf{L}} v_i^t v_i e^{-\tau\mathbf{L}^t} = \frac{e^{-2\tau\mathbf{L}}}{N}, \tag{3}$$

since $\sum_{i=1}^{N} v_i^t v_i = 1$ gives the unit matrix, and in the case of diffusion dynamics on undirected networks, we have $\mathbf{L} = \mathbf{L}^t$. Interestingly, because of this property, the statistical operator equals the time-evolution operator after a proper rescaling ($2\tau \rightarrow \tau$), providing equivalent interpretations of diffusion-based density matrices in terms of flow and node-node correlations. It is worth mentioning that while the perturbation probability of nodes is uniform ($1/N$) with the assumption of maximum lack of information, other distributions can be used to build the statistical propagator.

While the trace of the statistical propagator is the total signal energy in the system, it also provides a suitable partition function $Z = \text{Tr}\,(\mathcal{U}_\tau)$ counting the units of signal energy and enabling the definition of a network density matrix:

$$\boldsymbol{\rho}_\tau = \frac{\mathcal{U}_\tau}{Z} \tag{4}$$

Note that in this formalism, one fixes τ and inspects the signal propagation at a specific scale. Therefore, τ is more of a topological scale, though proxied by a temporal one.

Similar to the statistical propagator, the density matrix encodes the response to perturbations but in a probabilistic way— i.e., the diagonal elements encode the probability

that a unit of signal energy is on top of the nodes.

Therefore, to derive the network density matrices and their partition functions, there is no need for a maximum entropy principle or equilibrium assumptions. However, interestingly, the resulting density matrix has a Gibbsian form, suggesting connections with equilibrium statistical physics [7].

We thank the reviewer once again for raising this important point.

The work is original, and claims can be supported with a revision. The methodology is sound, and details are provided. I therefore ask to revise the paper accordingly.

end of response to referee #2

Referee #3 (Remarks to the Author):

We sincerely thank the referee for the in-depth analysis and valuable comments on our manuscript. The referee's insights have been instrumental over the past few months as we designed new experiments and conducted more comprehensive analyses to address each comment meticulously. However, given the content of our responses, we would provide a quick overview for easier navigation here. To this end, we have summarized our responses into three key points, as detailed below:

First of all, we thank the referee for pointing out an important issue in our work: the lack of a clear explanation of the similarity and difference between our method and Laplacian Renormalization Group(LRG). In the following sections, the referee will see our response from four different perspectives: theoretical foundation, application scope, method of forming supernodes, and structural properties of macro networks. We also added a new section in the main text to discuss the difference between our method and LRG, both theoretically and numerically.

Also, as the referee pointed out, we previously lacked experimental proof that our model could be applied to very large networks. Here, we have conducted experiments and reported the results. We trained our model on small network datasets (less than 1,000 nodes), validated its renormalization effects on medium-sized networks (1,000 to 10,000 nodes), and applied it to networks with 100,000 nodes. These experiments demonstrate the model's capability to compress networks of substantial size. Following the referee's comment, we designed experiments to renormalize real networks from various fields. We selected 25 typical real networks across seven fields for compression. The results demonstrate the model's ability to renormalize networks from different domains. Additionally, we found that despite the diversity in the fields of the network data, the transitivity of network structures affects the model's performance—the stronger the transitivity, the more challenging it is to preserve the complete partition function during compression. We be-

lieve this is because high transitivity creates more complex patterns in the local diffusion of information.

Finally, as the referee correctly pointed out, in the previous version of our paper, our attempt to open the black box was merely an analysis of the sensitivity of input features to the neural network. Here, we try to understand how the model works from a new perspective. We first discovered that the model compresses the network based on the principle that "nodes with similar local structures are grouped together." Further analysis proves the validity of this principle, as it reduces structural redundancy while preserving the function of local structures in the process of information diffusion. Additionally, based on this principle, our model, as an optimization algorithm, continuously fine-tunes the connectivity weights of the macroscopic network, thereby precisely controlling the information flow in the macroscopic network to maintain consistency with the microscopic network.

We hope that this summary provides a quick overview for the referee. We especially appreciate the referee's in-depth and meticulous analysis and comments on our paper, which guided us to conduct more experiments and analyses, improving the quality of this manuscript. In the detailed sections that follow, we provide one-on-one detailed responses to other comments, such as the impact of network structural properties on information flow behavior, whether the ten points chosen in the experiments sufficiently describe the partition function curve, and a detailed analysis of time complexity, among others.

The authors use a graph neural network approach to introduce a renormalization scheme that preserves global information dynamics in complex networks. Information dynamics is modeled as a diffusion process ruled by the Laplacian of the network. A multilayer perceptron is utilized to minimize the mean absolute error between the normalized partition function of the compressed network and the original one. In this man-

ner, macroscopic magnitudes derived from the partition function, such as the Von Neumann entropy and the free energy, are expected to be preserved in the renormalization flow.

Thank the referee for succinctly and clearly summarizing the main logic and core contributions of our work.

While the topic is certainly relevant, and the idea of preserving the partition function is appealing, I have major concerns. The main contribution of the paper appears to be portrayed as a mere technical improvement. In other words, the novelty and significance of the contribution is not clearly conveyed, and many claims lack supporting evidence. Additionally, the manuscript contains inaccuracies. I would not recommend accepting the manuscript in its present form. A new version may be reconsidered once the ideas introduced in the manuscript are thoroughly addressed, their significance and novelty clarified, and the inaccuracies corrected. Other aspects, such as the tendency to overstate, also need improvement.

The research in the manuscript is very close to the Laplacian renormalization method recently introduced in Ref. 48. It is fundamental to clearly and convincingly explain how the introduced method differs from Laplacian renormalization and advances the results obtained in that reference.

We thank the reviewer for pointing out this issue. Both our paper and Laplacian Renormalization Group(LRG) propose network renormalization methods are based on recent advances in statistical physics work on networks. However, the two methods differ significantly in their theoretical foundations, scope of application, underlying mechanisms, and performance in preserving information flow. We will explain these differences one by one in the following paragraphs.

The LRG method was among the first to introduce Kadanoff's decimation based on

diffusion dynamics on networks, which is the core of the renormalization process. In different systems, the ways to simplify the system can vary significantly. On a uniform space or lattice, due to the uniformity and translational invariance of space, Kadanoff's decimation can be easily implemented. However, complex networks do not exist in a uniform space. To overcome this problem, the LRG method proposed a way to create Kadanoff supernodes using diffusion dynamics on the network. It has successfully pushed the boundaries of RG to include structural heterogeneity.

However, our method and LRG are different in theoretical foundations: while LRG focuses on the specific heat at the specific scale of phase transition to define their information cores, we focus on replicating the full partition function profile at all scales of propagation, based on the fact that the partition function contains the full information of the macroscopic descriptors of the flow. Both LRG and our method can renormalize networks at any scale. However, LRG compresses network size by discarding the fast modes of diffusion processes (corresponding to eigenvalues greater than $\frac{1}{\tau}$), while our method tries to preserve the complete distribution of eigenvalues at any scale, thereby maintaining the full behavior of the partition function (see Fig. 3 for a comparison of the two methods in terms of their ability to preserve eigenvalues).

The spectrum coupled with the temporal scale of signal propagation, describes the flow. More technically, metrics built on top of the spectrum, like the network Von Neumann entropy and free energy, encode macroscopic properties like how diverse the flow pathways are and how fast the flow is [1]. Fortunately, and similarly to the statistical physics of particles, the network partition function contains full information of such metrics— i.e., they can be derived from the full partition function profile. That is why compressions preserving the partition function profile, simultaneously preserve the macroscopic descriptors of information flow across scales— the full derivation is in the text.

LRG performs well on heterogeneous networks, especially on BA networks and scale-

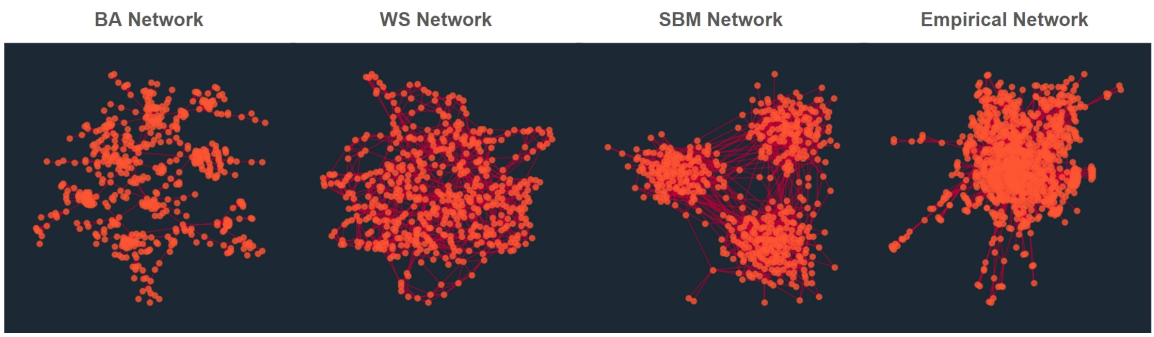

Figure 1: **A schematic diagram showing three typical synthetic networks and one empirical network:** They are a BA network(N=500, m=1), a WS network(N=500, k=3, p=0.1), a SBM network(N=500, mixing parameter $\mu = \frac{k_{in}}{k_{in}+k_{out}} = 0.067$, in which k_{in} denotes the number of edges inside the communities and k_{out} denotes the number of edges between communities, and an empirical network data from the field of biology.

free trees. However, the topology of real-world networks is not limited to what the BA model can generate. The two decades of network science has revealed that real-world networks also exhibit important features such as segregation while integration and order while disorder— i.e., modules, motifs, etc. That is why the RG methods must eventually go beyond the current model specific limiting assumptions.

As we show here, our method clearly outperforms LRG across synthetic and real-world networks. Remarkably, even in the case of the BA model, the richer expressive capacity of our model, like allowing for weighted compressions, leads to better flow preservation than LRG. As examples, please see our analysis of a BA (N=500, m=1), Watts-Strogatz (N=500, k=3, p=0.1), an SBM (N=500, mixing parameter μ=0.067), and a real biological network data: a visualization of the networks in the Fig. 1, and the comparitive results in terms of partition functions Fig. 2 and Laplacian spectrum Fig. 3.

In Figure 2, we show the partition functions of the compressed networks obtained at different compression ratios when applying our method and the LRG method to these 4 networks, respectively. We also present a subplot showing the relationship between the differences in the partition functions before and after renormalization and the com-

pression size. We chose to compress the networks to 50%, 25%, 12.5%, and 6.25% of the original network size. From Figure 2, we can observe that among all the network data, the LRG method performs best on BA networks, where the compressed network retains the partition function to the greatest extent, consistent with the effects claimed in their paper. However, when we move to other network structures, the performance of the LRG method starts to decline, with a larger change in the partition functions of the networks before and after compression. In contrast, our method ensures that the compressed network has a partition function similar to that of the original network in any type of network, and the degree of similarity is higher under any dataset(even including BA network) compared to the LRG method.

In Fig. 3, we display the distribution of eigenvalues of the Laplacian matrix for the original network and the networks compressed by the netIDRG and LRG methods. Since the network's partition function is defiened by $Z(\tau) = Tr(e^{-\tau L}) = \sum_{l=1}^{N} e^{-\tau\lambda_l}$, where λ_l denotes the Lth smallest eigen value of the laplacian matrix, all the eigenvalues are non-negative, $\lambda_l \geq 0$, and at least one of the eigenvalues is $\lambda_1 = 0$, the smaller eigenvalues will determine the network's diffusion behavior. For some networks, to highlight the differences in distributions, we only show the first 80% of the sorted eigenvalues. The figure shows that our method can more accurately maintain the consistency of the eigenvalue distribution while compressing the network size, thereby ensuring the consistency of information flow behavior.

Additionally, LRG and netIDRG fundamentally differ in their basic mechanisms, that is, principles for selecting supernodes. LRG selects a subnetwork of the original network to form a supernode based on the results of the diffusion process, and this strategy is discrete, meaning each node can only be part of one supernode. In contrast, netIDRG selects nodes with similar local structures to form a supernode, and this strategy is continuous; for example, node i can enter supernode A with an 80% proportion and supernode B with

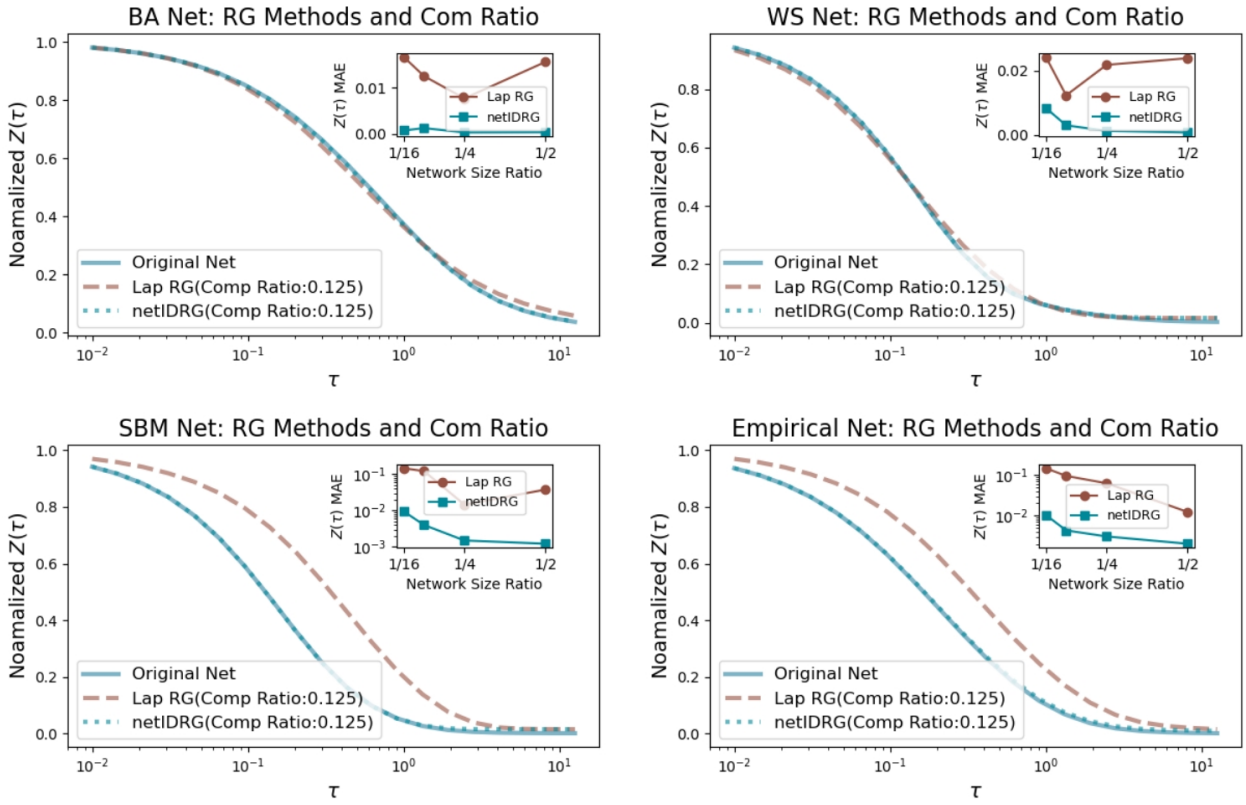

Figure 2: **Comparison for netIDRG method and the LRG method across different networks:** On each subplot, we plotted the partition function curve of the original network and the curves when the network is compressed to 12.5% of its original size by two methods. In the insets of each subplot, we show the mean absolute error of the normalized partition function of the original network after being compressed to 50%, 25%, 12.5%, and 6.25% by different methods, measured across a vector of values from 15 uniformly selected points in logarithmic space from 10^-2 to 10^1.

