

Statistical Physics Analysis of Graph Neural Networks: Approaching Optimality in the Contextual Stochastic Block Model

O. Duranthon^{*} and L. Zdeborová[✉]

Statistical Physics of Computation Laboratory, École polytechnique fédérale de Lausanne, Lausanne, Switzerland



(Received 5 March 2025; accepted 16 September 2025; published 10 November 2025)

Graph neural networks (GNNs) are designed to process data associated with graphs. They are finding an increasing range of applications; however, as with other modern machine learning techniques, their theoretical understanding is limited. GNNs can encounter difficulties in gathering information from nodes that are far apart by iterated aggregation steps. This situation is partly caused by so-called oversmoothing; and overcoming it is one of the practically motivated challenges. We consider the situation where information is aggregated by multiple steps of convolution, leading to graph convolutional networks (GCNs). We analyze the generalization performance of a basic GCN, trained for node classification on data generated by the contextual stochastic block model. We predict its asymptotic performance by deriving the free energy of the problem, using the replica method, in the high-dimensional limit. Calling *depth* the number of convolutional steps, we show the importance of going to large depth to approach the Bayes optimality. We detail how the architecture of the GCN has to scale with the depth to avoid oversmoothing. The resulting large depth limit can be close to the Bayes optimality and leads to a continuous GCN. Technically, we tackle this continuous limit via an approach that resembles dynamical mean-field theory with constraints at the initial and final times. An expansion around large regularization allows us to solve the corresponding equations for the performance of the deep GCN. This promising tool may contribute to the analysis of further deep neural networks.

DOI: [10.1103/PhysRevX.15.041026](https://doi.org/10.1103/PhysRevX.15.041026)

Subject Areas: Complex Systems,
Computational Physics,
Statistical Physics

I. INTRODUCTION

A. Summary of the narrative

Graph neural networks (GNNs) emerged as the leading paradigm when learning from data that are associated with a graph or a network. Given the ubiquity of such data in sciences and technology, GNNs are gaining importance in their range of applications, including chemistry [1], biomedicine [2], neuroscience [3], simulating physical systems [4], particle physics [5] and solving combinatorial problems [6,7]. As common in modern machine learning, the theoretical understanding of learning with GNNs is lagging behind their empirical success. In the context of GNNs, one pressing question concerns their ability to aggregate information from far away parts of the graph: the performance of GNNs often deteriorates as depth increases [8]. This issue is

often attributed to oversmoothing [9,10], a situation where a multilayer GNN averages out the relevant information. Consequently, mostly relatively shallow GNNs are used in practice or other strategies are designed to avoid oversmoothing [11,12].

Understanding the generalization properties of GNNs on unseen examples is a path towards yet more powerful models. Existing theoretical works addressed the generalization ability of GNNs mainly by deriving generalization bounds, with a minimal set of assumptions on the architecture and on the data, relying on VC dimension, Rademacher complexity or a PAC-Bayesian analysis; see, for instance, [13] and the references therein. Works along these lines that considered settings related to one of this work include [14,15], or [16]. However, they only derive loose bounds for the test performance of the GNN and they do not provide insights on the effect of the structure of data. Reference [14] provides sharper bounds; yet they do not take into account the data structure and depend on continuity constants that cannot be determined *a priori*. In order to provide more actionable outcomes, the interplay between the architecture of the GNN, the training algorithm, and the data needs to be understood better, ideally

^{*}Contact author: firstname.secondname@epfl.ch

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

including constant factors characterizing their dependencies on the variety of parameters.

Statistical physics traditionally plays a key role in understanding the behavior of complex dynamical systems in the presence of disorder. In the context of neural networks, the dynamics refers to the training, and the disorder refers to the data used for learning. In the case of GNNs, the data are related to a graph. The statistical physics research strategy defines models that are simplified and allow analytical treatment. One models both the data generative process, and the learning procedure. A key ingredient is a properly defined thermodynamic limit in which quantities of interest self-average. One then aims to derive a closed set of equations for the quantities of interest, akin to obtaining exact expressions for free energies from which physical quantities can be derived. While numerous other research strategies are followed in other theoretical works on GNNs, see above, the statistical physics strategy is the main one accounting for constant factors in the generalization performance and as such provides invaluable insight about the properties of the studied systems. This line of research has been very fruitful in the context of fully connected feed-forward neural networks, see, e.g., [17–19]. It is reasonable to expect that also in the context of GNNs this strategy will provide new actionable insights.

The analysis of generalization of GNNs in the framework of the statistical physics strategy was initiated recently in [20] where the authors studied the performance of a single-layer graph convolutional neural network (GCN) applied to data coming from the so-called contextual stochastic block model (CSBM). The CSBM, introduced in [21,22], is particularly suited as a prototypical generative model for graph-structured data, where each node belongs to one of several groups and is associated with a vector of attributes. The task is then the classification of the nodes into groups. Such data are used by practitioners as a benchmark for performance of GNNs [15,23–25]. On the theoretical side, the follow-up work [26] generalized the analysis of [20] to a broader class of loss functions but also alerted to the relatively large gap between the performance of a single-layer GCN and the Bayes-optimal performance.

In this paper, we show that the close-formed analysis of training a GCN on data coming from the CSBM can be extended to networks performing multiple layers of convolutions. With a properly tuned regularization and strength of the residual connection this allows us to approach the Bayes-optimal performance very closely. Our analysis sheds light on the interplay between the different parameters—mainly the depth, the strength of the residual connection and the regularization—and on how to select the values of the parameters to mitigate oversmoothing. On a technical level the analysis relies on the replica method, with the limit of large depth leading to a continuous formulation similar to neural ordinary differential equations [27] that can be treated analytically via an approach that resembles

dynamical mean-field theory with the position in the network playing the role of time. We anticipate that this type of infinite depth analysis can be generalized to studies of other deep networks with residual connections such a residual networks or multilayer attention networks.

B. Further motivations and related work

1. Graph neural networks

In this work we focus on graph neural networks. GNNs are neural networks designed to work on data that can be represented as graphs, such as molecules, knowledge graphs extracted from encyclopedias, interactions among proteins or social networks. GNNs can predict properties at the level of nodes, edges, or the whole graph. Given a graph $\mathcal{G}$ over N nodes, its adjacency matrix $A \in \mathbb{R}^{N \times N}$ and initial features $h_i^{(0)} \in \mathbb{R}^M$ on each node i , a GNN can be expressed as the mapping

$$h_i^{(k+1)} = f_{\theta^{(k)}}(h_i^{(k)}, \text{agg}(\{h_j^{(k)}, j \sim i\})) \quad (1)$$

for $k = 0, \dots, K$ with K being the depth of the network. where $f_{\theta^{(k)}}$ is a learnable function of parameters $\theta^{(k)}$, and $\text{agg}()$ is a function that aggregates the features of the neighboring nodes in a permutation-invariant way. A common choice is the sum function, akin to a convolution on the graph

$$\text{agg}(\{h_j, j \sim i\}) = \sum_{j \sim i} h_j = (Ah)_i. \quad (2)$$

Given this choice of aggregation the GNN is called graph convolutional network [28]. For a GNN of depth K the transformed features $h^{(K)} \in \mathbb{R}^{M'}$ can be used to predict the properties of the nodes, the edges or the graph by a learnt projection.

In this work we will consider a GCN with the following architecture, that we will define more precisely in the detailed setting Sec. II. We consider one trainable layer $w \in \mathbb{R}^M$, since dealing with multiple layers of learned weights is still a major issue [29], and since we want to focus on modeling the impact of numerous convolution steps on the generalization ability of the GCN.

$$h^{(k+1)} = \left(\frac{1}{\sqrt{N}} \tilde{A} + c_k I_N \right) h^{(k)}, \quad (3)$$

$$\hat{y} = \text{sign} \left(\frac{1}{\sqrt{N}} w^T h^{(K)} \right), \quad (4)$$

where $\tilde{A}$ is a rescaling of the adjacency matrix, I_N is the identity, $c_k \in \mathbb{R}$ for all k are the residual connection strengths and $\hat{y} \in \mathbb{R}^N$ are the predicted labels of each node. We will call the number of layers K the depth, but we reiterate that only the layer w is learned.

2. Analyzable model of synthetic data

Modeling the training data is a starting point to derive sharp predictions. A popular model of attributed graph, that we will consider in the present work and define in detail in Sec. II A, is the contextual stochastic block model (CSBM), introduced in [21,22]. It consists in N nodes with labels $y \in \{-1, +1\}^N$, in a binary stochastic block model (SBM) to model the adjacency matrix $A \in \mathbb{R}^{N \times N}$ and in features (or attributes) $X \in \mathbb{R}^{N \times M}$ defined on the nodes and drawn according to a Gaussian mixture. y has to be recovered given A and X . The inference is done in a semisupervised way, in the sense that one also has access to a train subset of y .

A key aspect in statistical physics is the thermodynamic limit, how should N and M scale together. In statistical physics we always aim at a scaling in which quantities of interest concentrate around deterministic values, and the performance of the system ranges between as bad as random guessing to as good as perfect learning. As we will see, these two requirements are satisfied in the high-dimensional limit $N \rightarrow \infty$ and $M \rightarrow \infty$ with $\alpha = N/M$ of order 1. This scaling limit also aligns well with the common graph datasets that are of interest in practice, for instance, Cora [30] ($N = 3.10^3$ and $M = 3.10^3$), Coauthor CS [31] ($N = 2.10^4$ and $M = 7.10^3$), CiteSeer [32] ($N = 4.10^3$ and $M = 3.10^3$), and PubMed [33] ($N = 2.10^4$ and $M = 5.10^2$).

A series of works that builds on the CSBM with lower dimensionality of features that is $M = o(N)$ exists. Authors of [34] consider a one-layer GNN trained on the CSBM by logistic regression and derive bounds for the test loss; however, they analyze its generalization ability on new graphs that are independent of the train graph and do not give exact predictions. In [35] they propose an architecture of GNN that is optimal on the CSBM with low-dimensional features, among classifiers that process local treelike neighborhoods, and they derive its generalization error. In [36] the authors analyze the structure and the separability of the convolved data $\tilde{A}^K X$, for different rescalings $\tilde{A}$ of the adjacency matrix, and provide a bound on the classification error. Compared to our work these articles consider a low-dimensional setting ([35]) where the dimension of the features M is constant, or a setting where M is negligible compared to N ([34,36]).

3. Tight prediction on GNNs in the high-dimensional limit

Little has been done as to tightly predicting the performance of GNNs in the high-dimensional limit where both the size of the graph and the dimensionality of the features diverge proportionally. The only pioneering references in this direction we are aware of are [20,26], where the authors consider a simple single-layer GCN that performs only one step of convolution, $K = 1$, trained on the CSBM in a semisupervised setting. In these works the authors express

the performance of the trained network as a function of a finite set of order parameters following a system of self-consistent equations.

There are two important motivations to extend these works and to consider GCNs with a higher depth K . First, the GNNs that are used in practice almost always perform several steps of aggregation, and a more realistic model should take this into account. Second, [26] shows that the GCN it considers is far from the Bayes-optimal (BO) performance and the Bayes-optimal rate for all common losses. The BO performance is the best that any algorithm can achieve knowing the distribution of the data, and the BO rate is the rate of convergence toward perfect inference when the signal strength of the graph grows to infinity. Such a gap is intriguing in the sense that previous works [37,38] show that a simple one-layer fully connected neural network can reach or be very close to the Bayes optimality on simple synthetic datasets, including Gaussian mixtures. A plausible explanation is that on the CSBM considering only one step of aggregation $K = 1$ is not enough to retrieve all information, and one has to aggregate information from further nodes. Consequently, even on this simple dataset, introducing depth and considering a GCN with several convolution layers, $K > 1$, is crucial.

In the present work we study the effect of the depth K of the convolution for the generalization ability of a simple GCN. A first part of our contribution consists in deriving the exact performance of a GCN performing several steps of convolution, trained on the CSBM, in the high-dimensional limit. We show that $K = 2$ is the minimal number of steps to reach the BO learning rate. As to the performance at moderate signal strength, it appears that, if the architecture is well tuned, going to larger and larger K increases the performance until it reaches a limit. This limit, if the adjacency matrix is symmetrized, can be close to the Bayes optimality. This is illustrated on Fig. 1, which highlights the importance of numerous convolution layers.

4. Oversmoothing and residual connections

Going to larger depth K is essential to obtain better performance. Yet, GNNs used in practice can be quite shallow, because of the several difficulties encountered at increasing depth, such that vanishing gradient, which is not specific to graph neural networks, or oversmoothing [9,10]. Oversmoothing refers to the fact that the GNN tends to act like a low-pass filter on the graph and to smooth the features h_i , which after too many steps may converge to the same vector for every node. A few steps of aggregation are beneficial but too many degrade the performance, as [40] shows for a simple GNN, close to the one we study, on a particular model. In the present work we show that the model we consider can suffer from oversmoothing at increasing K if its architecture is not well tuned and we precisely quantify it.

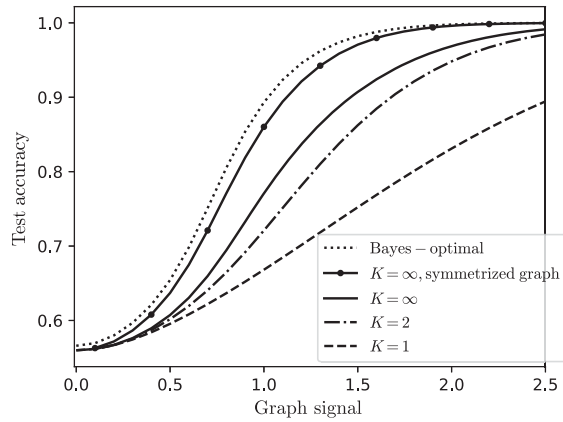


FIG. 1. Test accuracy of the graph neural network on data generated by the contextual stochastic block model vs the signal strength. We define the model and the network in Sec. II. The test accuracy is maximized over all the hyperparameters of the network. The Bayes-optimal performance is from [39]. The line $K = 1$ has been studied by [20,26]; we improve it to $K > 1$, $K = \infty$ and symmetrized graphs. All the curves are theoretical predictions we derive in this work.

A way to mitigate vanishing gradient and oversmoothing is to allow the nodes to remember their initial features $h_i^{(0)}$. This is done by adding residual (or skip) connections to the neural network, so the update function becomes

$$h_i^{(k+1)} = c_k h_i^{(k)} + f_{\theta^{(k)}}(h_i^{(k)}, \text{aggreg}(\{h_j^{(k)}, j \sim i\})), \quad (5)$$

where the c_k modulate the strength of the residual connections. The resulting architecture is known as residual network or resnet [41] in the context of fully connected and convolutional neural networks. As to GNNs, architectures with residual connections have been introduced in [42] and used in [11,12] to reach large numbers of layers with competitive accuracy. Reference [43] additionally shows that residual connections help gradient descent. In the setting we consider we prove that residual connections are necessary to circumvent oversmoothing, to go to larger K and to improve the performance.

5. Continuous neural networks

Continuous neural networks can be seen as the natural limit of residual networks, when the depth K and the residual connection strengths c_k go to infinity proportionally, if $f_{\theta^{(k)}}$ is smooth enough with respect to k . In this limit, rescaling $h^{(k+1)}$ with c_k and setting $x = k/K$ and $c_k = K/t$, the rescaled h satisfies the differential equation

$$\frac{dh_i}{dx}(x) = t f_{\theta(x)}(h_i(x), \text{aggreg}(\{h_j(x), j \sim i\})). \quad (6)$$

This equation is called a neural ordinary differential equation [27]. The convergence of a residual network to

a continuous limit has been studied, for instance, in [44]. Continuous neural networks are commonly used to model and learn the dynamics of time-evolving systems, by usually taking the update function f_{θ} independent of the time t . For an example, [45] uses a continuous fully connected neural network to model turbulence in a fluid. As such, they are a building block of scientific machine learning; see, for instance, [46] for several applications. As to the generalization ability of continuous neural networks, the only theoretical work we are aware of is [47], that derives loose bounds based on continuity arguments.

Continuous neural networks have been extended to continuous GNNs in [48,49]. For the GCN that we consider the residual connections are implemented by adding self-loops $c_k I_N$ to the graph. The continuous dynamic of h is then

$$\frac{dh}{dx}(x) = t \tilde{A}h(x), \quad (7)$$

with $t \in \mathbb{R}$; which is a diffusion on the graph. Other types of dynamics have been considered, such as anisotropic diffusion, where the diffusion factors are learnt, or oscillatory dynamics, that should avoid oversmoothing too; see, for instance, the review [50] for more details. No prior works predict their generalization ability. In this work we fill this gap by deriving the performance of the continuous limit of the simple GCN we consider.

C. Summary of the main results

We first generalize the work of [20,26] to predict the performance of a simple GCN with arbitrary number K of convolution steps. The network is trained in a semisupervised way on data generated by the CSBM for node classification. In the high-dimensional limit and in the limit of dense graphs, the main properties of the trained network concentrate onto deterministic values, that do not depend on the particular realization of the data. The network is described by a few order parameters (or summary statistics), that satisfy a set of self-consistent equations, that we solve analytically or numerically. We thus have access to the expected train and test errors and accuracies of the trained network.

From these predictions we draw several consequences. Our main guiding line is to search for the architecture and the hyperparameters of the GCN that maximize its performance, and check whether the optimal GCN can reach the Bayes-optimal performance on the CSBM. The main parameters we consider are the depth K , the residual connection strengths c_k , the regularization r and the loss function.

We consider the convergence rates towards perfect inference at large graph signal. We show that $K = 2$ is the minimal depth to reach the Bayes-optimal rate, after which increasing K or fine-tuning c_k only leads to

subleading improvements. In case of asymmetric graphs the GCN is not able to deal the asymmetry, for all K or c_k , and one has to pre-process the graph by symmetrizing it.

At finite graph signal the behavior of the GCN is more complex. We find that large regularization r maximizes the test accuracy in the case we consider, while the loss has little effect. The residual connection strengths c_k have to be tuned to a same optimal value c that depends on the properties of the graph.

An important point is that going to larger K seems to improve the test accuracy. Yet the residual connection c has to vary accordingly. If c stays constant with respect to K then the GCN will perform PCA on the graph A , oversmooth and discard the information from the features X . Instead, if c grows with K , the residual connections alleviate oversmoothing and the performance of the GCN keeps increasing with K , if the diffusion time $t = K/c$ is well tuned.

The limit $K \rightarrow \infty$, $c \propto K$ is thus of particular interest. It corresponds to a continuous GCN performing diffusion on the graph. Our analysis can be extended to this case by directly taking the limit in the self-consistent equations. One has to further jointly expand them around $r \rightarrow +\infty$ and we keep the first order. At the end we predict the performance of the continuous GCN in an explicit and closed form. To our knowledge this is the first tight prediction of the generalization ability of a continuous neural network, and in particular of a continuous graph neural network. The large regularization limit $r \rightarrow +\infty$ is important: On one hand it appears to lead to the optimal performance of the neural network; on another hand, it is instrumental to analyze the continuous limit $K \rightarrow \infty$ and it allows us to analytically solve the self-consistent equations describing the neural network.

We show that the continuous GCN at optimal time t performs better than any finite- K GCN. The optimal t depends on the properties of the graph, and can be negative for heterophilic graphs. This result is a step toward solving one of the major challenges identified by [8]; that is, creating benchmarks where depth is necessary and building efficient deep networks.

The continuous GCN as large r is optimal. Moreover, if run on the symmetrized graph, it approaches the Bayes optimality on a broad range of configurations of the CSBM, as exemplified in Fig. 1. We identify when the GCN fails to approach the Bayes optimality: This happens when most of the information is contained in the features and not in the graph, and has to be processed in an unsupervised manner.

We provide the code to evaluate our predictions in [51].

II. DETAILED SETTING

A. Contextual stochastic Block model for attributed graphs

We consider the problem of semisupervised node classification on an attributed graph, where the nodes have

labels and carry additional attributes, or features, and where the structure of the graph correlates with the labels. We consider a graph $\mathcal{G}$ made of N nodes; each node i has a binary label $y_i = \pm 1$ that is a Rademacher random variable.

The structure of the graph should be correlated with y . We model the graph with a binary stochastic block model: The adjacency matrix $A \in \mathbb{R}^{N \times N}$ is drawn according to

$$A_{ij} \sim \mathcal{B}\left(\frac{d}{N} + \frac{\lambda}{\sqrt{N}} \sqrt{\frac{d}{N} \left(1 - \frac{d}{N}\right)} y_i y_j\right), \quad (8)$$

where λ is the signal-to-noise ratio (SNR) of the graph, d is the average degree of the graph, $\mathcal{B}$ is a Bernoulli law, and the elements A_{ij} are independent for all i and j . It can be interpreted in the following manner: An edge between i and j appears with a higher probability if $\lambda y_i y_j > 1$, i.e., for $\lambda > 0$ if the two nodes are in the same group. The scaling with d and N is chosen so that this model does not have a trivial limit at $N \rightarrow \infty$ both for $d = \Theta(1)$ and $d = \Theta(N)$. Notice that we take A asymmetric.

Additionally to the graph, each node i carries attributes $X_i \in \mathbb{R}^M$, that we collect in the matrix $X \in \mathbb{R}^{N \times M}$. We set $\alpha = N/M$ the aspect ratio between the number of nodes and the dimension of the features. We model them by a Gaussian mixture: We draw M hidden Gaussian variables $u_\nu \sim \mathcal{N}(0, 1)$, the centroid $u \in \mathbb{R}^M$, and we set

$$X = \sqrt{\frac{\mu}{N}} y u^T + W, \quad (9)$$

where μ is the SNR of the features and W is noise whose components $W_{i\nu}$ are independent standard Gaussians. We use the notation $\mathcal{N}(m, V)$ for a Gaussian distribution or density of mean m and variance V . The whole model for (y, A, X) is called the contextual stochastic block model and was introduced in [21,22].

We consider the task of inferring the labels y given a subset of them. We define the training set R as the set of nodes whose labels are revealed; $\rho = |R|/N$ is the training ratio. The test set R' is selected from the complement of R ; we define the testing ratio $\rho' = |R'|/N$. We assume that R and R' are independent from the other quantities. The inference problem is to find back y and u given A , X , R and the parameters of the model.

[22,52] prove that the effective SNR of the CSBM is

$$\text{SNR}_{\text{CSBM}} = \lambda^2 + \mu^2/\alpha, \quad (10)$$

in the sense that in the unsupervised regime $\rho = 0$ for $\text{SNR}_{\text{CSBM}} < 1$ no information on the labels can be recovered while for $\text{SNR}_{\text{CSBM}} > 1$ partial information can be recovered. The information given by the graph is λ^2 while the information given by the features is μ^2/α . As soon as a finite fraction of nodes $\rho > 0$ is revealed the phase

transition between no recovery and weak recovery disappears.

We work in the high-dimensional limit $N \rightarrow \infty$ and $M \rightarrow \infty$ while the aspect ratio $\alpha = N/M$ is of order one. The average degree d should be of order N , but taking d growing with N should be sufficient for our results to hold, as shown by our experiments. The other parameters λ, μ, ρ and ρ' are of order one.

B. Analyzed architecture

In this work, we focus on the role of applying several data aggregation steps. With the current theoretical tools, the tight analysis of the generic GNN described in Eq. (1) is not possible: Dealing with multiple layers of learned weights is hard; and even for a fully connected two-layer perceptron this is a current and major topic [29]. Instead, we consider a one-layer GNN with a learned projection w . We focus on graph convolutional networks [28], where the aggregation is a convolution done by applying powers of a rescaling $\tilde{A}$ of the adjacency matrix. Last we remove the nonlinearities. As we will see, the fact that the GCN is linear does not prevent it approaching the optimality in some regimes. The resulting GCN is referred to as a simple graph convolutional network; it has been shown to have good performance while being much easier to train [53,54]. The network we consider transforms the graph and the features in the following manner:

$$h(w) = \prod_{k=1}^K \left(\frac{1}{\sqrt{N}} \tilde{A} + c_k I_N \right) \frac{1}{\sqrt{N}} Xw, \quad (11)$$

where $w \in \mathbb{R}^M$ is the layer of trainable weights, I_N is the identity, $c_k \in \mathbb{R}$ is the strength of the residual connections, and $\tilde{A} \in \mathbb{R}^{N \times N}$ is a rescaling of the adjacency matrix defined by

$$\tilde{A}_{ij} = \left(\frac{d}{N} \left(1 - \frac{d}{N} \right) \right)^{-1/2} \left(A_{ij} - \frac{d}{N} \right), \quad \text{for all } i, j. \quad (12)$$

The prediction $\hat{y}_i$ of the label of i by the GNN is then $\hat{y}_i = \text{sign} h(w)_i$.

$\tilde{A}$ is a rescaling of A that is centered and normalized. In the limit of dense graphs, where d is large, this will allow us to rely on a Gaussian equivalence property to analyze this GCN. The equivalence [20,22,55] states that in the high-dimensional limit, for d growing with N , $\tilde{A}$ can be approximated by the following spiked matrix A^g without changing the macroscopic properties of the GCN:

$$A^g = \frac{\lambda}{\sqrt{N}} yy^T + \Xi, \quad (13)$$

where the components of the $N \times N$ matrix Ξ are independent standard Gaussian random variables. The main

reason for considering dense graphs instead of sparse graphs $d = \Theta(1)$ is to ease the theoretical analysis. The dense model can be described by a few order parameters; while a sparse SBM would be harder to analyze because many quantities, such as the degrees of the nodes, do not self-average, and one would need to take into account all the nodes, by predicting the performance on one realization of the graph or by running population dynamics. We believe that it would lead to qualitatively similar results, as, for instance, [56] shows, for a related model.

The above architecture corresponds to applying K times a graph convolution on the projected features Xw . At each convolution step k a node i updates its features by summing those of its neighbors and adding c_k times its own features. In [20,26] the same architecture was considered for $K = 1$; we generalize these works by deriving the performance of the GCN for arbitrary numbers K of convolution steps. As we will show this is crucial to approach the Bayes-optimal performance.

Compared to [20,26], another important improvement towards the Bayes optimality is obtained by symmetrizing the graph, and we will also study the performance of the GCN when it acts by applying the symmetrized rescaled adjacency matrix $\tilde{A}^s$ defined by

$$\tilde{A}^s = \frac{1}{\sqrt{2}} (\tilde{A} + \tilde{A}^T), \quad A^{g,s} = \frac{\lambda^s}{\sqrt{N}} yy^T + \Xi^s. \quad (14)$$

$A^{g,s}$ is its Gaussian equivalent, with $\lambda^s = \sqrt{2}\lambda$, Ξ^s is symmetric and $\Xi_{i \leq j}^s$ are independent standard Gaussian random variables. In this article we derive and show the performance of the GNN both acting with $\tilde{A}$ and $\tilde{A}^s$ but in this first part we will mainly consider and state the expressions for $\tilde{A}$ because they are simpler. We will consider $\tilde{A}^s$ second while taking the continuous limit. To deal with both cases, asymmetric or symmetrized, we define $\tilde{A}^e \in \{\tilde{A}, \tilde{A}^s\}$ and $\lambda^e \in \{\lambda, \lambda^s\}$.

The continuous limit of the above network (11) is defined by

$$h(w) = e^{\frac{t}{\sqrt{N}} \tilde{A}^e} \frac{1}{\sqrt{N}} Xw, \quad (15)$$

where t is the diffusion time. It is obtained at large K when the update between two convolutions becomes small, as follows:

$$\left(\frac{t}{K\sqrt{N}} \tilde{A}^e + I_N \right)^K \xrightarrow{K \rightarrow \infty} e^{\frac{t}{\sqrt{N}} \tilde{A}^e}. \quad (16)$$

h is the solution at time t of the time-continuous diffusion of the features on the graph $\mathcal{G}$ with Laplacian $\tilde{A}^e$, defined by $\partial_x X(x) = (1/\sqrt{N}) \tilde{A}^e X(x)$ and $X(0) = X$. The discrete GCN can be seen as the discretization of the differential

equation in the forward Euler scheme. The mapping with Eq. (11) is done by taking $c_k = K/t$ for all k and by rescaling the features of the discrete GCN $h(w)$ as $h(w) \prod_k c_k^{-1}$ so they remain of order one when K is large. For the discrete GCN we do not directly consider the update $h_{k+1} = (I_N + c_k^{-1} \tilde{A}/\sqrt{N})h_k$ because we want to study the effect of having no residual connections, i.e., $c_k = 0$. The case where the diffusion coefficient depends on the position in the network is equivalent to a constant diffusion coefficient. Indeed, because of commutativity, the solution at time t of $\partial_x X(x) = (1/\sqrt{N})a(x)\tilde{A}^e X(x)$ for $a: \mathbb{R} \rightarrow \mathbb{R}$ is $\exp[\int_0^t dx a(x)(1/\sqrt{N})\tilde{A}^e]X(0)$.

The discrete and the continuous GCNs are trained by empirical risk minimization. We define the regularized loss

$$L_{A,X}(w) = \frac{1}{\rho N} \sum_{i \in R} \ell(y_i h_i(w)) + \frac{r}{\rho N} \sum_{\nu} \gamma(w_{\nu}) \quad (17)$$

where γ is a strictly convex regularization function, r is the regularization strength and ℓ is a convex loss function. The regularization ensures that the GCN does not overfit the train data and has good generalization properties on the test set. We will focus on l_2 regularization $\gamma(x) = x^2/2$ and on the square loss $\ell(x) = (1-x)^2/2$ (ridge regression) or the logistic loss $\ell(x) = \log(1 + e^{-x})$ (logistic regression). Since L is strictly convex it admits a unique minimizer w^* . The key quantities we want to estimate are the average train and test errors and accuracies of this model, which are

$$E_{\text{train/test}} = \mathbb{E} \frac{1}{|\hat{R}|} \sum_{i \in \hat{R}} \ell[y_i h(w^*)_i], \quad (18)$$

$$\text{Acc}_{\text{train/test}} = \mathbb{E} \frac{1}{|\hat{R}|} \sum_{i \in \hat{R}} \delta_{y_i = \text{sign} h(w^*)_i}, \quad (19)$$

where $\hat{R}$ stands either for the train set R or the test set R' and the expectation is taken over y , u , A , X , R , and R' . $\text{Acc}_{\text{train/test}}$ is the proportion of train or test nodes that are correctly classified. A main part of the present work is dedicated to the derivation of exact expressions for the errors and the accuracies. We will then search for the architecture of the GCN that maximizes the test accuracy Acc_{test} . The parameters of the model are summarized in Table I.

Notice that one could treat the residual connection strengths c_k as supplementary parameters, jointly trained with w to minimize the train loss. Our analysis can straightforwardly be extended to deal with this case. Yet, as we will show, to take trainable c_k degrades the test performances; and it is better to consider them as hyperparameters optimizing the test accuracy.

TABLE I. Summary of the parameters of the model.

N	Number of nodes
M	Dimension of the attributes
$\alpha = N/M$	Aspect ratio
d	Average degree of the graph
λ	Signal strength of the graph
μ	Signal strength of the features
$\rho = R /N$	Fraction of training nodes
ℓ, γ	Loss and regularization functions
r	Regularization strength
K	Number of aggregation steps
c_k, c, t	Residual connection strengths, diffusion time

C. Bayes-optimal performance

An interesting consequence of modeling the data as we propose is that one has access to the Bayes-optimal performance on this task. The BO performance is defined as the upper bound on the test accuracy that any algorithm can reach on this problem, knowing the model and its parameters α , λ , μ and ρ . It is of particular interest since it will allow us to check how far the GCNs are from the optimality and how much improvement one can hope for.

The BO performance on this problem has been derived in [22,39]. It is expressed as a function of the fixed-point of an algorithm based on approximate message-passing (AMP). In the limit of large degrees $d = \Theta(N)$ this algorithm can be tracked by a few scalar state-evolution (SE) equations that we reproduce in Appendix C.

III. ASYMPTOTIC CHARACTERIZATION OF THE GCN

In this section we provide an asymptotic characterization of the performance of the GCNs previously defined. It relies on a finite set of order parameters that satisfy a system of self-consistent, or fixed-point, equations that we obtain thanks to the replica method in the high-dimensional limit at finite K . In a second part, for the continuous GCN, we show how to take the limit $K \rightarrow \infty$ for the order parameters and for their self-consistent equations. The continuous GCN is still described by a finite set of order parameters, but these are now continuous functions and the self-consistent equations are integral equations.

Notice that for a quadratic loss function ℓ there is an analytical expression for the minimizer w^* of the regularized loss $L_{A,X}$ Eq. (17), given by the regularized least-squares formula. Based on that, a computation of the performance of the GCN with random matrix theory (RMT) is possible. It would not be straightforward in the sense that the convolved features, the weights w^* and the labels y are correlated, and such a computation would have to take into account these correlations. Instead, we prefer to use the replica method, which has already been successfully applied to analyze several architectures of one (learnable) layer neural networks in articles such

that [17,38]. Compared to RMT, the replica method allows us to seamlessly handle the regularized pseudo-inverse of the least squares and to deal with logistic regression, where no explicit expression for w^* exists.

We compute the average train and test errors and accuracies Eqs. (18) and (19) in the high-dimensional limit N and M large. We define the Hamiltonian

$$H(w) = s \sum_{i \in R} \ell[y_i h(w)_i] + r \sum_{\nu} \gamma(w_{\nu}) + s' \sum_{i \in R'} \ell[y_i h(w)_i], \quad (20)$$

where s and s' are external fields to probe the observables. The loss of the test samples is in H for the purpose of the analysis; we will take $s' = 0$ later and the GCN is minimizing the training loss (17). The free energy f is defined as

$$Z = \int dw e^{-\beta H(w)}, \quad f = -\frac{1}{\beta N} \mathbb{E} \log Z. \quad (21)$$

β is an inverse temperature; we consider the limit $\beta \rightarrow \infty$, where the partition function Z concentrates over w^* at $s = 1$ and $s' = 0$. The train and test errors are then obtained according to

$$E_{\text{train}} = \frac{1}{\rho} \frac{\partial f}{\partial s}, \quad E_{\text{test}} = \frac{1}{\rho'} \frac{\partial f}{\partial s'}, \quad (22)$$

both evaluated at $(s, s') = (1, 0)$. One can, in the same manner, compute the average accuracies by introducing the observables $\sum_{i \in R} \delta_{y_i = \text{sign} h(w)_i}$ in H . To compute f we introduce n replica:

$$\mathbb{E} \log Z = \mathbb{E} \frac{\partial Z^n}{\partial n} (n=0) = \left(\frac{\partial}{\partial n} \mathbb{E} Z^n \right) (n=0). \quad (23)$$

To pursue the computation we need to precise the architecture of the GCN.

A. Discrete GCN

1. Asymptotic characterization

In this section, we work at finite K . We consider only the asymmetric graph. We define the state of the GCN after the k th convolution step as

$$h_k = \left(\frac{1}{\sqrt{N}} \tilde{A} + c_k I_N \right) h_{k-1}, \quad h_0 = \frac{1}{\sqrt{N}} X w. \quad (24)$$

$h_K = h(w) \in \mathbb{R}^N$ is the output of the full GCN. We introduce h_k in the replicated partition function Z^n and we integrate over the fluctuations of A and X . This couples the variables across the different layers $k = 0 \dots K$ and one has to take into account the correlations between the different h_k , which will result into order parameters of dimension K . One has to keep separate the indices $i \in R$

and $i \notin R$, whether the loss ℓ is active or not; consequently, the free entropy of the problem will be a linear combination of ρ times a potential with ℓ and $(1 - \rho)$ times without ℓ . The limit $N \rightarrow \infty$ is taken thanks to Laplace's method. The extremization is done in the space of the replica-symmetric ansatz, which is justified by the convexity of H . The detailed computation is given in Appendix A.

The outcome of the computation is that this problem is described by a set of twelve order parameters (or summary statistics). They are $\Theta = \{m_w \in \mathbb{R}, Q_w \in \mathbb{R}, V_w \in \mathbb{R}, m \in \mathbb{R}^K, Q \in \mathbb{R}^{K \times K}, V \in \mathbb{R}^{K \times K}\}$ and their conjugates $\hat{\Theta} = \{\hat{m}_w \in \mathbb{R}, \hat{Q}_w \in \mathbb{R}, \hat{V}_w \in \mathbb{R}, \hat{m} \in \mathbb{R}, \hat{Q} \in \mathbb{R}^{K \times K}, \hat{V} \in \mathbb{R}^{K \times K}\}$, where

$$m_w = \frac{1}{N} u^T w, \quad m_k = \frac{1}{N} y^T h_k, \quad (25)$$

$$Q_w = \frac{1}{N} w^T w, \quad Q_{k,l} = \frac{1}{N} h_k^T h_l, \quad (26)$$

$$V_w = \frac{\beta}{N} \text{Tr}(\text{Cov}_{\beta}(w, w)), \quad V_{k,l} = \frac{\beta}{N} \text{Tr}[\text{Cov}_{\beta}(h_k, h_l)]. \quad (27)$$

m_w and m_k are the magnetizations (or overlaps) between the weights and the hidden variables and between the k th layer and the labels; the Q s are the self-overlaps (or scalar products) between the different layers; and, writing Cov_{β} for the covariance under the density $e^{-\beta H}$, the V 's are the covariances between different trainings on the same data, after rescaling by β .

The order parameters Θ and $\hat{\Theta}$ satisfy the property that they extremize the following free entropy ϕ :

$$\begin{aligned} \phi = & \frac{1}{2} (\hat{V}_w V_w + \hat{V}_w Q_w - V_w \hat{Q}_w) - \hat{m}_w m_w \\ & + \frac{1}{2} \text{tr}(\hat{V} V + \hat{V} Q - V \hat{Q}) - \hat{m}^T m \\ & + \frac{1}{\alpha} \mathbb{E}_{u, \zeta} \left(\log \int dw e^{\psi_w(w)} \right) \\ & + \rho \mathbb{E}_{y, \xi, \zeta, \chi} \left(\log \int \prod_{k=0}^K dh_k e^{\psi_h(h; s)} \right) \\ & + (1 - \rho) \mathbb{E}_{y, \xi, \zeta, \chi} \left(\log \int \prod_{k=0}^K dh_k e^{\psi_h(h; s')} \right), \end{aligned} \quad (28)$$

the potentials being

$$\psi_w(w) = -r \gamma(w) - \frac{1}{2} \hat{V}_w w^2 + \left(\sqrt{\hat{Q}_w \zeta} + u \hat{m}_w \right) w, \quad (29)$$

$$\begin{aligned} \psi_h(h; \bar{s}) = & -\bar{s} \ell(y h_K) - \frac{1}{2} h_{<K}^T \hat{V} h_{<K} + (\xi^T \hat{Q}^{1/2} + y \hat{m}^T) h_{<K} \\ & + \log \mathcal{N}(h_0 | \sqrt{\mu} y m_w + \sqrt{\hat{Q}_w \zeta}; V_w) \\ & + \log \mathcal{N}(h_{>0} | c \odot h_{<K} + \lambda y m + Q^{1/2} \chi; V), \end{aligned} \quad (30)$$

for $w \in \mathbb{R}$ and $h \in \mathbb{R}^{K+1}$, where we introduced the Gaussian random variables $\varsigma \sim \mathcal{N}(0, 1)$, $\xi \sim \mathcal{N}(0, I_K)$, $\zeta \sim \mathcal{N}(0, 1)$ and $\chi \sim \mathcal{N}(0, I_K)$, take y Rademacher and $u \sim \mathcal{N}(0, 1)$, where we set $h_{>0} = (h_1, \dots, h_K)^T$, $h_{<K} = (h_0, \dots, h_{K-1})^T$ and $c \odot h_{<K} = (c_1 h_0, \dots, c_K h_{K-1})^T$ and where $\bar{s} \in \{0, 1\}$ controls whether the loss ℓ is active or not. We use the notation $\mathcal{N}(\cdot|m; V)$ for a Gaussian density of mean m and variance V . We emphasize that ψ_w and ψ_h are effective potentials taking into account the randomness of the model and that are defined over a finite number of variables, contrary to the initial loss function H .

The extremality condition $\nabla_{\Theta, \hat{\Theta}} \phi = 0$ can be stated in terms of a system of self-consistent equations that we give here. In the limit $\beta \rightarrow \infty$ one has to consider the extremizers of ψ_w and ψ_h defined as

$$w^* = \underset{w}{\operatorname{argmax}} \psi_w(w) \in \mathbb{R}, \quad (31)$$

$$h^* = \underset{h}{\operatorname{argmax}} \psi_h(h; \bar{s} = 1) \in \mathbb{R}^{K+1}, \quad (32)$$

$$h'^* = \underset{h}{\operatorname{argmax}} \psi_h(h; \bar{s} = 0) \in \mathbb{R}^{K+1}. \quad (33)$$

We also need to introduce $\operatorname{Cov}_{\psi_h}(h)$ and $\operatorname{Cov}_{\psi_h}(h')$ the covariances of h under the densities $e^{\psi_h(h, \bar{s}=1)}$ and $e^{\psi_h(h, \bar{s}=0)}$. In the limit $\beta \rightarrow \infty$ they read

$$\operatorname{Cov}_{\psi_h}(h) = \nabla \nabla \psi_h(h^*; \bar{s} = 1), \quad (34)$$

$$\operatorname{Cov}_{\psi_h}(h') = \nabla \nabla \psi_h(h'^*; \bar{s} = 0), \quad (35)$$

$\nabla \nabla$ being the Hessian with respect to h . Last, for compactness we introduce the operator $\mathcal{P}$ that, for a function g in h , acts according to

$$\mathcal{P}[g(h)] = \rho g(h^*) + (1 - \rho) g(h'^*). \quad (36)$$

For instance $\mathcal{P}(hh^T) = \rho h^*(h^*)^T + (1 - \rho) h'^*(h'^*)^T$ and $\mathcal{P}[\operatorname{Cov}_{\psi_h}(h)] = \rho \operatorname{Cov}_{\psi_h}(h) + (1 - \rho) \operatorname{Cov}_{\psi_h}(h')$. Then the extremality condition gives the following self-consistent, or fixed-point, equations on the order parameters:

$$m_w = \frac{1}{\alpha} \mathbb{E}_{u, \varsigma} u w^*, \quad (37)$$

$$Q_w = \frac{1}{\alpha} \mathbb{E}_{u, \varsigma} (w^*)^2, \quad (38)$$

$$V_w = \frac{1}{\alpha} \frac{1}{\sqrt{Q_w}} \mathbb{E}_{u, \varsigma} \varsigma w^*, \quad (39)$$

$$m = \mathbb{E}_{y, \xi, \zeta, \chi} y \mathcal{P}(h_{<K}), \quad (40)$$

$$Q = \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{P}(h_{<K} h_{<K}^T), \quad (41)$$

$$V = \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{P}[\operatorname{Cov}_{\psi_h}(h_{<K})], \quad (42)$$

$$\hat{m}_w = \frac{\sqrt{\mu}}{V_w} \mathbb{E}_{y, \xi, \zeta, \chi} y \mathcal{P}(h_0 - \sqrt{\mu} y m_w), \quad (43)$$

$$\hat{Q}_w = \frac{1}{V_w^2} \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{P}((h_0 - \sqrt{\mu} y m_w - \sqrt{Q_w} \zeta)^2), \quad (44)$$

$$\hat{V}_w = \frac{1}{V_w} - \frac{1}{V_w^2} \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{P}[\operatorname{Cov}_{\psi_h}(h_0)], \quad (45)$$

$$\hat{m} = \lambda V^{-1} \mathbb{E}_{y, \xi, \zeta, \chi} y \mathcal{P}(h_{>0} - c \odot h_{<K} - \lambda y m), \quad (46)$$

$$\hat{Q} = V^{-1} \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{P}((h_{>0} - c \odot h_{<K} - \lambda y m - Q^{1/2} \chi)^{\otimes 2}) V^{-1}, \quad (47)$$

$$\hat{V} = V^{-1} - V^{-1} \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{P}[\operatorname{Cov}_{\psi_h}(h_{>0} - c \odot h_{<K})] V^{-1}. \quad (48)$$

Once this system of equations is solved, the expected errors and accuracies can be expressed as

$$E_{\text{train}} = \mathbb{E}_{y, \xi, \zeta, \chi} \ell(y h_K^*), \quad \text{Acc}_{\text{train}} = \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y=\text{sign}(h_K^*)}, \quad (49)$$

$$E_{\text{test}} = \mathbb{E}_{y, \xi, \zeta, \chi} \ell(y h_K'^*), \quad \text{Acc}_{\text{test}} = \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y=\text{sign}(h_K'^*)}. \quad (50)$$

2. Analytical solution

In general the system of self-consistent equations (37)–(48) has to be solved numerically. The equations are applied iteratively, starting from arbitrary Θ and $\hat{\Theta}$, until convergence.

An analytical solution can be computed in some special cases. We consider ridge regression (i.e., quadratic ℓ) and take $c = 0$ no residual connections. Then $\operatorname{Cov}_{\psi_h}(h)$, $\operatorname{Cov}_{\psi_h'}(h)$, V , and $\hat{V}$ are diagonal. We obtain that

$$\text{Acc}_{\text{test}} = \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{\lambda q_{y, K-1}}{\sqrt{2}} \right) \right), \quad q_{y, k} = \frac{m_k}{\sqrt{Q_{k, k}}}. \quad (51)$$

The test accuracy only depends on the angle (or overlap) $q_{y, k}$ between the labels y and the last hidden state h_{K-1} of the GCN. $q_{y, k}$ can easily be computed in the limit $r \rightarrow \infty$. In Appendix A 3 we explicit the equations (37)–(50) and give their solution in that limit. In particular we obtain for any k

$$m_k = \frac{\rho}{\alpha r} \left(\mu \lambda^{K+k} + \sum_{l=0}^k \lambda^{K-k+2l} \right), \quad (52)$$

$$Q_{k,k} = \frac{\rho}{\alpha^2 r^2} \left(\alpha \left(1 + \rho \mu \lambda^{2K} + \rho \sum_{l=1}^K \lambda^{2l} \right) + \sum_{l=0}^k \left(1 + \rho \sum_{l'=1}^{K-1-l} \lambda^{2l'} + \frac{\alpha^2 r^2}{\rho} m_l^2 \right) \right). \quad (53)$$

3. Consequences: Going to large K is necessary

We derive consequences from the previous theoretical predictions. We numerically solve Eqs. (37)–(48) for some plausible values of the parameters of the data model. We keep balanced the signals from the graph, λ^2 , and from the features, μ^2/α ; we take $\rho = 0.1$ to stick to the common case where few train nodes are available. We focus on searching the architecture that maximizes the test accuracy by varying the loss ℓ , the regularization r , the residual connections c_k and K . For simplicity we will mostly consider the case where $c_k = c$ for all k and for a given c . We compare our theoretical predictions to simulations of the GCN for $N = 10^4$ in Fig. 2; as expected, the predictions are within the statistical errors. Details on the numerics are provided in Appendix D. The code that allows us to evaluate our predictions is openly available [51].

Reference [26] already studies in detail the effect of ℓ , r , and c at $K = 1$. It reaches the conclusion that the optimal regularization is $r \rightarrow \infty$, that the choice of the loss ℓ has

little effect, and that there is an optimal $c = c^*$ of order 1. According to Fig. 2, it seems that these results can be extrapolated to $K > 1$. We indeed observe that, for both the quadratic and the logistic loss, at $K \in \{1, 2, 3\}$, $r \rightarrow \infty$ seems optimal. Then the choice of the loss has little effect, because at large r the output $h(w)$ of the network is small and only the behavior of ℓ around 0 matters. Notice that, though $h(w)$ is small and the error $E_{\text{train/test}}$ is trivially equal to $\ell(0)$, the sign of $h(w)$ is mostly correct and the accuracy $\text{Acc}_{\text{train/test}}$ is not trivial. Last, according to the inset of Fig. 2 for $K = 2$, to take $c_1 = c_2$ is optimal and our assumption $c_k = c$ for all k is justified.

To take trainable c_k would degrade the test performances. We show this in Fig. 15 in Appendix E, where optimizing the train error E_{train} over $c_k = c$ trivially leads to $c = +\infty$. Indeed, in this case the graph is discarded and the convolved features are proportional to the features X ; which, if $\alpha\rho$ is small enough, are separable, and lead to a null train error. Consequently, c_k should be treated as a hyperparameter, tuned to maximize Acc_{test} , as we do in the rest of the article.

Finite K . We focus on the effect of varying the number K of aggregation steps. Reference [26] shows that at $K = 1$ there is a large gap between the Bayes-optimal test accuracy and the best test accuracy of the GCN. We find that, according to Fig. 2, for $K \in \{1, 2, 3\}$, to increase K reduces more and

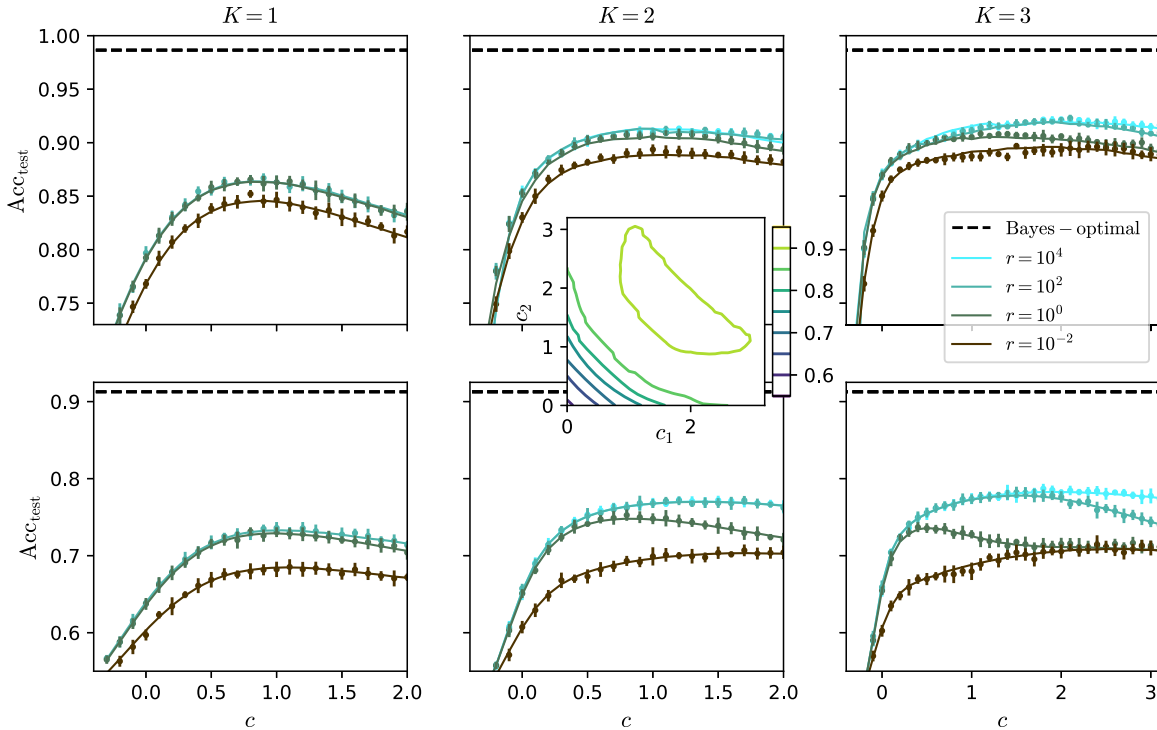


FIG. 2. Predicted test accuracy Acc_{test} for different values of K . Top: for $\lambda = 1.5$, $\mu = 3$ and logistic loss; bottom: for $\lambda = 1$, $\mu = 2$ and quadratic loss; $\alpha = 4$ and $\rho = 0.1$. We take $c_k = c$ for all k . Inset: Acc_{test} vs c_1 and c_2 at $K = 2$ and at large r . Dots: numerical simulation of the GCN for $N = 10^4$ and $d = 30$, averaged over ten experiments.

more the gap. Thus going to higher depth allows one to approach the Bayes optimality.

This also stands as to the learning rate when the signal λ of the graph increases. At $\lambda \rightarrow \infty$ the GCN is consistent and correctly predicts the labels of all the test nodes, that is $\text{Acc}_{\text{test}} \xrightarrow{\lambda \rightarrow \infty} 1$. The learning rate $\tau > 0$ of the GCN is defined as

$$\log(1 - \text{Acc}_{\text{test}}) \underset{\lambda \rightarrow \infty}{\sim} -\tau\lambda^2. \quad (54)$$

As shown in [39], the rate τ_{BO} of the Bayes-optimal test accuracy is

$$\tau_{\text{BO}} = 1. \quad (55)$$

For $K = 1$ [26] proves that $\tau \leq \tau_{\text{BO}}/2$ and that $\tau \rightarrow \tau_{\text{BO}}/2$ when the signal from the features μ^2/α diverges. We obtain that if $K > 1$ then $\tau = \tau_{\text{BO}}/2$ for any signal from the features. This is shown on Fig. 3, where for $K = 1$ the slope of the residual error varies with μ and α but does not reach half of the Bayes-optimal slope; while for $K > 1$ it does, and the features only contribute with a subleading order.

Analytically, taking the limit in Eqs. (52) and (53), at $c = 0$ and $r \rightarrow \infty$ we have that

$$\lim_{\lambda \rightarrow \infty} q_{y,K-1} \begin{cases} = 1 & \text{if } K > 1 \\ < 1 & \text{if } K = 1 \end{cases}. \quad (56)$$

Since $\log[1 - \text{erf}(\lambda q_{y,K-1}/\sqrt{2})] \underset{\lambda \rightarrow \infty}{\sim} -\lambda^2 q_{y,K-1}^2/2$ we recover the leading behavior depicted in Fig. 3. c has little effect on the rate τ ; it only seems to vary the test accuracy by a subleading term.

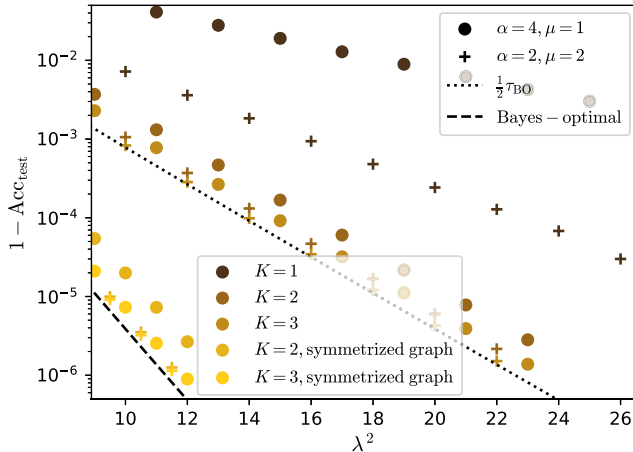


FIG. 3. Predicted misclassification error $1 - \text{Acc}_{\text{test}}$ at large λ for two strengths of the feature signal. $r = \infty$, $c = c^*$ is optimized by grid search and $\rho = 0.1$. The dots are theoretical predictions given by numerically solving the self-consistent equations (37)–(48) simplified in the limit $r \rightarrow \infty$. For the symmetrized graph the self-consistent equations are Eqs. (83)–(90) in the next part.

Symmetrization. We found that in order to reach the Bayes-optimal rate one has to further symmetrize the graph, according to Eq. (14), and to perform the convolution steps by applying $\tilde{A}^s$ instead of $\tilde{A}$. Then, as shown in Fig. 3, the GCN reaches the BO rate for any $K > 1$, at any signal from the features.