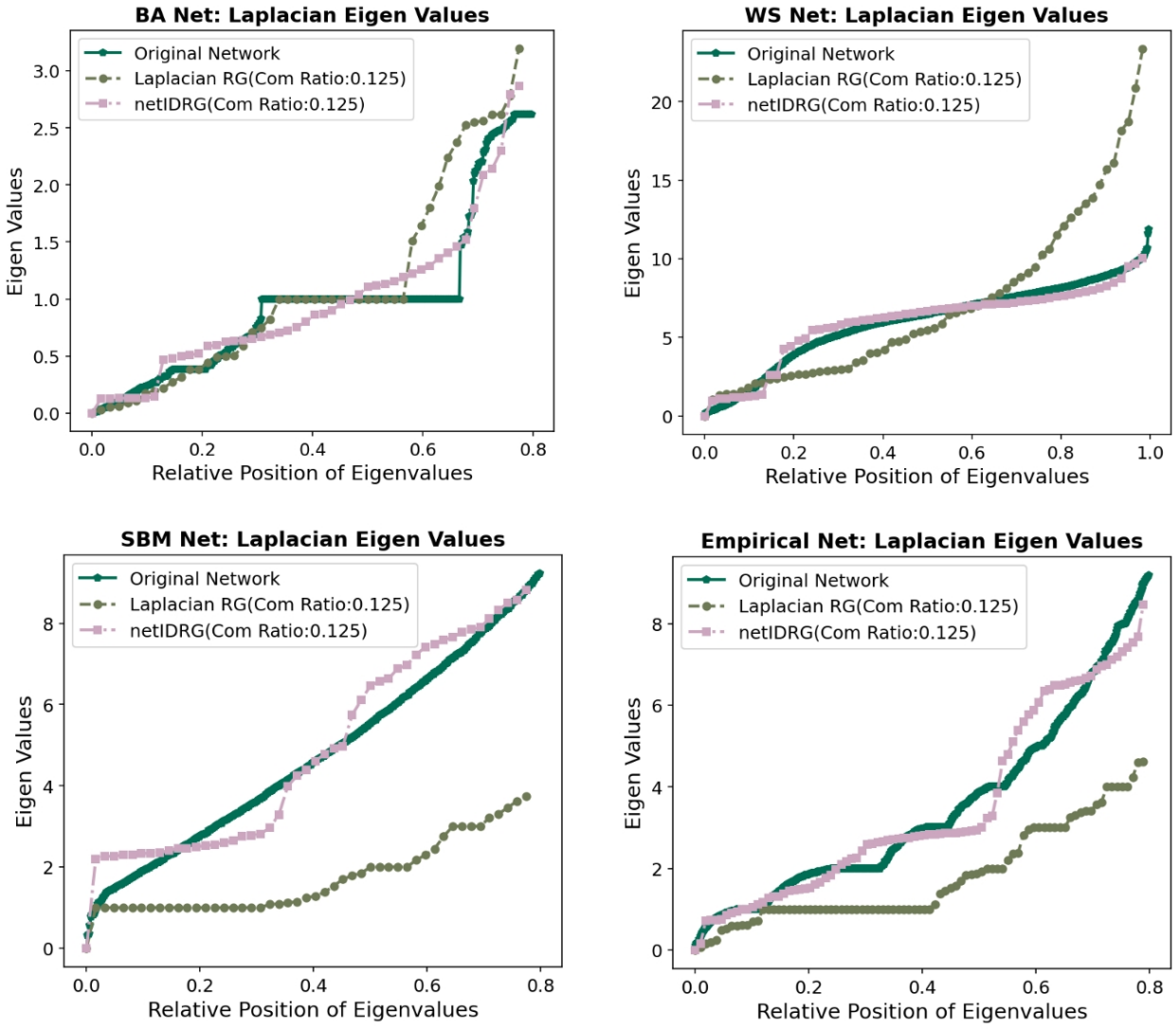

Figure 3: **Distribution of Laplacian eigenvalues:** We present the distribution of the Laplacian matrix's eigenvalues of networks, sorted from smallest to largest, on three synthetic networks and one empirical network. This includes the original network, networks compressed by the Laplacian RG method, and networks compressed by the netIDRG method. Since the partition function is primarily determined by the smaller eigenvalues, we only display the smallest 80% of the eigenvalues for some networks.

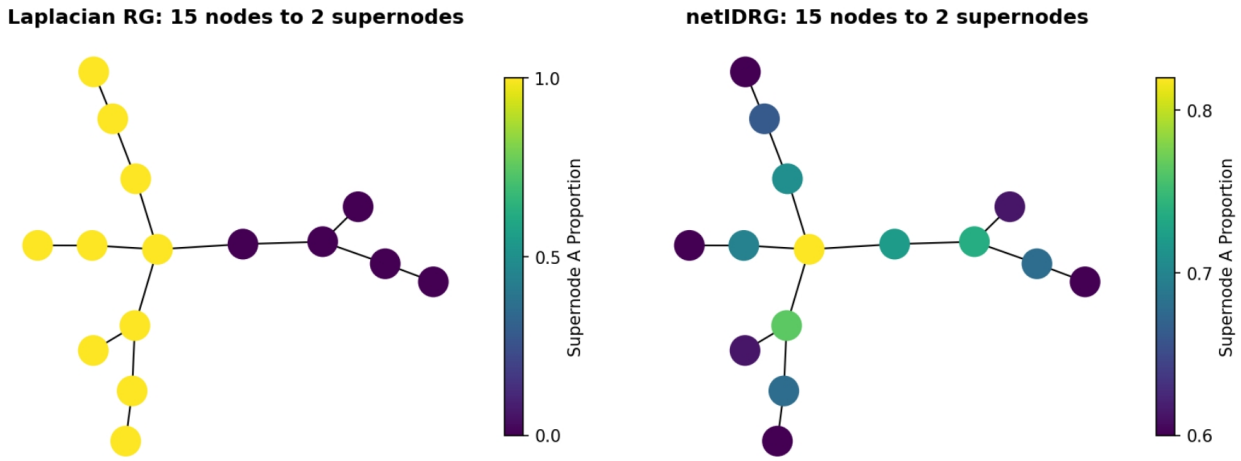

Figure 4: **Differences in supernode composition between methods Laplacian RG and netIDRG:** In this figure, we take a BA network with 15 nodes and renormalize it to two nodes using methods Laplacian RG and netIDRG: supernode A and supernode B. In the left subplot, Laplacian RG groups connected subnetworks into Supernodes. In the right subplot, each node belongs to Supernode A at a certain proportion.

a 20% proportion. As shown in Fig 4, we chose a typical BA network with 15 nodes and $m = 1$, and used LRG and netIDRG to renormalize it, compressing it into a macroscopic network with two supernodes (A and B). The left subplot shows LRG's grouping strategy for nodes, where supernodes correspond to two subnetworks formed by adjacent nodes. The right subplot shows netIDRG's strategy, where nodes with similar local structures are given similar grouping results that are continuous, thus finely controlling the structure of the macroscopic network.

Moreover, an intuitive display of the structure of the compressed network can also reflect the differences between the netIDRG method and the Laplacian RG method, as shown in Fig. 5: The Laplacian RG method compresses the original network into a smaller unweighted graph without self-loops. In contrast, the netIDRG method produces a weighted graph that may include self-loops.

To resolve the ambiguities of our previous submission, we add a section to our text comparing our approach and LRG, laying out the provided arguments and attaching the

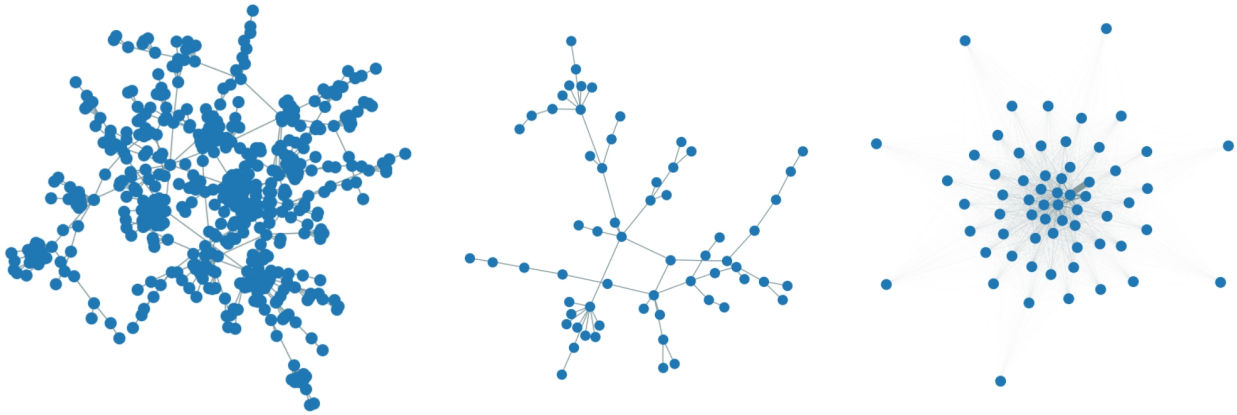

Figure 5: **Visualization of network structures before and after compression:** In the left subplot, we display a BA network with N = 500 and m = 1. In the middle subplot, we show the network structure obtained using the Laplacian RG method, while the network structure obtained with the netIDRG method is presented in the right subplot.

supplementary analyses to the supplementary materials.

We are confident that our theoretical and numerical arguments answer the concerns raised by the reviewer, as it demonstrates that our approach is theoretically and fundamentally different from LRG and that enables flow preservations that work across network models, outperforming LRG by an order of magnitude.

It is also fundamental to expand the evidence provided to support the claims. There are many unsupported claims in the manuscript. For instance:

* Abstract: "Our work offers a low-complexity renormalization method breaking the size barrier for meaningful compressions of extremely large networks". I did not see extremely large networks, say of 100000 nodes, addressed in this work. The synthetic networks under analysis have only 256 nodes, and the empirical networks have a maximum of 2500 nodes. Results of the analysis of large networks, say of about 100000 nodes, should be reported.

We thank the referee for raising this question. In our previous version of the paper,

we indeed did not sufficiently discuss the renormalization of large networks. Here, we've added insights into the challenges we might face when renormalizing large networks, analyzed the time complexity of different calculation processes, and included an experiment on renormalizing a network with 100,000 nodes.

When we train our model to renormalize a network with N nodes, it needs to compress the original network and calculate the error of the normalized partition function before and after compression as the loss to be optimized. Here, the most time-consuming part is calculating the partition function of the original network to serve as supervisory information, which has a time complexity of $O(N^3)$. Therefore, calculating the partition function for extremely large networks is almost impossible. To address this, we train the renormalization model on smaller network datasets, validate the model's generalizability on larger network datasets, and apply the model to the renormalization of extremely large networks (100,000 nodes). The time required to calculate the partition function for networks of different sizes and the sizes of the network data used for training and testing our model are shown in Figure 6.

In the left subplot of Figure 6, we show the time required to calculate the partition function for networks of various sizes. Since the calculation of the partition function involves eigen-decomposition of the Laplacian matrix, its time complexity is $O(N^3)$. By performing linear fitting in log-log scale and extrapolating, we estimate that the time required to calculate the partition function for a network with 100K nodes exceeds 10^4 seconds, making repeated experiments at this scale very impractical, not to mention for larger networks. Therefore, we adopted the strategy of training on smaller networks and applying the trained model to larger networks. In the left subplot of Figure 1, we can see that we train our model on networks of sizes 10^2 to 10^3. To assess the model's generalization ability, we validate the model's performance on networks sized 10^3 to 10^4. If the validation error is within a reasonable range, we can consider the model capable

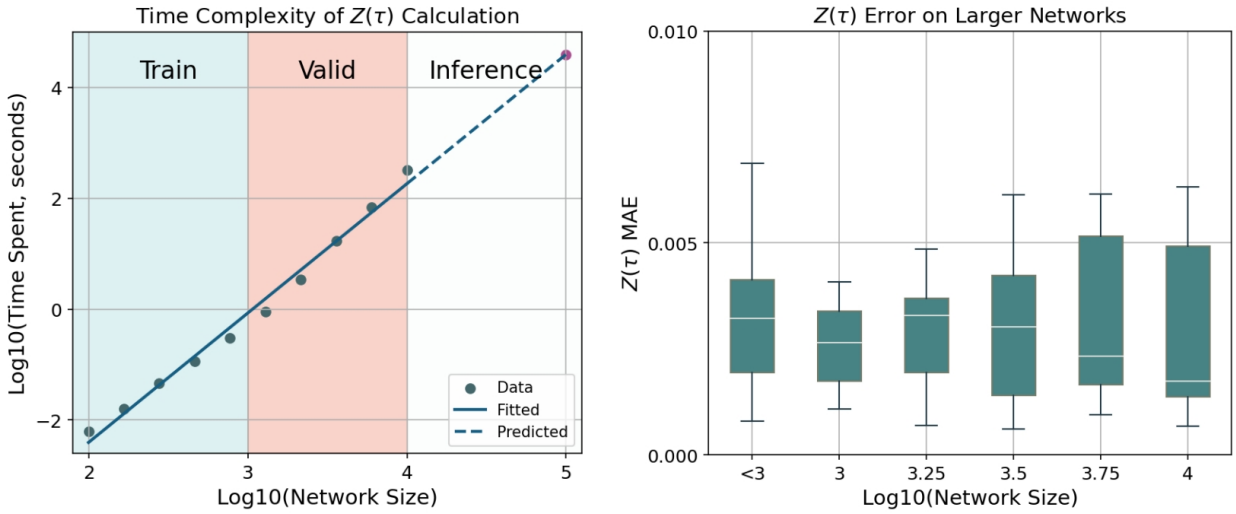

Figure 6: **Time complexity of $Z(\tau)$ calculation and Generalization Results on Larger Network:**In the left subplot, we present the relationship between network size in logarithmic space and the time consumed for calculating the partition function, and based on this relationship, we divide the model into three phases: training, validation, and inference. In the right subplot, we show the error variation during validation on models trained on networks smaller than 10^3 in size, and tested across networks of various sizes. We selected network sizes for testing that are similar to the training set size ($< 10^3$), and also chose different sizes of networks uniformly in the logarithmic space between 10^3 and 10^4 for testing. The model's testing error does not show a significant growth relationship with the size of the network.

of compressing larger networks, and thus, apply it to the renormalization of larger large networks in the inference phase. The networks we used here are Watts-Strogatz (WS) networks with different sizes, their number of neighbors ranges from 2 to 8, and rewiring probabilities ranges from 0 to 0.6.

In the right subplot of Figure 6, we showcase the training of our model on networks smaller than 10^3 in size, and we tested this model on both a test set of networks of the same size and larger networks (from 10^3 to 10^4) to compare the Mean Absolute Error (MAE) of the network's partition function before and after compression. We observed that compared to smaller network data, the error did not significantly increase when our model was applied to larger networks. This indicates that the model has strong ability for generalization. Therefore, we further applied the model to the renormalization of a network with 100K nodes. Since the original network has 100K nodes and cannot be visualized, we can only visualize the network after compression. In Figure 7, we display the structure of compressed network with 50 nodes and its partition function. Notably, while the computation of the original network's partition function would have taken more than 10^4 seconds, compressing the network first and then calculating the partition function within an acceptable range of partition function error took only a few seconds.

* An analysis of a more extensive set of real networks is required. Given the authors' claim about the speed of computation and the already collected dataset of 1100 empirical networks, extending the analysis beyond the two analyzed real networks should not be overly challenging. Statistics regarding the performance of the method across various networks should be provided. Additionally, is there any observable difference depending on the network domain?

We thank the referee for raising this issue, which highlights a limitation in our existing work. We have conducted further analysis on various real networks and attempt to understand the differences in renormalization outcomes across different domains. In our

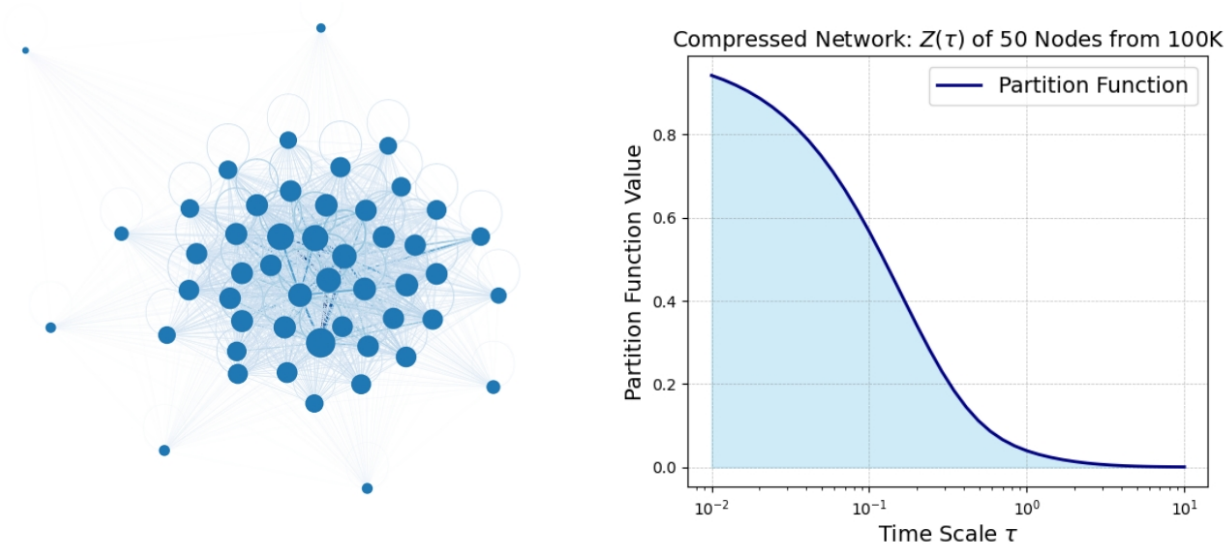

Figure 7: **Compressed WS Network: From 100K Nodes to 50 Nodes:** Visualization of the structure of the compressed network(left subplot) and Partition Function of the compressed network(right subplot).

upcoming experiments, we have selected 25 networks from 7 different fields: biological networks, economic networks, brain networks, collaboration networks, email networks, power networks, road networks and social networks. Their node number ranges from 242 to 5,908, and the number of edges ranges from 906 to 41,706. Their basic information is shown in Table 1.