The reason of this improvement is the following. The GCN we consider is not able to deal with the asymmetry of the graph and the supplementary information it gives. This is shown in Fig. 14 in Appendix E for different values of K at finite signal, in agreement with [20]. There is little difference in the performance of the simple GCN whether the graph is symmetric or not with the same λ . As to the rates, as shown by the computation in Appendix C, a symmetric graph with signal λ would lead to a BO rate $\tau_{\text{BO}}^s = 1/2$, which is the rate the GCN achieves on the asymmetric graph. It is thus better to let the GCN process the symmetrized the graph, which has a higher signal $\lambda^s = \sqrt{2}\lambda$, and which leads to $\tau = 1 = \tau_{\text{BO}}$.

Symmetrization is an important step toward the optimality and we will detail the analysis of the GCN in the symmetrized graph in Sec. III B.

Large K and scaling of c . Going to larger K is beneficial and allows the network to approach the Bayes optimality. Yet $K = 3$ is not enough to reach it at finite λ , and one can ask what happens at larger K . An important point is that c has to be well tuned. In Fig. 2 we observe that c^* , the optimal c , is increasing with K . To make this point more precise, in Fig. 4 we show the predicted test accuracy for larger K for different scalings of c . We take $r = \infty$ since it appears to be the optimal regularization. We consider no residual connections, $c = 0$; constant residual connections, $c = 1$; or growing residual connections, $c \propto K$.

A main observation is that, in Fig. 4 for $K \rightarrow \infty$, $c = 0$ or $c = 1$ converge to the same limit, while $c \propto K$ converge to a different limit, which has higher accuracy.

In the case where $c = 0$ or $c = 1$ the GCN oversmooths at large K . The limit it converges to corresponds to the accuracy of principal component analysis (PCA) on the sole graph; that is, it corresponds to the accuracy of the estimator $\hat{y}_{\text{PCA}} = \text{sign}(\text{Re}(y_1))$, where y_1 is the leading eigenvector of $\tilde{A}$. The overlap q_{PCA} between y and $\hat{y}_{\text{PCA}}$ and the accuracy are

$$q_{\text{PCA}} = \begin{cases} \sqrt{1 - \lambda^{-2}} & \text{if } \lambda \geq 1 \\ 0 & \text{if } \lambda \leq 1 \end{cases}, \quad (57)$$

$$\text{Acc}_{\text{test,PCA}} = \frac{1}{2} \left(1 + \text{erf} \left(\frac{\lambda q_{\text{PCA}}}{\sqrt{2}} \right) \right). \quad (58)$$

Consequently, if c does not grow, the GCN will oversmooth at large K , in the sense that all the information from the features X vanishes. Only the information from the graph remains, that can still be informative if $\lambda > 1$.

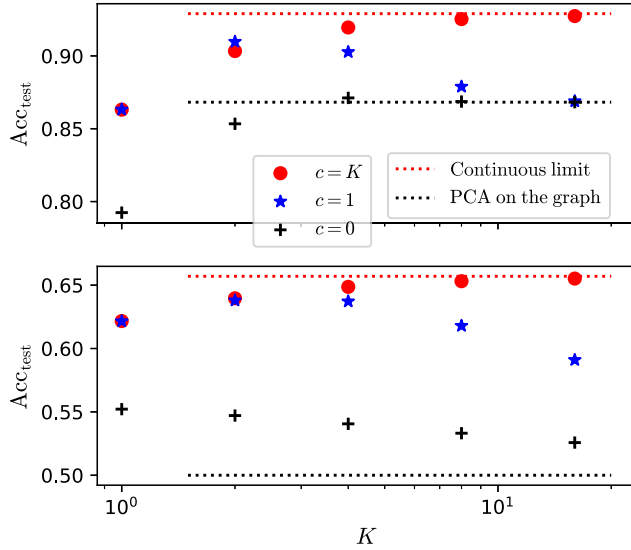


FIG. 4. Predicted test accuracy Acc_{test} vs K for different scalings of c , at $r = \infty$. Top: for $\lambda = 1.5$, $\mu = 3$; bottom: for $\lambda = 0.7$, $\mu = 1$; $\alpha = 4$, $\rho = 0.1$. The predictions are given either by the explicit expression Eqs. (51)–(53) for $c = 0$, either by solving the self-consistent equations (37)–(48) simplified in the limit $r \rightarrow \infty$. The performance for the continuous limit are derived and given in Sec. III B, while the performance of PCA on the graph is given by Eqs. (57)–(58).

The formula (57)–(58) is obtained by taking the limit $K \rightarrow \infty$ in Eqs. (51)–(53), for $c = 0$. For any constant c it can also be recovered by considering the leading eigenvector y_1 of $\tilde{A}$. At large K , $(\tilde{A}/\sqrt{N} + cI)^K$ is dominated by y_1 and the output of the GCN is $h(w) \propto y_1$ for any w . Consequently, the GCN exactly acts like thresholded PCA on $\tilde{A}$. The sharp transition at $\lambda = 1$ corresponds to the BBP phase transition in the spectrum of A^g and $\tilde{A}$ [57]. According to Eqs. (51)–(53) the convergence of $q_{y,K-1}$ toward q_{PCA} is exponentially fast in K if $\lambda > 1$; it is like $1/\sqrt{K}$, much slower, if $\lambda < 1$.

The fact that the oversmoothed features can be informative differs from several previous works where they are fully noninformative, such as [9,10,40]. This is mainly due to the normalization $\tilde{A}$ of A we use and that these works do not use. It allows us to remove the uniform eigenvector $(1, \dots, 1)^T$, that otherwise dominates A and leads to non-informative features. Reference [36] emphasizes this point and compares different ways of normalizing and correcting A . This work concludes, as we do, that for a correct rescaling $\tilde{A}$ of A , similar to ours, going to higher K is always beneficial if λ is high enough, and that the convergence to the limit is exponentially fast. Yet, at large K it obtains bounds on the test accuracy that do not depend on the features: the network they consider still oversmooths in the precise sense we defined. This can be expected since it does not have residual connections, i.e., $c = 0$, that appear to be decisive.

In the case where $c \propto K$ the GCN does not oversmooth and it converges to a continuous limit, obtained as $(cI + \tilde{A}/\sqrt{N})^K \propto (I + t\tilde{A}/K\sqrt{N})^K \rightarrow e^{\frac{t\tilde{A}}{\sqrt{N}}}$. We study this limit in detail in the next part where we predict the resulting accuracy for all constant ratios $t = c/K$. In general, the continuous limit has a better performance than the limit at constant c that relies only on the graph, performing PCA, because it can take into account the features, which bring additional information.

Figure 4 suggests that Acc_{test} is monotonically increasing with K if $c \propto K$ and that the continuous limit is the upper bound on the performance at any K . We will make this point more precise in the next part. Yet we can already see that, for this to be true, one has to correctly tune the ratio c/K : for instance, if λ is small $\tilde{A}$ mostly contains noise and applying it to X will mostly lower the accuracy. Shortly, if c/K is optimized then $K \rightarrow \infty$ is better than any fixed K . Consequently, the continuous limit is the correct limit to maximize the test accuracy and it is of particular relevance.

B. Continuous GCN

In this section we present the asymptotic characterization of the continuous GCN, both for the asymmetric graph and for its symmetrization. The continuous GCN is the limit of the discrete GCN when the number of convolution steps K diverges while the residual connections c become large. The order parameters that describe it, as well as the self-consistent equations they follow, can be obtained as the limit of those of the discrete GCN. We give a detailed derivation of how the limit is taken, since it is of independent interest.

The outcome is that the state h of the GCN across the convolutions is described by a set of equations resembling the dynamical mean-field theory. The order parameters of the problem are continuous functions and the self-consistent equations can be expressed by expansion around large regularization $r \rightarrow \infty$ as integral equations, that specialize to differential equations in the asymmetric case. The resulting equations can be solved analytically; for asymmetric graphs, the covariance and its conjugate are propagators (or resolvent) of the two-dimensional Klein-Gordon equation. We show numerically that our approach is justified and agrees with simulations. Last we show that going to the continuous limit while symmetrizing the graph corresponds to the optimum of the architecture and allows to approach the Bayes optimality.

1. Asymptotic characterization

To deal with both cases, asymmetric or symmetrized, we define $(\delta_e, \tilde{A}^e, \lambda^e) \in \{(0, \tilde{A}, \lambda), (1, \tilde{A}^s, \lambda^s)\}$, where we remind that $\tilde{A}^s$ is the symmetrized $\tilde{A}$ with effective signal $\lambda^s = \sqrt{2}\lambda$. In particular $\delta_e = 0$ for the asymmetric and $\delta_e = 1$ for the symmetrized.

The continuous GCN is defined by the output function

$$h(w) = e^{\frac{t}{\sqrt{N}}\tilde{A}^e} \frac{1}{\sqrt{N}} Xw. \quad (59)$$

We first derive the free entropy of the discretization of the GCN and then take the continuous limit. The discretization at finite K is

$$h(w) = h_K, \quad (60)$$

$$h_{k+1} = \left(I_N + \frac{t}{K\sqrt{N}} \tilde{A}^e \right) h_k, \quad (61)$$

$$h_0 = \frac{1}{\sqrt{N}} Xw. \quad (62)$$

In the case of the asymmetric graph this discretization can be mapped to the discrete GCN of the previous section A as detailed in eq. (16) and the following paragraph; the free entropy and the order parameters of the two models are the same, up to a rescaling by c .

The order parameters of the discretization of the GCN are $m_w \in \mathbb{R}$, $Q_w \in \mathbb{R}$, $V_w \in \mathbb{R}$, $m \in \mathbb{R}^K$, $Q_h \in \mathbb{R}^{K \times K}$, $V_h \in \mathbb{R}^{K \times K}$, their conjugates $\hat{m}_w \in \mathbb{R}$, $\hat{Q}_w \in \mathbb{R}$, $\hat{V}_w \in \mathbb{R}$, $\hat{m} \in \mathbb{R}$, $\hat{Q}_h \in \mathbb{R}^{K \times K}$, $\hat{V}_h \in \mathbb{R}^{K \times K}$ and the two additional order parameters $Q_{qh} \in \mathbb{R}^{K \times K}$ and $V_{qh} \in \mathbb{R}^{K \times K}$ that account for the supplementary correlations the symmetry of the graph induces; $Q_{qh} = V_{qh} = 0$ for the asymmetric case.

The free entropy and its derivation are given in Appendix B. The outcome is that h is described by the effective low-dimensional potential ψ_h over $\mathbb{R}^{K+1}$ that is

$$\psi_h(h; \bar{s}) = -\frac{1}{2} h^T G h + h^T (B_h + D_{qh}^T G_0^{-1} B); \quad (63)$$

where

$$G = G_h + D_{qh}^T G_0^{-1} D_{qh}, \quad (64)$$

$$G_h = \begin{pmatrix} \hat{V}_h & 0 \\ 0 & \bar{s} \end{pmatrix}, \quad (65)$$

$$G_0 = \begin{pmatrix} K^2 V_w & 0 \\ 0 & t^2 V_h \end{pmatrix}, \quad (66)$$

$$D_{qh} = D - t \begin{pmatrix} 0 & 0 \\ -i\delta_e V_{qh}^T & 0 \end{pmatrix} \quad (67)$$

are $(K+1) \times (K+1)$ block matrices;

$$D = K \begin{pmatrix} 1 & & & 0 \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{pmatrix} \quad (68)$$

is the $(K+1) \times (K+1)$ discrete derivative;

$$B = \begin{pmatrix} K\sqrt{Q_w}\chi \\ i t (\hat{Q}^{1/2} \zeta)_q \end{pmatrix} + y \begin{pmatrix} K\sqrt{\mu} m_w \\ \lambda^e t m \end{pmatrix}, \quad (69)$$

$$B_h = \begin{pmatrix} (\hat{Q}^{1/2} \zeta)_h \\ 0 \end{pmatrix} + y \begin{pmatrix} \hat{m} \\ \bar{s} \end{pmatrix}, \quad (70)$$

$$\begin{pmatrix} (\hat{Q}^{1/2} \zeta)_q \\ (\hat{Q}^{1/2} \zeta)_h \end{pmatrix} = \begin{pmatrix} -Q_h & -\delta_e Q_{qh}^T \\ -\delta_e Q_{qh} & \hat{Q}_h \end{pmatrix}^{1/2} \begin{pmatrix} \zeta_q \\ \zeta_h \end{pmatrix} \quad (71)$$

are vectors of size $K+1$, where $y = \pm 1$ is Rademacher and $\zeta_q \sim \mathcal{N}(0, I_{K+1})$, $\zeta_h \sim \mathcal{N}(0, I_{K+1})$, and $\chi \sim \mathcal{N}(0, 1)$ are standard Gaussians. $\bar{s}$ determines whether the loss is active $\bar{s} = 1$ or not $\bar{s} = 0$. We assumed that ℓ is quadratic. Later we will take the limit $r \rightarrow \infty$ where h is small and where ℓ can effectively be expanded around 0 as a quadratic potential. Notice that in the case $\delta_e = 0$ we recover the potential ψ_h Eq. (30) of the previous part.

This potential Eq. (63) corresponds to a one-dimensional interacting chain, involving the positions h and their effective derivative $D_{qh}h$, and with constraints at the two ends, for the loss on h_K and the regularized weights on h_0 . Its extremizer h^* is

$$h^* = G^{-1} (B_h + D_{qh}^T G_0^{-1} B). \quad (72)$$

The order parameters are determined by the following fixed-point equations, obtained by extremizing the free entropy. As before $\mathcal{P}$ acts by linearly combining quantities evaluated at h^* , taken with $\bar{s} = 1$ and $\bar{s} = 0$ with weights ρ and $1 - \rho$.

$$m_w = \frac{1}{\alpha r + \hat{V}_w} \hat{m}_w, \quad (73)$$

$$Q_w = \frac{1}{\alpha (r + \hat{V}_w)^2} \hat{Q}_w + \hat{m}_w^2, \quad (74)$$

$$V_w = \frac{1}{\alpha r + \hat{V}_w}, \quad (75)$$

$$\begin{pmatrix} \hat{m}_w \\ \hat{m} \\ m \\ \cdot \end{pmatrix} = \begin{pmatrix} K\sqrt{\mu} & 0 \\ & \lambda^e t I_K \\ 0 & I_{K+1} \end{pmatrix} \mathbb{E}_{y,\xi,\zeta} \mathcal{P} \left(\begin{pmatrix} G_0^{-1}(D_{qh}h - B) \\ h \end{pmatrix} \right), \quad (76)$$

$$\begin{pmatrix} \hat{Q}_w \\ \hat{Q}_h & Q_{qh} \\ Q_{qh}^T & Q_h \\ \cdot \end{pmatrix} = \begin{pmatrix} K & 0 \\ t I_K & \\ 0 & I_{K+1} \end{pmatrix} \mathbb{E}_{y,\xi,\zeta} \mathcal{P} \left(\left(\begin{pmatrix} G_0^{-1}(D_{qh}h - B) \\ h \end{pmatrix} \right)^{\otimes 2} \right) \begin{pmatrix} K & 0 \\ t I_K & \\ 0 & I_{K+1} \end{pmatrix}, \quad (77)$$

$$\begin{pmatrix} \cdot \\ -iV_{qh} & \cdot \end{pmatrix} = t \mathcal{P}(G_0^{-1} D_{qh} G^{-1}), \quad (78)$$

$$\begin{pmatrix} V_h & \cdot \\ \cdot & \cdot \end{pmatrix} = \mathcal{P}(G^{-1}), \quad (79)$$

$$\begin{pmatrix} \hat{V}_w & \cdot \\ \cdot & \hat{V}_h \end{pmatrix} = \begin{pmatrix} K^2 & 0 \\ 0 & t^2 I_K \end{pmatrix} \mathcal{P}(G_0^{-1} - G_0^{-1} D_{qh} G^{-1} D_{qh}^T G_0^{-1}), \quad (80)$$

where $\cdot$ are unspecified elements that pad the vector to the size $2(K+1)$ and the matrices to the size $2(K+1) \times 2(K+1)$ and $(K+1) \times (K+1)$. On w we assumed l_2 regularization and obtained the same equations as in Sec. III A.

Once a solution to this system is found the train and test accuracies are expressed as

$$\text{Acc}_{\text{train/test}} = \mathbb{E}_{y,\xi,\zeta} \delta_{y=\text{sign}(h_K^*)}, \quad (81)$$

taking $\bar{s} = 1$ or $\bar{s} = 0$.

2. Expansion around large regularization r and continuous limit

Solving the above self-consistent equations (73)–(80) is difficult as such. One can solve them numerically by repeated updates; but this does not allow to go to large K because of numerical instability. One has to invert G Eq. (64) and to make sense of the continuous limit of matrix inverses. This is an issue in the sense that, for a generic $K \times K$ matrix $(M)_{ij}$ whose elements vary smoothly with i and j in the limit of large K , the elements of its inverse M^{-1} are not necessarily continuous with respect to their indices and can vary with a large magnitude.

Our analysis from the previous Sec. III A gives an insight on how to achieve this. It appears that the limit of large regularization $r \rightarrow \infty$ is of particular relevance. In this limit the above system can be solved analytically thanks to an expansion around large r . This expansion is natural in the

sense that it leads to several simplifications and corresponds to expanding the matrix inverses in Neumann series. Keeping the first terms of the expansion the limit $K \rightarrow \infty$ is then well defined. In this section we detail this expansion; we take the continuous limit and, keeping the first constant order, we solve (73)–(80).

In the limit of large regularization h and w are of order $1/r$; the parameters m_w , m , V_w , and V are of order $1/r$ and Q_w and Q are of order $1/r^2$, while all their conjugates, Q_{qh} and V_{qh} are of order one. Consequently, we have $G_0^{-1} \sim r \gg G_h \sim 1$ and we expand G^{-1} around G_0 :

$$G^{-1} = D_{qh}^{-1} G_0 D_{qh}^{-1,T} \sum_{a \geq 0} (-G_h D_{qh}^{-1} G_0 D_{qh}^{-1,T})^a. \quad (82)$$

Constant order. We detail how to solve the self-consistent equations (73)–(80) taking the continuous limit $K \rightarrow \infty$ at the constant order in $1/r$. As we will show later, truncating G^{-1} to the constant order gives predictions that are close to the simulations at finite r , even for $r \approx 1$ if t is not too large. Considering higher orders is feasible but more challenging and we will only provide insights on how to pursue the computation.

The truncated expansion gives, starting from the variances:

$$\begin{pmatrix} \cdot \\ -iV_{qh} & \cdot \end{pmatrix} = t D_{qh}^{-1,T}, \quad (83)$$

$$\begin{pmatrix} V_h & \cdot \\ \cdot & \cdot \end{pmatrix} = D_{qh}^{-1} \begin{pmatrix} K^2 V_w & 0 \\ 0 & t^2 V_h \end{pmatrix} D_{qh}^{-1,T}, \quad (84)$$

$$\begin{pmatrix} \hat{V}_w & \cdot \\ \cdot & \hat{V}_h \end{pmatrix} = \begin{pmatrix} K^2 & 0 \\ 0 & t^2 I_K \end{pmatrix} D_{qh}^{-1,T} \begin{pmatrix} \hat{V}_h & 0 \\ 0 & \rho \end{pmatrix} D_{qh}^{-1}. \quad (85)$$

We kept the order $a = 0$ for V_{qh} and V_h , and the orders $a \leq 1$ for $\hat{V}_w$ and $\hat{V}_h$. We expand $h^* \approx D_{qh}^{-1} G_0 D_{qh}^{-1,T} B_h + D_{qh}^{-1} B$ keeping the order $a = 0$ and obtain the remaining self-consistent equations

$$\begin{pmatrix} \hat{m}_w \\ \hat{m} \end{pmatrix} = \begin{pmatrix} K\sqrt{\mu} & 0 \\ 0 & \lambda^e t I_K \end{pmatrix} D_{qh}^{-1,T} \begin{pmatrix} \hat{m} \\ \rho \end{pmatrix}, \quad (86)$$

$$\begin{pmatrix} m \\ \cdot \end{pmatrix} = D_{qh}^{-1} G_0 D_{qh}^{-1,T} \begin{pmatrix} \hat{m} \\ \rho \end{pmatrix} + D_{qh}^{-1} \begin{pmatrix} K\sqrt{\mu} m_w \\ \lambda^e t m \end{pmatrix}, \quad (87)$$

$$\begin{pmatrix} \hat{Q}_w & \cdot \\ \cdot & \hat{Q}_h \end{pmatrix} = \begin{pmatrix} K & 0 \\ 0 & t I_K \end{pmatrix} D_{qh}^{-1,T} \left(\begin{pmatrix} \hat{Q}_h & 0 \\ 0 & 0 \end{pmatrix} + \rho \begin{pmatrix} \hat{m} \\ 1 \end{pmatrix}^{\otimes 2} + (1 - \rho) \begin{pmatrix} \hat{m} \\ 0 \end{pmatrix}^{\otimes 2} \right) D_{qh}^{-1} \begin{pmatrix} K & 0 \\ 0 & t I_K \end{pmatrix}, \quad (88)$$

$$\begin{pmatrix} \cdot & \cdot \\ -i Q_{qh} & \cdot \end{pmatrix} = t D_{qh}^{-1,T} \left[\left(t \delta_e \begin{pmatrix} 0 & -i Q_{qh} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \hat{m} \\ \rho \end{pmatrix} \begin{pmatrix} K\sqrt{\mu} m_w \\ \lambda^e t m \end{pmatrix}^T \right) D_{qh}^{-1,T} \right. \\ \left. + \left(\begin{pmatrix} \hat{Q}_h & 0 \\ 0 & 0 \end{pmatrix} + \rho \begin{pmatrix} \hat{m} \\ 1 \end{pmatrix}^{\otimes 2} + (1 - \rho) \begin{pmatrix} \hat{m} \\ 0 \end{pmatrix}^{\otimes 2} \right) D_{qh}^{-1} G_0 D_{qh}^{-1,T} \right], \quad (89)$$

$$\begin{pmatrix} Q_h & \cdot \\ \cdot & \cdot \end{pmatrix} = D_{qh}^{-1} G_0 D_{qh}^{-1,T} \left(\begin{pmatrix} \hat{Q}_h & 0 \\ 0 & 0 \end{pmatrix} + \rho \begin{pmatrix} \hat{m} \\ 1 \end{pmatrix}^{\otimes 2} + (1 - \rho) \begin{pmatrix} \hat{m} \\ 0 \end{pmatrix}^{\otimes 2} \right) D_{qh}^{-1} G_0 D_{qh}^{-1,T} \\ + D_{qh}^{-1} \left(\begin{pmatrix} K^2 Q_w & 0 \\ 0 & t^2 Q_h \end{pmatrix} + \begin{pmatrix} K\sqrt{\mu} m_w \\ \lambda^e t m \end{pmatrix}^{\otimes 2} \right) D_{qh}^{-1,T} \\ + D_{qh}^{-1} G_0 D_{qh}^{-1,T} \left(t \delta_e \begin{pmatrix} 0 & -i Q_{qh} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \hat{m} \\ \rho \end{pmatrix} \begin{pmatrix} K\sqrt{\mu} m_w \\ \lambda^e t m \end{pmatrix}^T \right) D_{qh}^{-1,T} \\ + D_{qh}^{-1} \left(t \delta_e \begin{pmatrix} 0 & 0 \\ -i Q_{qh}^T & 0 \end{pmatrix} + \begin{pmatrix} K\sqrt{\mu} m_w \\ \lambda^e t m \end{pmatrix} \begin{pmatrix} \hat{m} \\ \rho \end{pmatrix}^T \right) D_{qh}^{-1} G_0 D_{qh}^{-1,T}. \quad (90)$$

We see that all these self-consistent equations (84)–(90) are vectorial or matricial equations of the form $x = \lambda^e t D_{qh}^{-1} x$ or $X = t^2 D_{qh}^{-1} X D_{qh}^{-1,T}$, over x or X , plus inhomogeneous terms and boundary conditions at 0 or (0,0). The equations are recursive in the sense that each equation only depends on the previous ones and they can be solved one by one. It is thus enough to compute the resolvents of these two equations. Last, Eq. (83) shows how to invert D_{qh} and express D_{qh}^{-1} . These different properties make the system of self-consistent equations easily solvable, provided one can compute D_{qh} and the two resolvents. This furthermore highlights the relevance of the $r \rightarrow \infty$ limit.

We take the continuous limit $K \rightarrow \infty$. We translate the above self-consistent equations into functional equations. Thanks to the expansion around large r we have a well

defined limit, that does not involve any matrix inverse. We set $x = k/K$ and $z = l/K$ continuous indices ranging from 0 to 1. We extend the vectors and the matrices by continuity to match the correct dimensions. We apply the following rescaling to obtain quantities that are independent of K in that limit:

$$\hat{m} \rightarrow K \hat{m}, \quad \hat{Q}_h \rightarrow K^2 \hat{Q}_h, \quad \hat{V}_h \rightarrow K^2 \hat{V}_h, \quad (91)$$

$$Q_{qh} \rightarrow K Q_{qh}, \quad V_{qh} \rightarrow K V_{qh}. \quad (92)$$

We first compute the effective derivative $D_{qh} = D - t \begin{pmatrix} 0 & 0 \\ -i \delta_e V_{qh}^T & 0 \end{pmatrix}$ and its inverse. In the asymmetric case we have $D_{qh} = D$, the usual derivative. In the symmetric case we have $D_{qh} = D - t V_{qh}^T$, where V_{qh} satisfies

Eq. (83), which reads

$$\begin{aligned} \partial_z V_{qh}(x, z) + \delta(z) V_{qh}(x, z) \\ = t\delta(z-x) + t \int_0^1 dx' V_{qh}(x, x') V_{qh}(x', z), \end{aligned} \quad (93)$$

where we multiplied both sides by D_{qh}^T and took $V_{qh}(x, z)$ for $-iV_{qh}$. The solution to this integro-differential equation is

$$V_{qh}(x, z) = \theta(z-x) \frac{I_1[2t(z-x)]}{z-x} \quad (94)$$

with θ the step function and I_ν the modified Bessel function of the second kind of order ν . Consequently, we obtain the effective inverse derivative

$$D_{qh}^{-1}(x, z) = D_{qh}^{-1,T}(z, x) = \begin{cases} \theta(x-z) & \text{if } \delta_e = 0 \\ \frac{1}{t} V_{qh}(z, x) & \text{if } \delta_e = 1 \end{cases} \quad (95)$$

We then define the resolvents (or propagators) φ and Φ of the integral equations as

$$D_{qh}\varphi(x) = \lambda^e t\varphi(x) + \delta(x), \quad (96)$$

$$D_{qh}\Phi(x, z) D_{qh}^T = t^2 \Phi(x, z) + \delta(x, z). \quad (97)$$

Notice that in the asymmetric case, $D_{qh} = \partial_x$, $D_{qh}^T = \partial_z$, and Φ is the propagator of the two-dimensional Klein-Gordon equation up to a change of variables. The resolvents can be expressed as

$$\varphi(x) = \begin{cases} e^{\lambda^e tx} & \text{if } \delta_e = 0 \\ \sum_{\nu>0} \nu (\lambda^e)^{\nu-1} \frac{I_\nu(2tx)}{tx} & \text{if } \delta_e = 1 \end{cases}, \quad (98)$$

$$\Phi(x, z) = \begin{cases} I_0(2t\sqrt{xz}) & \text{if } \delta_e = 0 \\ \frac{I_1(2t(x+z))}{t(x+z)} & \text{if } \delta_e = 1 \end{cases}. \quad (99)$$

We obtain the solution of the self-consistent equations by convolving φ or Φ with the nonhomogeneous terms. We flip $\hat{m}$ along its axis to match the vectorial equation with boundary condition at $x = 0$; we do the same for $\hat{V}_h$ and $\hat{Q}_h$ along their two axes, and for Q_{qh} along its first axis. This gives the following expressions for the order parameters:

$$V_w = \frac{1}{r\alpha}, \quad (100)$$

$$V_h(x, z) = V_w \Phi(x, z), \quad (101)$$

$$\hat{V}_h(1-x, 1-z) = t^2 \rho \Phi(x, z), \quad (102)$$

$$\hat{V}_w = t^{-2} \hat{V}_h(0, 0), \quad (103)$$

$$\hat{m}(1-x) = \rho \lambda^e t \varphi(x), \quad (104)$$

$$\hat{m}_w = \sqrt{\mu} \frac{1}{\lambda^e t} \hat{m}(0), \quad (105)$$

$$m_w = \frac{\hat{m}_w}{r\alpha}, \quad (106)$$

$$\begin{aligned} m(x) = (1+\mu) \frac{m_w}{\sqrt{\mu}} \varphi(x), \\ + \frac{t}{\lambda^e} \int_0^x dx' \int_0^1 dx'' \varphi(x-x') V_h(x', x'') \hat{m}(x''), \end{aligned} \quad (107)$$

$$\hat{Q}_w = t^{-2} \hat{Q}_h(0, 0), \quad (108)$$

$$Q_w = \frac{\hat{Q}_w + \hat{m}_w^2}{r^2 \alpha}, \quad (109)$$

$$\begin{aligned} \hat{Q}_h(1-x, 1-z) = t^2 \int_{0^-, 0^-}^{x, z} dx' dz' \Phi(x-x', z-z') \\ \times [\mathcal{P}(\hat{m}^{\otimes 2})(1-x', 1-z')], \end{aligned} \quad (110)$$

$$\begin{aligned} Q_{qh}(1-x, z) = t \int_{0^-, 0^-}^{x, z} dx' dz' \Phi(x-x', z-z') \\ \times \left[\mathcal{P}(\hat{m})(1-x') [\lambda^e t m(z') + \sqrt{\mu} m_w \delta(z')] \right. \\ \left. + \int_{0, 0^-}^{1^+, 1} dx'' dz'' (\hat{Q}_h(1-x', x'') \right. \\ \left. + \mathcal{P}(\hat{m}^{\otimes 2})(1-x', x'')) D_{qh}^{-1}(x'', z'') G_0(z'', z') \right]. \end{aligned} \quad (111)$$

$$\begin{aligned}
 Q_h(x, z) = & \int_{0^-, 0^-}^{x, z} dx' dz' \Phi(x - x', z - z') \left[\hat{Q}_w \delta(x', z') + [\lambda^e t m(x') + \sqrt{\mu} m_w \delta(x')] [\lambda^e t m(z') + \sqrt{\mu} m_w \delta(z')] \right. \\
 & + \int_{0^-, 0}^{1, 1^+} dx'' dx''' G_0(x', x'') D_{qh}^{-1, T}(x'', x''') (t \delta_e Q_{qh}(x''', z') + \mathcal{P}(\hat{m})(x''') [\lambda^e t m(z') + \sqrt{\mu} m_w \delta(z')]) \\
 & + \int_{0^-, 0}^{1^+, 1} dz''' dz'' (t \delta_e Q_{qh}(z''', x') + [\lambda^e t m(x') + \sqrt{\mu} m_w \delta(x')]) \mathcal{P}(\hat{m})(z''') D_{qh}^{-1}(z''', z'') G_0(z'', z') \\
 & \left. + \int_{0^-, 0, 0, 0^-}^{1, 1^+, 1^+, 1} dx'' dx''' dz''' dz'' G_0(x', x'') D_{qh}^{-1, T}(x'', x''') (\hat{Q}_h(x''', z''') + \mathcal{P}(\hat{m}^{\otimes 2})(x''', z''')) D_{qh}^{-1}(z''', z'') G_0(z'', z') \right]; \quad (112)
 \end{aligned}$$

where we set

$$\mathcal{P}(\hat{m})(x) = \hat{m}(x) + \rho \delta(1 - x), \quad (113)$$

$$\mathcal{P}(\hat{m}^{\otimes 2})(x, z) = \rho(\hat{m}(x) + \delta(1 - x))(\hat{m}(z) + \delta(1 - z)) + (1 - \rho)\hat{m}(x)\hat{m}(z), \quad (114)$$

$$G_0(x, z) = t^2 V_h(x, z) + V_w \delta(x, z), \quad (115)$$

and take $Q_{qh}(x, z)$ for $-iQ_{qh}$. The accuracies are, with $\bar{s} = 1$ for train and $\bar{s} = 0$ for test:

$$\text{Acc}_{\text{train/test}} = \frac{1}{2} \left(1 + \text{erf} \left(\frac{m(1) + (\bar{s} - \rho)V_h(1, 1)}{\sqrt{2}\sqrt{Q_h(1, 1) - m(1)^2 - \rho(1 - \rho)V_h(1, 1)^2}} \right) \right). \quad (116)$$

Notice that we fully solved the model, in a certain limit, by giving an explicit expression of the performance of the GCN. This is an uncommon result in the sense that, in several works analyzing the performance of neural networks in a high-dimensional limit, the performance are only expressed as the function of the self-consistent of a system of equations similar to ours (73)–(80). These systems have to be solved numerically, which may be unsatisfactory for the understanding of the studied models.

So far, we dealt with infinite regularization r keeping only the first constant order. The predicted accuracy (116) does not depend on r . We briefly show how to pursue the computation at any order in Appendix B 4, by a perturbative approach with expansion in powers of $1/r$.

Interpretation in terms of dynamical mean-field theory. The order parameters V_h , V_{qh} , and $\hat{V}_h$ come from the replica computation and were introduced as the covariances between h and its conjugate q . Their values are determined by extremizing the free entropy of the problem. In the above lines we derived that $V_h(x, z) \propto \Phi(x, z)$ is the forward propagator, from the weights to the loss, while $\hat{V}_h(x, z) \propto \Phi(1 - x, 1 - z)$ is the backward propagator, from the loss to the weights.

In this section we state an equivalence between these order parameters and the correlation and response functions of the dynamical process followed by h .

We introduce the tilting field $\eta(x) \in \mathbb{R}^N$ and the tilted Hamiltonian as

$$\frac{dh}{dx}(x) = \frac{t}{\sqrt{N}} \tilde{A}^e h(x) + \eta(x) + \delta(x) \frac{1}{\sqrt{N}} Xw, \quad (117)$$

$$h(x) = \int_{0^-}^x dx' e^{(x-x')\frac{t}{\sqrt{N}} \tilde{A}^e} \left(\eta(x') + \delta(x') \frac{1}{\sqrt{N}} Xw \right), \quad (118)$$

$$H(\eta) = \frac{1}{2} [y - h(1)]^T R [y - h(1)] + \frac{r}{2} w^T w, \quad (119)$$

where $R \in \mathbb{R}^{N \times N}$ diagonal accounts for the train and test nodes. We write $\langle \cdot \rangle_\beta$ the expectation under the density $e^{-\beta H(\eta)} / Z$ (normalized only at $\eta = 0$).

Then we have

$$V_h(x, z) = \frac{\beta}{N} \text{Tr} [\langle h(x) h(z)^T \rangle_\beta - \langle h(x) \rangle_\beta \langle h(z)^T \rangle_\beta] \Big|_{\eta=0}, \quad (120)$$

$$V_{qh}(x, z) = \frac{t}{N} \text{Tr} \frac{\partial}{\partial \eta(z)} \langle h(x) \rangle_\beta \Big|_{\eta=0}, \quad (121)$$

$$\hat{V}_h(x, z) = \frac{t^2}{\beta^2 N} \text{Tr} \frac{\partial^2}{\partial \eta(x) \partial \eta(z)} \langle 1 \rangle_\beta \Big|_{\eta=0}; \quad (122)$$

that is to say V_h is the correlation function, $V_{qh} \approx tD_{qh}^{-1,T}$ is the response function and $\hat{V}_h$ is the correlation function of the responses of h . We prove these equalities at the constant order in r using random matrix theory in Appendix B 5.

3. Consequences

Convergences. We compare our predictions to numerical simulations of the continuous GCN for $N = 10^4$ and $N = 7 \times 10^3$ in Fig. 5 and Figs. 8, 10, and 11 in Appendix E. The predicted test accuracies are well within the statistical errors. On these figures we can observe the convergence of Acc_{test} with respect to r . The interversion of the two limits $r \rightarrow \infty$ and $K \rightarrow \infty$ we did to obtain (116) seems valid. Indeed on the figures we simulate the continuous GCN with $\frac{r\tilde{A}}{e\sqrt{N}}$ or $\frac{r\tilde{A}^S}{e\sqrt{N}}$ and take $r \rightarrow \infty$ after the continuous limit $K \rightarrow \infty$; and we observe that the simulated accuracies converge well toward the predicted ones. To keep only the constant order in $1/r$ gives a good approximation of the continuous GCN. Indeed, the convergence with respect to $1/r$ can be fast: at $t \lesssim 1$ not too large, $r \gtrsim 1$ is enough to reach the continuous limit.

The convergence with respect to $K \rightarrow \infty$, taken after $r \rightarrow \infty$, is depicted in Figs. 6 and 9 in Appendix E. Again the continuous limit enjoys good convergence properties since $K \gtrsim 16$ is enough if t is not too large.

To summarize, Figs. 5, 6, and Appendix E validate our method that consists in deriving the self-consistent equations at finite K with replica, expanding them with respect to $1/r$, taking the continuous limit $K \rightarrow \infty$ and then solving the integral equations.

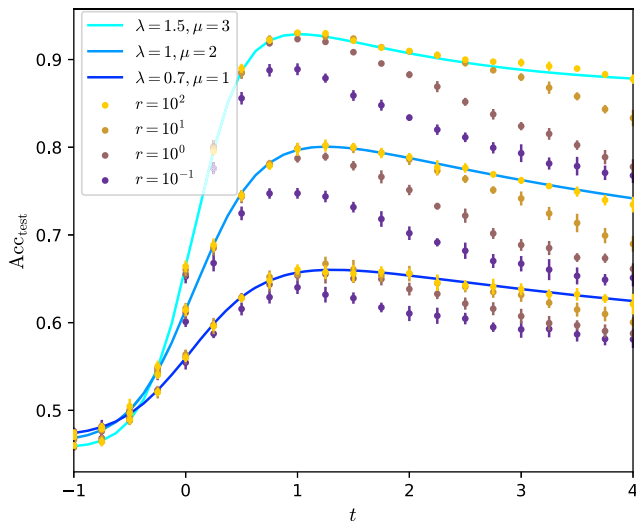


FIG. 5. Predicted test accuracy Acc_{test} of the continuous GCN on the asymmetric graph, at $r = \infty$, $\alpha = 4$ and $\rho = 0.1$. The performance of the continuous GCN are given by Eq. (116). Dots: numerical simulation of the continuous GCN for $N = 10^4$ and $d = 30$, trained with quadratic loss, averaged over ten experiments.

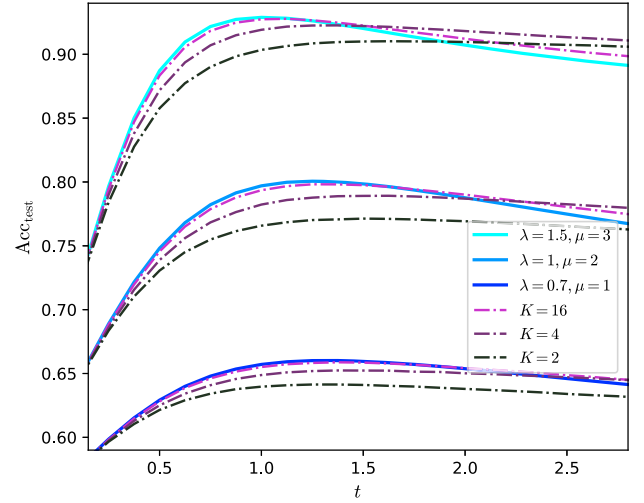


FIG. 6. Predicted test accuracy Acc_{test} of the continuous GCN and of its discrete counterpart with depth K on the asymmetric graph, at $r = \infty$, $\alpha = 1$ and $\rho = 0.1$. The performance of the continuous GCN are given by Eq. (116) while for the discrete GCN they are given by numerically solving the fixed-point Eqs. (84)–(90).

Optimal diffusion time t^* . We observe on the previous figures that there is an optimal diffusion time t^* that maximizes Acc_{test} . Though we are able to solve the self-consistent equations and to obtain an explicit and analytical expression (116), it is hard to analyze it in order to evaluate t^* . We have to consider further limiting cases or to compute t^* numerically. The derivation of the following equations is detailed in Appendix B 6.

We first consider the case $t \rightarrow 0$. Expanding (116) to the first order in t we obtain

$$\text{Acc}_{\text{test}} \underset{t \rightarrow 0}{=} \frac{1}{2} \left(1 + \text{erf} \left(\frac{1}{\sqrt{2}} \sqrt{\frac{\rho \mu + \lambda^e t (2 + \mu)}{\alpha \sqrt{1 + \rho \mu}}} \right) \right) + o(t). \quad (123)$$

This expression shows in particular that $t^* > 0$, i.e., some diffusion on the graph is always beneficial compared to no diffusion, as long as $\lambda t > 0$, i.e., the diffusion is done forward if the graph is homophilic $\lambda > 0$ and backward if it is heterophilic $\lambda < 0$. We recover the result of [40] for the discrete case in a slightly different setting. This holds even if the features of the graph are not informative $\mu = 0$. Notice the explicit invariance by the change $(\lambda, t) \rightarrow (-\lambda, -t)$ in the potential (63) and in (123), which allows us to focus on $\lambda \geq 0$. The case $t = 0$ no diffusion corresponds to performing ridge regression on the Gaussian mixture X alone. Such a model has been studied in [37]; we checked we obtain the same expression as theirs at large regularization.

We now consider the case $t \rightarrow +\infty$ and $\lambda \geq 0$. Taking the limit in (116) we obtain

$$\text{Acc}_{\text{test}} \xrightarrow{t \rightarrow \infty} \frac{1}{2} \left(1 + \text{erf} \left(\frac{\lambda^e q_{\text{PCA}}}{\sqrt{2}} \right) \right), \quad (124)$$

where q_{PCA} is the same as for the discrete GCN, defined in Eq. (57). This shows that the continuous GCN will over-smooth at large diffusion times. Thus, if the features are informative, if $\mu^2/\alpha > 0$, the optimal diffusion time should be finite, $t^* < +\infty$. The continuous GCN does exactly as does the discrete GCN at $K \rightarrow \infty$ if c is fixed. This is not surprising because of the mapping $c = K/t$: taking t large is equivalent to take c small with respect to K . $e^{\frac{t}{\sqrt{N}}\tilde{A}}$ is dominated by the same leading eigenvector y_1 .

These two limits show that at finite time t the GCN avoids oversmoothing and interpolates between an estimator that is only function of the features at $t = 0$ and an estimator only function of the graph at $t = \infty$. t has to be fine-tuned to reach the best trade-off t^* and the optimal performance.

In the insets of Figs. 7 and 12 in Appendix E we show how t^* depends on λ . In particular, t^* is finite for any λ : some diffusion is always beneficial but too much diffusion leads to oversmoothing. We have $t^* \xrightarrow{\lambda \rightarrow 0} 0$. This is expected since if $\lambda = 0$ then A is not informative and any diffusion $t > 0$ would degrade the performance. The nonmonotonicity of t^* with respect to λ is less expected and we do not have a clear interpretation for it. Last t^* decreases when the feature signal μ^2/α increases: the more informative X the less needed diffusion is.

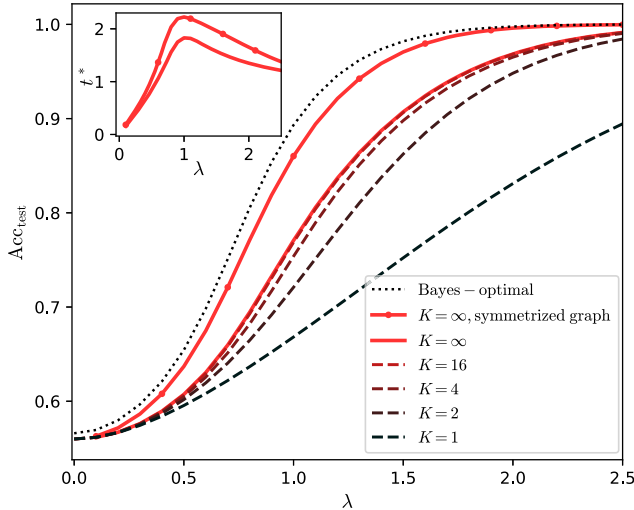


FIG. 7. Predicted test accuracy Acc_{test} of the continuous GCN and of its discrete counterpart with depth K , at optimal times t^* and $r = \infty$. $\alpha = 4$, $\mu = 1$, and $\rho = 0.1$. The performance of the continuous GCN $K = \infty$ are given by Eq. (116) while for its discretization at finite K they are given by numerically solving Eqs. (83)–(90). Inset: t^* the maximizer at $K = \infty$.

Optimality of the continuous limit. A major result is that, at $t = t^*$, the continuous GCN is better than any fixed- K GCN. Taking the continuous limit of the simple GCN is the way to reach its optimal performance. This was suggested by Fig. 4 in the previous part; we show this more precisely in Figs. 7 and 12 in Appendix E. We compare the continuous GCN to its discretization at different depths K for several configurations α , λ , μ , and ρ of the data model. The result is that at t^* the test accuracy appears to be always an increasing function of K , and that its value at $K \rightarrow \infty$ and t^* is an upper bound for all K and t .

Additionally, if the GCN is run on the symmetrized graph it can approach the Bayes optimality and almost close the gap that [26] describes, as shown by Figs. 7, 12, and 13 right. For all the considered λ and μ the GCN is less than a few percents of accuracy far from the optimality.

However, we shall precise this statement: the GCN approaches the Bayes optimality only for a certain range of the parameters of the CSBM, as exemplified by Figs. 12 and 13 left. In these figures, the GCN is far from the Bayes optimality when λ is small but μ is large. In this regime we have $\text{SNR}_{\text{CSBM}} > 1$; even at $\rho = 0$ information can be retrieved on the labels and the problem is closer to an unsupervised classification of the sole features X . On X the GCN acts as a supervised classifier, and as long as $\rho \neq 1$ it cannot catch all information. As previously highlighted by [39] the comparison with the Bayes optimality is more relevant at $\text{SNR}_{\text{CSBM}} < 1$ where supervision is necessary. Then, as shown by Figs. 7, 12, and 13 the symmetrized continuous GCN is close to the Bayes optimality. The GCN is also able to close the gap in the region where λ is large because, as we saw, it can perform unsupervised PCA on A .

IV. CONCLUSION

In this article we derived the performance of a simple GCN trained for node classification in a semisupervised way on data generated by the CSBM in the high-dimensional limit. We first studied a discrete network with a finite number K of convolution steps. We showed the importance of going to large K to approach the Bayes optimality, while scaling accordingly the residual connections c of the network to avoid oversmoothing. The resulting limit is a continuous GCN.

In a second part we were able to explicitly derive the performance of the continuous GCN. We highlighted the importance of the double limit $r, K \rightarrow \infty$, which allows us to reach the optimal architecture and which can be analyzed thanks to an expansion in powers of $1/r$. It is an interesting question for future work whether this approach could allow the study of fully connected large-depth neural networks.

Though the continuous GCN can be close to the Bayes optimality, it has to better handle the features, especially when they are the main source of information.

ACKNOWLEDGMENTS

We acknowledge useful discussions with J. Zavattone-Veth, F. Zamponi, and V. Erba. This work is supported by the Swiss National Science Foundation under grant SNSF SMArtNet (Grant No. 212049).

DATA AVAILABILITY

The data that support the findings of this article are openly available [51].

APPENDIX A: ASYMPTOTIC CHARACTERIZATION OF THE DISCRETE GCN

In this Appendix we compute the free energy of the discrete finite- K GCN using replica. We derive the fixed-point equations for the order parameters of the problem and the asymptotic characterization of the errors and accuracies in function of the order parameters. We consider only the asymmetric graph $\tilde{A}$; the symmetrized case $\tilde{A}^s$ is analyzed in Appendix B together with the continuous GCN.

The free energy of the problem is $-\beta N f = \partial_n \mathbb{E}_{u,\Xi,W,y} Z^n(n=0)$, where the partition function is

$$Z = \int \prod_{\nu} dw_{\nu} e^{-\beta r_{\nu}(w_{\nu})} e^{-\beta s \sum_{i \in R} \ell[y_i h(w)_i] - \beta s' \sum_{i \in R'} \ell[y_i h(w)_i]}. \quad (\text{A1})$$

To lighten the notations we take $\rho' = 1 - \rho$, i.e., the test set is the whole complementary of the train set. This does not change the result since the performances do not depend on the size of the test set.

We recall that $\tilde{A}$ admits the following Gaussian equivalent:

$$\tilde{A} \approx A^g = \frac{\lambda}{\sqrt{N}} y y^T + \Xi, \quad \Xi_{ij} \sim \mathcal{N}(0, 1). \quad (\text{A2})$$

$\tilde{A}$ can be approximated by A^g with a vanishing change in the free energy f .

1. Derivation of the free energy

We define the intermediate states of the GCN as

$$h_k = \left(\frac{1}{\sqrt{N}} \tilde{A} + c_k I_N \right) h_{k-1}, \quad h_0 = \frac{1}{\sqrt{N}} X w. \quad (\text{A3})$$

We introduce them in Z thanks to Dirac deltas. The expectation of the replicated partition function is

$$\begin{aligned} \mathbb{E} Z^n &\propto \mathbb{E}_{u,\Xi,W,y} \int \prod_a^n \prod_{\nu} dw_{\nu}^a e^{-\beta r_{\nu}(w_{\nu}^a)} \prod_a^n \prod_i \prod_{k=0}^K dh_{i,k}^a dq_{i,k}^a e^{-\beta s \sum_{a,i \in R} \ell(y_i h_{i,K}^a) - \beta s' \sum_{a,i \in R'} \ell(y_i h_{i,K}^a)} \\ &\times e^{\sum_{a,i} \sum_{k=1}^K i q_{i,k}^a (h_{i,k}^a - \frac{1}{\sqrt{N}} \sum_j (\frac{\lambda}{\sqrt{N}} y_i y_j + \Xi_{ij}) h_{j,k-1}^a - c_k h_{i,k-1}^a) + \sum_{a,i} i q_{i,0}^a (h_{i,0}^a - \frac{1}{\sqrt{N}} \sum_{\nu} (\sqrt{\frac{E}{N}} y_i u_{\nu} + W_{i\nu}) w_{\nu}^a)} \end{aligned} \quad (\text{A4})$$

$$\begin{aligned} &= \mathbb{E}_{u,y} \int \prod_{a,\nu} dw_{\nu}^a e^{-\beta r_{\nu}(w_{\nu}^a)} \prod_{a,i,k} dh_{i,k}^a e^{-\beta s \sum_{a,i \in R} \ell(y_i h_{i,K}^a) - \beta s' \sum_{a,i \in R'} \ell(y_i h_{i,K}^a)} \\ &\times \prod_i \mathcal{N}\left(h_{i,>0} | c \odot h_{i,<K} + y_i \frac{\lambda}{N} \sum_j y_j h_{j,<K}; \tilde{Q}\right) \prod_i \mathcal{N}\left(h_{i,0} | y_i \frac{\sqrt{\mu}}{N} \sum_{\nu} u_{\nu} w_{\nu}; \frac{1}{N} \sum_{\nu} w_{\nu} w_{\nu}^T\right). \end{aligned} \quad (\text{A5})$$

$\mathcal{N}(\cdot | m; V)$ is the Gaussian density of mean m and covariance V . We integrated over the random fluctuations Ξ and W and then over the q s. We collected the replica in vectors of size n and assembled them as

$$h_{i,>0} = \begin{pmatrix} h_{i,1} \\ \vdots \\ h_{i,K} \end{pmatrix} \in \mathbb{R}^{nK}, \quad h_{i,<K} = \begin{pmatrix} h_{i,0} \\ \vdots \\ h_{i,K-1} \end{pmatrix} \in \mathbb{R}^{nK}, \quad c \odot h_{i,<K} = \begin{pmatrix} c_1 h_{i,0} \\ \vdots \\ c_K h_{i,K-1} \end{pmatrix}, \quad (\text{A6})$$

$$\tilde{Q}_{k,l} = \frac{1}{N} \sum_j h_{j,k} h_{j,l}^T, \quad \tilde{Q} = \begin{pmatrix} \tilde{Q}_{0,0} & \cdots & \tilde{Q}_{0,K-1} \\ \vdots & & \vdots \\ \tilde{Q}_{K-1,0} & \cdots & \tilde{Q}_{K-1,K-1} \end{pmatrix} \in \mathbb{R}^{nK \times nK}. \quad (\text{A7})$$

We introduce the order parameters

$$m_w^a = \frac{1}{N} \sum_{\nu} u_{\nu} w_{\nu}^a, \quad Q_w^{ab} = \frac{1}{N} \sum_{\nu} w_{\nu}^a w_{\nu}^b, \quad (\text{A8})$$

$$m_k^a = \frac{1}{N} \sum_j y_j k_{j,k}^a, \quad Q_k^{ab} = (\tilde{Q}_{k,k})_{a,b} = \frac{1}{N} \sum_j h_{j,k}^a h_{j,k}^b, \quad Q_{k,l}^{ab} = (\tilde{Q}_{k,l})_{a,b} = \frac{1}{N} \sum_j h_{j,k}^a h_{j,l}^b. \quad (\text{A9})$$

m_k is the magnetization (or overlap) between the k th layer and the labels; m_w is the magnetization between the weights and the hidden variables and the Q 's are the self-overlaps across the different layers. In the following we write $\tilde{Q}$ for the matrix with elements $(\tilde{Q})_{ak,bl} = Q_{k,l}^{ab}$. We introduce these quantities thanks to new Dirac deltas. This allows us to factorize the spacial i and ν indices.

$$\begin{aligned} \mathbb{E} Z^n &\propto \int \prod_a \prod_{k=0}^{K-1} d\hat{m}_k^a d m_k^a e^{N \hat{m}_k^a m_k^a} \prod_a d\hat{m}_w^a d m_w^a e^{N \hat{m}_w^a m_w^a} \prod_{a \leq b} \prod_{k=0}^{K-1} d\hat{Q}_k^{ab} d Q_k^{ab} e^{N \hat{Q}_k^{ab} Q_k^{ab}} \prod_{a,b} \prod_{k < l} d\hat{Q}_{k,l}^{ab} d Q_{k,l}^{ab} e^{N \hat{Q}_{k,l}^{ab} Q_{k,l}^{ab}} \\ &\times \prod_{a \leq b} d\hat{Q}_w^{ab} d Q_w^{ab} e^{N \hat{Q}_w^{ab} Q_w^{ab}} \left[\mathbb{E}_u \int \prod_a d w^a e^{\psi_w^{(n)}(w)} \right]^{\frac{N}{2}} \left[\mathbb{E}_y \int \prod_{a,k} d h_k^a e^{\psi_h^{(n)}(h;s)} \right]^{\rho N} \left[\mathbb{E}_y \int \prod_{a,k} d h_k^a e^{\psi_h^{(n)}(h;s')} \right]^{(1-\rho)N}, \end{aligned} \quad (\text{A10})$$

where we defined the two potentials

$$\psi_w^{(n)}(w) = -\beta r \sum_a \gamma(w^a) - \sum_{a \leq b} \hat{Q}_w^{ab} w^a w^b - \sum_a \hat{m}_w^a u w^a, \quad (\text{A11})$$

$$\begin{aligned} \psi_h^{(n)}(h; \bar{s}) &= -\beta \bar{s} \sum_a \ell(y h_K^a) - \sum_{a \leq b} \sum_{k=0}^{K-1} \hat{Q}_k^{ab} h_k^a h_k^b - \sum_{a,b} \sum_{k < l} \hat{Q}_{k,l}^{ab} h_k^a h_l^b - \sum_a \sum_{k=0}^{K-1} \hat{m}_k^a y h_k^a \\ &+ \log \mathcal{N}(h_{>0} | c \odot h_{<K} + \lambda y m_{<K}; \tilde{Q}) + \log \mathcal{N}(h_0 | \sqrt{\mu} y m_w; Q_w). \end{aligned} \quad (\text{A12})$$