We compressed all networks at various scales, with compression ratios ranging from 1% and 2% up to 25% of the original network size. We aimed to explore how the effectiveness of compression varies with the compression ratio across networks from different domains. The results are shown in the left subplot of Figure 8. First, we observed that in almost all cases, the error of the normalized partition function (measured by Mean Absolute Error) is less than 10^{-2}, indicating that our model performs well across networks from different domains. Moreover, regarding the differences between networks from various domains, we did observe some patterns, such as economic networks having the smallest error in the partition function before and after compression. However, beyond this,

Network Domain	Network	Nodes	Edges
Biology	yeast-protein-inter	1458	1948
	grid-plant	1272	2726
	grid-mouse	791	1098
	CE-GT	878	3181
	diseasome	516	1188
Economic	poli	2243	2667
	mahindas	1258	7513
Brain	mouse-kasthuri_graph_v4	987	1536
	macaque-rhesus_brain_1	242	3054
Collaboration	netscience	379	913
	CSphd	1024	1042
Email	univ	1133	5451
	dnc-corecipient	849	10384
Power	662-bus	662	906
	685-bus	685	1282
	bcspwr09	1723	2394
	inf-power	4941	6594
Road	euroroad	1039	1305
	minnesota	2640	3302
Social	pages-politician	5908	41706
	advogato	5054	39374
	pages-tvshow	3892	17239
	hamsterster	2000	16097
	wiki-Vote	889	2914
	pages-food	620	2091

Table 1: **Information about the networks:** Basic information for 25 typical networks from 7 different domains. All networks were downloaded from the Network Repository(https://networkrepository.com/).

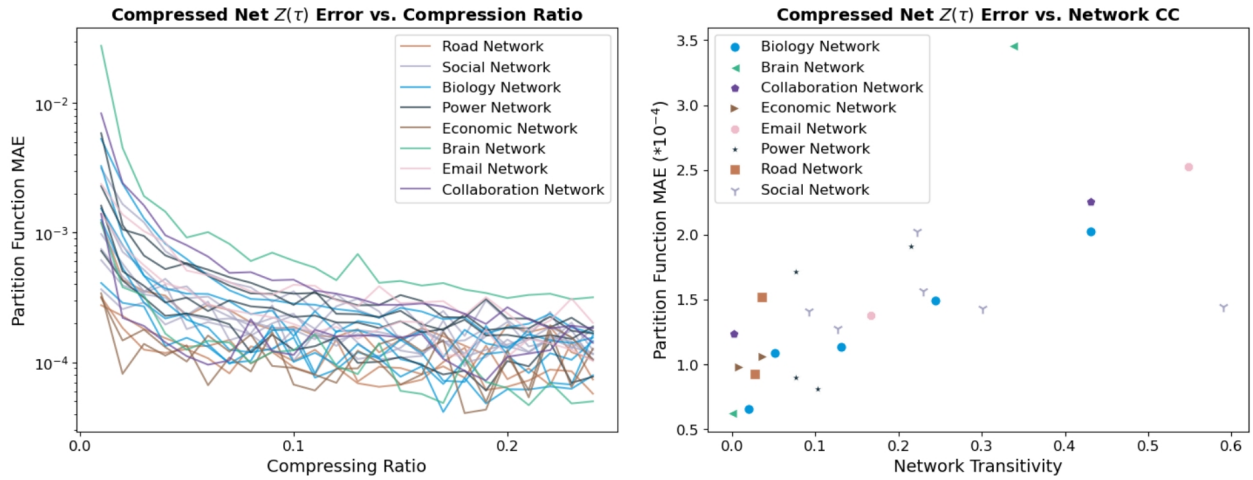

Figure 8: **Compressing empirical networks from different domains:** In the left subplot, we show the pre- and post-compression partition function error (measured by MAE) for 25 networks from 7 different domains at various compression ratios, ranging from 1% to 25%. In the right subplot, we display the relationship between the average compression error and the network's transitivity when the compression ratio is greater than 0.1 (at this point, the compression error is not sensitive to the compression size, which can be considered as the best compression error achievable by the model without being limited by size).

there were no significant differences between all networks from different domains. This prompted us to seek more general principles to explain when and why our model performs better, and why it is particularly effective on economic networks.

We discovered that regardless of the network's size or the field it belongs to, its transitivity (the fraction of all possible triangles present in the network) has a major impact on compression efficiency. As shown in the right subplot of Figure 8, we plotted the transitivity against the error in the partition function before and after compression for all networks, and observed a positive correlation(correlation coefficient of 0.676): a network with lower transitivity tends to be more easily compressed.

We believe this may be due to the fact that increased transitivity enhances the complexity of information diffusion across the network, making it more difficult to compress. On one hand, networks with high transitivity boost the speed of local information spread

in some cases, as information or influence can propagate through multiple paths. On the other hand, this type of network might show lower efficiency in certain aspects, such as potentially creating an "echo chamber" effect during the process of information spread, which limits the diffusion of new information.

No details are provided on how the alternative renormalization methods were implemented. A "See Supplementary Materials for details" was included in the main text, but I could not find the information there.

Thank the referee for pointing this out, we have updated the supplenmentary materials to add more relavent details.

More details about training are fundamental to understand the computational cost of the GNN, but are not provided. In particular, what does it mean "with sufficient training" in section Neural Network Architecture?

We thank the reviewer for raising this question; we do need to provide a more detailed explanation of the computation time for our model.

We use Graph Isomorphism Networks (GIN) for network renormalization. To run a trained model we first need to calculate node features based on the topological structure, all node features used here are local characteristics of the nodes, with the most time-consuming feature being k-coreness. The computational time complexity for k-coreness is $O(N+E)$, where N and E are number of nodes and edges of the original network, respectively. The model then calculate the node grouping information by aggregating neighbor information and updating node information, which can be represented by Equation 5.

$$h_v^{(k)} = MLP^{(k)}((1+\epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}), \tag{5}$$

in which $h_v^{(k)}$ denotes the vector representation of node v in layer k, $\mathcal{N}(v)$ denotes a set of node v's neighbors, $MLP^{(k)}$ means a multi-layer perceptron and $\epsilon^{(k)}$ denotes a learnable parameter.

Subsequently, the updated node information needs to be mapped to grouping information through a softmax operation, as represented by Equation 6.

$$S_v^i = \frac{e^{h_v^i}}{\sum_{j=1}^{M} e^{h_v^j}}, \tag{6}$$

in which S_v^i denotes the ratio that node v belongs to the super-node i and M denotes the number of a super-nodes, which is a pre-defined constant here.

Finally, the model uses the obtained grouping matrix to compress the original network and generate the adjacency matrix of the compressed network, as shown in Equation 7

$$\hat{A} = \frac{M}{N} S^T \cdot A \cdot S, \tag{7}$$

in which A and $\hat{A}$ denotes the micro and macro adjacency matrix, respectively.

In the aggregation function of GIN represented by Equation 5, it involves two operations: aggregation of neighbor information and feature update. For the aggregation of neighbor information, it requires summing up the features of all neighbors for each node. The time complexity of this part is proportional to the number of edges E, as each edge contributes to one addition operation of neighbor features, that is, $O(E)$. For the feature update part, we use an MLP (Multi-Layer Perceptron) for updates, with the time complexity of each layer being $O(N * d_{in} * d_{out})$, where N is the number of nodes, d_{in} is the input dimension to the MLP, and d_{out} is its output dimension. Therefore, the total complexity of the GIN model is $O(E + N * d_{in} * dout)$. Since d_{in} and d_{out} are manually defined constant

hyperparameters, the time complexity can generally be approximated as $O(N + E)$. This shows that the GIN model is highly efficient when processing large-scale graph data.

In the softmax operation represented by Equation 6, for each node, we need to perform M exponential operations, one summation operation of M elements, and M multiplication operations. Therefore, its time complexity is $O(N * 3M) = O(N)$.

In the process shown in Equation 7, which involves using a grouping matrix to compress the original adjacency matrix, there are two matrix multiplication operations. Since the original adjacency matrix is often represented sparsely, the time complexity of the first matrix multiplication is $O(M * E)$, and the time complexity of the second matrix multiplication is $O(M^2 * N)$. Given that M is an manually set constant representing the macro network size, the final time complexity is $O(N + E)$. Therefore, considering these three computational steps together, compressing a network with a trained model ultimately requires time proportional to $O(N + E)$.

Regarding the referee's question about training time, it involves the interpretation of what constitutes "sufficient training." In the process of training our model, we typically define a state of 'sufficient training' as one where the loss function has nearly ceased to decrease. Unfortunately, to our knowledge, there is currently no theoretical work that guarantees the convergence of the training process of graph neural networks within a certain timeframe. However, a statistical law known as the 'neural scaling law'[8, 9, 10] has been discovered in recent years. It has been observed across many models that the time required to train a model is related to the size of the dataset by a power-law relationship. Similarly, we tested the relationship between network size and training time in 25 real networks, and our results generally conform to the power-law relationship of the neural scaling law, as shown in Figure 9.

In Figure 9, we present the time required for our model to train on different real

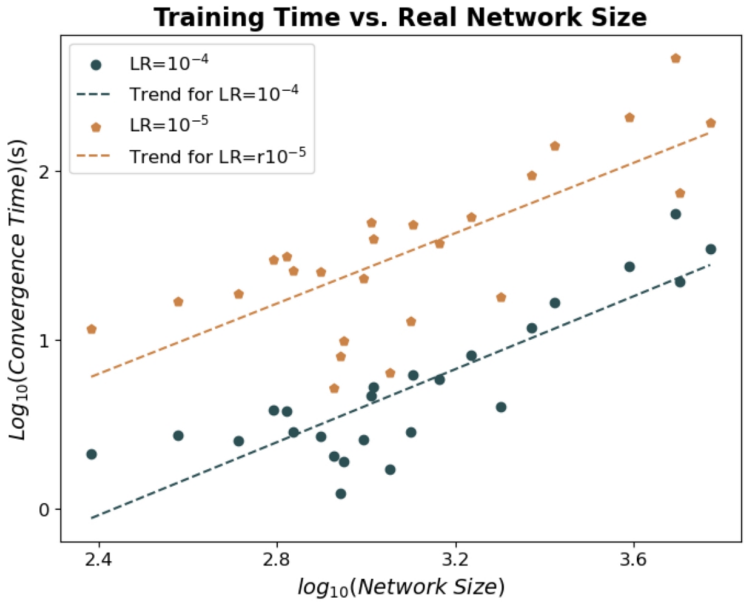

Figure 9: **Training time vs. real-world network size:** In this figure, we show the relationship between the time required for the model's loss curve to converge during training on different networks and the network size. If the average value of the last M steps of the error curve is not less than the average value between the second to last M steps and the last M steps, it indicates that the error is no longer decreasing with training, and we consider the model to have converged. Here, we chose $M = 50$.

networks at various learning rates, with the network data sourced from Table 1. We found that even with different learning rates, the required convergence time for training shows a positive correlation with network size on a log-log scale. This means our model can converge within a reasonable timeframe on large networks, finding an appropriate renormalization strategy. However, it's important to reiterate that while this figure shows a scaling law relationship between the network size and convergence time for some real networks.To our knowledge, most current studies on neural scaling laws focus on transformer-based language models rather than the graph neural networks discussed here. Moreover, this is a statistical law; there is no strict theoretical guarantee that the neural scaling law applies to the tasks discussed in this paper.

Also in reference to the training process, the authors measure the deviation of the partition function of the compressed layer from the original layer by uniformly selecting

ten points from the logarithmic interval of time before the partition function converges. These points compose a vector representing the partition function for minimizing the mean absolute error between the two. More details should be provided regarding the evolution of the partition function. Is it always smooth? Are the ten selected points truly representative of the curve?

We thank the referee for pointing out this issue; we indeed need to provide a deeper explanation of the properties of the partition function. Next, we will first demonstrate the smoothness of the partition function, and then explore how many points are sufficient to describe the partition function.

First, we can understand why the partition function is smooth from its physical meaning. We define the partition function on the network through the diffusion process occurring on the network, considering a simple yet common diffusion process whose dynamics is governed by $\frac{d\mathbf{s}(\tau)}{d\tau} = -L\mathbf{s}(\tau)$, where $\mathbf{s}(\tau)$ denotes the network state at time τ, L refers to the Laplacian matrix of the undirected network. The solution to the network's diffusion dynamics is given by $\mathbf{s}(\tau) = e^{-\tau L}\mathbf{s}(0)$. The network propagator $K = e^{-\tau L}$ describe the volume of information spreading on the network, with its ith row and jth column representing the total amount of information traveling from node i to node j through all possible paths at time τ. The partition function, denoted as $Z(\tau) = Tr(e^{-\tau L})$, intuitively represents the amount of information trapped at all initial nodes. Since information diffusion is a continuous process, the amount of information trapped at the initial nodes does not undergo "jumps," and therefore, the partition function is always smooth. Moreover, it is a monotonically decreasing function because, as diffusion progresses, the information trapped at the initial nodes inevitably decreases.

We can also demonstrate from a mathematical perspective that the partition function is smooth. Generally, if a function has all its derivatives existing and continuous, we can prove that the function is smooth. The definition of the network's partition function

is $Z(\tau) = Tr(e^{-\tau L}) = \sum_{l=1}^{N} e^{-\tau\lambda_l}$, where λ_l denotes the lth smallest eigenvalue of the laplacian matrix, all the eigenvalues are non-negative, $\lambda_l \geq 0$. Its first derivative is $Z'(\tau) = \sum_{l=1}^{N} -\lambda_l e^{-\tau\lambda_l}$, Its second derivative is $Z''(\tau) = \sum_{l=1}^{N} \lambda_l^2 e^{-\tau\lambda_l}$, and so on, Its nth derivative is $Z^{(n)}(\tau) = \sum_{l=1}^{N} (-\lambda_l)^n e^{-\tau\lambda_l}$. Since the exponential function is continuous, all derivatives are continuous, making the partition function smooth for $\tau \geq 0$.

Here, to compare partition functions of different sizes, we have normalized the partition function, making it a curve that transitions from 1 to 0.

Considering whether selecting 10 points from the partition function curve to form a vector can adequately describe the curve is indeed an important question, as fully describing the curve is a prerequisite for all our optimization work. For this purpose, we designed the experiment shown in Figure 10. We uniformly selected different numbers of points from the same logarithmic space interval to describe the partition function and set the optimization objective as minimizing the error of the vectors formed by these points when compressing the network. For the compressed partition function, we chose a sufficiently large number of points (here, 50 points uniformly in logarithmic space) to form vectors and calculate the error with the original vector. We observed that less than 5 points are insufficient to describe the shape of the partition function, resulting in larger errors in the compression outcome. However, beyond 8 points, there is no significant decrease in error, indicating that 8 points provide a fairly comprehensive description of the partition function. Despite the different partition function errors achievable on different empirical and synthetic networks, we observed a similar phenomenon across all cases. We finally chose 10 points as a trade-off between the accuracy of describing the partition function and computational time.