We leverage the replica-symmetric ansatz. It is justified by the convexity of the Hamiltonian H . We assume that for all a and b

$$m_k^a = m_k, \quad \hat{m}_k^a = -\hat{m}_k, \quad m_w^a = m_w, \quad \hat{m}_w^a = -\hat{m}_w, \quad (\text{A13})$$

$$Q_k^{ab} = Q_k J + V_k I, \quad \hat{Q}_k^{ab} = -\hat{Q}_k J + \frac{1}{2}(\hat{V}_k + \hat{Q}_k) I, \quad Q_w^{ab} = Q_w J + V_w I, \quad \hat{Q}_w^{ab} = -\hat{Q}_w J + \frac{1}{2}(\hat{V}_w + \hat{Q}_w) I, \quad (\text{A14})$$

$$Q_{k,l}^{ab} = Q_{k,l} J + V_{k,l} I, \quad \hat{Q}_{k,l}^{ab} = -\hat{Q}_{k,l} J + \hat{V}_{k,l} I. \quad (\text{A15})$$

I is the $n \times n$ identity and J is the $n \times n$ matrix filled with ones. We introduce the $K \times K$ symmetric matrices Q and V , filled with $(Q_k)_{0 \leq k \leq K-1}$ and $(V_k)_{0 \leq k \leq K-1}$ on the diagonal, and $(Q_{k,l})_{0 \leq k < l \leq K-1}$ and $(V_{k,l})_{0 \leq k < l \leq K-1}$ off the diagonal, such that $\tilde{Q}$ can be written in terms of Kronecker products as

$$\tilde{Q} = Q \otimes J + V \otimes I. \quad (\text{A16})$$

The entropic terms of $\psi_w^{(n)}$ and $\psi_h^{(n)}$ can be computed. Since we will take $n = 0$ we discard subleading terms in n . We obtain

$$\sum_a \hat{m}_w^a m_w^a = n \hat{m}_w m_w, \quad \sum_{a \leq b} \hat{Q}_w^{ab} Q_w^{ab} = \frac{n}{2} (\hat{V}_w V_w + \hat{V}_w Q_w - V_w \hat{Q}_w), \quad (\text{A17})$$

$$\sum_a \hat{m}_k^a m_k^a = n \hat{m}_k m_k, \quad \sum_{a \leq b} \hat{Q}_k^{ab} Q_k^{ab} = \frac{n}{2} (\hat{V}_k V_k + \hat{V}_k Q_k - V_k \hat{Q}_k), \quad \sum_{a,b} \hat{Q}_{k,l}^{ab} Q_{k,l}^{ab} = n (\hat{V}_{k,l} V_{k,l} + \hat{V}_{k,l} Q_{k,l} - V_{k,l} \hat{Q}_{k,l}). \quad (\text{A18})$$

The Gaussian densities can be explicitied, keeping again the main order in n and using the formula for a rank-1 update to a matrix (Sherman-Morrison formula):

$$Q_w^{-1} = \frac{1}{V_w} I - \frac{Q_w}{V_w^2} J, \quad \log \det Q_w = n \frac{Q_w}{V_w} + n \log V_w, \quad (\text{A19})$$

$$\tilde{Q}^{-1} = V^{-1} \otimes I - (V^{-1} Q V^{-1}) \otimes J, \quad \log \det \tilde{Q} = n \text{Tr}(V^{-1} Q) + n \log V. \quad (\text{A20})$$

Then we can factorize the replica by introducing random Gaussian variables:

$$\int \prod_a dw^a e^{\psi_w^{(n)}(w)} = \int \prod_a dw^a e^{\sum_a \log P_w(w^a) + \frac{1}{2} \hat{Q}_w w^T J w - \frac{1}{2} \hat{V}_w w^T w + u \hat{m}_w^T w} = \mathbb{E}_\zeta \left(\int dw e^{\psi_w(w)} \right)^n, \quad (\text{A21})$$

where $\zeta \sim \mathcal{N}(0, 1)$ and the potential is

$$\psi_w(w) = \log P_w(w) - \frac{1}{2} \hat{V}_w w^2 + \left(\sqrt{\hat{Q}_w \zeta} + u \hat{m}_w \right) w; \quad (\text{A22})$$

and, similarly,

$$\begin{aligned} & \int \prod_{a,k} dh_k^a e^{\psi_h^{(n)}(h; \bar{s})} \\ &= \int \prod_{a,k} dh_k^a e^{-\beta \bar{s} \sum_a \mathcal{L}(y k_K^a) + \sum_{k=0}^{K-1} \left(\frac{1}{2} \hat{Q}_k h_k^T J h_k - \frac{1}{2} \hat{V}_k h_k^T h_k + y \hat{m}_k^T h_k \right) + \sum_{k < l}^{K-1} \left(\hat{Q}_{k,l} h_k^T J h_l - \hat{V}_{k,l} h_k^T h_l \right)} e^{-\frac{1}{2} (h_0 - \sqrt{\mu} y m_w)^T \left(\frac{1}{V_w} I - \frac{Q_w}{V_w^2} J \right) (h_0 - \sqrt{\mu} y m_w) - \frac{1}{2} n \frac{Q_w}{V_w} - \frac{1}{2} n \log V_w} \\ & \times e^{-\frac{1}{2} \sum_{k,l} (h_k - c_k h_{k-1} - \lambda y m_{k-1})^T ((V^{-1})_{k-1,l-1} I - (V^{-1} Q V^{-1})_{k-1,l-1} J) (h_l - c_l h_{l-1} - \lambda y m_{l-1}) - \frac{n}{2} \text{Tr}(V^{-1} Q) - \frac{n}{2} \log \det V} \end{aligned} \quad (\text{A23})$$

$$= \mathbb{E}_{\xi, \chi, \zeta} \left(\int \prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s})} \right)^n, \quad (\text{A24})$$

where $\xi \sim \mathcal{N}(0, I_K)$, $\chi \sim \mathcal{N}(0, I_K)$, $\zeta \sim \mathcal{N}(0, 1)$ and the potential is

$$\begin{aligned} \psi_h(h; \bar{s}) &= -\beta \bar{s} \mathcal{L}(y h_K) - \frac{1}{2} h_{<K}^T \hat{V} h_{<K} + (\xi^T \hat{Q}^{1/2} + y \hat{m}^T) h_{<K} \\ &+ \log \mathcal{N}(h_0 | \sqrt{\mu} y m_w + \sqrt{Q_w} \zeta; V_w) + \log \mathcal{N}(h_{>0} | c \odot h_{<K} + \lambda y m + Q^{1/2} \chi; V); \end{aligned} \quad (\text{A25})$$

where $h_{>0} = (h_1, \dots, h_K) \in \mathbb{R}^K$, $h_{<K} = (h_0, \dots, h_{K-1}) \in \mathbb{R}^K$, $c \odot h_{<K} = (c_1 h_0, \dots, c_K h_{K-1})$, $\hat{m} = (\hat{m}_0, \dots, \hat{m}_{K-1}) \in \mathbb{R}^K$, $m = (m_0, \dots, m_{K-1}) \in \mathbb{R}^K$, $\hat{Q}$ and $\hat{V}$ are the $K \times K$ symmetric matrix filled with $(\hat{Q}_k)_{0 \leq k \leq K-1}$ and $(\hat{V}_k)_{0 \leq k \leq K-1}$ on the diagonal, and $(\hat{Q}_{k,l})_{0 \leq k < l \leq K-1}$ and $(\hat{V}_{k,l})_{0 \leq k < l \leq K-1}$ off the diagonal. We used that $\mathbb{E}_\zeta e^{-\frac{n Q_w \zeta^2}{2 V_w}} = e^{-\frac{n Q_w}{2 V_w}}$ in the limit $n \rightarrow 0$ to factorize $\sqrt{Q_w} \zeta$ and the same for $Q^{1/2} \chi$.

We pursue the computation

$$\begin{aligned} \mathbb{E} Z^n &\propto \int d\hat{m}_w dm_w e^{N n \hat{m}_w^T m_w} d\hat{Q}_w dQ_w d\hat{V}_w dV_w e^{N \frac{n}{2} (\hat{V}_w V_w + \hat{V}_w Q_w - V_w \hat{Q}_w)} \\ &\times \prod_{k=0}^{K-1} d\hat{m}_k dm_k e^{N n \hat{m}_k^T m_k} \prod_{k=0}^{K-1} d\hat{Q}_k dQ_k d\hat{V}_k dV_k \prod_{k < l}^{K-1} d\hat{Q}_{k,l} dQ_{k,l} d\hat{V}_{k,l} dV_{k,l} e^{N \frac{n}{2} \text{tr}(\hat{V} V + \hat{V} Q - V \hat{Q})} \\ &\times \left[\mathbb{E}_{u, \zeta} \left(\int dw e^{\psi_w(w)} \right)^n \right]^{N/\alpha} \left[\mathbb{E}_{y, \xi, \chi, \zeta} \left(\int \prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s})} \right)^n \right]^{\rho N} \left[\mathbb{E}_{y, \xi, \chi, \zeta} \left(\int \prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s}')} \right)^n \right]^{(1-\rho)N} \end{aligned} \quad (\text{A26})$$

$$:= \int d\Theta d\hat{\Theta} e^{N \phi^{(n)}(\Theta, \hat{\Theta})}, \quad (\text{A27})$$

where $\Theta = \{m_w, Q_w, V_w, m, Q, V\}$ and $\hat{\Theta} = \{\hat{m}_w, \hat{Q}_w, \hat{V}_w, \hat{m}, \hat{Q}, \hat{V}\}$ are the sets of the order parameters. We can now take the limit $N \rightarrow \infty$ thanks to Laplace's method.

$$-\beta f \propto \frac{1}{N} \frac{\partial}{\partial n} (n=0) \int d\Theta d\hat{\Theta} e^{N\phi^{(n)}(\Theta, \hat{\Theta})} \quad (\text{A28})$$

$$= \text{extr}_{\Theta, \hat{\Theta}} \frac{\partial}{\partial n} (n=0) \phi^{(n)}(\Theta, \hat{\Theta}) \quad (\text{A29})$$

$$:= \text{extr}_{\Theta, \hat{\Theta}} \phi(\Theta, \hat{\Theta}), \quad (\text{A30})$$

where we extremize the following free entropy ϕ :

$$\begin{aligned} \phi = & \frac{1}{2} (\hat{V}_w V_w + \hat{V}_w Q_w - V_w \hat{Q}_w) - \hat{m}_w m_w \\ & + \frac{1}{2} \text{tr}(\hat{V} V + \hat{V} Q - V \hat{Q}) - \hat{m}^T m \\ & + \frac{1}{\alpha} \mathbb{E}_{u, \xi} \left(\log \int dw e^{\psi_w(w)} \right) \\ & + \rho \mathbb{E}_{y, \xi, \zeta, \chi} \left(\log \int \prod_{k=0}^K dh_k e^{\psi_h(h; s)} \right) \\ & + (1 - \rho) \mathbb{E}_{y, \xi, \zeta, \chi} \left(\log \int \prod_{k=0}^K dh_k e^{\psi_h(h; s')} \right). \quad (\text{A31}) \end{aligned}$$

We take the limit $\beta \rightarrow \infty$. Later we will differentiate ϕ with respect to the order parameters or to $\bar{s}$ and these derivatives will simplify in that limit. We introduce the measures

$$\begin{aligned} dP_w &= \frac{dw e^{\psi_w(w)}}{\int dw e^{\psi_w(w)}}, \quad dP_h = \frac{\prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s}=1)}}{\int \prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s}=1)}}, \\ dP'_h &= \frac{\prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s}=0)}}{\int \prod_{k=0}^K dh_k e^{\psi_h(h; \bar{s}=0)}}. \quad (\text{A32}) \end{aligned}$$

We have to rescale the order parameters not to obtain a degenerated solution when $\beta \rightarrow \infty$ (we recall that, in ψ_w , $\log P_w(w) \propto \beta$). We take

$$\hat{m}_w \rightarrow \beta \hat{m}_w, \quad \hat{Q}_w \rightarrow \beta^2 \hat{Q}_w, \quad \hat{V}_w \rightarrow \beta \hat{V}_w, \quad V_w \rightarrow \beta^{-1} V_w, \quad (\text{A33})$$

$$\hat{m} \rightarrow \beta \hat{m}, \quad \hat{Q} \rightarrow \beta^2 \hat{Q}, \quad \hat{V} \rightarrow \beta \hat{V}, \quad V \rightarrow \beta^{-1} V. \quad (\text{A34})$$

So we obtain that $f = -\phi$. Then dP_w , dP_h , and dP'_h are picked around their maximum and can be approximated by Gaussian measures. We define

$$\begin{aligned} w^* &= \underset{w}{\text{argmax}} \psi_w(w), \quad h^* = \underset{h}{\text{argmax}} \psi_h(h; \bar{s} = 1), \\ h'^* &= \underset{h}{\text{argmax}} \psi_h(h; \bar{s} = 0). \quad (\text{A35}) \end{aligned}$$

Then we have the expected value of a function g in h $\mathbb{E}_{P_h} g(h) = g(h^*)$ and the covariance $\text{Cov}_{P_h}(h) = -\frac{1}{2} [\nabla \nabla \psi_h(h^*)]^{-1}$ with $\nabla \nabla$ the Hessian; and similarly for dP_w and dP'_h .

Last we compute the expected errors and accuracies. We differentiate the free energy f with respect to s and s' to obtain that

$$E_{\text{train}} = \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{L}(y h_K^*), \quad E_{\text{test}} = \mathbb{E}_{y, \xi, \zeta, \chi} \mathcal{L}(y h_K'^*). \quad (\text{A36})$$

Augmenting H with the observable $(1/|\hat{R}|) \times \sum_{i \in \hat{R}} \delta_{y_i = \text{sign}(w)_i}$ and following the same steps gives the expected accuracies

$$\text{Acc}_{\text{train}} = \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y = \text{sign}(h_K^*)}, \quad (\text{A37})$$

$$\text{Acc}_{\text{test}} = \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y = \text{sign}(h_K'^*)}. \quad (\text{A38})$$

2. Self-consistent equations

The above two formulas (A36) and (A37) are valid only at the values of the order parameters that extremize the free entropy. We seek the extremizer of ϕ . The extremality condition $\nabla_{\Theta, \hat{\Theta}} \phi = 0$ gives the following self-consistent equations:

$$m_w = \frac{1}{\alpha} \mathbb{E}_{u, \xi} u w^*, \quad (\text{A39})$$

$$Q_w = \frac{1}{\alpha} \mathbb{E}_{u, \xi} (w^*)^2, \quad (\text{A40})$$

$$V_w = \frac{1}{\alpha} \frac{1}{\sqrt{\hat{Q}_w}} \mathbb{E}_{u, \xi} \zeta w^*, \quad (\text{A41})$$

$$m = \mathbb{E}_{y, \xi, \zeta, \chi} y (\rho h_{<K}^* + (1 - \rho) h_{<K}'^*), \quad (\text{A42})$$

$$Q = \mathbb{E}_{y, \xi, \zeta, \chi} (\rho (h_{<K}^*)^{\otimes 2} + (1 - \rho) (h_{<K}'^*)^{\otimes 2}), \quad (\text{A43})$$

$$V = \mathbb{E}_{y, \xi, \zeta, \chi} (\rho \text{Cov}_{P_h}(h_{<K}) + (1 - \rho) \text{Cov}_{P'_h}(h_{<K})), \quad (\text{A44})$$

$$\begin{aligned} \hat{m}_w &= \frac{\sqrt{\mu}}{V_w} \mathbb{E}_{y, \xi, \zeta, \chi} y (\rho (h_0^* - \sqrt{\mu} y m_w) \\ &+ (1 - \rho) (h_0'^* - \sqrt{\mu} y m_w)), \quad (\text{A45}) \end{aligned}$$

$$\begin{aligned} \hat{Q}_w &= \frac{1}{V_w^2} \mathbb{E}_{y, \xi, \zeta, \chi} (\rho (h_0^* - \sqrt{\mu} y m_w - \sqrt{Q_w} \zeta)^2 \\ &+ (1 - \rho) (h_0'^* - \sqrt{\mu} y m_w - \sqrt{Q_w} \zeta)^2), \quad (\text{A46}) \end{aligned}$$

$$\hat{V}_w = \frac{1}{V_w} - \frac{1}{V_w^2} \mathbb{E}_{y,\xi,\zeta,\chi}(\rho \text{Cov}_{P_h}(h_0) + (1-\rho) \text{Cov}_{P_h}(h_0)), \quad (\text{A47})$$

$$\hat{m} = \lambda V^{-1} \mathbb{E}_{y,\xi,\zeta,\chi} y (\rho(h_{>0}^* - c \odot h_{<K}^* - \lambda y m) + (1-\rho)(h_{>0}^* - c \odot h_{<K}^* - \lambda y m)), \quad (\text{A48})$$

$$\hat{Q} = V^{-1} \mathbb{E}_{y,\xi,\zeta,\chi} (\rho(h_{>0}^* - c \odot h_{<K}^* - \lambda y m - Q^{1/2} \chi)^{\otimes 2} + (1-\rho)(h_{>0}^* - c \odot h_{<K}^* - \lambda y m - Q^{1/2} \chi)^{\otimes 2}) V^{-1}, \quad (\text{A49})$$

$$\hat{V} = V^{-1} - V^{-1} \mathbb{E}_{y,\xi,\zeta,\chi} (\rho \text{Cov}_{P_h}(h_{>0} - c \odot h_{<K}) + (1-\rho) \text{Cov}_{P_h}(h_{>0} - c \odot h_{<K})) V^{-1}. \quad (\text{A50})$$

We introduced the covariance $\text{Cov}_P(x) = \mathbb{E}_P(xx^T) - \mathbb{E}_P(x)\mathbb{E}_P(x^T)$ and the tensorial product $x^{\otimes 2} = xx^T$. We

used Stein's lemma to simplify the differentials of $Q^{1/2}$ and $\hat{Q}^{1/2}$ and to transform the expression of $\hat{V}_w$ into a more accurate expression for numerical computation in terms of covariance. We used the identities $2x^T Q^{1/2}[(\partial Q^{1/2})/(\partial E_{k,l})]x = x^T E_{k,l}x$ and $-x^T V[(\partial V^{-1})/(\partial E_{k,l})]Vx = x^T E_{k,l}x$ for any element matrix $E_{k,l}$ and for any vector x . We have also that $\nabla_V \log \det V = V^{-1}$, considering its comatrix. Last we kept the first order in β with the approximations $Q + V \approx Q$ and $\hat{Q} - \hat{V} \approx \hat{Q}$.

These self-consistent equations are reproduced in Sec. III A 1.

3. Solution for ridge regression

We take quadratic ℓ and γ . Moreover we assume there is no residual connections $c = 0$; this simplifies largely the analysis in the sense that the covariances of h under P_h or P'_h become diagonal. We have

$$\text{Cov}_{P_h}(h) = \text{diag}\left(\frac{V_w}{1+V_w\hat{V}_0}, \frac{V_0}{1+V_0\hat{V}_1}, \dots, \frac{V_{K-2}}{1+V_{K-2}\hat{V}_{K-1}}, \frac{V_{K-1}}{1+V_{K-1}}\right), \quad (\text{A51})$$

$$\text{Cov}_{P'_h}(h) = \text{diag}\left(\frac{V_w}{1+V_w\hat{V}_0}, \frac{V_0}{1+V_0\hat{V}_1}, \dots, \frac{V_{K-2}}{1+V_{K-2}\hat{V}_{K-1}}, V_{K-1}\right), \quad (\text{A52})$$

$$h^* = \text{Cov}_{P_h}(h) \left(\begin{pmatrix} \hat{Q}^{1/2}\xi + y\hat{m} \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{V_w}(\sqrt{\mu}ym_w + \sqrt{Q_w}\zeta) \\ V^{-1}(\lambda ym + Q^{1/2}\chi) \end{pmatrix} \right), \quad (\text{A53})$$

$$h'^* = \text{Cov}_{P'_h}(h) \left(\begin{pmatrix} \hat{Q}^{1/2}\xi + y\hat{m} \\ 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{V_w}(\sqrt{\mu}ym_w + \sqrt{Q_w}\zeta) \\ V^{-1}(\lambda ym + Q^{1/2}\chi) \end{pmatrix} \right), \quad (\text{A54})$$

where diag means the diagonal matrix with the given diagonal. We packed elements into block vectors of size $K+1$. The self-consistent equations can be explicated:

$$m_w = \frac{1}{\alpha r + \hat{V}_w} \hat{m}_w, \quad V_w = \frac{1}{\alpha r + \hat{V}_w}, \quad Q_w = \frac{1}{\alpha} \frac{\hat{Q}_w + \hat{m}_w^2}{(r + \hat{V}_w)^2}, \quad (\text{A55})$$

$$m = V \left(\hat{m} + \begin{pmatrix} \sqrt{\mu} \frac{m_w}{V_w} \\ \lambda V_{<K-1}^{-1} m_{<K-1} \end{pmatrix} \right), \quad V = \text{diag}\left(\frac{V_w}{1+V_w\hat{V}_0}, \frac{V_0}{1+V_0\hat{V}_1}, \dots, \frac{V_{K-2}}{1+V_{K-2}\hat{V}_{K-1}}\right), \quad (\text{A56})$$

$$\hat{m}_w = \frac{\sqrt{\mu}}{V_w} (m_0 - \sqrt{\mu} m_w), \quad \hat{V}_w = \frac{\hat{V}_0}{1+V_w\hat{V}_0}, \quad (\text{A57})$$

$$\hat{m} = \lambda \hat{V} \left(\begin{pmatrix} \hat{V}_{>0}^{-1} \hat{m}_{>0} \\ 1 \end{pmatrix} - \lambda m \right), \quad \hat{V} = \text{diag}\left(\frac{\hat{V}_1}{1+V_0\hat{V}_1}, \dots, \frac{\hat{V}_{K-1}}{1+V_{K-2}\hat{V}_{K-1}}, \frac{\rho}{1+V_{K-1}}\right) \quad (\text{A58})$$

and

$$\hat{Q}_w = \frac{V_0^2}{V_w^2} \hat{Q}_{0,0} + \left(\frac{V_0}{V_w} - 1\right)^2 \frac{Q_w}{V_w^2} + \frac{\hat{m}_w^2}{\mu}, \quad (\text{A59})$$

$$Q = V \left(\hat{Q} + \begin{pmatrix} \frac{Q_w}{V_w^2} & 0 \\ 0 & V_{<K-1}^{-1} Q_{<K-1} V_{<K-1}^{-1} \end{pmatrix} \right) V + m^{\otimes 2}, \quad (\text{A60})$$

$$\begin{aligned} \hat{Q} = \hat{V} & \left(\begin{pmatrix} \hat{V}_{>0}^{-1} \hat{Q}_{>0} \hat{V}_{>0}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \rho \begin{pmatrix} I_{K-1} & 0 \\ 0 & \rho^{-1} \end{pmatrix} Q \begin{pmatrix} I_{K-1} & 0 \\ 0 & \rho^{-1} \end{pmatrix} + (1-\rho) \begin{pmatrix} I_{K-1} & 0 \\ 0 & 0 \end{pmatrix} Q \begin{pmatrix} I_{K-1} & 0 \\ 0 & 0 \end{pmatrix} \right) \hat{V} \\ & + \rho \begin{pmatrix} \frac{1}{\lambda} \hat{m}_{<K-1} \\ \frac{1}{\lambda \rho} \hat{m}_{K-1} \end{pmatrix}^{\otimes 2} + (1-\rho) \begin{pmatrix} \frac{1}{\lambda} \hat{m}_{<K-1} \\ 0 \end{pmatrix}^{\otimes 2}. \end{aligned} \quad (\text{A61})$$

We used the notations $m_{<K-1} = (m_k)_{0 \leq k < K-1}$, $\hat{m}_{<K-1} = (\hat{m}_k)_{0 \leq k < K-1}$, $\hat{m}_{>0} = (\hat{m}_k)_{0 < k \leq K-1}$, $Q_{<K-1} = (Q_{k,l})_{0 \leq k, l < K-1}$, $Q_{>0} = (Q_{k,l})_{0 < k, l \leq K-1}$, $V_{<K-1} = (V_{k,l})_{0 \leq k, l < K-1}$, and $V_{>0} = (V_{k,l})_{0 < k, l \leq K-1}$. We simplified the equations by combining the expressions of V , $\hat{V}$, m , and $\hat{m}$: the above system of equations is equivalent to the generic equations only at the fixed point. The expected losses and accuracies are

$$E_{\text{train}} = \frac{1}{2\rho} \hat{Q}_{K-1, K-1}, \quad E_{\text{test}} = \frac{1}{2\rho} (1 + V_{K-1, K-1})^2 \hat{Q}_{K-1, K-1}, \quad (\text{A62})$$

$$\text{Acc}_{\text{train}} = \frac{1}{2} \left(1 + \text{erf} \left(\frac{V_{K-1, K-1} + \lambda m_{K-1}}{\sqrt{2Q_{K-1, K-1}}} \right) \right), \quad \text{Acc}_{\text{test}} = \frac{1}{2} \left(1 + \text{erf} \left(\frac{\lambda m_{K-1}}{\sqrt{2Q_{K-1, K-1}}} \right) \right). \quad (\text{A63})$$

To obtain a simple solution we take the limit $r \rightarrow \infty$. The solution of this system is then

$$m_w = \frac{\rho \sqrt{\mu}}{\alpha r} \lambda^K, \quad V_w = \frac{1}{\alpha r}, \quad Q_w = \frac{1}{\alpha r^2} \left(\rho + \rho^2 \mu \lambda^{2K} + \rho^2 \sum_{l=1}^K \lambda^{2l} \right), \quad (\text{A64})$$

$$m_k = \frac{\rho}{\alpha r} \left(\mu \lambda^{K+k} + \sum_{l=0}^k \lambda^{K-k+2l} \right), \quad V_{k,k} = \frac{1}{\alpha r} \quad (\text{A65})$$

$$\hat{m}_w = \rho \sqrt{\mu} \lambda^K, \quad \hat{V}_w = \rho, \quad \hat{Q}_w = \rho + \rho^2 \sum_{l=1}^K \lambda^{2l}, \quad (\text{A66})$$

$$\hat{m}_k = \rho \lambda^{K-k}, \quad \hat{V}_{k,k} = \rho, \quad \hat{Q}_{k,k} = \rho + \rho^2 \sum_{l=1}^{K-1-k} \lambda^{2l} \quad (\text{A67})$$

and

$$Q_{k,k} = \frac{\rho}{\alpha^2 r^2} \left(\alpha \left(1 + \rho \mu \lambda^{2K} + \rho \sum_{l=1}^K \lambda^{2l} \right) + \sum_{m=0}^k \left(1 + \rho \sum_{l=1}^{K-1-m} \lambda^{2l} + \rho \left(\mu \lambda^{K+m} + \sum_{l=0}^m \lambda^{K-m+2l} \right)^2 \right) \right). \quad (\text{A68})$$

We did not precise the off-diagonal parts of Q and $\hat{Q}$ since they do not enter in the computation of the losses and accuracies. The expressions for m and Q are reproduced in Sec. III A 2.

APPENDIX B: ASYMPTOTIC CHARACTERIZATION OF THE CONTINUOUS GCN, FOR ASYMMETRIC AND SYMMETRIZED GRAPHS

Here we derive the asymptotic characterization of the continuous GCN for both the asymmetric and symmetrized

graphs $\tilde{A}$ and $\tilde{A}^s$. As shown in Sec. III B this architecture is particularly relevant since it can be close to the Bayes optimality.

We start by discretizing the GCN and deriving its free energy and the self-consistent equations on its order parameters. Then we take the continuous limit $K \rightarrow \infty$, jointly with an expansion around large regularization r . The derivation of the free energy and of the self-consistent equations follows the same steps as in Appendix A; in particular, for the asymmetric case the expressions are identical up to the point where the continuous limit is taken.