The section on interpretability does not open the black box of the neural network used. It essentially serves as a sensitivity analysis of the contribution of structural features to the model's output, as acknowledged by the authors. From this analysis, the authors

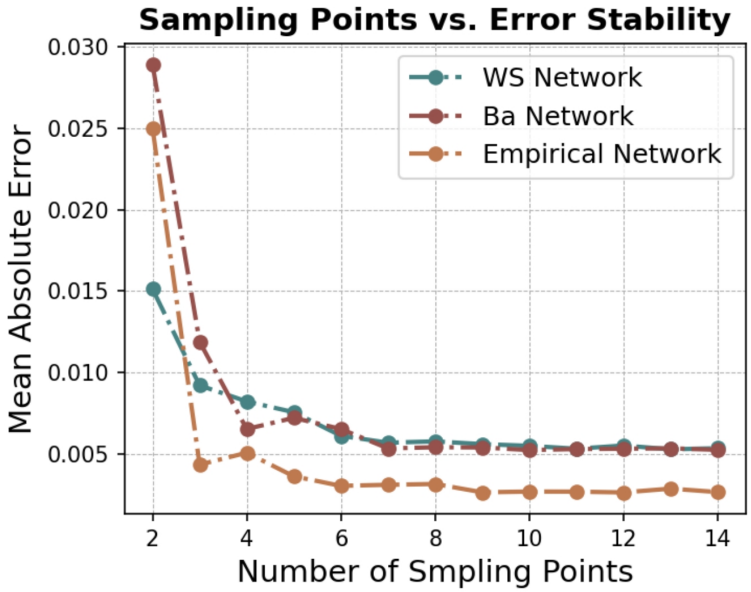

Figure 10: **Observing Error Stability by Number of Sampling Points:** The x-axis of this figure represents the length of the vector chosen during network optimization, which corresponds to how many points are uniformly selected in the logarithmic time interval to form a vector of partition function values for training. The y-axis does not represent the corresponding training error but rather the MAE between the normalized partition function of the macroscopic network obtained after sufficient training and the normalized partition function of the original network. Here in the y-axis, the MAE is measured using a sufficiently large number of points (50 points uniformly selected in logarithmic space).

suggest that nodes with a similar ”ecological niche” are grouped together. This claim should be substantiated further.

We thank the reviewer for raising this question. We acknowledge that the section on interpretability actually pertains more to sensitivity analysis of structural features affecting model outputs, rather than serving as a comprehensive part of model interpretability. However, we would first point out that although feature sensitivity contributes limitedly to interpretability, this approach is still widely accepted due to the inherent challenges in explaining deep learning models. Secondly, we will show how, following the reviewers' suggestions, we have enhanced our understanding of how our model works.

The complex decision pathways between inputs and outputs in deep learning confer powerful capabilities but also make it difficult to explain how deep learning models function. Post-hoc explanation methods, typically represented by feature sensitivity analysis, are often model-agnostic, meaning they can be applied to any deep learning model. Moreover, feature sensitivity analysis does not require modifications to the original model, which is particularly important for the practical application of deep learning models. Consequently, feature sensitivity analysis methods, such as LIME[11], SHAP[12], and Integrated Gradients[13], are widely used in deep learning applications across fields like biology[14, 15, 16], medicine[17, 18], social science[19] and computer vision[20, 21, 22]

However, as the reviewers have pointed out, feature sensitivity analysis provides a very limited explanation of how the model works. We need to further explore why the model is effective.

Here, the 'ecological niche' we refer to is actually a function of a node's local structure. The term ”ecological niche” is not precise enough; we have removed the relevant description from the main text and replaced it with ”the local structure of nodes.” This information of the node serves as the input to this function, and the node's grouping in-

formation is the output. The mapping process inside this function is learned by a graph isomorphism network model. Neural network models are often considered to be a black box; here, we attempt to reveal how this model works.

The essence of graph neural networks is to train a learnable function that takes local structures as input and outputs groupings. Thus, our model has a natural property: nodes with similar local structures will be grouped together, which is actually beneficial for preserving the network's information flow. This can be intuitively understood through the analysis of several typical structures. **High Clustering Coefficient Nodes:** Nodes with a high clustering coefficient often have neighbors that are interconnected, acting to trap information locally during the diffusion of information in the network. Grouping nodes with a high clustering coefficient together, their numerous interconnections will become self-loops on the supernode, similarly playing a role in trapping information locally, as shown in Fig. 11, subplot a. **Hub Nodes:** Hub nodes act as bridges for global information diffusion within the network. By clustering hub nodes together, the edges between hub nodes and other nodes transform into connections between supernodes in the compressed network, still ensuring the smooth passage of global information. Suppose we group a hub node and its neighbors into one; then, their interconnections become self-loops of that supernode, acting as a barrier to information transmission, as shown in Fig. 11, subplot b. **Nodes in Communities:** Networks with distinct community structures experience slow global information diffusion (due to few inter-community edges). Grouping nodes of the same community together transforms the numerous intra-community edges into self-loops on the supernode, trapping information locally. Meanwhile, the few inter-community edges become low-weight edges between supernodes, similarly serving to slow down the speed of global information diffusion as shown in Fig. 11, subplot c.

However, based on the principle that "nodes with similar local structures are grouped together," the graph neural network still needs to optimize the specific grouping strategy.

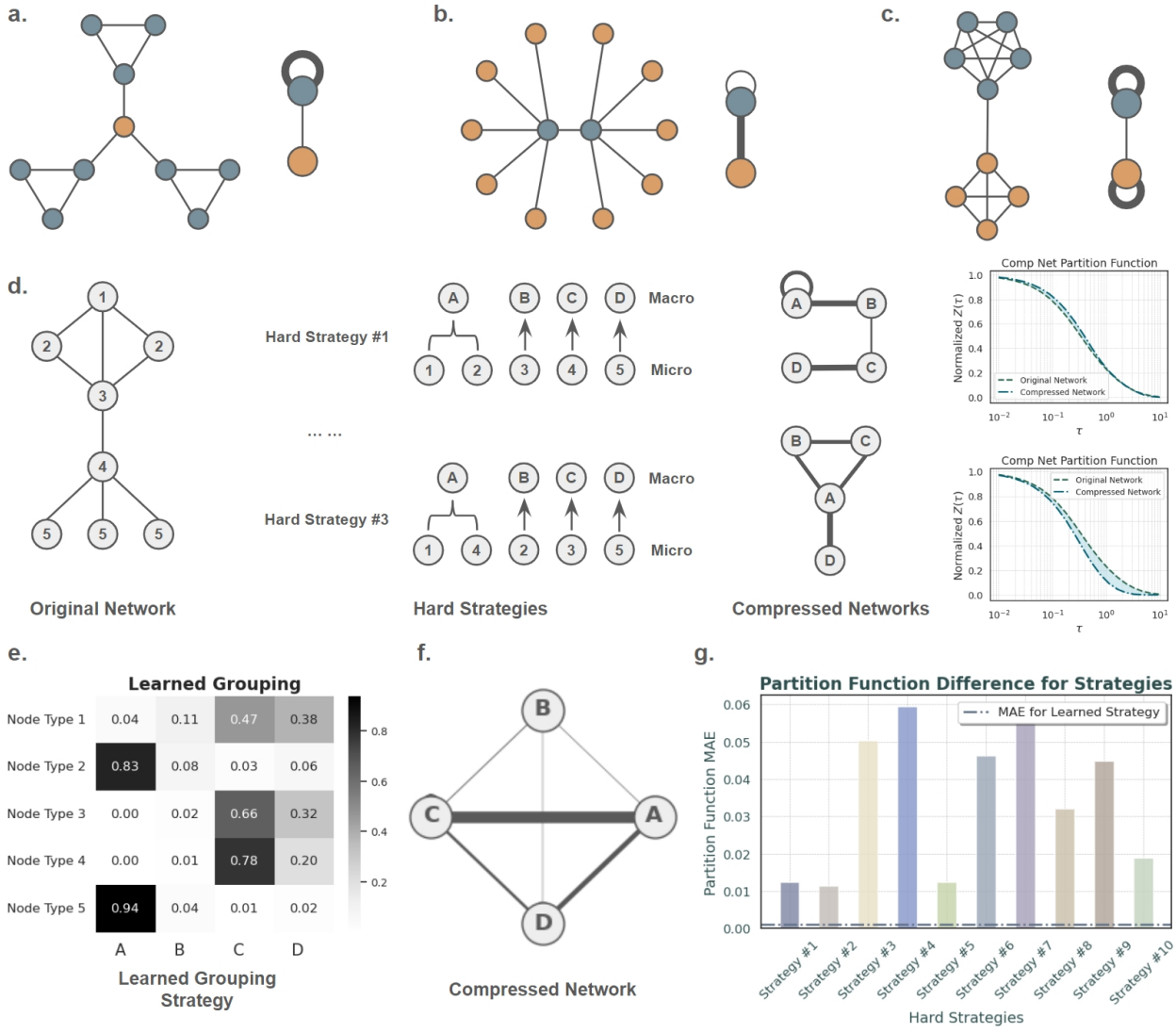

Figure 11: **Grouping strategies and the structure of the macroscopic network:** Subplots $a-c$: Compressing nodes with similar local structures into a supernode reduces structural redundancy and preserves the role of local structures in information diffusion. Nodes with high clustering coefficients and community structures tend to trap information locally, while hub nodes act as global bridges. Subplot d: we show the macroscopic network after compression using different hard(discrete) grouping strategies for a toy network with 8 nodes and 5 types of different local structures, and the changes in the corresponding partition function. Subplot e: the soft(continuous) grouping strategy learned by our model. In subplot f, we present the macroscopic network structure given by the grouping strategy from subplot e. Subplot g: comparison of the normalized partition function error before and after compression under different grouping strategies.

This is mainly because, on one hand, the number of groups is often far less than the number of types of nodes with different local structures (imagine a network with 3 different types of nodes, whereas we need to compress it into a macro network with 2 nodes). On the other hand, our model produces continuous grouping results, for example, a certain type of node might enter group A with an 80% proportion and group B with a 20% proportion. Each different proportion represents a new grouping strategy, and we need to find the most effective one from infinitely many continuous grouping strategies, which is an optimization problem, hence requiring the aid of a machine learning model. Subplots $d - f$ of Fig. 11 show discrete grouping strategies and the learned continuous grouping strategy for compressing a network with 5 types of nodes into a macroscopic network with 4 supernodes.

In subplot d of Figure 11, we know that since there are 5 types of nodes with different local structural features. When we want to compress them into 4 groups, if each node clearly belongs to a certain group and every group must have at least one node, we will produce C_5^2 different grouping strategies. We call these strategies "hard grouping." However, if we allow a type of node to belong to multiple groups at the same time (for example, belonging to group A with a proportion of 0.3 and to group B with a proportion of 0.7), then there are infinitely many grouping strategies. We call these strategies "soft grouping." Each grouping strategy will produce a macroscopic network and generate a new partition function. In subplot d of Figure 11, we show the macroscopic networks and corresponding partition functions produced by two hard grouping strategies. In subplots e and f, we respectively show the grouping matrix of the soft grouping strategy learned by our model and the macroscopic network structure. In subplot g, we present the differences between the partition functions of all reduced networks(including the networks generated by all hard grouping strategies and our learned soft grouping strategy) and the original network's partition function. We find that the soft grouping strategy, due to its stronger expressive power, can produce a macroscopic network with a more similar par-

tition function. This type of grouping strategy is not manually designed but optimized through the gradient descent method. By taking gradients of the GNN's internal parameters, our model automatically discovered the grouping strategy shown in subplot e. In summary, this is how our model works: based on the fundamental idea that "nodes with similar local structures enter the same group," our model uses the gradient descent algorithm to select one of the infinitely many solutions (grouping strategies) that results in a lower error of the partition function.

There are inaccurate statements in the manuscript. For instance:

* "This non-local effect has no structural counterpart in other methods, as in the classical network renormalization methods, super-nodes always consist of connected subnetworks." This is simply wrong. There are network renormalization methods that coarsen groups of nodes even without the need that those nodes are connected.

We thank the referee for pointing this out, we have changed this wrong statement in the manuscript.

* The authors assert that geometric methods are order N^3, while in the relevant papers it is stated that the method is N^2.

Thank the referee for pointing out this issue; we have changed the relavent statement in the main text.

The investigation leaves many relevant questions unanswered:

* What is the behavior of the network topology in the flow? Are the degree distribution and the clustering spectrum preserved? What about degree-degree correlations? Which are the effects of having a homogeneous/heterogeneous connectivity, high/low levels of clustering, etc.?

Thank the referee for raising this question. We will discuss in greater detail here the

impact of a network's topology on information flow. The information flow of a network is described by its partition function, which is the counterpart of the partition function in statistical mechanics applied to complex networks, but with a clear physical meaning. Assuming simple and common diffusion dynamics are occurring on the network, described by $\frac{d\mathbf{s}(\tau)}{dt} = -L\mathbf{s}(\tau)$, where $\mathbf{s}(\tau)$ denotes the network state at time τ and L denotes the laplacian matrix of the network. with the solution being $\mathbf{s}(\tau) = e^{-\tau L}\mathbf{s}(0)$. The network propogator $K = e^{-\tau L}$ represents the overall flow of information across the network. For instance, the ith row and jth column of K at time τ represents the total amount of information reaching node j from node i through all possible paths at time τ. The partition function of the network $Z(\tau) = Tr(e^{-\tau L})$ is the trace of the network propagator K, whose physical meaning is the total amount of information trapped at the initial nodes up to time τ. At the start of diffusion ($\tau = 0$), all information is at the initial nodes, and the partition function is at its maximum value. At the end of diffusion ($\tau = \infty$), information is uniformly distributed across all nodes, and the partition function reaches its minimum value. The partition function is a function of the network structure at a given time interval, making it crucial to discuss how different topological properties affect the partition function curve.

In Figure 12, we demonstrate the impact of different network topological properties on the partition function. Here, we focus on the effects of network density, clustering spectrum, community structure, degree distribution, degree-degree correlation, and homogeneity/heterogeneity on the network structure. Specifically, we observe the shape of the partition function by altering one topological property of the network while trying to keep other properties unchanged, to gain insights into the behavior of information flow on top of the network.