To deal with both cases, asymmetric or symmetrized, we define $(\delta_e, \tilde{A}^e, A^{g,e}, \lambda^e, \Xi^e) \in \{(0, \tilde{A}, A^g, \lambda, \Xi), (1, \tilde{A}^s, A^{g,s}, \lambda^s, \Xi^s)\}$. In particular, $\delta_e = 0$ for the asymmetric and $\delta_e = 1$ for the symmetrized. We remind that $\tilde{A}^s$ is the symmetrized $\tilde{A}$ with effective signal $\lambda^s = \sqrt{2}\lambda$. $\tilde{A}^e$ admits the following Gaussian equivalent [20,22,55]:

$$\tilde{A}^e \approx A^{g,e} = \frac{\lambda^e}{\sqrt{N}} y y^T + \Xi^e, \quad (B1)$$

with $(\Xi)_{ij}$ i.i.d. for all i and j while Ξ^s is taken from the Gaussian orthogonal ensemble.

1. Derivation of the free energy

The continuous GCN is defined by the output function

$$h(w) = e^{\frac{\lambda^e}{\sqrt{N}} \tilde{A}^e} \frac{1}{\sqrt{N}} X w. \quad (B2)$$

Its discretization at finite K is

$$h(w) = h_K, \quad h_k = \left(I_N + \frac{t}{\sqrt{N}} \tilde{A}^e \right) h_{k-1}, \quad h_0 = \frac{1}{\sqrt{N}} X w. \quad (B3)$$

It can be mapped to the discrete GCN of Appendix A by taking $c = t/K$.

The free energy is $-\beta N f = \partial_n \mathbb{E}_{u, \Xi^e, W, y} Z^n (n = 0)$, where the partition function is

$$Z = \int \prod_{\nu}^M dw_{\nu} e^{-\beta r \gamma(w_{\nu})} e^{-\beta s \sum_{i \in R} \ell(y_i h(w)_i) - \beta s' \sum_{i \in R'} \ell(y_i h(w)_i)}. \quad (B4)$$

The expectation of the replicated partition function is

$$\begin{aligned} \mathbb{E} Z^n &\propto \mathbb{E}_{u, \Xi^e, W, y} \int \prod_a^n \prod_{\nu}^M dw_{\nu}^a e^{-\beta r \gamma(w_{\nu}^a)} \prod_a^n \prod_i^N \prod_{k=0}^K dh_{i,k}^a dq_{i,k}^a e^{-\beta s \sum_{a,i \in R} \ell(y_i h_{i,K}^a) - \beta s' \sum_{a,i \in R'} \ell(y_i h_{i,K}^a)} \\ &\times e^{\sum_{a,i} \sum_{k=1}^K i q_{i,k}^a \left(\frac{K}{i} h_{i,k}^a - \frac{1}{\sqrt{N}} \sum_j (\sqrt{N} \frac{K}{i} \delta_{i,j} + \frac{c}{\sqrt{N}} y_i y_j + \Xi_{ij}^e) h_{j,k-1}^a \right) + \sum_{a,i} i q_{i,0}^a \left(h_{i,0}^a - \frac{1}{\sqrt{N}} \sum_{\nu} (\sqrt{\frac{K}{N}} y_i u_{\nu} + W_{j\nu}) w_{\nu}^a \right)} \end{aligned} \quad (B5)$$

$$\begin{aligned} &= \mathbb{E}_{u, y} \int \prod_{a,\nu} dw_{\nu}^a e^{-\beta r \gamma(w_{\nu}^a)} \prod_{a,i,k} dh_{i,k}^a dq_{i,k}^a e^{-\beta s \sum_{a,i \in R} \ell(y_i h_{i,K}^a) - \beta s' \sum_{a,i \in R'} \ell(y_i h_{i,K}^a) + i \sum_{a,i,k>0} q_{i,k}^a \left(\frac{K}{i} (h_{i,k}^a - h_{i,k-1}^a) - \frac{c}{\sqrt{N}} y_i \sum_j y_j h_{j,k-1}^a \right)} \\ &\times e^{-\frac{1}{2N} \sum_{i,j} \sum_{a,b} \sum_{k>0,l>0} (q_{i,k}^a h_{j,k-1}^a q_{j,l-1}^b h_{i,l}^b + \delta_e q_{i,k}^a h_{j,k-1}^a q_{j,l-1}^b h_{i,l-1}^b) - i \sum_{a,i} \frac{\sqrt{K}}{N} y_i q_{i,0}^a \sum_{\nu} u_{\nu} w_{\nu}^a - \frac{1}{2N} \sum_{i,\nu,a,b} q_{i,0}^a q_{i,0}^b w_{\nu}^a w_{\nu}^b}. \end{aligned} \quad (B6)$$

Compared to Appendix A, because of the symmetry the expectation over Ξ^s gives an additional cross-term. We symmetrized $\sum_{i<j}$ by neglecting the diagonal terms. We introduce new order parameters between h and its conjugate q . We set for all a and b and for $0 < k \leq K$ and $0 < l \leq K$

$$m_w^a = \frac{1}{N} \sum_{\nu} u_{\nu} w_{\nu}^a, \quad Q_w^{ab} = \frac{1}{N} \sum_{\nu} w_{\nu}^a w_{\nu}^b, \quad (B7)$$

$$m_k^a = \frac{1}{N} \sum_j y_j h_{j,k-1}^a, \quad Q_{h,kl}^{ab} = \frac{1}{N} \sum_j h_{j,k-1}^a h_{j,l-1}^b, \quad (B8)$$

$$Q_{q,kl}^{ab} = \frac{1}{N} \sum_j q_{j,k}^a q_{j,l}^b, \quad Q_{qh,kl}^{ab} = \frac{1}{N} \sum_j q_{j,k}^a h_{j,l-1}^b. \quad (B9)$$

We introduce these quantities via δ -Dirac functions. Their conjugates are $\hat{m}_w^a$, $\hat{Q}_w^{ab}$, $\hat{V}_w^{ab}$, $\hat{m}_k^a$, $\hat{Q}^{ab}$, and $\hat{V}^{ab}$. We factorize the ν and i indices. We leverage the replica-symmetric ansatz. We assume that for all a and b

$$m_w^a = m_w, \quad \hat{m}_w^a = -\hat{m}_w, \quad m_k^a = m_k, \quad \hat{m}_k^a = -\hat{m}_k \quad (B10)$$

and

$$Q_w^{ab} = Q_w + V_w \delta_{a,b}, \quad \hat{Q}_w^{ab} = -\hat{Q}_w + \frac{1}{2} (\hat{V}_w + \hat{Q}_w) \delta_{a,b}, \quad (B11)$$

$$Q_{h,kl}^{ab} = Q_{h,kl} + V_{h,kl} \delta_{a,b}, \quad \hat{Q}_{h,kl}^{ab} = -\hat{Q}_{h,kl} + \frac{1}{2} (\hat{V}_{h,kl} + \hat{Q}_{h,kl}) \delta_{a,b}, \quad \hat{Q}_{h,kl}^{ab} = -\hat{Q}_{h,kl} + \hat{V}_{h,kl} \delta_{a,b}, \quad (B12)$$

$$Q_{q,kl}^{ab} = Q_{q,kl} + V_{q,kl} \delta_{a,b}, \quad \hat{Q}_{q,kl}^{ab} = -\hat{Q}_{q,kl} + \frac{1}{2} (\hat{V}_{q,kl} + \hat{Q}_{q,kl}) \delta_{a,b}, \quad Q_{q,kl}^{ab} = -\hat{Q}_{q,kl} + \hat{V}_{q,kl} \delta_{a,b}, \quad (B13)$$

$$Q_{qh,kl}^{ab} = Q_{qh,kl} + V_{qh,kl} \delta_{a,b}, \quad \hat{Q}_{qh,kl}^{ab} = -\hat{Q}_{qh,kl} + \hat{V}_{qh,kl} \delta_{a,b}, \quad \hat{Q}_{qh,kl}^{ab} = -\hat{Q}_{qh,kl} + \hat{V}_{qh,kl} \delta_{a,b}. \quad (B14)$$

$\delta_{a,b}$ is a Kronecker delta between a and b . $Q_h, Q_q, Q_{qh}, V_h, V_q, V_{qh}$, and their conjugates, written with a hat, are $K \times K$ matrices that we pack into the following $2K \times 2K$ symmetric block matrices:

$$Q = \begin{pmatrix} Q_q & Q_{qh} \\ Q_{qh}^T & Q_h \end{pmatrix}, \quad V = \begin{pmatrix} V_q & V_{qh} \\ V_{qh}^T & V_h \end{pmatrix}, \quad (\text{B15})$$

$$\hat{Q} = \begin{pmatrix} \hat{Q}_q & \hat{Q}_{qh} \\ \hat{Q}_{qh}^T & \hat{Q}_h \end{pmatrix}, \quad \hat{V} = \begin{pmatrix} \hat{V}_q & \hat{V}_{qh} \\ \hat{V}_{qh}^T & \hat{V}_h \end{pmatrix}. \quad (\text{B16})$$

We obtain that

$$\begin{aligned} \mathbb{E} Z^n &\propto \int d\hat{Q}_w d\hat{V}_w dQ_w dV_w d\hat{Q}_d d\hat{V}_d dQ_d dV_d e^{\frac{nN}{2}[V_w V_w + \hat{V}_w Q_w - V_w \hat{Q}_w + \text{tr}(\hat{V}V + \hat{V}Q - V\hat{Q}) - \text{tr}(V_q V_h + V_q Q_h + V_h Q_q + \delta_e V_{qh}^2 + 2\delta_e V_{qh} Q_{qh})]} \\ &\quad \times d\hat{m}_w dm_w d\hat{m}_\sigma dm_\sigma e^{-nN(\hat{m}_w m_w + \hat{m}_\sigma m_\sigma)} \left[\mathbb{E}_u \int \prod_a dw^a e^{\psi_w^{(n)}(w)} \right]^{N/\alpha} \\ &\quad \times \left[\mathbb{E}_y \int \prod_{a,k} dh_k^a dq_k^a e^{\psi_h^{(n)}(h,q;s)} \right]^{\rho N} \left[\mathbb{E}_y \int \prod_{a,k} dh_k^a dq_k^a e^{\psi_h^{(n)}(h,q;s')} \right]^{(1-\rho)N} \end{aligned} \quad (\text{B17})$$

$$:= \int d\Theta d\hat{\Theta} e^{N\phi^{(n)}(\Theta, \hat{\Theta})}, \quad (\text{B18})$$

with $\Theta = \{m_w, Q_w, V_w, m, Q, V\}$ and $\hat{\Theta} = \{\hat{m}_w, \hat{Q}_w, \hat{V}_w, \hat{m}, \hat{Q}, \hat{V}\}$ the sets of order parameters and

$$\psi_w^{(n)}(w) = -\beta r \sum_a \gamma(w^a) - \frac{1}{2} \hat{V}_w \sum_a (w^a)^2 + \hat{Q}_w \sum_{a,b} w^a w^b + u \hat{m}_w \sum_a w^a, \quad (\text{B19})$$

$$\begin{aligned} \psi_h^{(n)}(h, q; \bar{s}) &= -\beta \bar{s} \sum_a \ell(y h_K^a) - \frac{1}{2} V_w \sum_a (q_0^a)^2 + Q_w \sum_{a,b} q_0^a q_0^b - \frac{1}{2} \sum_a \begin{pmatrix} q_{>0}^a \\ h_{<K}^a \end{pmatrix}^T \hat{V} \begin{pmatrix} q_{>0}^a \\ h_{<K}^a \end{pmatrix} + \sum_{a,b} \begin{pmatrix} q_{>0}^a \\ h_{<K}^a \end{pmatrix}^T \hat{Q} \begin{pmatrix} q_{>0}^b \\ h_{<K}^b \end{pmatrix} \\ &\quad + y \hat{m}^T \sum_a h_{<K}^a + i \sum_a (q_{>0}^a)^T \left(\frac{K}{t} (h_{>0}^a - h_{<K}^a) - \lambda^e y m^a \right) - i \sqrt{\mu} y m_w \sum_a q_0^a \end{aligned} \quad (\text{B20})$$

u is a scalar standard Gaussian and y is a scalar Rademacher variable. We use the notation $q_{>0}^a \in \mathbb{R}^K$ for $(q_k^a)_{k>0}$ and similarly as to $h_{>0}^a$ and $h_{<K}^a = (h_k^a)_{k<K}$. We packed them into vectors of size $2K$.

We take the limit $N \rightarrow \infty$ thanks to Laplace's method.

$$-\beta f \propto \frac{1}{N} \frac{\partial}{\partial n} (n=0) \int d\Theta d\hat{\Theta} e^{N\phi^{(n)}(\Theta, \hat{\Theta})} \quad (\text{B21})$$

$$= \text{extr}_{\Theta, \hat{\Theta}} \frac{\partial}{\partial n} (n=0) \phi^{(n)}(\Theta, \hat{\Theta}) \quad (\text{B22})$$

$$:= \text{extr}_{\Theta, \hat{\Theta}} \phi(\Theta, \hat{\Theta}), \quad (\text{B23})$$

where we extremize the following free entropy ϕ :

$$\begin{aligned} \phi &= \frac{1}{2} (V_w \hat{V}_w + \hat{V}_w Q_w - V_w \hat{Q}_w) + \frac{1}{2} \text{Tr}(V_q \hat{V}_q + \hat{V}_q Q_q - V_q \hat{Q}_q) + \frac{1}{2} \text{Tr}(V_h \hat{V}_h + \hat{V}_h Q_h - V_h \hat{Q}_h) \\ &\quad + \text{Tr}(V_{qh} \hat{V}_{qh}^T + \hat{V}_{qh} Q_{qh}^T - V_{qh} \hat{Q}_{qh}^T) - \frac{1}{2} \text{Tr}(V_q V_h + V_q Q_h + Q_q V_h + \delta_e V_{qh}^2 + 2\delta_e V_{qh} Q_{qh}) - m_w \hat{m}_w - m^T \hat{m} \\ &\quad + \mathbb{E}_{u,\zeta} \int dw e^{\psi_w(w)} + \rho \mathbb{E}_{y,\zeta,\chi} \int dq dh e^{\psi_{qh}(q,h;s)} + (1-\rho) \mathbb{E}_{y,\zeta,\chi} \int dq dh e^{\psi_{qh}(q,h;s')}. \end{aligned} \quad (\text{B24})$$

We factorized the replica and took the derivative with respect to n by introducing independent standard Gaussian random variables $\varsigma \in \mathbb{R}$, $\zeta = \begin{pmatrix} \zeta_q \\ \zeta_h \end{pmatrix} \in \mathbb{R}^{2K}$ and $\chi \in \mathbb{R}$. The potentials are

$$\psi_w(w) = -\beta r \gamma(w) - \frac{1}{2} \hat{V}_w w^2 + \left(\sqrt{\hat{Q}_w} \varsigma + u \hat{m}_w \right) w \quad (\text{B25})$$

$$\begin{aligned} \psi_{qh}(q, h; \bar{s}) = & -\beta \bar{s} \ell(y h_K) - \frac{1}{2} V_w q_0^2 - \frac{1}{2} \begin{pmatrix} q_{>0} \\ h_{<K} \end{pmatrix}^T \hat{V} \begin{pmatrix} q_{>0} \\ h_{<K} \end{pmatrix} + \begin{pmatrix} q_{>0} \\ h_{<K} \end{pmatrix}^T \hat{Q}^{1/2} \begin{pmatrix} \zeta_q \\ \zeta_h \end{pmatrix} \\ & + y h_{<K}^T \hat{m} + i q^T \left(\begin{pmatrix} 1/K \\ I/t \end{pmatrix} D h - \begin{pmatrix} y \sqrt{\mu} m_w + \sqrt{\bar{Q}_w} \chi \\ y \lambda^e m \end{pmatrix} \right). \end{aligned} \quad (\text{B26})$$

We already extremize ϕ with respect to Q and V to obtain the following equalities:

$$\hat{V}_q = V_h, \quad V_q = \hat{V}_h, \quad \hat{V}_{qh} = \delta_e V_{qh}^T, \quad (\text{B27})$$

$$\hat{Q}_q = -Q_h, \quad Q_q = -\hat{Q}_h, \quad \hat{Q}_{qh} = -\delta_e Q_{qh}^T. \quad (\text{B28})$$

In particular, this shows that in the asymmetric case where $\delta_e = 0$ one has $\hat{V}_{qh} = \hat{Q}_{qh} = 0$ and as a consequence $V_{qh} = Q_{qh} = 0$; and we recover the potential ψ_h previously derived in Appendix A.

We assume that ℓ is quadratic so ψ_{qh} can be written as the following quadratic potential. Later we will take the limit $r \rightarrow \infty$ where h is small and where ℓ can effectively be expanded around 0 as a quadratic potential.

$$\psi_{qh}(q, h; \bar{s}) = -\frac{1}{2} \begin{pmatrix} q \\ h \end{pmatrix}^T \begin{pmatrix} G_q & -i G_{qh} \\ -i G_{qh}^T & G_h \end{pmatrix} \begin{pmatrix} q \\ h \end{pmatrix} + \begin{pmatrix} q \\ h \end{pmatrix}^T \begin{pmatrix} -i B_q \\ B_h \end{pmatrix} \quad (\text{B29})$$

with

$$G_q = \begin{pmatrix} V_w & 0 \\ 0 & \hat{V}_q \end{pmatrix}, \quad G_h = \begin{pmatrix} \hat{V}_h & 0 \\ 0 & \beta \bar{s} \end{pmatrix}, \quad G_{qh} = \begin{pmatrix} 1/K & 0 \\ 0 & I_K/t \end{pmatrix} D + \begin{pmatrix} 0 & 0 \\ i \hat{V}_{qh} & 0 \end{pmatrix}, \quad D = K \begin{pmatrix} 1 & & & 0 \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{pmatrix}, \quad (\text{B30})$$

$$B_q = \begin{pmatrix} \sqrt{\bar{Q}_w} \chi \\ i(\hat{Q}^{1/2} \zeta)_q \end{pmatrix} + y \begin{pmatrix} \sqrt{\mu} m_w \\ \lambda^e m \end{pmatrix}, \quad B_h = \begin{pmatrix} (\hat{Q}^{1/2} \zeta)_h \\ 0 \end{pmatrix} + y \begin{pmatrix} \hat{m} \\ \beta \bar{s} \end{pmatrix}, \quad \begin{pmatrix} (\hat{Q}^{1/2} \zeta)_q \\ (\hat{Q}^{1/2} \zeta)_h \end{pmatrix} = \begin{pmatrix} \hat{Q}_q & \hat{Q}_{qh} \\ \hat{Q}_{qh}^T & \hat{Q}_h \end{pmatrix}^{1/2} \begin{pmatrix} \zeta_q \\ \zeta_h \end{pmatrix}. \quad (\text{B31})$$

G_q, G_h, G_{qh} , and D are in $\mathbb{R}^{(K+1) \times (K+1)}$. D is the discrete derivative. B_q, B_h , and $\begin{pmatrix} q \\ h \end{pmatrix}$ are in $\mathbb{R}^{2(K+1)}$. We can marginalize $e^{\psi_{qh}}$ over q :

$$\int dq dh e^{\psi_{qh}(q, h; \bar{s})} = \int dh e^{\psi_h(h; \bar{s})}, \quad (\text{B32})$$

$$\psi_h(h; \bar{s}) = -\frac{1}{2} h^T G_h h + h^T B_h - \frac{1}{2} \log \det G_q - \frac{1}{2} (G_{qh} h - B_q)^T G_q^{-1} (G_{qh} h - B_q) \quad (\text{B33})$$

$$= -\frac{1}{2} h^T G h + h^T (B_h + D_{qh}^T G_0^{-1} B) - \frac{1}{2} \log \det G_q, \quad (\text{B34})$$

where we set

$$G = G_h + D_{qh}^T G_0^{-1} D_{qh}, \quad (\text{B35})$$

$$G_0 = \begin{pmatrix} K^2 V_w & 0 \\ 0 & t^2 V_h \end{pmatrix}, \quad (\text{B36})$$

$$D_{qh} = D - t \begin{pmatrix} 0 & 0 \\ -i\delta_e V_{qh}^T & 0 \end{pmatrix}, \quad (\text{B37})$$

$$B = \begin{pmatrix} K\sqrt{Q_w\chi} \\ it(\hat{Q}^{1/2}\zeta)_q \end{pmatrix} + y \begin{pmatrix} K\sqrt{\mu}m_w \\ \lambda^e tm \end{pmatrix}. \quad (\text{B38})$$

Equation (B34) is the potential Eq. (63) given in the main part, up to a term independent of h .

We take the limit $\beta \rightarrow \infty$. As before we introduce the measures dP_w , dP_{qh} , and dP'_h whose unnormalized densities are $e^{\psi_w(w)}$, $e^{\psi_{qh}(h,q;s)}$, $e^{\psi_{qh}(h,q;s')}$, $e^{\psi_h(h;s)}$, and $e^{\psi_h(h;s')}$. We use Laplace's method to evaluate them. We have to rescale the order parameters not to obtain a degenerated solution. We take

$$m_w \rightarrow m_w, \quad Q_w \rightarrow Q_w, \quad V_w \rightarrow V_w/\beta, \quad (\text{B39})$$

$$\hat{m}_w \rightarrow \beta \hat{m}_w, \quad \hat{Q}_w \rightarrow \beta^2 \hat{Q}_w, \quad \hat{V}_w \rightarrow \beta \hat{V}_w, \quad (\text{B40})$$

$$m \rightarrow m, \quad Q_h \rightarrow Q_h, \quad V_h \rightarrow V_h/\beta, \quad (\text{B41})$$

$$\hat{m} \rightarrow \beta \hat{m}, \quad \hat{Q}_h \rightarrow \beta^2 \hat{Q}_h, \quad \hat{V}_h \rightarrow \beta \hat{V}_h, \quad (\text{B42})$$

$$Q_{qh} \rightarrow \beta Q_{qh}, \quad V_{qh} \rightarrow V_{qh}. \quad (\text{B43})$$

We take this scaling for Q_{qh} and V_{qh} because we want D_{qh} and B to be of order one while G , B_h , and G_0^{-1} to be of order β . Taking the matrix square root we obtain the blockwise scaling

$$\begin{pmatrix} \hat{m}_w \\ \hat{m} \\ m \end{pmatrix} = \begin{pmatrix} K\sqrt{\mu} & 0 \\ & \lambda^e t I_K \\ 0 & I_{K+1} \end{pmatrix} \mathbb{E}_{y,\xi,\zeta} y \mathcal{P} \left(\begin{pmatrix} G_0^{-1}(D_{qh}h - B) \\ h \end{pmatrix} \right), \quad (\text{B50})$$

$$\begin{pmatrix} \hat{Q}_w \\ \hat{Q}_h & Q_{qh} \\ Q_{qh}^T & Q_h \end{pmatrix} = \begin{pmatrix} K & 0 \\ & t I_K \\ 0 & I_{K+1} \end{pmatrix} \mathbb{E}_{y,\xi,\zeta} \mathcal{P} \left(\left(\begin{pmatrix} G_0^{-1}(D_{qh}h - B) \\ h \end{pmatrix} \right)^{\otimes 2} \right) \begin{pmatrix} K & 0 \\ & t I_K \\ 0 & I_{K+1} \end{pmatrix}, \quad (\text{B51})$$

$$\begin{pmatrix} \hat{V}_w \\ \hat{V}_h & V_{qh} \\ V_{qh}^T & V_h \end{pmatrix} = \mathcal{P} \left(\text{Cov}_{\psi_{qh}} \begin{pmatrix} q \\ h \end{pmatrix} \right). \quad (\text{B52})$$

We use the notation $\cdot$ for unspecified padding to reach vectors of size $2(K+1)$ and matrices of size $2(K+1) \times 2(K+1)$.

$$\hat{Q}^{1/2} \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & \beta \end{pmatrix} \odot \hat{Q}^{1/2}, \quad (\text{B44})$$

which does give $(\hat{Q}^{1/2}\zeta)_q$ of order one and $(\hat{Q}^{1/2}\zeta)_h$ of order β . As a consequence we obtain that $f = -\phi$ and that P_w , P_h , and P'_h are peaked around their respective maximum w^* , h^* , and h'^* , and that they can be approximated by Gaussian measures. Notice that P_{qh} is not peaked as to its q variable, which has to be integrated over all its range, which leads to the marginals P_h and the potential ψ_h Eq. (B34).

Last, differentiating the free energy f with respect to s and s' we obtain the expected errors and accuracies:

$$E_{\text{train}} = \mathbb{E}_{y,\xi,\zeta} \mathcal{L}(yh_K^*), \quad \text{Acc}_{\text{train}} = \mathbb{E}_{y,\xi,\zeta} \delta_{y=\text{sign}(h_K^*)}, \quad (\text{B45})$$

$$E_{\text{test}} = \mathbb{E}_{y,\xi,\zeta} \mathcal{L}(yh_K'^*), \quad \text{Acc}_{\text{test}} = \mathbb{E}_{y,\xi,\zeta} \delta_{y=\text{sign}(h_K'^*)}. \quad (\text{B46})$$

2. Self-consistent equations

The extremality condition $\nabla_{\Theta,\hat{\Theta}}\phi$ gives the following self-consistent equations on the order parameters. $\mathcal{P}$ is the operator that acts by linearly combining quantities evaluated at h^* , taken with $\bar{s} = 1$ and $\bar{s} = 0$ with weights ρ and $1 - \rho$, according to $\mathcal{P}[g(h)] = \rho g(h^*) + (1 - \rho)g(h'^*)$. We assume l_2 regularization, i.e., $\gamma(w) = w^2/2$.

$$m_w = \frac{1}{\alpha} \frac{\hat{m}_w}{r + \hat{V}_w}, \quad (\text{B47})$$

$$Q_w = \frac{1}{\alpha} \frac{\hat{Q}_w + \hat{m}_w^2}{(r + \hat{V}_w)^2}, \quad (\text{B48})$$

$$V_w = \frac{1}{\alpha} \frac{1}{r + \hat{V}_w}, \quad (\text{B49})$$

The extremizer h^* of ψ_h is

$$h^* = G^{-1}(B_h + D_{qh}^T G_0^{-1} B). \quad (\text{B53})$$

It has to be plugged in to the fixed-point equations (B50)–(B51) and the expectation over the disorder has to be taken.

As to the variances Eq. (B52), we have $\text{Cov}_{\psi_{qh}}(q) = (G_{qh} - iG_{qh}^T G_h)^{-1}$ and using Schur's complement on G_q invertible, one obtains

$$\begin{pmatrix} \cdot & \cdot \\ -iV_{qh} & \cdot \end{pmatrix} = t\mathcal{P}(G_0^{-1} D_{qh} G^{-1}), \quad (\text{B54})$$

$$\begin{pmatrix} V_h & \cdot \\ \cdot & \cdot \end{pmatrix} = \mathcal{P}(G^{-1}), \quad (\text{B55})$$

$$\begin{pmatrix} \hat{V}_w & \cdot \\ \cdot & \hat{V}_h \end{pmatrix} = \begin{pmatrix} K^2 & 0 \\ 0 & t^2 I_K \end{pmatrix} \mathcal{P}(G_0^{-1} - G_0^{-1} D_{qh} G^{-1} D_{qh}^T G_0^{-1}). \quad (\text{B56})$$

The continuation of the computation and how to solve these equations is detailed in Sec. III B 2.

3. Solution in the continuous limit at large r

We report the final values of the order parameters, given in Sec. III B 1. We set $x = k/K$ and $z = l/K$ continuous indices ranging from 0 to 1. We define the resolvents

$$\varphi(x) = \begin{cases} e^{\lambda^e t x} & \text{if } \delta_e = 0 \\ \sum_{\nu>0} \nu (\lambda^e)^{\nu-1} \frac{I_\nu(2tx)}{tx} & \text{if } \delta_e = 1 \end{cases}, \quad (\text{B57})$$

$$\Phi(x, z) = \begin{cases} I_0(2t\sqrt{xz}) & \text{if } \delta_e = 0 \\ \frac{I_1(2t(x+z))}{t(x+z)} & \text{if } \delta_e = 1 \end{cases}, \quad (\text{B58})$$

with I_ν the modified Bessel function of the second kind of order ν . The effective inverse derivative is

$$V_{qh}(x, z) = \theta(z - x)(z - x)^{-1} I_1[2t(z - x)], \quad (\text{B59})$$

$$D_{qh}^{-1}(x, z) = D_{qh}^{-1,T}(z, x) = \begin{cases} \theta(x - z) & \text{if } \delta_e = 0 \\ \frac{1}{t} V_{qh}(z, x) & \text{if } \delta_e = 1 \end{cases}, \quad (\text{B60})$$

with θ the step function.

The solution to the fixed-point equations, in the continuous limit $K \rightarrow \infty$, at first constant order in $1/r$, is

$$V_w = \frac{1}{r\alpha}, \quad (\text{B61})$$

$$V_h(x, z) = V_w \Phi(x, z), \quad (\text{B62})$$

$$\hat{V}_h(1 - x, 1 - z) = t^2 \rho \Phi(x, z), \quad (\text{B63})$$

$$\hat{V}_w = t^{-2} \hat{V}_h(0, 0), \quad (\text{B64})$$

$$\hat{m}(1 - x) = \rho \lambda^e t \varphi(x), \quad (\text{B65})$$

$$\hat{m}_w = \sqrt{\mu} \frac{1}{\lambda^e t} \hat{m}(0), \quad (\text{B66})$$

$$m_w = \frac{\hat{m}_w}{r\alpha}, \quad (\text{B67})$$

$$m(x) = (1 + \mu) \frac{m_w}{\sqrt{\mu}} \varphi(x) + \frac{t}{\lambda^e} \int_0^x dx' \int_0^1 dz' \varphi(x - x') V_h(x', z') \hat{m}(z'), \quad (\text{B68})$$

$$\hat{Q}_w = t^{-2} \hat{Q}_h(0, 0), \quad (\text{B69})$$

$$Q_w = \frac{\hat{Q}_w + \hat{m}_w^2}{r^2 \alpha}, \quad (\text{B70})$$

$$\hat{Q}_h(1 - x, 1 - z) = t^2 \int_{0^-, 0^-}^{x, z} dx' dz' \Phi(x - x', z - z') [\mathcal{P}(\hat{m}^{\otimes 2})(1 - x', 1 - z')], \quad (\text{B71})$$

$$Q_{qh}(1 - x, z) = t \int_{0^-, 0^-}^{x, z} dx' dz' \Phi(x - x', z - z') \left[\mathcal{P}(\hat{m})(1 - x') [\lambda^e t m(z') + \sqrt{\mu} m_w \delta(z')] + \int_{0, 0^-}^{1^+, 1} dx'' dz'' (\hat{Q}_h(1 - x', x'') + \mathcal{P}(\hat{m}^{\otimes 2})(1 - x', x'')) D_{qh}^{-1}(x'', z'') G_0(z'', z') \right], \quad (\text{B72})$$

$$\begin{aligned}
Q_h(x, z) = & \int_{0^-, 0^-}^{x, z} dx' dz' \Phi(x - x', z - z') \left[\hat{Q}_w \delta(x', z') + (\lambda^e t m(x') + \sqrt{\mu} m_w \delta(x')) (\lambda^e t m(z') + \sqrt{\mu} m_w \delta(z')) \right. \\
& + \int_{0^-, 0}^{1, 1^+} dx'' dx''' G_0(x', x'') D_{qh}^{-1, T}(x'', x''') (t \delta_e Q_{qh}(x''', z') + \mathcal{P}(\hat{m})(x''') (\lambda^e t m(z') + \sqrt{\mu} m_w \delta(z'))) \\
& + \int_{0, 0^-}^{1^+, 1} dz''' dz'' (t \delta_e Q_{qh}(z''', x') + (\lambda^e t m(x') + \sqrt{\mu} m_w \delta(x')) \mathcal{P}(\hat{m})(z''')) D_{qh}^{-1}(z''', z'') G_0(z'', z') \\
& \left. + \int_{0^-, 0, 0, 0^-}^{1, 1^+, 1^+, 1} dx'' dx''' dz''' dz'' G_0(x', x'') D_{qh}^{-1, T}(x'', x''') (\hat{Q}_h(x''', z''') + \mathcal{P}(\hat{m}^{\otimes 2})(x''', z''')) D_{qh}^{-1}(z''', z'') G_0(z'', z') \right], \quad (B73)
\end{aligned}$$

where we set

$$\mathcal{P}(\hat{m})(x) = \hat{m}(x) + \rho \delta(1 - x), \quad (B74)$$

$$\mathcal{P}(\hat{m}^{\otimes 2})(x, z) = \rho(\hat{m}(x) + \delta(1 - x))(\hat{m}(z) + \delta(1 - z)) + (1 - \rho)\hat{m}(x)\hat{m}(z), \quad (B75)$$

$$G_0(x, z) = t^2 V_h(x, z) + V_w \delta(x, z). \quad (B76)$$

The test and train accuracies are

$$\text{Acc}_{\text{test}} = \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y = \text{sign}[h^*(1)]} \quad (B77)$$

$$= \mathbb{E}_{\xi, \zeta, \chi} \delta_{0 < \sqrt{\mu} m_w + K \int_0^1 dx V(1, x) \hat{m}(x) + \lambda t \int_0^1 dx m(x) + \sqrt{Q_w} \zeta + K \int_0^1 dx dz V(1, x) \hat{Q}^{1/2}(x, z) \xi(z) + t \int_0^1 dx dz Q^{1/2}(x, z) \chi(z)} \quad (B78)$$

$$= \frac{1}{2} \left(1 + \text{erf} \left(\frac{\sqrt{\mu} m_w + K \int_0^1 dx V(1, x) \hat{m}(x) + \lambda t \int_0^1 dx m(x)}{\sqrt{2} \sqrt{Q_w + K^2 \int_0^1 dx dz V(1, x) \hat{Q}(x, z) V(z, 1) + t^2 \int_0^1 dx dz Q(x, z)}} \right) \right) \quad (B79)$$

$$= \frac{1}{2} \left(1 + \text{erf} \left(\frac{m(1) - \rho V(1, 1)}{\sqrt{2} \sqrt{Q(1, 1) - m(1)^2 - \rho(1 - \rho) V(1, 1)^2}} \right) \right) \quad (B80)$$

and

$$\text{Acc}_{\text{train}} = \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y = \text{sign}[h^*(1)]} \quad (B81)$$

$$= \mathbb{E}_{y, \xi, \zeta, \chi} \delta_{y = \text{sign}[h^*(1) + V(1, 1)y]} \quad (B82)$$

$$= \frac{1}{2} \left(1 + \text{erf} \left(\frac{m(1) + (1 - \rho) V(1, 1)}{\sqrt{2} \sqrt{Q(1, 1) - m(1)^2 - \rho(1 - \rho) V(1, 1)^2}} \right) \right). \quad (B83)$$

To obtain the last expressions we integrated m and Q by parts thanks to the self-consistent conditions they satisfy.

4. Higher orders in $1/r$: How to pursue the computation

The solution given in Sec. III B 1 and reproduced above are for infinite regularization r , keeping only the first constant order. We briefly show how to pursue the computation at any order.

The self-consistent equations for V_{qh} , V_h , and $\hat{V}_h$ at any order can be phrased as, rewriting Eqs. (B54)–(B56) and extending the matrices by continuity:

$$\frac{1}{t} V_{qh} = \mathcal{P} \left(D_{qh}^{-1, T} \sum_{a \geq 0} (-G_h D_{qh}^{-1} G_0 D_{qh}^{-1, T})^a \right), \quad (B84)$$

$$V_h = D_{qh}^{-1} \mathcal{P} \left(G_0 \sum_{a \geq 0} (-D_{qh}^{-1, T} G_h D_{qh}^{-1} G_0)^a \right) D_{qh}^{-1, T}, \quad (B85)$$

$$\hat{V}_h = t^2 D_{qh}^{-1, T} \mathcal{P} \left(G_h \sum_{a \geq 0} (-D_{qh}^{-1} G_0 D_{qh}^{-1, T} G_h)^a \right) D_{qh}^{-1}, \quad (B86)$$

where we remind that $G_0 = t^2 V_h + V_w \delta(x, z) = \mathcal{O}(1/r)$, $G_h = \hat{V}_h + \bar{s} \delta(1 - x, 1 - z)$, and $D_{qh} = D - t \delta_e V_{qh}^T$.

These equations form a system of nonlinear integral equations. A perturbative approach with expansion in powers of $1/r$ should allow to solve it. At each order one has to solve linear integral equations whose resolvent is Φ for V_h and $\hat{V}_h$, the previously determined resolvent to the constant order. The perturbations have to be summed and the resulting V_{qh} , V_h and $\hat{V}_h$ can be used to express h^* , h'^* and the other order parameters.

5. Interpretation of terms of dynamical mean-field theory: Computation

We prove the relations given in Sec. III B 2 b, that state an equivalence between the order parameters V_h , V_{qh} , and $\hat{V}_h$ stemming from the replica computation and the correlation and response functions of the dynamical process that h follows. We assume that the regularization r is large and we derive the equalities to the constant order.

We introduce the tilting field $\eta(x) \in \mathbb{R}^N$ and the tilted Hamiltonian as

$$\frac{dh}{dx}(x) = \frac{t}{\sqrt{N}} \tilde{A}^e h(x) + \eta(x), \quad (\text{B87})$$

$$h(x) = \int_0^x dx' e^{(x-x')\frac{t}{\sqrt{N}}\tilde{A}^e} \left(\eta(x') + \delta(x') \frac{1}{\sqrt{N}} Xw \right), \quad (\text{B88})$$

$$H(\eta) = \frac{1}{2} (y - h(1))^T R (y - h(1)) + \frac{r}{2} w^T w, \quad (\text{B89})$$

where $R \in \mathbb{R}^{N \times N}$ diagonal accounts for the train and test nodes. We write $\langle \cdot \rangle_\beta$ the expectation under the density $e^{-\beta H(\eta)} / Z$ (normalized only at $\eta = 0$, Z is not a function of η).

For V_h we have

$$\frac{\beta}{N} \text{Tr}[\langle h(x)h(z)^T \rangle_\beta - \langle h(x) \rangle_\beta \langle h(z)^T \rangle_\beta] \Big|_{\eta=0} \quad (\text{B90})$$

$$= \frac{1}{N} \text{Tr} \left(e^{\frac{tx}{\sqrt{N}}\tilde{A}^e} \frac{1}{N} X((\langle ww^T \rangle_\beta - \langle w \rangle_\beta \langle w^T \rangle_\beta) X^T e^{\frac{tz}{\sqrt{N}}\tilde{A}^e}) \right) \quad (\text{B91})$$

$$= \frac{V_w}{N} \begin{cases} \text{Tr}(e^{\frac{tx}{\sqrt{N}}\tilde{A}} e^{\frac{tz}{\sqrt{N}}\tilde{A}^T}) & \text{if } \delta_e = 0 \\ \text{Tr}(e^{\frac{tx+tz}{\sqrt{N}}\tilde{A}^s}) & \text{if } \delta_e = 1 \end{cases} \quad (\text{B92})$$

We used that in the large regularization limit the covariance of w is I_M/r and $V_w = r\alpha$. We distinguish the two cases symmetrized or not. For the symmetrized case we have

$$\frac{V_w}{N} \text{Tr} \left(e^{\frac{tx+tz}{\sqrt{N}}\tilde{A}^s} \right) = \int_{-2}^{+2} \frac{d\hat{\lambda}}{2\pi} \sqrt{4 - \hat{\lambda}^2} e^{\hat{\lambda}t(x+z)} \quad (\text{B93})$$

$$= V_w \frac{I_1[2t(x+z)]}{t(x+z)}, \quad (\text{B94})$$

where we used that the spectrum of $\tilde{A}^s/\sqrt{N}$ follows the semicircle law up to negligible corrections. For the asymmetric case we expand the two exponentials. $\tilde{A} \approx \Xi$ has independent Gaussian entries.

$$\begin{aligned} & \frac{V_w}{N} \text{Tr} \left(e^{\frac{tx}{\sqrt{N}}\tilde{A}} e^{\frac{tz}{\sqrt{N}}\tilde{A}^T} \right) \\ &= \sum_{n,m \geq 0} \frac{V_w}{N^{1+\frac{n+m}{2}}} \frac{(tx)^n (tz)^m}{n!m!} \\ & \times \sum_{i_1, \dots, i_n, j_1, \dots, j_m} \Xi_{i_1 i_2} \dots \Xi_{i_{n-1} i_n} \Xi_{i_n j_1} \Xi_{j_2 j_1} \Xi_{j_3 j_2} \dots \Xi_{i_1 j_m} \end{aligned} \quad (\text{B95})$$

$$= V_w \sum_n \frac{(t^2 xz)^n}{(n!)^2} = V_w I_0(2t\sqrt{xz}). \quad (\text{B96})$$

In the sum only contribute the terms where $j_2 = i_n, \dots, j_m = i_2$ for $m = n$. Consequently, in both cases we obtain that

$$V_h(x, z) = \frac{\beta}{N} \text{Tr}[\langle h(x)h(z)^T \rangle_\beta - \langle h(x) \rangle_\beta \langle h(z)^T \rangle_\beta] \Big|_{\eta=0}. \quad (\text{B97})$$

V_h is the correlation function between the states $h(x) \in \mathbb{R}^N$ of the network, under the dynamic defined by the Hamiltonian (20).

This derivation can be used to compute the resolvent $\Phi = V_h/V_w$ in the symmetrized case, instead of solving the integral equation that defines it Eq. (97), that is $\Phi(x, z) = D_{qh}^{-1}[t^2 \Phi(x, z) + \delta(x, z)] D_{qh}^{-1, T}$. As a consequence of the two equivalent definitions we obtain the following mathematical identity, for all x and z :

$$\begin{aligned} & \int_{0,0}^{x,z} dx' dz' \frac{I_1[2(x-x')]}{x-x'} \frac{I_1[2(x'+z')]}{x'+z'} \frac{I_1[2(z-z')]}{z-z'} \\ &= \frac{I_1[2(x+z)]}{x+z} - \frac{I_1(2x)I_1(2z)}{xz}. \end{aligned} \quad (\text{B98})$$

For V_{qh} we have

$$\begin{aligned} & \frac{t}{N} \text{Tr} \frac{\partial}{\partial \eta(z)} \langle h(x) \rangle_\beta \Big|_{\eta=0} \\ &= \frac{t}{N} \text{Tr} e^{(x-z)\frac{t}{\sqrt{N}}\tilde{A}^e} \theta(x-z) \end{aligned} \quad (\text{B99})$$

$$= \begin{cases} \theta(x-z) & \text{if } \delta_e = 0 \\ \theta(x-z)(x-z)^{-1} I_1(2t(x-z)) & \text{if } \delta_e = 1 \end{cases} \quad (\text{B100})$$

$$= V_{qh}(x, z). \quad (\text{B101})$$

We neglected the terms of order $1/r$ stemming from w . We integrated over the spectrum of $\tilde{A}^e$, which follows the semicircle law (symmetric case) or the circular law (asymmetric) up to negligible corrections. We obtain that V_{qh} is the response function oh .

Last for $\hat{V}_h$ we have

$$\begin{aligned} & \frac{t^2}{\beta^2 N} \text{Tr} \frac{\partial^2}{\partial \eta(x) \partial \eta(z)} \langle 1 \rangle_\beta \Big|_{\eta=0} \\ &= \frac{t^2}{N} \text{Tr} \left[R \langle (y - h(1))^{\otimes 2} \rangle_\beta \Big|_{\eta=0} R e^{(1-z)\frac{t}{\sqrt{N}}\tilde{A}^e} e^{(1-x)\frac{t}{\sqrt{N}}(\tilde{A}^e)^T} \right] \end{aligned} \quad (\text{B102})$$

$$= \frac{\rho t^2}{N} \text{Tr} e^{(1-z)\frac{t}{\sqrt{N}}\tilde{A}^e} e^{(1-x)\frac{t}{\sqrt{N}}(\tilde{A}^e)^T} \quad (\text{B103})$$

$$= \hat{V}_h(x, z). \quad (\text{B104})$$

We neglected the terms of order $1/\beta$ obtained by differentiating only once $e^{-\beta H}$ and these of order $1/r$, i.e., $y - h(1) \approx y$. We obtain that $\hat{V}_h$ is the correlation function between the responses.

6. Limiting cases

To obtain insights on the behaviour of the test accuracy and to make connections with already studied models we expand (B80) around the limiting cases $t \rightarrow 0$ and $t \rightarrow \infty$.

At $t \rightarrow 0$ we use that $\varphi(x) = 1 + \lambda^e t x + O(t^2)$ and $\Phi(x, z) = 1 + O(t^2)$; this simplifies several terms. We obtain the following expansions at the first order in t :

$$V_w = \frac{1}{r\alpha}, \quad V(x, z) = \frac{1}{r\alpha}, \quad (\text{B105})$$

$$\hat{m}_w = \rho\sqrt{\mu}, \quad \hat{m}(x) = \rho\lambda^e t, \quad (\text{B106})$$

$$m_w = \frac{\rho}{r\alpha}\sqrt{\mu}, \quad m(x) = \frac{\rho}{r\alpha}(1 + \mu)[1 + \lambda^e t(x + 1)], \quad (\text{B107})$$

$$\hat{Q}_w = \rho, \quad \hat{Q}_h(x, z) = 0, \quad (\text{B108})$$

$$Q_w = \frac{\rho + \rho^2 \mu}{r\alpha}, \quad Q_{qh} = O(t), \quad (\text{B109})$$

$$Q_h(1, 1) = Q_w + m(0)^2 + \rho(1 - \rho)V_w^2 + 2\frac{\rho^2}{r^2\alpha^2}(1 + \mu)^2\lambda^e t. \quad (\text{B110})$$

Plugging them in Eq. (B80) we obtain the expression given in Sec. III B 3:

$$\text{Acc}_{\text{test}} = \frac{1}{2} \left(1 + \text{erf} \left(\frac{1}{\sqrt{2}} \sqrt{\frac{\rho\mu + \lambda^e t(2 + \mu)}{\alpha\sqrt{1 + \rho\mu}}} \right) \right). \quad (\text{B111})$$

At $t \rightarrow \infty$ we assume that $\lambda^e > 1$. We distinguish the two cases asymmetric or symmetrized. For asymmetric we have $\varphi(x) = \exp(\lambda^e t x)$ and $\log \Phi(x, z) = \Theta(2t\sqrt{xz})$. For the symmetrized we have

$$\varphi(x) = \frac{1}{tx} \frac{\partial}{\partial \lambda^e} \sum_{\nu \geq 0} (\lambda^e)^\nu I_\nu(2tx) \quad (\text{B112})$$

$$\approx \frac{1}{tx} \frac{\partial}{\partial \lambda^e} \sum_{\nu=-\infty}^{+\infty} (\lambda^e)^\nu I_\nu(2tx) \quad (\text{B113})$$

$$= \frac{1}{tx} \frac{\partial}{\partial \lambda^e} e^{tx(\lambda^e + 1/\lambda^e)} \quad (\text{B114})$$

$$= (1 - (\lambda^e)^{-2}) e^{tx(\lambda^e + 1/\lambda^e)}. \quad (\text{B115})$$

and $\log \Phi(x, z) = \Theta(2t(x + z))$. In the two cases, only the few dominant terms scaling like $e^{2\lambda^e t}$ or $e^{2(\lambda^e + 1/\lambda^e)t}$ dominate in (B80). We obtain

$$\text{Acc}_{\text{test}} \approx \frac{1}{2} \left(1 + \text{erf} \left(\frac{m(1)}{\sqrt{2}\sqrt{Q(1, 1) - m(1)^2}} \right) \right), \quad (\text{B116})$$

$$m(x) = \frac{\rho}{r\alpha} \varphi(1)\varphi(x)[1 + \mu + C(\lambda^e)], \quad (\text{B117})$$

$$C(\lambda^e) = \int_0^\infty dx' dz' \begin{cases} I_0(2\sqrt{x'z'}) e^{-(x'+z')\lambda^e} & \text{if } \delta_e = 0 \\ \frac{I_1(2(x'+z'))}{x'+z'} e^{-(x'+z')(\lambda^e + 1/\lambda^e)} & \text{if } \delta_e = 1 \end{cases}, \quad (\text{B118})$$

$$Q(1, 1) \approx \int_0^1 dx' dz' \Phi(1 - x', 1 - z') (\lambda^e)^2 t^2 m(x') m(z'), \quad (\text{B119})$$

where in m we performed the changes of variables $x' \rightarrow x'/t$ and $z' \rightarrow z'/t$ and took the limit $t \rightarrow \infty$ in the integration bounds to remove the dependency in t and x . Performing a change of variables $1 - x' \rightarrow x'/t$ and $1 - z' \rightarrow z'/t$ in $Q(1, 1)$ we can express Acc_{test} solely in terms of $C(\lambda^e)$. Last, we use the identity

$$C(\lambda^e) = \frac{1}{(\lambda^e)^2 - 1}, \quad (\text{B120})$$

valid in the two cases asymmetric or not, to obtain the expression given in Sec. III B 3:

$$\text{Acc}_{\text{test}} \xrightarrow{t \rightarrow \infty} \frac{1}{2} \left(1 + \text{erf} \left(\frac{\lambda^e q_{\text{PCA}}}{\sqrt{2}} \right) \right), \quad (\text{B121})$$

$$q_{\text{PCA}} = \sqrt{1 - (\lambda^e)^{-2}}. \quad (\text{B122})$$

APPENDIX C: STATE-EVOLUTION EQUATIONS FOR THE BAYES-OPTIMAL PERFORMANCE

The Bayes-optimal performance for semisupervised classification on the binary CSBM can be computed thanks to the following iterative state-evolution equations, that have been derived in [22,39].

The equations have been derived for a symmetric graph. We map the asymmetric $\tilde{A}$ to a symmetric matrix by the symmetrization $(\tilde{A} + \tilde{A}^T)/\sqrt{2}$. Thus the BO performance on A asymmetric are the BO performance on A symmetrized and effective signal $\lambda^s = \sqrt{2}\lambda$.

Let m_y^0 and m_u^0 be the initial condition. The state-evolution equations are

$$m_u^{t+1} = \frac{\mu m_y^t}{1 + \mu m_y^t}, \quad (C1)$$

$$m^t = \frac{\mu}{\alpha} m_u^t + (\lambda^s)^2 m_y^{t-1}, \quad (C2)$$

$$m_y^t = \rho + (1 - \rho) \mathbb{E}_W[\tanh(m^t + \sqrt{m^t} W)], \quad (C3)$$

where W is a standard scalar Gaussian. These equations are iterated until convergence to a fixed point (m, m_y, m_u) . Then the BO test accuracy is

$$\text{Acc}_{\text{test}} = \frac{1}{2} (1 + \text{erf} \sqrt{m/2}). \quad (C4)$$

In the large λ limit we have $m_y \rightarrow 1$ and

$$\log(1 - \text{Acc}_{\text{test}}) \underset{\lambda \rightarrow \infty}{\sim} -\lambda^2. \quad (C5)$$

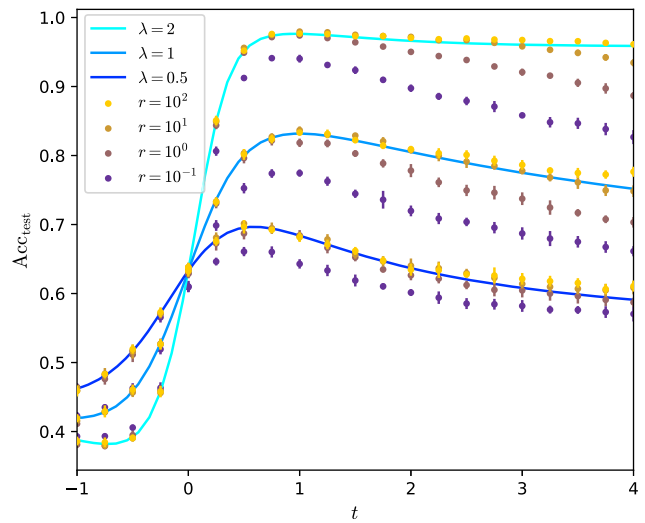
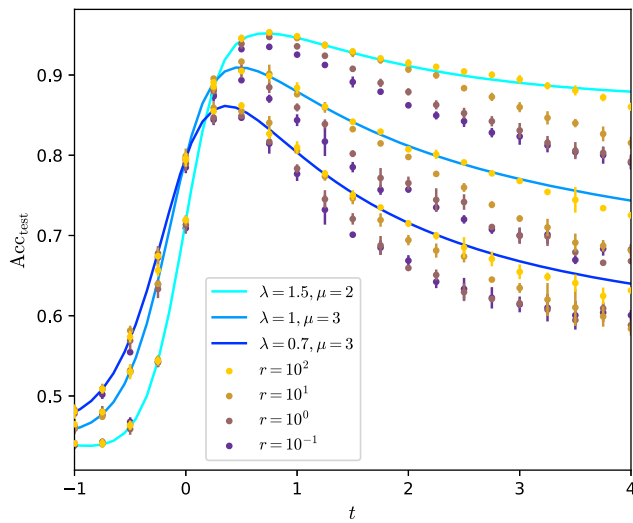


FIG. 8. Predicted test accuracy Acc_{test} of the continuous GCN, at $r = \infty$. Left: for $\alpha = 1$ and $\rho = 0.1$; right: for $\alpha = 2$, $\mu = 1$ and $\rho = 0.3$. The performance of the continuous GCN are given by Eq. (116). Dots: numerical simulation of the continuous GCN for $N = 7 \times 10^3$ and $d = 30$, trained with quadratic loss, averaged over ten experiments.

APPENDIX D: DETAILS ON THE NUMERICS

For the discrete GCN, the system of fixed-point equations (37)–(48) is solved by iterating it until convergence. The iterations are stable up to $K \approx 4$ and no damping is necessary. The integration over (ξ, ζ, χ) is done by Hermite quadrature (quadratic loss) or Monte Carlo sampling (logistic loss) over about 10^6 samples. For the quadratic loss h^* has to be computed by Newton's method. Then the whole computation takes around one minute on a single CPU.

For the continuous GCN the Eq. (116) is evaluated by a trapezoidal integration scheme with a hundred of discretization points. In the nested integrals of $Q(1, 1)$, $\hat{Q}$ can be evaluated only once at each discretization point. The whole computation takes a few seconds.

We provide the code to evaluate our predictions in [51].

APPENDIX E: SUPPLEMENTARY FIGURES

In this section we provide the supplementary figures of Sec. III B 3. They show the convergence to the continuous limit with respect to K and r , and that the continuous limit can be close to the optimality. We also provide the supplementary figures of Sec. III A 3, that compare the GCN on symmetric and asymmetric graphs, and that show the train error versus the residual connection strength.

1. Asymmetric graph

The following figures support the discussion Sec. III B 3 a for the asymmetric graph. They compare the theoretical predictions for the continuous GCN to numerical simulations of the trained network. They show the convergence towards the limit $r \rightarrow \infty$ and the optimality of the continuous GCN over its discretization at finite K .