In subplot a of Figure 12 we generated networks of different densities by changing the connection probability p in ER networks. We directly observed that density has a signifi-

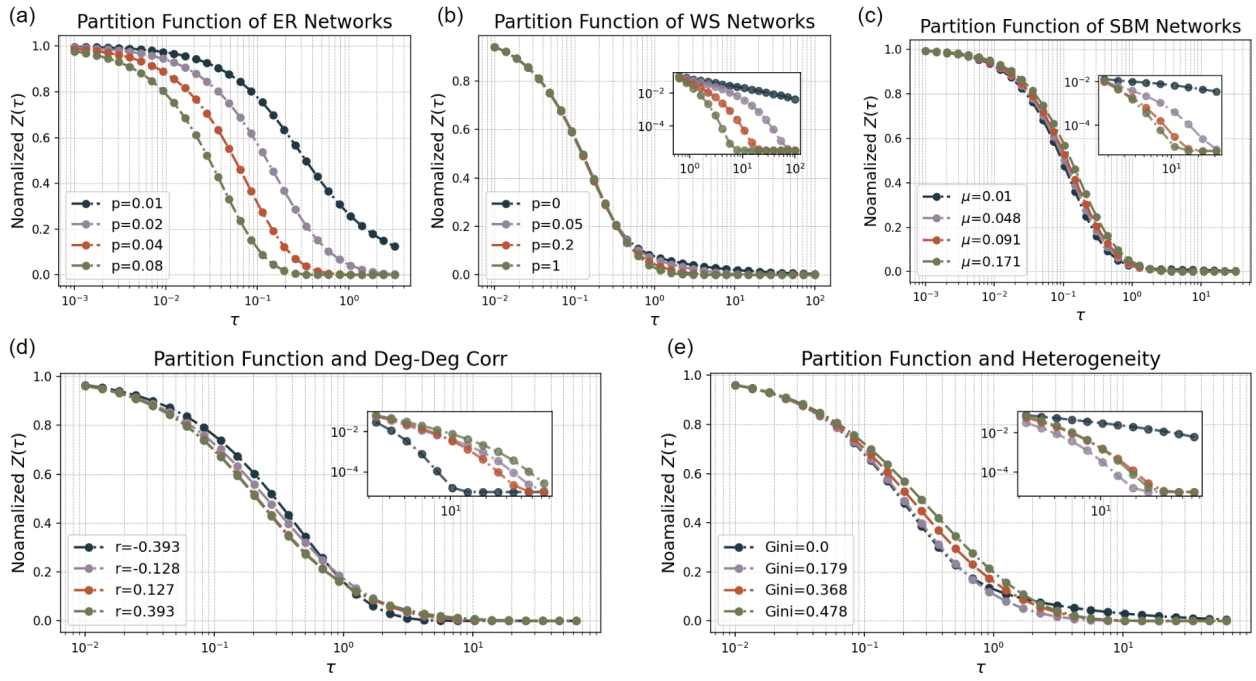

Figure 12: **Network topology and information flow:**In this figure, we show how the network's partition function responds to changes in structural properties. In the subplot a, we set different network densities by adjusting the probability of edge creation for an Erdos Renyi graph with 400 nodes. In subplot b, we observe the effect of the small-world phenomenon on the partition function by adjusting the probability of rewiring each edge in a WS graph (with 300 nodes, each having 6 neighbors). In subplot c, we examine the impact of links between community structures on the partition function by setting the mixing parameter μ for a stochastic block network (with 3 communities, each containing 50 nodes). Here, $\mu = \frac{k_{out}}{k_{in}+k_{out}}$, where k_{in} denotes the number of connections within a community, and k_{out} denotes the number of connections between communities. In subplot d, we observe the impact of the network's degree-degree correlation on the partition function, which can be measured by the assortativity coefficient r. With constant network density, we increase the network's assortativity by adding edges between nodes of similar degrees and removing edges between nodes of differing degrees. We reduce assortativity by removing edges between nodes of similar degrees and adding edges between nodes of differing degrees. In subplot e, we gradually rewire the edges of a regular network with 100 nodes, each with 4 neighbors, using the principle of preferential attachment to transition it into a scale-free network. During this process, homogeneity decreases while heterogeneity increases, and we use the Gini coefficient of the degree distribution to measure the level of heterogeneity.

cant impact on the partition function: an increase in density causes the partition function to drop more quickly because higher density facilitates information diffusion across the network, making diffusion easier to complete. In subplot b, we observe the impact of the clustering spectrum on the partition function by changing the rewiring probability in small-world networks. When the rewiring probability of the small-world network is very low, each node is connected to a few neighbors, resulting in a high average clustering coefficient. Increasing the rewiring probability breaks these triangles, reducing the network's average clustering coefficient. With the total number of connections remaining unchanged, this affects the diffusion dynamics as follows: each node can initially diffuse information to its locality with the same difficulty, but at larger time scales, as the clustering coefficient decreases and long-range connections increase, global information diffusion becomes easier. Therefore, reducing the clustering coefficient makes large-scale information diffusion easier to complete. This is reflected in the partition function, meaning it approaches zero faster at large τ.

In subplot c of Figure 12, we generated networks using the SBM model and altered the network's μ to observe the effect of inter-community connections on information diffusion. In the SBM model, $\mu = \frac{k_{out}}{k_{in}+k_{out}}$, in which k_{out} denotes the number of connections between communities and k_{in} denotes the number of connections inside the communities. By removing intra-community edges and adding inter-community edges, while keeping the network density constant, we notice that as the mixing parameter increases, the initial decline of the partition function slows down. This is due to the reduction of intra-community edges, which slows down the network's local (within-community) diffusion speed. Conversely, the latter part of the partition function declines faster because the increase in inter-community edges accelerates the global diffusion speed.

In subplot d of Figure 12, we measured the impact of degree-degree correlation on the network's partition function, which can be quantified by the assortativity coefficient r.

When the assortativity coefficient is high, the initial decline of the partition function is faster. This is because the distribution of node degrees is relatively uniform, with each node having a sufficient number of neighbors, facilitating local information diffusion. When the assortativity coefficient is low, the decline in the latter part of the partition function is faster. This is because a few high-degree nodes connect to many low-degree nodes, acting as bridges and increasing the speed of global information diffusion.

Finally, in subplot e of Figure 12, we studied the impact of homogeneity/heterogeneity on the partition function. In a homogeneous network, different nodes have similar local structures(i.e. regular networks). In a heterogeneous network, the distribution of node degrees is broad, with certain nodes (referred to as central or hub nodes) having many more connections than others, leading to a diversity in the network's structure and connection patterns. The BA network is a typical example of a heterogeneous network. The homogeneity/heterogeneity of a network can be measured by various indicators, and here we chose the Gini coefficient of degree distribution to measure the heterogeneity. In economics, the Gini coefficient is used to measure income inequality; here, we use it to measure the unevenness of the network's degree distribution. The Gini coefficient ranges from 0 to 1, where 0 represents complete equality (total homogeneity) and 1 represents maximum inequality (total heterogeneity). In this experiment, we started with a regular network and rewired it with preferential attachment mechanism. After enough rewirings, this homogeneous regular network transformed into a heterogeneous scale-free network. We observe that in completely homogeneous networks, the partition function decreases more quickly at smaller τ but slows down at larger τ. Conversely, heterogeneous networks show the opposite pattern, with a slower decline at smaller τ and a faster decrease at larger τ. This difference can be understood from the dynamics of information diffusion. when τ is small, representing local information spread, homogeneous networks, where each node has the same number of neighbors, facilitate easy local information diffusion. In contrast, in heterogeneous networks, many nodes have only a few neighbors, making

local information spread more challenging. When τ become larger, the network come to the global diffusion phase, the heterogeneity of networks, due to the presence of hub nodes, accelerates global information communication, whereas homogeneous networks lack such channels for global communication, hence slowing down at this stage.

The partition function effectively describes the flow of information under diffusion dynamics, but it's important to note that although many topological properties can influence the partition function, our renormalization method can preserve the partition function with high accuracy. This does not mean that our renormalization method will necessarily preserve these structural properties. This is fundamentally because our model produces a graph with weights and self-loops. The addition of weights and self-loops allows the model to maintain the information flow without using some of the structural features (for example, by controlling the amount of information that stays at the node itself through weighted self-loops). Therefore, we did not directly compare the structural properties of networks before and after compression.

* In relation to the point above, results in Fig. 2 suggest that the method works much better for trees. This point needs to be checked more carefully.

We thank the referee for the the sharp observation. In Fig. 2 in the main text, we conducted experiments on four different types of synthetic networks, where the error in the partition function before and after compression on the BA network (m=1) was lower compared to other networks. To address this question, we need to investigate under what circumstances the model performs better. This question essentially relates to the similar issue the referee mentioned earlier about "the model's renormalization effects on networks in different domains". Through our investigation into this matter, we found in Fig 8 that the model performs better on networks with lower transitivity. We believe this is because high transitivity increases the complexity of information diffusion dynamics. Since the transitivity of a tree is 0, we think this explains why the compression effect on

trees differs from that on other synthetic networks.

Less critical but still important, the terminology is occasionally ambiguous and confusing, and there is some language misuse. For instance:

*"upon reaching a certain compression size that plays the role of a transition point, further compressions lead to significantly higher costs", and "transition point" is used in several other places. In our field, a transition has a very specific meaning. I could not see any transition point in the reported results.

We thank the referee for pointing out this issue. "Transition point" typically refers to a critical point where there is a significant change in the system's state, such as a phase transition. In our research, the "transition point" does not involve a fundamental change in the system's state but rather a specific point where the error rate significantly increases as the compression size decreases. Indeed, using this term might lead to misunderstandings. Therefore, we have replaced all related expressions with "critical size."

* The niches the authors talk about in the paper are not ecological in any form.

We thank the referee for pointing out this issue. Indeed, the term "ecological niche" was not precise enough. We have replaced all instances of this term in the paper with "local structures".

* Second paragraph of the introduction "to search for proper compressions of networks...that preserve the flow.", and the same idea again in the third paragraph. Meaningless sentence. What is preserved is not the flow, but some magnitude in the flow.

We thank the referee for the comment. We would like to take this chance to provide arguments for the claim that our method is not merely preserving the magnitude.

As the referee knows, network density matrices are rich objects capable of incorporating myriads of topological network properties, coupled with dynamical processes, and

give compact descriptions of network flow at different topological scales.

This generality makes network density matrices powerful tools for analyzing complex systems. How fast can the quantities of interest flow? How diverse are the carrying pathways? What is the probability that a unit of the flow returns to its initial location? Statistical physics measures obtained from network density matrices, like network entropy and free energy, have been widely used to provide a quantitative proxy answering these questions [1, 3, 4, 5]. Essentially, all these macroscopic measures are functions of the spectrum and the propagation scale. Yet it's worth noting that these functions are non-linear, and their properties can't be simply reduced to their input variables— e.g., the degree distribution of the graph or even the eigenvalue distribution on its own.

To assess whether a compression preserves the flow properties, one can compare it against the original flow captured by measures like entropy and free energy. Yet, as we show in the text, a shortcut exists: if a compression replicates the full partition function curve (across τ), it also replicates all the metrics. This is the cornerstone of our approach.

Therefore, by replicating the full partition function curve, our method ensures preserving all these macroscopic measures of the flow.

We thank the referee once again and to resolve the ambiguity, modified the second paragraph of the introduction part.

Also, be attentive to inflated self-citations. Specifi all theecally, in the first paragraph of the introduction, there are citations on multiplexity and interdependence, a topic not directly relevant to the topic of the manuscript. These citations are used to introduce seven references, of which four are self-citations.

There are typos in the text, for instance "open the doors" in the first paragraph of the introduction, etc..

We thank the referee for raising this issue. We have tried our best to find and correct all the typos in the paper. Also, we have removed two self-citations: now there are 5 citations, as many as for the higher-order networks (where there are no self-citations). Note that both topics are very important in the literature of network science and deserve to be cited for building an adequate context about the complexity of empirical networks.

References

[1] Ghavasieh, A. & De Domenico, M. Diversity of information pathways drives sparsity in real-world networks. *Nature Physics* **20**, 512–519 (2024). URL http://dx.doi.org/10.1038/s41567-023-02330-x.

[2] Ghavasieh, A., Stella, M., Biamonte, J. & De Domenico, M. Unraveling the effects of multiscale network entanglement on empirical systems. *Communications Physics* **4** (2021). URL http://dx.doi.org/10.1038/s42005-021-00633-0.

[3] Ghavasieh, A., Nicolini, C. & De Domenico, M. Statistical physics of complex information dynamics. *Physical Review E* **102**, 052304 (2020).

[4] Ghavasieh, A. & De Domenico, M. Generalized network density matrices for analysis of multiscale functional diversity. *Phys. Rev. E* **107**, 044304 (2023). URL https://link.aps.org/doi/10.1103/PhysRevE.107.044304.

[5] Ghavasieh, A. & De Domenico, M. Enhancing transport properties in interconnected systems without altering their structure. *Phys. Rev. Res.* **2**, 013155 (2020). URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.013155.

[6] De Domenico, M. & Biamonte, J. Spectral entropies as information-theoretic tools for complex network comparison. *Phys. Rev. X* **6**, 041062 (2016). URL https://link.aps.org/doi/10.1103/PhysRevX.6.041062.

[7] Ghavasieh, A. & De Domenico, M. Maximum entropy network states for coalescence processes (2022). URL https://arxiv.org/abs/2212.02392.

[8] Kaplan, J. *et al.* Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361* (2020).

[9] Henighan, T. *et al.* Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701* (2020).

[10] Ghorbani, B. *et al.* Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740* (2021).

[11] Ribeiro, M. T., Singh, S. & Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144 (2016).

[12] Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems* **30** (2017).

[13] Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328 (PMLR, 2017).

[14] Keyl, P. *et al.* Single-cell gene regulatory network prediction by explainable ai. *Nucleic Acids Research* **51**, e20–e20 (2023).

[15] Liang, J. *et al.* Deep learning supported discovery of biomarkers for clinical prognosis of liver cancer. *Nature Machine Intelligence* **5**, 408–420 (2023).

[16] Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).

[17] Holzinger, A., Langs, G., Denk, H., Zatloukal, K. & Müller, H. Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **9**, e1312 (2019).

[18] Jiménez-Luna, J., Grisoni, F. & Schneider, G. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence* **2**, 573–584 (2020).

[19] Ferrara, E. Social bot detection in the age of chatgpt: Challenges and opportunities. *First Monday* (2023).

[20] Dehghani, M. *et al.* Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 7480–7512 (PMLR, 2023).

[21] Cornia, M., Stefanini, M., Baraldi, L. & Cucchiara, R. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10578–10587 (2020).

[22] Chefer, H., Gur, S. & Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 782–791 (2021).

Letter for the Referees:

Detailed Response

Referee #1 (Remarks to the Author):

We thank the reviewer for the time and effort and the constructive comments. Here, we respond to each comment in detail. Yet, before that, we provide a quick summary of them, as three key points:

Firstly, we discuss the theoretical transformations guaranteeing that our method outperforms LRG in preserving the macroscopic features of information flow (in all network types, not only those possessing scale-free topological features) across wide ranges of τ (not at a manually selected specific τ value). Additionally, we would like to mention that although we previously provided a wide range of numerical results to support this claim, we admit that they can be thought of as indirect evidence. Therefore, we add new figures showing the preservation of entropy curves through compression guided by our method, regardless of the network type, and its superiority over other methods. This result is expected because while our method is designed for this aim, the other methods are designed for other aims.

Secondly, we thank the reviewer for discussing the concept and definition of RG, which has been useful to guide us how to better highlight the differences between our methods. While the term RG is used in a broader way in the temporary literature, we fully agree with the reviewer that higher precision can clarify the differences and similarities of methods. Therefore, agreeing with the reviewer that what we do is not exactly RG but a coarse-graining aimed to recover certain properties in the network, we renamed our manuscript to Reducing the Dimension of Network-Driven Processes Using Generalized Thermodynamics. Moreover, in the revised manuscript, we opted for using compression and to compress rather than renormalization and to renormalize, to further stress the

aforementioned difference.

Finally, we thank the reviewer for the questions about BA networks compressions. We have to emphasize that our method's aim is not to recover/preserve topological properties, but it's aimed to preserved curves like entropy and free energy across τ. Also, since our compressions are weighted and full of self-loops, comparing them with the binary graphs would be unfair. Yet, we were positively surprised to find that the scale-freeness of degree distribution is retained through compresssion.

We hope this summary captures the essence of our response, and we appreciate the valuable comments from the reviewer. For more detailed point-by-point responses, please refer to the content in the following sections.

I have appreciated the meticulous and detailed response of the referees. Still, as my first referee report explained, the paper does not propose any substantial novelty to the actual RG problem on complex networks. Indeed, in the present form, the manuscript can add some confusion to the global picture of the problem. Again, after addressing a major revision I can recommend it to be published in other specific journals such as Scientific Reports or Communications Physics. In what follows, the authors can find my comments on the revised version of the manuscript, which has not essentially changed from its original form.

I suggest the authors do a major revision to different new parts of the manuscript.

1) "LRG focuses on the specific heat at the specific scale of phase transition to define their information cores, we focus on replicating the full partition function profile at all scales of propagation." "LRG compresses network size by discarding the fast modes of diffusion processes (corresponding to eigenvalues greater than $\frac{1}{\tau^*}$), while our method tries to preserve the complete distribution of eigenvalues at any scale".

I want to clarify about the nature and the spirit of the LRG. The authors make some

statements that are not true both in the rebuttal letter and across the manuscript. They must be softened.

The LRG maps the Fourier space information contained in the eigenvalues of the Laplacian onto the density matrix the authors use here. In fact, tightly coupling spatial and temporal scales, it uses specific times to perform different network reductions. But, still, I insist it is a general theory does not depend on specific scales but that allows to analyze of different network mesoscopic scales.

We thank the referee for this comment. However, we would like to emphasize that our approach is designed to take a network as input and give a compressed network as output, where the output network preserves the entropy and free energy curves at all scales, in contrast with methods focusing on a manually selected τ value. We remind the reviewer that replicating the partition function curves guarantees the preservation of entropy, free energy, etc. — achieved by our method as shown for various network types in Fig 1.

Note that since the density matrix is Gibbsian-like, the classic statistical mechanics relations for free energy F, internal energy U and the entropy S of the system

$$\begin{aligned} F &= -\log Z/\tau, \\ U &= -\partial_\tau \log Z, \\ F &= U - S/\tau, \\ S &= \tau U - \tau F, \end{aligned}$$

hold true.

To derive the transformations required for compression, we consider a compression where the system changes size from N to N', leading to a new Laplacian (L') and a new

partition function (Z'). Here, we aim to reach a compression that follows:

$$Z = \gamma Z', \tag{1}$$

regardless of τ, with minimum error.