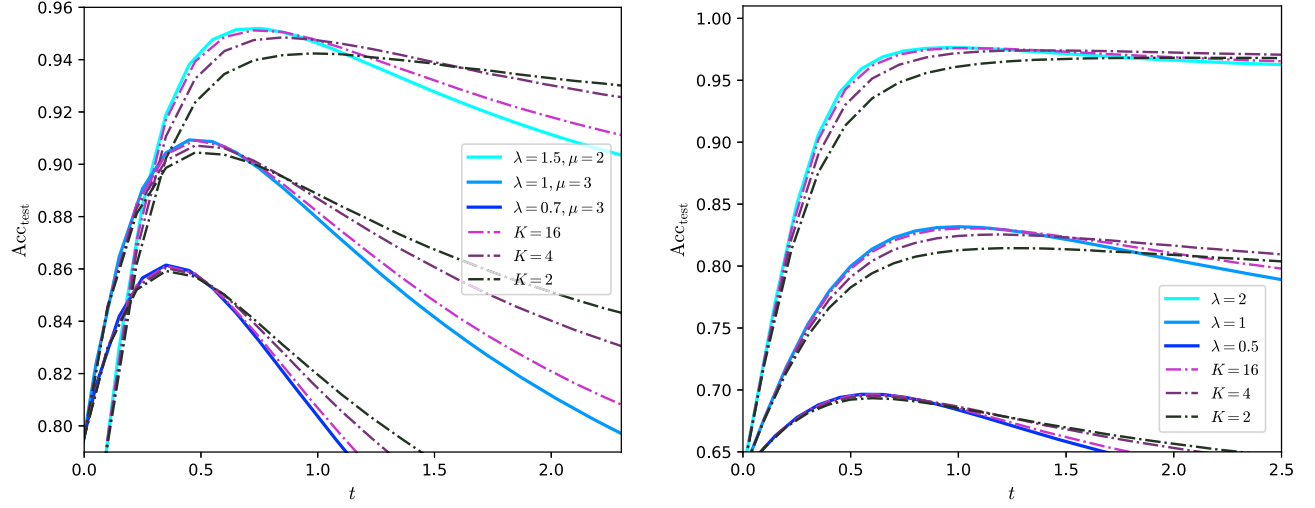


FIG. 9. Predicted test accuracy Acc_{test} of the continuous GCN, at $r = \infty$. Left: for $\alpha = 1$ and $\rho = 0.1$; right: for $\alpha = 2$, $\mu = 1$ and $\rho = 0.3$. The performance of the continuous GCN are given by Eq. (116) while for its discretization at finite K they are given by numerically solving the fixed-point equations (83)–(90).

2. Symmetrized graph

The following figures support the discussion of Sec. III B 3 a for the symmetrized graph. They compare the theoretical predictions for the continuous GCN to numerical simulations of the trained network. They show the convergence towards the limit $r \rightarrow \infty$.

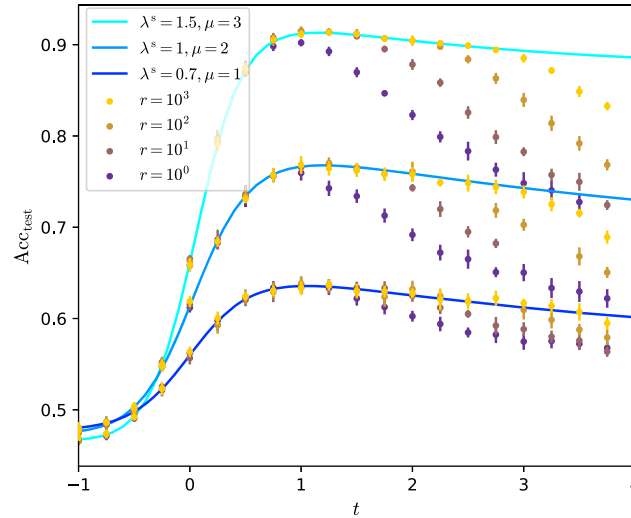


FIG. 10. Predicted test accuracy Acc_{test} of the continuous GCN, at $r = \infty$ for a symmetrized graph. $\alpha = 4$, $\rho = 0.1$. We remind that $\lambda^s = \sqrt{2}\lambda$. The performance of the continuous GCN are given by Eq. (116). Dots: numerical simulation of the continuous GCN for $N = 10^4$ and $d = 30$, trained with quadratic loss, averaged over ten experiments.

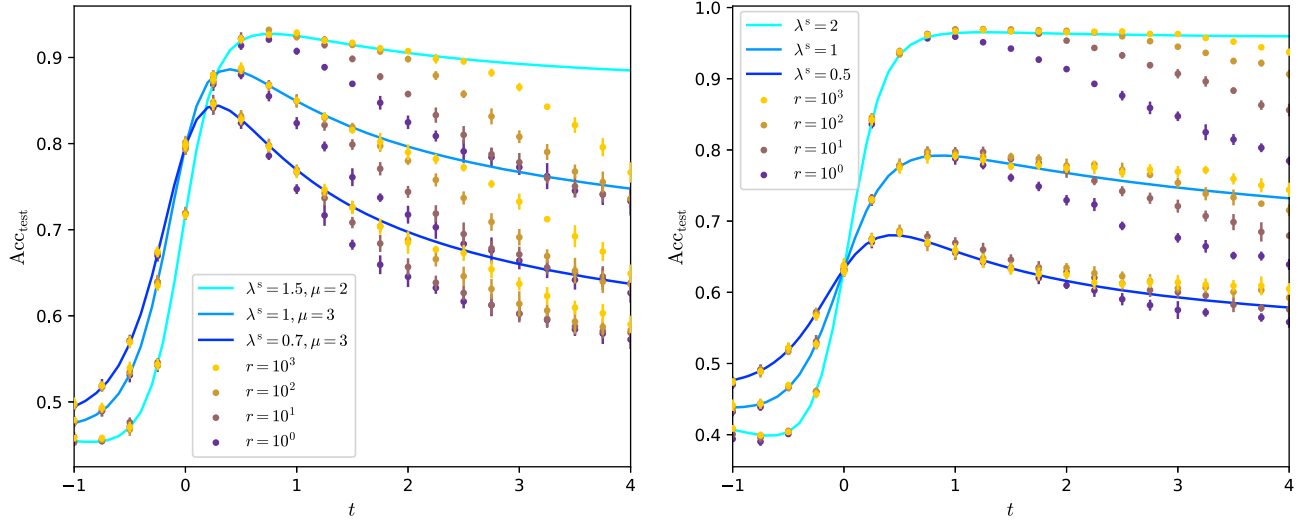


FIG. 11. Predicted test accuracy Acc_{test} of the continuous GCN, at $r = \infty$ for a symmetrized graph. Left: for $\alpha = 1$ and $\rho = 0.1$; right: for $\alpha = 2$, $\mu = 1$ and $\rho = 0.3$. We remind that $\lambda^s = \sqrt{2}\lambda$. The performance of the continuous GCN are given by Eq. (116). Dots: numerical simulation of the continuous GCN for $N = 7 \times 10^3$ and $d = 30$, trained with quadratic loss, averaged over ten experiments.

3. Comparison with optimality

The following figures support the discussion of Secs. III B 3 b and III B 3 c. They show how the optimal diffusion time t^* varies with respect to the parameters of the model and they compare the performance of the optimal continuous GCN and its discrete counterpart to the Bayes optimality.

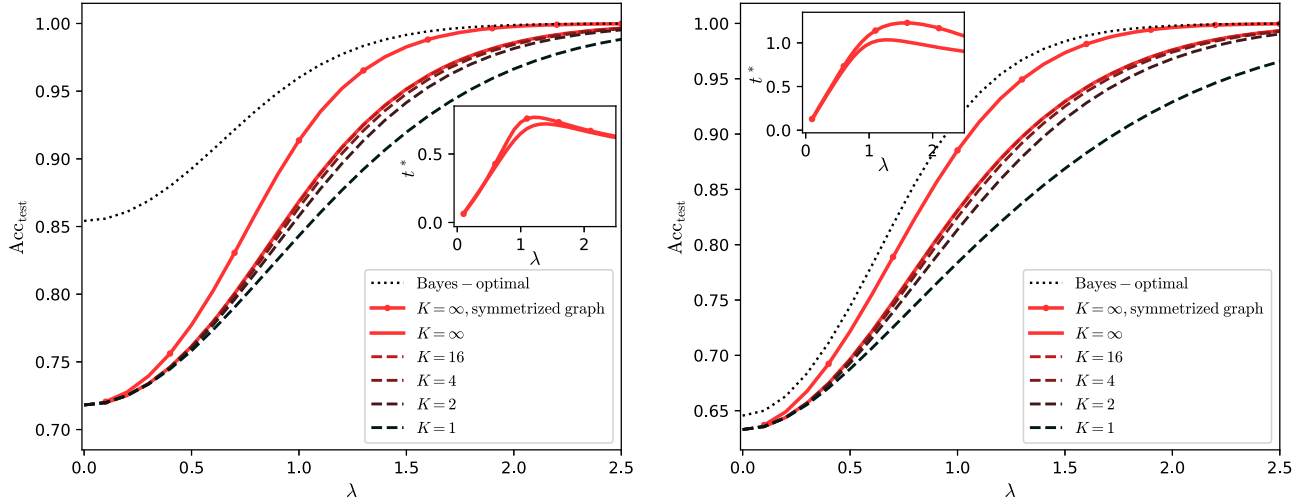


FIG. 12. Predicted test accuracy Acc_{test} of the continuous GCN and of its discrete counterpart with depth K , at optimal times t^* and $r = \infty$. Left: for $\alpha = 1$, $\mu = 2$ and $\rho = 0.1$; right: for $\alpha = 2$, $\mu = 1$ and $\rho = 0.3$. The performance of the continuous GCN $K = \infty$ are given by Eq. (116) while for its discretization at finite K they are given by numerically solving the fixed-point Eqs. (83)–(90). Inset: t^* the maximizer at $K = \infty$.

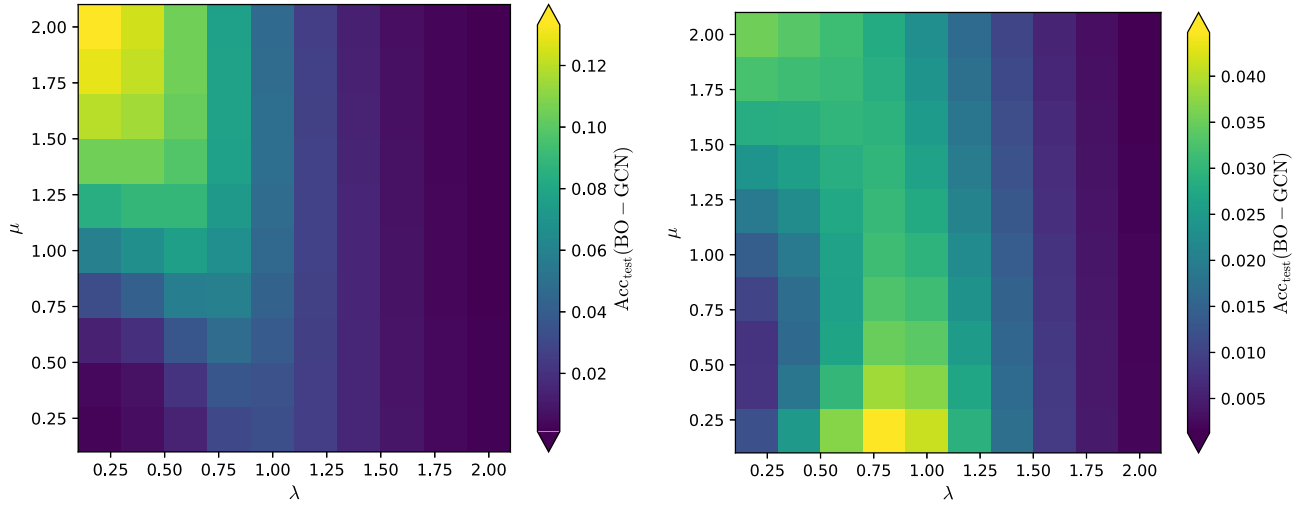


FIG. 13. Gap to the Bayes optimality. Predicted difference between the Bayes-optimal test accuracy and the test accuracy of the continuous GCN at optimal time t^* and $r = \infty$, vs the two signals λ and μ . Left: for $\alpha = 1$ and $\rho = 0.1$; right: for $\alpha = 2$ and $\rho = 0.3$. The performance of the continuous GCN are given by Eq. (116).

4. Comparison between symmetric and asymmetric graphs

The following figure supports the claim of Sec. III A 3 b, that the performance of the GCN depends little whether the graph is symmetric or not at same λ , and that it is not able to deal with the supplementary information the asymmetry gives.

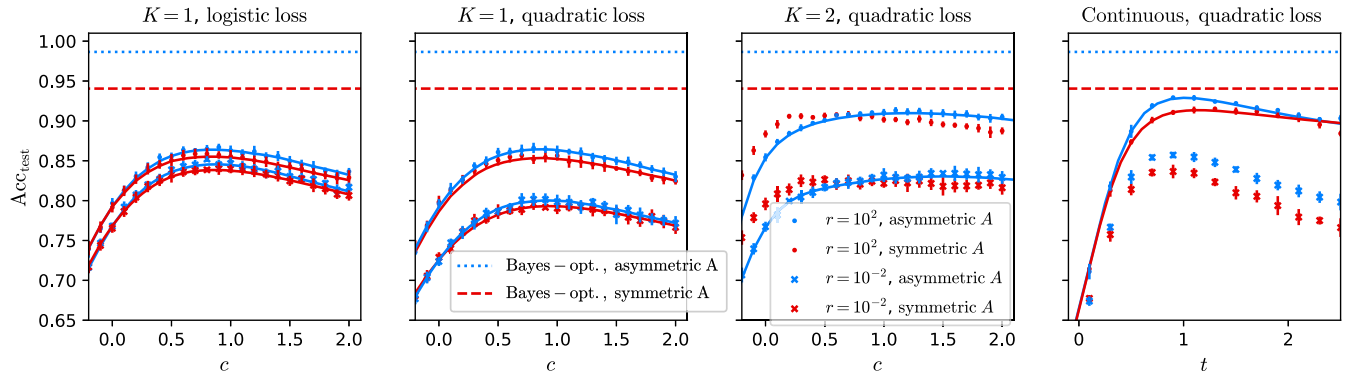


FIG. 14. Test accuracy of the GCN, on asymmetric A and its symmetric counterpart, obtained by equaling A_{ij} with A_{ji} for all $i < j$. $\alpha = 4$, $\lambda = 1.5$, $\mu = 3$ and $\rho = 0.1$. Lines: predictions. Dots: numerical simulation of the GCN for $N = 10^4$ and $d = 30$, averaged over ten experiments.

5. Train error

The following figure displays the train error E_{train} Eq. (18) vs the self-loop intensity c , in the same settings as Fig. 2 of Sec. III A. It shows in particular that to treat c as a parameter trained to minimize the train error would degrade the performance, since it would lead to $c \rightarrow \infty$. As a consequence, c should be treated as a hyperparameter, tuned to maximize the test accuracy, as done in the main part of the article.

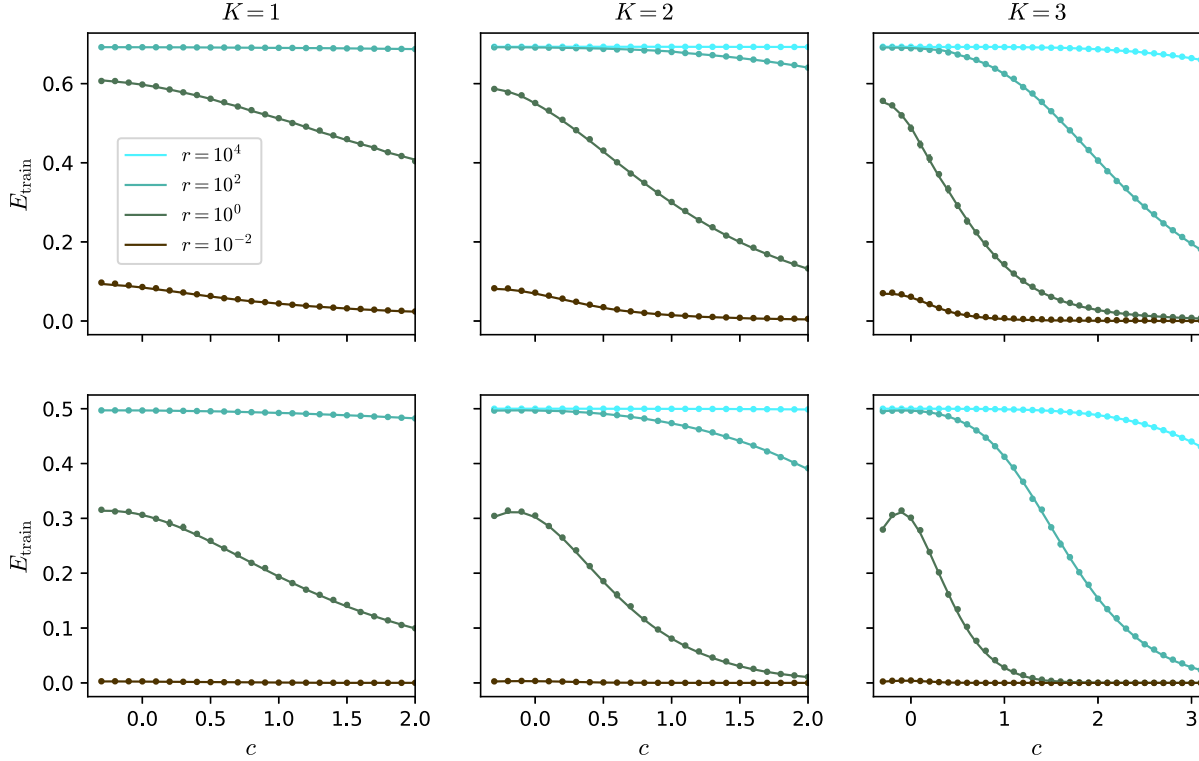


FIG. 15. Predicted train error E_{test} for different values of K . Top: for $\lambda = 1.5$, $\mu = 3$ and logistic loss; bottom: for $\lambda = 1$, $\mu = 2$ and quadratic loss; $\alpha = 4$ and $\rho = 0.1$. We take $c_k = c$ for all k . Dots: numerical simulation of the GCN for $N = 10^4$ and $d = 30$, averaged over ten experiments.

-
- [1] Y. Wang, Z. Li, and A. Barati Farimani, *Graph neural networks for molecules*, in *Machine Learning in Molecular Sciences* (Springer International Publishing, Cham, 2023), pp. 21–66, [arXiv:2209.05582](#).
 - [2] M. M. Li, K. Huang, and M. Zitnik, *Graph representation learning in biomedicine and healthcare*, *Nat. Biomed. Eng.* **6**, 1353 (2022).
 - [3] A. Bessadok, M. A. Mahjoub, and I. Rekik, *Graph neural networks in network neuroscience*, [arXiv:2106.03535](#).
 - [4] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, *Learning to simulate complex physics with graph networks*, in *Proceedings of the 37th International Conference on Machine Learning* (2020), [arXiv:2002.09405](#).
 - [5] J. Shlomi, P. Battaglia, and J.-R. Vlimant, *Graph neural networks in particle physics*, *Mach. Learn.* **2**, 021001 (2020).
 - [6] Y. Peng, B. Choi, and J. Xu, *Graph learning for combinatorial optimization: A survey of state-of-the-art*, *Data Sci. Eng.* **6**, 119 (2021).
 - [7] Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Veličković, *Combinatorial optimization and reasoning with graph neural networks*, *J. Mach. Learn. Res.* **24**, 1 (2023).
 - [8] C. Morris, F. Frasca, N. Dym, H. Maron, I. I. Ceylan, R. Levie, D. Lim, M. Bronstein, M. Grohe, and S. Jegelka, *Position: Future directions in the theory of graph machine learning*, in *Proceedings of the 41st International Conference on Machine Learning* (2024).
 - [9] Q. Li, Z. Han, and X.-M. Wu, *Deeper insights into graph convolutional networks for semi-supervised learning*, in *Thirty-Second AAAI Conference on Artificial Intelligence* (2018) [arXiv:1801.07606](#).

- [10] K. Oono and T. Suzuki, *Graph neural networks exponentially lose expressive power for node classification*, in *International Conference on Learning Representations* (2020), [arXiv:1905.10947](#).
- [11] G. Li, M. Müller, A. Thabet, and B. Ghanem, *DeepGCNs: Can GCNs go as deep as CNNs?*, in *ICCV* (2019), [arXiv:1904.03751](#).
- [12] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, *Simple and deep graph convolutional networks*, in *Proceedings of the 37th International Conference on Machine Learning* (2020), [arXiv:2007.02133](#).
- [13] H. Ju, D. Li, A. Sharma, and H. R. Zhang, *Generalization in graph neural networks: Improved PAC-Bayesian bounds on graph diffusion*, in *AISTATS* (2023), [arXiv:2302.04451](#).
- [14] H. Tang and Y. Liu, *Towards understanding the generalization of graph neural networks* (2023), [arXiv:2305.08048](#).
- [15] W. Cong, M. Ramezani, and M. Mahdavi, *On provable benefits of depth in training graph convolutional networks*, in *35th Conference on Neural Information Processing Systems* (2021), [arXiv:2110.15174](#).
- [16] P. M. Esser, L. C. Vankadara, and D. Ghoshdastidar, *Learning theory can (sometimes) explain generalisation in graph neural networks*, in *35th Conference on Neural Information Processing Systems* (2021), [arXiv:2112.03968](#).
- [17] H. S. Seung, H. Sompolinsky, and N. Tishby, *Statistical mechanics of learning from examples*, *Phys. Rev. A* **45**, 6056 (1992).
- [18] B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, and L. Zdeborová, *Learning curves of generic features maps for realistic datasets with a teacher-student model*, *Adv. Neural Inf. Process. Syst.* **34**, 18137 (2021).
- [19] S. Mei and A. Montanari, *The generalization error of random features regression: Precise asymptotics and the double descent curve*, *Commun. Pure Appl. Math.* **75**, 667 (2022).
- [20] C. Shi, L. Pan, H. Hu, and I. Dokmanić, *Homophily modulates double descent generalization in graph convolution networks*, *Proc. Natl. Acad. Sci. U.S.A.* **121**, e2309504121 (2023).
- [21] B. Yan and P. Sarkar, *Covariate regularized community detection in sparse graphs*, *J. Am. Stat. Assoc.* **116**, 734 (2021).
- [22] Y. Deshpande, S. Sen, A. Montanari, and E. Mossel, *Contextual stochastic block models*, in *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (2018), Vol. 31, [arXiv:1807.09596](#).
- [23] E. Chien, J. Peng, P. Li, and O. Milenkovic, *Adaptive universal generalized pagerank graph neural network*, in *Proceedings of the 39th International Conference on Learning Representations* (2021), [arXiv:2006.07988](#).
- [24] G. Fu, P. Zhao, and Y. Bian, *p-Laplacian based graph neural networks*, in *Proceedings of the 39th International Conference on Machine Learning* (2021), [arXiv:2111.07337](#).
- [25] R. Lei, Z. Wang, Y. Li, B. Ding, and Z. Wei, *EvenNet: Ignoring odd-hop neighbors improves robustness of graph neural networks*, in *36th Conference on Neural Information Processing Systems* (2022), [arXiv:2205.13892](#).
- [26] O. Duranthon and L. Zdeborová, *Asymptotic generalization error of a single-layer graph convolutional network*, in *The Learning on Graphs Conference* (2024), [arXiv:2402.03818](#).
- [27] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural ordinary differential equations*, in *32nd Conference on Neural Information Processing Systems* (2018), [arXiv:1806.07366](#).
- [28] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, in *International Conference on Learning Representations* (2017), [arXiv:1609.02907](#).
- [29] H. Cui, F. Krzakala, and L. Zdeborová, *Bayes-optimal learning of deep random networks of extensive-width*, in *Proceedings of the 40th International Conference on Machine Learning* (2023), [arXiv:2302.00375](#).
- [30] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, *Automating the construction of internet portals with machine learning*, *Information Retrieval* **3**, 127 (2000).
- [31] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, *Pitfalls of graph neural network evaluation*, [arXiv:1811.05868](#).
- [32] C. L. Giles, K. D. Bollacker, and S. Lawrence, *CiteSeer: An automatic citation indexing system*, in *Proceedings of the Third ACM Conference on Digital Libraries* (1998), pp. 89–98.
- [33] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, *Collective classification in network data*, *AI Magazine* **29**, 93 (2008).
- [34] A. Baranwal, K. Fountoulakis, and A. Jagannath, *Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization*, in *Proceedings of the 38th International Conference on Machine Learning* (2021), [arXiv:2102.06966](#).
- [35] A. Baranwal, K. Fountoulakis, and A. Jagannath, *Optimality of message-passing architectures for sparse graphs*, in *37th Conference on Neural Information Processing Systems* (2023), [arXiv:2305.10391](#).
- [36] R. Wang, A. Baranwal, and K. Fountoulakis, *Analysis of corrected graph convolutions*, [arXiv:2405.13987](#).
- [37] F. Mignacco, F. Krzakala, Y. M. Lu, and L. Zdeborová, *The role of regularization in classification of high-dimensional noisy Gaussian mixture*, in *International Conference on Learning Representations* (2020), [arXiv:2002.11544](#).
- [38] B. Aubin, F. Krzakala, Y. M. Lu, and L. Zdeborová, *Generalization error in high-dimensional perceptrons: Approaching Bayes error with convex optimization*, in *Advances in Neural Information Processing Systems* (2020), [arXiv:2006.06560](#).
- [39] O. Duranthon and L. Zdeborová, *Optimal inference in contextual stochastic block models*, [arXiv:2306.07948](#) [Trans. Mach. Learn. Res. (to be published)], <https://openreview.net/forum?id=Pe6hldOUkw>.
- [40] N. Keriven, *Not too little, not too much: A theoretical analysis of graph (over)smoothing*, in *36th Conference on Neural Information Processing Systems* (2022), [arXiv:2205.12156](#).
- [41] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in *IEEE Conference on Computer Vision and Pattern Recognition* (2016), [arXiv:1512.03385](#).

- [42] T. Pham, T. Tran, D. Phung, and S. Venkatesh, *Column networks for collective classification*, in *AAAI* (2017), [arXiv:1609.04508](#).
- [43] K. Xu, M. Zhang, S. Jegelka, and K. Kawaguchi, *Optimization of graph neural networks: Implicit acceleration by skip connections and more depth*, in *Proceedings of the 38th International Conference on Machine Learning* (2021), [arXiv:2105.04550](#).
- [44] M. E. Sander, P. Ablin, and G. Peyré, *Do residual neural networks discretize neural ordinary differential equations?*, in *36th Conference on Neural Information Processing Systems* (2022), [arXiv:2205.14612](#).
- [45] J. Ling, A. Kurzawski, and J. Templeton, *Reynolds averaged turbulence modelling using deep neural networks with embedded invariance*, *J. Fluid Mech.* **807**, 155 (2016).
- [46] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, *Universal differential equations for scientific machine learning*, [arXiv:2001.04385](#).
- [47] P. Marion, *Generalization bounds for neural ordinary differential equations and deep residual networks*, [arXiv:2305.06648](#).
- [48] M. Poli, S. Massaroli, J. Park, A. Yamashita, H. Asama, and J. Park, *Graph neural ordinary differential equations*, [arXiv:1911.07532](#).
- [49] L.-P. A. C. Xhonneux, M. Qu, and J. Tang, *Continuous graph neural networks*, in *Proceedings of the 37th International Conference on Machine Learning* (2020), [arXiv:1912.00967](#).
- [50] A. Han, D. Shi, L. Lin, and J. Gao, *From continuous dynamics to graph neural networks: Neural diffusion and beyond*, [arXiv:2310.10121](#).
- [51] O. Duranthon and L. Zdeborová (2025), <https://arxiv.org/src/2503.01361v2/anc>.
- [52] C. Lu and S. Sen, *Contextual stochastic block model: Sharp thresholds and contiguity*, [arXiv:2011.09841](#).
- [53] F. Wu, T. Zhang, A. H. de Souza Jr., C. Fifty, T. Yu, and K. Q. Weinberger, *Simplifying graph convolutional networks*, in *Proceedings of the 36th International Conference on Machine Learning* (2019), [arXiv:1902.07153](#).
- [54] H. Zhu and P. Koniusz, *Simple spectral graph convolution*, in *International Conference on Learning Representations* (2021).
- [55] T. Lesieur, F. Krzakala, and L. Zdeborová, *Constrained low-rank matrix estimation: Phase transitions, approximate message passing and applications*, *J. Stat. Mech.* (2017) 073403.
- [56] O. Duranthon and L. Zdeborová, *Neural-prior stochastic block model*, *Mach. Learn.* **4**, 035017 (2023).
- [57] J. Baik, G. B. Arous, and S. Péché, *Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices*, *Ann. Probab.* **33**, 1643 (2005).