In fact, if the two networks have different sizes, the only valid value for the multiplier is $\gamma = N/N'$. A simple argument to support this claim is that at $\tau = 0$ it can be shown that $Z = N$ and $Z' = N'$, regardless of the network topology. Therefore, the only solution not rejected by the counter example of $\tau = 0$, which gives the maximum value for the partition function, is $\gamma = N/N'$. If $\gamma = N/N'$ with N and N' being the sizes of the original and the reduced systems, these transformations can be viewed as size adjustments.

We show that such a compression simultaneously preserves other important functions, like entropy and free energy, after size adjustment. For instance, the system's entropy has its maximum $S = \log N$ at $\tau = 0$, which can be transformed into the compressed entropy with the subtraction of the factor $\log \gamma$ as size adjustment $S \to S - \log \gamma$. Similarly, for free energy $F = -\log Z/\tau$, the size required adjustment reads $-\log Z/\tau \to -\log Z/\tau + \log \gamma/\tau$.

The partition function transformation we discussed earlier ($Z \to \gamma Z$, and therefore $Z' = Z/\gamma$), automatically guarantees both transformations:

$$
\begin{aligned}
S' &= -\tau\partial_\tau \log Z' - \tau F' \\
&= -\tau\partial_\tau \log Z' + \tau \log Z' \\
&= -\tau\partial_\tau [\log Z - \log \gamma] + \log Z - \log \gamma \\
&= S - \log \gamma. \qquad (2)
\end{aligned}
$$

Also, since the von Neumann entropy of the coarse-grained system is given by

$$
S' = \tau U' + \log Z', \qquad (3)
$$

the transformation of the internal energy can be obtained as:

$$
\begin{aligned}
S + \log \gamma &= \tau U' + \log \gamma + \log Z, \qquad (4) \\
\tau U + \log Z &= \tau U' + \log Z, \qquad (5) \\
U &= U'. \qquad (6)
\end{aligned}
$$

All in all, our work uses machine-learning technique to find compressions that approximate $Z' = Z/\gamma$, for all network types across all values of τ.

Our approach performs the mathematically severe task of approximating a solution that truly replicates the flow, regardless of the scale (τ), and the network type.

On the other hand, LRG is designed to target a different task, as the referee lays out in the previous and following comments. In fact, LRG does its job nearly perfectly. And, of course, our approach can not replace LRG since we have designed to solve a totally

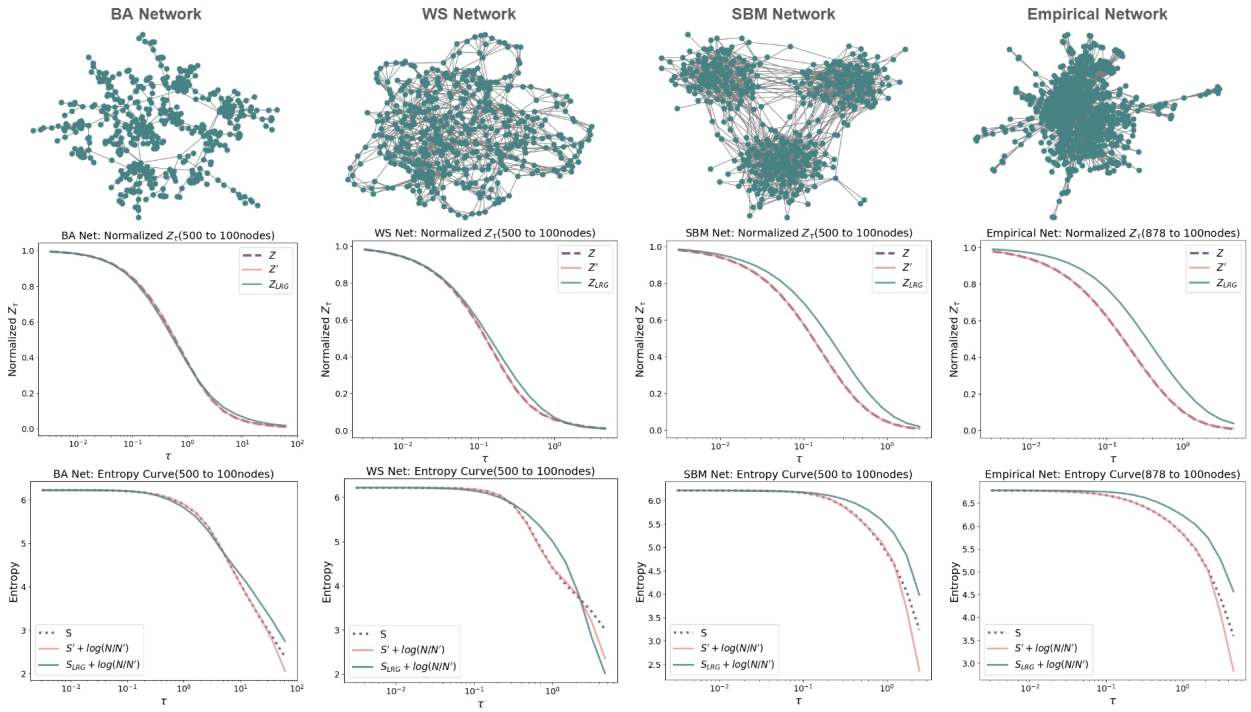

Figure 1: **Effectiveness of preserving Entropy after network compression:** In this figure we demonstrate the preservation of the partition function and entropy after compressing 3 synthetic networks and a empirical network using our method and LRG. Here, Z, Z', and Z_{LRG} denote the partition functions of the original network, our compressed network, and the LRG-compressed network, respectively. Similarly, S, S', and S_{LRG} represent the entropies of the original network, our compressed network, and the LRG-compressed network, respectively.

different problem.

We highlight again that the main question is: "can LRG take any network and return a single compressed version that keeps the flow properties (e.g., entropy) at all scales simultaneously?" The answer is clearly no, given our numerical and theoretical arguments.

In fact, in the following comments, the referee him/herself correctly mentions how the structures emerging from LRG differ in their flow properties with the original network if, for example, the network has characteristic scales.

We thank the referee for the time and effort and the valuable comments once again.

Instead, I want to emphasize here what the RG is. The renormalization group (RG) is the fundamental tool in statistical physics and statistical field theory to study and describe the physical behavior of systems showing a multi-scale and/or scale-invariant behavior as, for instance, it happens in the proximity of a critical point of a second-order phase transition at and out of equilibrium. It comprises three fundamental ingredients

i) A coarse-graining procedure: it can be performed either (a) in real space, through the derivation of mesoscopic collective variables (i.e., block variables) from a local resummation of the microscopic ones inside identical mesoscopic cells of size Λ tiling the whole space, or (b) in the conjugate Fourier space by rewriting the model through the Fourier modes of the microscopic variables and explicitly integrating all the modes with wavelength smaller than an arbitrary mesoscopic value Λ. In both cases, the new lower cut-off of the model is given, respectively by Λ. For instance, in the Ising model, we can either define à la Kadanoff mesoscopic magnetic moments summing the microscopic spins inside identical cells, or à la Wilson, through the Hubbard-Stratonovich transformation, we can rewrite the model in terms of a continuous spatial field whose Gaussian approximation is diagonal in Fourier space.

ii) Reformulation of the model (i.e., redefinition of the interaction constants of the system) in terms of the block variables in real space and of the variable modes with wavelength larger than the new lower cut-off.

iii) Spatial rescaling, so that the new lower cut-off becomes the new unit length.

To extend this approach to models defined on heterogeneous graphs, the fundamental step to overcome to formulate a correct RG approach was the first one. Indeed, given a homogeneous model (i.e., with interaction constants identical for all spatial points as the Ising model with nearest-neighbor interactions, for instance), the definition of coarse-graining in identical mesoscopic cells or integrating small wavelength modes all over the

space is immediate both in homogeneous spaces (e.g., Rd) or homogeneous lattices or trees due to the continuous or discrete translational invariance of the space itself. This fundamental property is completely lost in heterogeneous networks. Consequently, both the definition of real space block variables and the resummation procedure of small wavelengths of the model in Fourier space may appear completely arbitrary or even meaningless. The LRG scheme for connected undirected graphs tackles this point as a natural extension of the coarse-graining in homogeneous spaces. First, I want to notice that the Laplacian operator in such spaces can be strictly related to the progressive coarse-graining procedure in the RG: it is an operator with the spectrum of eigenvalues given by eigenvalues $-k^2$, with k Fourier wave-vector, i.e., the inverse of the wavelength, which has no characteristic scale. The corresponding eigenvectors are exactly given by the Fourier basis of plane waves e^{ikx} , which are also eigenfunctions of the translation group. I think that the authors can easily see that a network reduction automatically implies a rescaling function of the eigenvalues distribution function, which needs to be properly rescaled if the network maintains its intrinsic properties, but that CANNOT remains constant when changing the system scale.

The basic concepts I have explained here can be easily followed in the classical books of Kardar, M. Statistical Physics of Fields [Kardar, 2007]. or Amit, D. J. and Martin-Mayor, V. Field Theory, the Renormalization Group, and Critical Phenomena 3rd edn [Amit and Martin-Mayor, 2005].

We thank the referee for clearly outlining his view of the matter, including the core elements of the RG in general. We believe, at this point, we better understand the referee's main concern. Also, we are glad to agree with the referee to a great extent.

We would like to mention that the term RG has been used in an increasingly broad way in recent years, and some models that may not strictly adhere to the original definition. However, we fully agree with the referee's view on the precision of terminology.

Therefore, instead of using the term RG, we describe our method as a coarse-graining procedure that preserves the macroscopic flow properties. Consequently, we change the title of our paper to **Reducing the dimension of network-driven processes using generalized thermodynamics**.

We thank the referee once again for this crucial comment. We definitely like to adhere to the most precise terminology. Also, this new way of naming our approach can eliminate some confusion we discussed in our response to the referee's previous comment. In short, what we provide is a different approach to solve a different problem. To avoid confusion, we add this paragraph to the main text:

> It is worth remarking that our method serves an ultimately different purpose than other available techniques. Current methods designed for network renormalization group tend to uncover the scaling behavior of systems. For instance, in such frameworks, a scale-free network should keep its main properties through coarse-graining, while systems with characteristic scales are expected to show varying behavior through reduction. In contrast, our method is not directly concerned with revealing the existence of scales in a network. Rather, it tends to find a coarse-graining that preserves the macroscopic flow properties regardless of the network type and topological scales. While the two approaches might lead to similar in some cases, as we will show, their aims are distinct and we will show that important differences are manifest.
>
> To further stress this important point, we compare the results of our approach against widely used methods, including Box-Covering, Geometric Renormalization, and Laplacian Renormalization, to show that they can not fulfill our aim (See Supplementary Materials for details).

2) "LRG performs well on heterogeneous networks, especially on BA networks and

scale-free trees. Therefore, LRG successfully delivers what it promises— preserving the flow of information in those cases"

It is not true. BA or random trees are specific examples of some networks that do not change their intrinsic properties when a network rescaling is performed. The LRG works for every network, leading to scale-dependent properties in such networks presenting characteristic scales.

We thank you for highlighting the wide applicability of LRG. We acknowledge that LRG can renormalize any network at any scale. However, we would also like to point out the distinct contributions of our method from a different perspective.

LRG can be applied to network type and at any selected scale τ. For networks with scale-invariant intrinsic properties, LRG preserves the properties, reflecting the structural scale-invariance. For network structures lacking scale-invariant properties, LRG can identify their characteristic scales, making significant contributions both theoretically and practically. However, LRG is not designed to fulfill the purpose of our work and, therefore, fails to do so, as we have extensively shown at this point, experimentally.

On the other hand, our approach aims to preserve the flow properties of any network type (even those that are not scale-free) across the scales, and it fulfills the task, outperforming other methods, including LRG, as we have shown in our figures.

While the problem we attack is different, we compare our results against other methods to show the current gap and the need for our approach, in a distinct and complementary perspective. We think that this should be a common scientific practice which, instead, is not widespread as it should.

3) Figure 5. Barabasi network compression.

I have some specific comments about the figure the authors have presented in the

rebuttal letter. In particular, with BA networks fixing the parameter m=1, we know these networks are trees. The final network does not maintain the intrinsic properties of the original one. I would like to see, for instance, some plots showing the diameter for a growing process and a reduction process of those networks.

We thank the referee for this comment. We believe that, again, this concern can be answered by highlighting the difference between what is expected from an RG algorithm—e.g., preserving topological properties in cases like the one mentioned by the referee—and what our method is designed to do, as a coarse-graining that gives compressions preserving the flow at all scales simultaneously, regardless of the network type.

Please note that our method's output is a weighted network with self-loops. Therefore, assessing the topological preservation in our coarse-graining wouldn't be straightforward.

Surprisingly, however, our method can retain the scale-free nature of the network's degree distribution, as shown in Fig 2. This may be because the scale-free nature of the degree distribution is key to preserving the information flow in BA networks.

We thank the referee once again for this comment.

4) How we theoretically differ from and operationally outperform LRG

I have some significant comments regarding this specific new paragraph. The authors ensure that: "We base our method on the fact that the partition function contains the full information of the macroscopic descriptors of the flow, including network entropy and free energy— as demonstrated in the previous sections. Therefore, by replicating the full partition function profile at all propagation scales (all values of τ), we replicate the macroscopic features like the diversity of diffusion pathways and the flow speed in reaching distant parts of the network." This is right what the LRG does. So, this paragraph must be removed from the paper or completely rewritten.

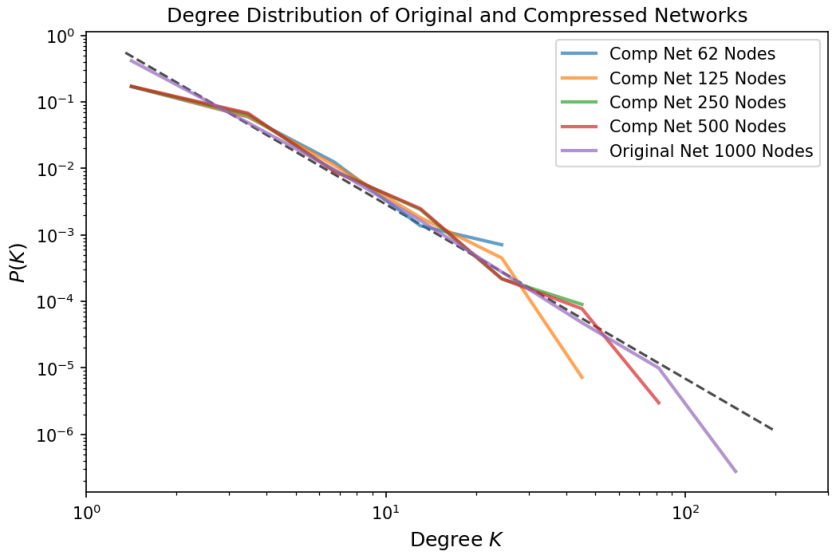

Figure 2: **Degree distribution of the original and compressed networks:** This plot displays the degree distribution for a BA network with 1000 nodes (m=1) compressed to various sizes (from 1/2 to 1/16 of the original network). X-axis: Node degree, Y-axis: Frequency of degrees. As can be seen, the degree distribution remains the same before and after compression.

It is important to note that the LRG is not limited to heterogeneous networks like Barabasi-Albert (BA) and scale-free trees, as the authors suggest. In fact, the LRG is designed to work with any general network or topology, including those found in real-world scenarios.

"Instead, our method preserves the complete distribution of eigenvalues at all scales simultaneously by maintaining the full behavior of the partition function." As I have explained before, general RG procedures MUST rescale the distribution of eigenvalues.

We thank the referee for this comment. We think that, again, this concern stems from the first one discussed above.

Here, we have two conflicting statements that can not be true simultaneously.

Either the referee is right and LRG truly "replicate(s) the macroscopic features like the diversity of diffusion pathways and the flow speed," in all network types at all scales

simultaneously. Or, we are correct, and LRG fails to do so for many network types.

We have provided several numerical experiments to demonstrate that we are correct (for instance, see Fig. 3 for the ability to preserve eigenvalue distribution), in our first rebuttal and also in our manuscript, and here we add an additional experiment on entropy preservation (see Fig 1). We would be more than glad to share our code with the referee, if he/she does not trust our results, as he/she has every right to.

Our final result is simply that the two methods are different since they are designed to solve different problems. And, of course, they outperform each other in their own grounds, as it should be.

Due to this different theoretical aim, our method also differs in practical outcomes from LRG: our model can not identify characteristic scales of network structures like LRG does. However, preserving the eigenvalue distribution allows our model to ensure consistency in the network diffusion behavior before and after compression.

Furthermore, the use of a machine learning pre-trained model provides computational advantages to our approach, allowing it to be trained on smaller networks and applied to larger ones.

Minor comments:

I still think the authors do not use at all the free energy nor it is useful for the specific statements they treat here. I recommend to move it to a methods section of the manuscript or tu Supplementary Information.

We thank the reviewer for this comment. Following other comments we recieved from the reviewer, we change that theoretical section with the transformations we discussed in our first comment. Also, we add the entropy case to support the validity of the theoretical transformations. We think these theoretical derivations lay the theoretical ground of our

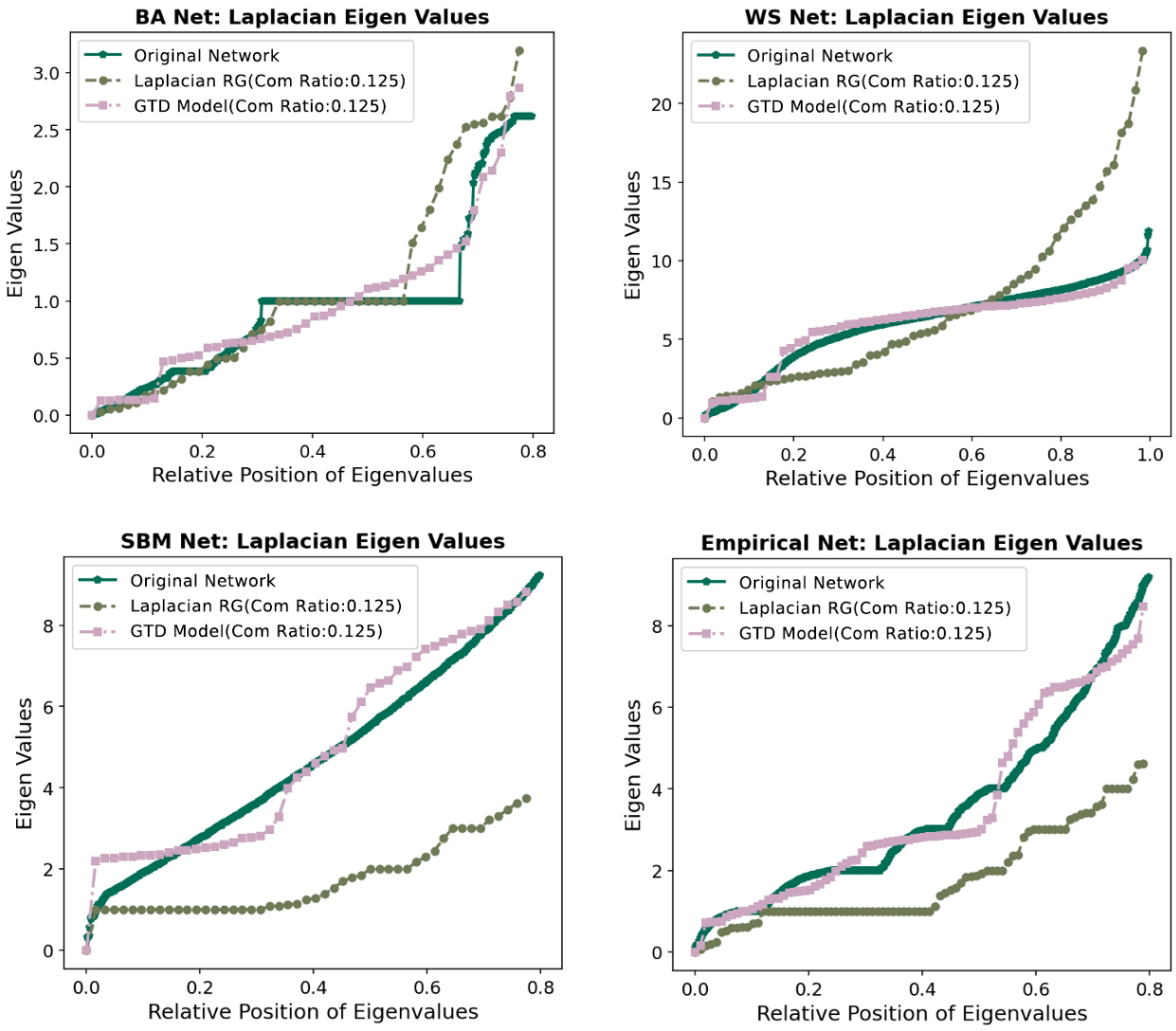

Figure 3: **Eigenvalue distribution before and after compression:** This figure shows the eigenvalue distributions of four typical artificial networks compressed using our method and LRG. The horizontal axis represents the relative ordering of eigenvalues, and the vertical axis represents the eigenvalues themselves. As it can be seen, our method better preserves the overall distribution of eigenvalues.

work.

We thank the referee once again for their brilliant comments and the time and effort spent to improve our work.

end of response to referee #1

Referee #2 (Remarks to the Author):

I am glad to see that the authors had a positive attitude against the comments outlined in my report. My opinion is that the author successfully answered to criticisms raised and modified the paper accordingly. I am in favour of publication of paper in the present form

Thank you for your encouraging feedback and support for the publication of our paper. We greatly appreciate your positive remarks and are glad that our revisions have satisfactorily addressed the criticisms raised. Your constructive comments were instrumental in enhancing the quality and clarity of our work. Thank you once again for your thoughtful and detailed review.

end of response to referee #2

Referee #3 (Remarks to the Author):

We thank the referee for the thoughtful feedback on our revised manuscript that undoubtedly have enhanced the quality and presentation of our work. Before proving a point by point answer, we summarize our main responses as follows:

Firstly, we thank the reviewer for helping us refine our key points and suggesting we highlight them in the main text. Additionally, following the referee's comment, we have found that the degree distribution's scale-free nature can be preserved in scale-free networks. This is surprising because our method is not designed to preserve topological properties.

Furthermore, we have provided a more elaborate discussion of the complexity of time complexity with numerical experiments to support the conclusions we made in our previous rebuttal.

In the following we provide a point by point response to the referee's comments.

The authors have addressed most of my comments, and I thank them for the thorough answers. The work has been significantly improved. However, many of these improvements are not reflected in the main text, and there are still some issues that need to be addressed.

In the experiments designed to answer some of my previous comments, the authors found:

"Regardless of the network's size or the field it belongs to, its transitivity (the fraction of all possible triangles present in the network) has a major impact on compression efficiency. As shown in the right subplot of Figure 8, we plotted the transitivity against the error in the partition function before and after compression for all networks and observed a positive correlation (correlation coefficient of 0.676): a network with lower transitivity

tends to be more easily compressed."

This effect is important, and I urge the authors to describe this result and the corresponding figure in the main text, along with the explanations given in the rebuttal:

"as the clustering coefficient decreases and long-range connections increase, global information diffusion becomes easier. Therefore, reducing the clustering coefficient makes large-scale information diffusion easier to complete. This is reflected in the partition function, meaning it approaches zero faster at large scales", and "This means that the model performs better on networks with lower transitivity. This is likely because high transitivity increases the complexity of information diffusion dynamics. Since the transitivity of a tree is 0, this explains why the compression effect on trees differs from that on other synthetic networks."

We thank the reviewer for highlighting this key point, which is indeed crucial for understanding the nature and applicable scenarios of our model. To better address this question, we systematically analyzed the impact of more structural variables on compression effectiveness. We found that networks with higher density are more challenging to preserve the properties of the partition function after compression.

Before that, we would like to highlight the fact that our work offers two similar but slightly different ways of optimization tending to replicate the partition function. Assuming N is the number of nodes, the upper bound for the partition function of any network is $Z = N$ that happens at $\tau = 0$ and the lower bound is $Z = 1$ happening at $\tau = \infty$. To replicate the partition function curve while reducing size, we must perform size adjustments. To this aim, we have explored two different optimization targets: $\frac{Z}{N} = \frac{Z'}{N'}$ and $\frac{Z-1}{N-1} = \frac{Z'-1}{N'-1}$. We implemented both methods and analyzed their differences from theoretical and practical perspectives. The first target allows us to replicate macroscopic properties like entropy and free energy, after size adjustment. For instance, entropy curve

of the reduced system indicated by S' is recovered as $S' = S - log(N/N')$. In our revised version, we theoretically derive these size adjusted replications and provide numerical experiments that they work. On the other hand, for the second optimization target, the method does not preserve the curves like entropy or free energy, yet it succeeds to preserve the distribution of the original Laplacian's eigenvalues with greater accuracy. We would like to mention that the first approach still outperforms others in preserving the eigenvalue distribution.

The previous results were obtained using the second optimization target that we moved to the supplementary, since the first target achieves much higher accuracy for our goals—i.e., replicating the macroscopic functions. Therefore, from here on, we focus on the first optimization target and its results.

The correlation coefficient between density and partition function error is 0.856(see Fig 1), higher than the 0.676 coefficient for transitivity we highlighted in our last response, thus providing a better explanation. We believe this is expected, because an increase in the number of connections makes the pathways for information transmission more abundant and, at the same time, it increases the difficulty of compression.

We thank the reviewer for this important comment and put this figure and the corresponding explanation in the main text, as the reviewer suggested.

My question about the behavior of network topology in the flow remains unresolved. Instead of directly addressing the question, the authors demonstrated the impact of different network topological properties on the partition function in Figure 12 of the rebuttal. While this is interesting, it does not answer my original question. The explanation provided on page 63 of the rebuttal states:

"it's important to note that although many topological properties can influence the partition function, our renormalization method can preserve the partition function with

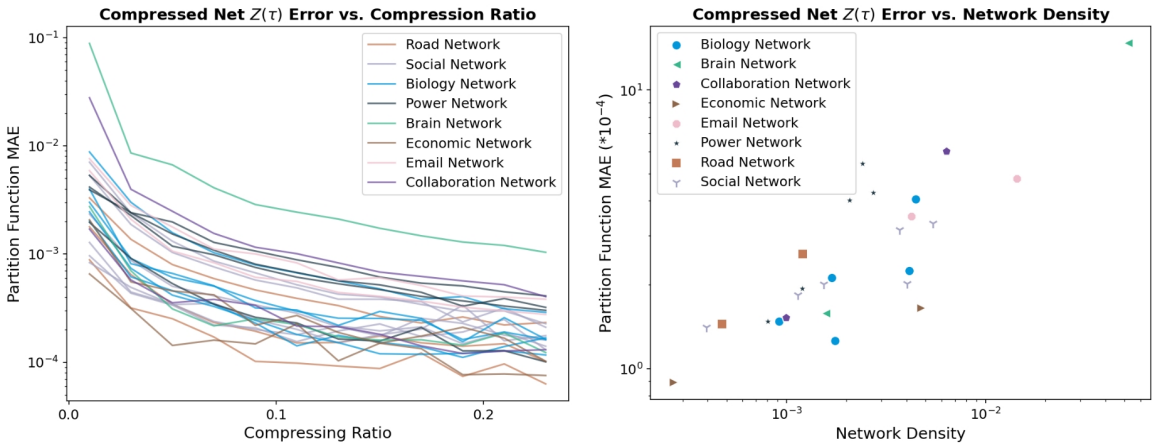

Figure 1: **Compressing empirical networks from different domains:** In the left subplot, we show the pre- and post-compression partition function error (measured by MAE) for 25 networks from 7 different domains at various compression ratios, ranging from 1% to 25%. In the right subplot, we display the relationship between the average compression error and the network's density when the compression ratio is greater than 0.1 (at this point, the compression error is not quite sensitive to the compression size, which can be considered as the best compression error achievable by the model without being limited by size).

high accuracy. This does not mean that our renormalization method will necessarily preserve these structural properties. This is fundamentally because our model produces a graph with weights and self-loops. The addition of weights and self-loops allows the model to maintain the information flow without using some of the structural features (for example, by controlling the amount of information that stays at the node itself through weighted self-loops). Therefore, we did not directly compare the structural properties of networks before and after compression."

This explanation is important and relevant for clarifying the limitations of the proposed methodology and should be included in the main text.

We thank the reviewer for this comment. We add the discussion of topology to the main text.

Also, as discussed previously, our compressions are weighted graphs that may in-

clude self-loops. Therefore, it is challenging to compare their topology directly against the original network or any other type of binary network.

Yet, following referee's comment, we found that, for instance, if the original network has a high clustering coefficient trapping information locally, the compressed model often achieves the same effect by generating more self-loops.

One the other hand, for simple structural properties like the degree distribution, we obtained new results that were surprising to us. When compressing BA networks, we found that the degree distribution is preserved (as shown in Fig 2). We believe this is because scale-free networks have strong heterogeneity, and adjacent nodes generally do not cluster together, thus the compressed network does not form many self-loops. This is surprising because our approach is not focused on preserving structural properties: we have to deduce that if scale-freeness is preserved then it is functional to preserve the information flow.

We thank the referee once again for this important comment and add the required and additional finding to the main text.

Another significant point clarified in the rebuttal letter but not fully addressed in the manuscript concerns the complexity of the method. In the abstract, the authors state, "Our work offers a low-complexity renormalization method", which is not entirely accurate. As clarified in the rebuttal: "the most time-consuming part is calculating the partition function of the original network to serve as supervisory information, which has a time complexity of $O(N^3)$. Therefore, calculating the partition function for extremely large networks is almost impossible, even if the machine learning renormalization model can be trained on smaller network datasets."

These clarifications make the section "One Renormalization Model for Many Real-world Networks" specially relevant. In this section, the authors assert: "The results

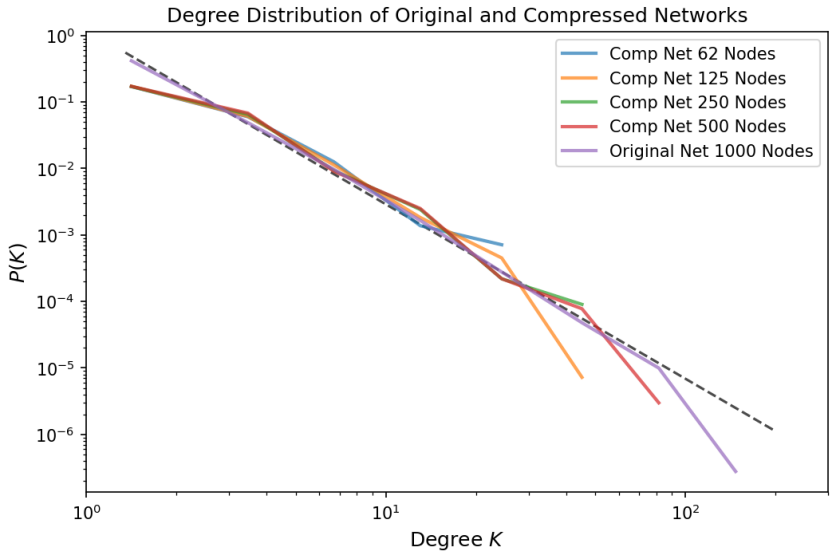

Figure 2: **Degree distribution of the original and compressed networks:** This plot displays the degree distribution for a BA network with 1000 nodes (m=1) compressed to various sizes (from 1/2 to 1/16 of the original network). X-axis: Node degree, Y-axis: Frequency of degrees. As can be seen, the degree distribution remains the same before and after compression.

demonstrate highly effective learning that can successfully compress the test set, generating macro-networks that preserve the flow of information". This statement should be accompanied by quantitative analyses.

We thank the referee for this important comment.

In the previous version, we only conducted a theoretical analysis of time complexity without providing concrete numerical evidence. In this revision, we have added a quantitative experiment to demonstrate that the time complexity of a pre-trained model operating on different networks is $O(N + E)$, where N denotes the number of nodes and E denotes the number of edges. Although the data sizes used in section "One Renormalization Model for Many Real-world Networks" varied (from 100 to 2495 nodes), they did not span multiple orders of magnitude, which is insufficient for quantifying the time complexity of the model. To address this, we conducted a quantitative analysis on a broader

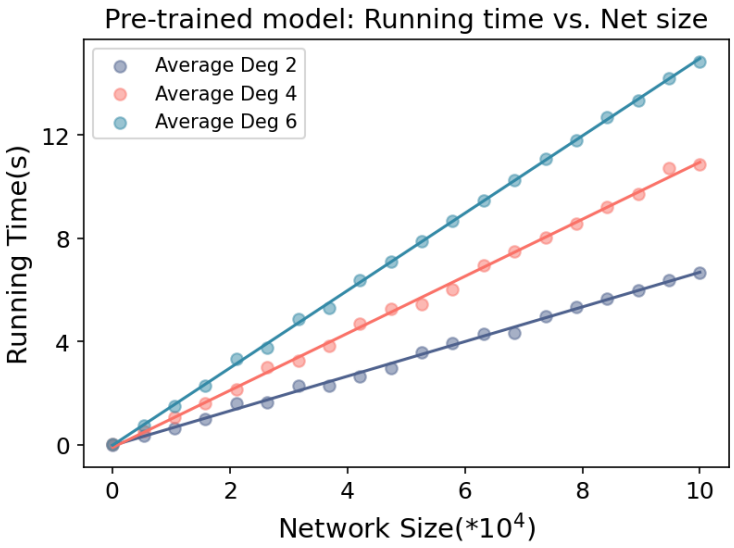

Figure 3: **Pre-Trained Mode: Running Time vs. Network Size:** In this figure, we show the time required for a pre-trained model to run on W-S networks with different numbers of nodes and connections. The X-axis represents the number of nodes, and the Y-axis shows the time needed to compress the network.

dataset, using networks with node counts ranging from 100 to 100,000. Specifically, we ran a trained model on synthetic W-S networks of varying sizes and average degrees, compressing them to 50 nodes. We found that the model's running time increases proportionally with network size. Additionally, networks of the same size require different amounts of time due to varying numbers of connections. The results are shown in Fig 3.

We thank the reviewer once again and add the new finding to the main text.

Minor:

My previous criticism regarding the use of the term ”transition point” applies to its substitution with ”critical size”.

We thank the reviewer for this point. We fully agree. Our error transition points appeared in the second optimization target (explained previously) and now are in the supplementary materials. We have changed the name of that point to “error threshold”.

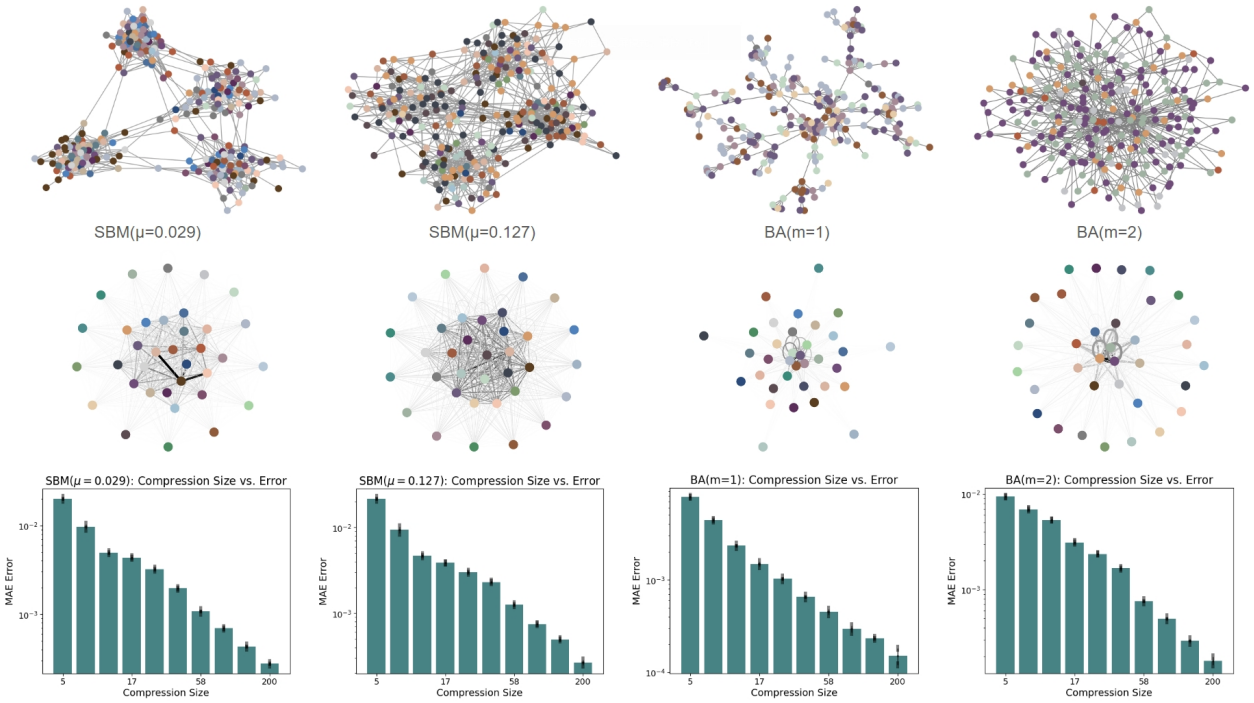

Figure 4: **Compression results for synthetic networks:** The first row shows visualizations of two Stochastic Block Model (SBM) networks (mixing parameters μ of 0.029 and 0.127, respectively) and Barabasi-Albert (BA) networks (numbers of links per node m of 1 and2) all having 256 nodes. The second row visualizes the structure of the macro network that was compressed to 35 nodes. The third row shows the Mean Absolute Error in the partition function before and after compression at different sizes. We selected 10 different sizes uniformly in the logarithmic space between 5 and 200.

On the other hand, the error bars related to the first optimization target follow a pattern similar to power-law with no characteristic scale or transition point. This result is presented in the main text.

We thank the reviewer for this comment once again.

On page 9 of the manuscript, the placement of punctuation marks in the sentence. "other models that often perform at $O(N^3)$–including box-covering, Laplacian RG, or $O(N^2)$– for instance, geometric RG" is incorrect. It should be revised as: "other models that often perform at $O(N^3)$–including box-covering and Laplacian RG–, or $O(N^2)$, for instance, geometric RG."

Thank you for pointing out this issue and suggesting how to address it. We have made the modifications in the revised version of the paper.

Reviewer #3 (Remarks on code availability):

The code is available at the address provided by the authors.

Again, we thank the referee for the meticulous and constructive comments and the time and effort spent reviewing our work.

References

[Amit and Martin-Mayor, 2005] Amit, D. J. and Martin-Mayor, V. (2005). Field theory, the renormalization group, and critical phenomena: graphs to computers. World Scientific Publishing Company.

[Kardar, 2007] Kardar, M. (2007). Statistical physics of fields. Cambridge University Press.

Letter for the Referee:

Detailed Response

Referee #3 (Remarks to the Author):

We would like to sincerely thank the reviewer for the valuable feedback and positive evaluation of our manuscript. The comments provided have greatly enhanced the quality and clarity of the paper. In order to quickly highlight the key points, we have summarized the main changes we made in response to the review.

This revised manuscript addresses the following major points:

- **Title Update:** Based on the reviewer's suggestion, we changed the title to "Coarse-graining network flow through statistical physics and machine learning" to better reflect the focus of our work on network flow and coarse-graining rather than dimension reduction.
- **Clarification of Coarsening Method:** We clarified the novelty of our approach in the coarsening process. While coarsening nodes with similar local structures is not new, our method's innovation lies in how machine learning automatically learns these similarity features, making the process more adaptive and effective.
- **Comparison with Existing Renormalization Methods:** We revised our discussion of traditional network renormalization methods to emphasize the distinction between our approach, which focuses on preserving macroscopic flow properties, and methods that aim to preserve topological properties. We also updated Figure 2 to provide a clearer comparison of the results.

We hope the changes made in the manuscript effectively address the reviewer's concerns. Below, we provide a detailed response to each comment.

I recommend the publication of the paper, provided the authors address some final remarks:

We thank the referee for his positive assessment of the work. Our manuscript has been greatly improved because of the previous and current comments. Accordingly, we take this chance to sincerely thank the referee once again.We are confident that the revised manuscript addresses the raised issues.

The new title, "Reducing the Dimension of Network-Driven Processes Using Generalized Thermodynamics", is still not very appropriate. Dimension is not a representative word for this text, neither is dimensionality. How are dimension or dimensionality defined in this paper? I understand that this terminology is used as in computer science, but network compression or coarse-graining could be more suitable instead reducing network dimension. In addition, not all network-driven processes are tractable with the proposed method, which focuses on diffusion. And the way thermodynamics is generalized here is quite specific, using machine learning. I suggest that the authors find a more appropriate title.

We agree with the referee that coarse-graining is a more precise term to describe what we do. Therefore, we change the title to "Coarse-graining network flow through statistical physics and machine learning." We would like to mention that while we are focused on diffusion dynamics implementation, the framework is open to many other dynamical processes such as random walks, epidemics, and neural, thanks to the recent generalization of network density matrices[1], However, we emphasize our focus on diffusion in the abstract and use the term network flow in the title to avoid confusion. We thank the reviewer once again for helping us find a more representative title.

In page 4, the text "Note that many network renormalization methods coarsen subnetworks into supernodes in a (compresssed) macro network. In contrast, our machine

learning model has automatically learned a novel renormalization strategy: it coarsens nodes with similar local structures into a supernode." is not clear. The idea of coarsening nodes (independently of whether they are connected or not) with similar local structures into a supernode is not novel. The existing methods coarsen nodes with similar local structures into supernodes. At this level, the present work belongs to the same category and there is nothing novel in this respect. The methods differ in the precise way they define what is a similar local structure. The present work introduces an alternative way to define what is a similar local structure.

We thank the reviewer for spotting this imprecision. Of course, we acknowledge that the coarsening of non-connected nodes with similar local structures into supernodes is not inherently novel— e.g., see methods focused on geometric embedding. We revised the text to provide a more precise description of our contribution. Specifically, we have clarified that our work introduces a new way to define similar local structures based on the learned features. The updated text reads:

"While most coarse-graning methods define and identify supernodes by grouping nodes with similar local structures, their notion of similarity is typically predefined[2, 3]. In contrast, our approach leverages a machine learning model to automatically learn the vital similarity features that can be used to compress the network while preserving the macroscopic flow properties."

The following text in page 5: "In those frameworks, for instance, a scale-free system is desired to keep its main properties through coarse-graining, while systems with characteristic scales are expected to show varying behavior through reduction. In contrast, our approach is not concerned with revealing the existence of scales in the system. Rather, it tends to find a coarse-graining that preserves the macroscopic flow properties regardless of the network type and topological scales. However, we compare the results of our work against these widely used methods, including Box-Covering, Geometric Renormal-

ization, and Laplacian Renormalization, to show that they can not fulfill our aim (See Supplementary Materials for details).” is confusing and unclear. For instance, ”a scale-free system is desired to keep its main properties through coarse-graining”, which properties and under which renormalization scheme? I do not think that the results presented in this manuscript support the claim that the mentioned methods cannot preserve the macroscopic flow properties of networks.

We thank the referee for this comment. To better clarify the main point, we updated Figure 2 in main text with comparison results and rewrote that paragraph as follows:

“Our work serves a purpose that is ultimately different from the network renormalization group methods. In fact, these methods aim to uncover the scaling behavior of systems. For instance, renormalization group methods preserve specific topological properties when compressing networks, like the degree distribution of scale-free networks. However, in other cases, for instance, in network with one or multiple characteristic topological scales, these methods are expected to capture the changes in network properties, including the flow, as it gets compressed. In contrast, our approach seeks to find a coarse-graining that preserves the flow properties, regardless of the specific features of a topology. Therefore, it is expected to outperform others in preserving the flow, when compared across distinct network families. We confirm this expectation through a series of numerical experiments demonstrated in Figure 2.”

The updated figure is shown below:

We thank the reviewer for this comment that help better clarifying our method’s distinction.

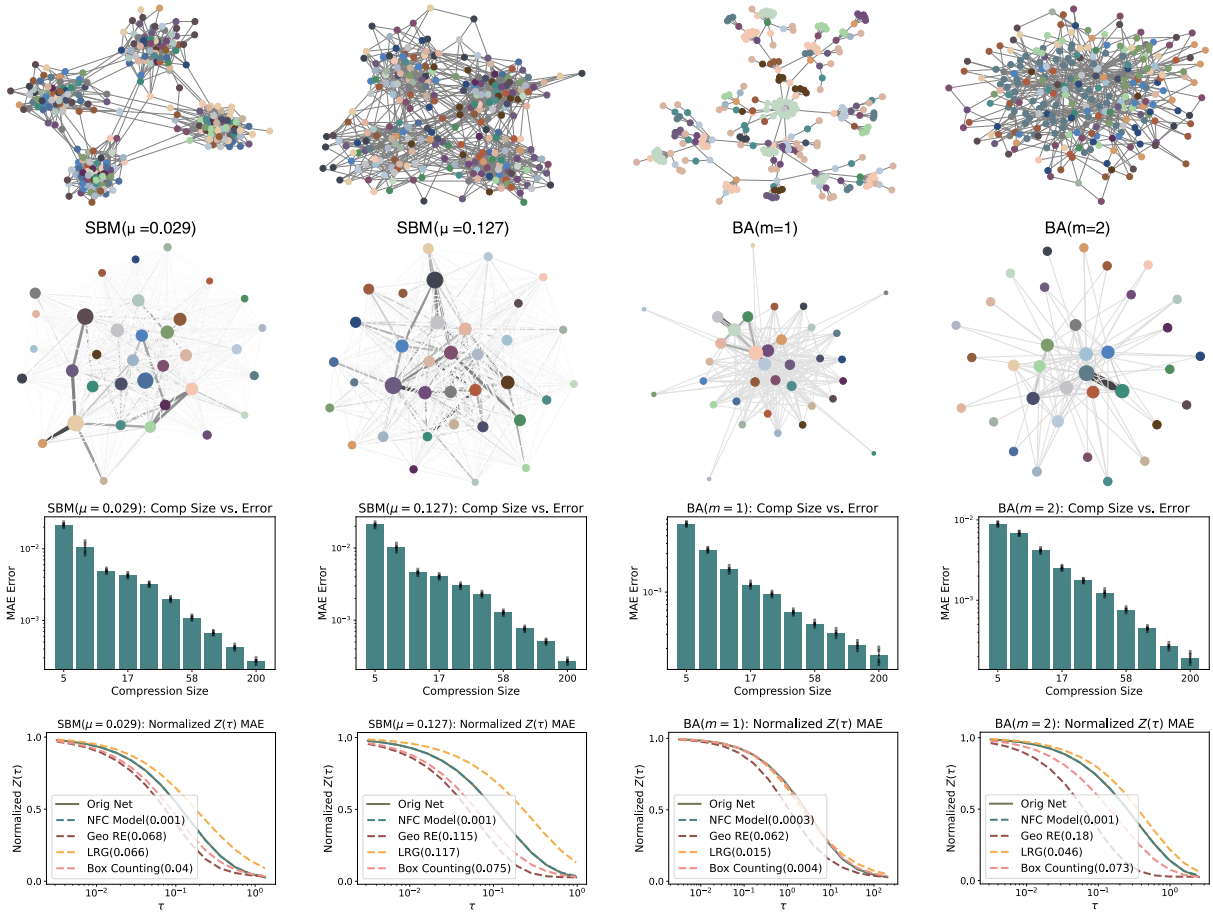

Figure 1: **Compression for Synthetic Networks** Visual representation of the compression process applied to two representative Stochastic Block Model (SBM) networks (mixing parameters μ of 0.029 and 0.127, respectively) and Barabasi-Albert (BA) networks (numbers of links per node m of 1 and 2). The second row visualizes the structure of the macro network that was compressed to 35 nodes. The third row shows the Mean Absolute Error in the partition function before and after compression at different sizes. We selected 10 different sizes uniformly in the logarithmic space between 5 and 200. The last subplot row shows the Mean Absolute Error of the partition function curve for different methods when compressing the network to 35 nodes.

References

[1] Ghavasieh, A. & De Domenico, M. Generalized network density matrices for analysis of multiscale functional diversity. Physical Review E **107**, 044304 (2023).

[2] Zheng, M., García-Pérez, G., Boguñá, M. & Serrano, M. Á. Geometric renormalization of weighted networks. Communications Physics **7**, 97 (2024).

[3] García-Pérez, G., Boguñá, M. & Serrano, M. Á. Multiscale unfolding of real networks by geometric renormalization. Nature Physics **14**, 583–589 (2018).