Graph Convolution for Semi-Supervised Classification: Improved Linear Separability and Out-of-Distribution Generalization

Aseem Baranwal [1] Kimon Fountoulakis [1] Aukosh Jagannath [2]

Abstract

Recently there has been increased interest in semi-supervised classification in the presence of graphical information. A new class of learning models has emerged that relies, at its most basic level, on classifying the data after first applying a graph convolution. To understand the merits of this approach, we study the classification of a mixture of Gaussians, where the data corresponds to the node attributes of a stochastic block model. We show that graph convolution extends the regime in which the data is linearly separable by a factor of roughly $1/\sqrt{D}$, where D is the expected degree of a node, as compared to the mixture model data on its own. Furthermore, we find that the linear classifier obtained by minimizing the cross-entropy loss after the graph convolution generalizes to out-of-distribution data where the unseen data can have different intra- and inter-class edge probabilities from the training data.

1. Introduction

Semi-supervised classification is one of the most important topics in machine learning and artificial intelligence. Recently, researchers extended classification models to include relational information (Hamilton, 2020), where relations are captured by a graph. The attributes of the nodes capture information about the nodes, while the edges of the graph capture relations among the nodes. The reason behind this trend is that many applications require the combination of both the graph and the node attributes, such as recommendation systems (Ying et al., 2018), predicting the properties of compounds or molecules (Gilmer et al., 2017; Scarselli et al., 2009), predicting states of physical objects (Battaglia et al., 2016), and classifying types of nodes in knowledge graphs (Kipf & Welling, 2017).

The most popular models use graph convolution (Kipf & Welling, 2017) where one averages the attributes of a node with those of its neighbors.[1] This allows the model to make predictions about a node using the attributes of its neighbors instead of only using the node's attributes. Despite the common perception among practitioners (Chen et al., 2019) that graph convolution can improve the performance of models for semi-supervised classification, we are not aware of any work that studies the benefits of graph convolution in improving classifiability of the data as compared to traditional classification methods, such as logistic regression, nor are we aware of any work on its generalization performance on out-of-distribution data for semi-supervised classification.

To understand these issues, we study the performance of a graph convolution on a simple classification model with node attributes that are correlated with the class information, namely semi-supervised classification for the contextual stochastic block model (Binkiewicz et al., 2017; Deshpande et al., 2018). The contextual stochastic block model (CSBM) is a coupling of the standard stochastic block model (SBM) (Holland et al., 1983) with a Gaussian mixture model. In this model, each class in the graph corresponds to a different Gaussian component of the mixture model, which yields the distribution for the attributes of the nodes. For a precise definition of the model see Section 3. The CSBM allows us to explore a range of questions related to linear separability and, in particular, to probe how various methods perform as one varies both the noise level of the mixture model, namely the distance between the means, and the noise level of the underlying graph, namely the difference between intra- and inter-class edge probabilities. We focus here on the simple case of two classes where the key issues are particularly transparent. We expect that our methods apply readily to the multi-class setting (see Section 6 for more on this).

Let us now briefly summarize our main findings. In the following, let d be the dimension of the mixture model (the number of attributes of a node in the graph), n the number

[1] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada [2] Department of Statistics and Actuarial Science, Department of Applied Mathematics, University of Waterloo, Waterloo, Canada. Correspondence to: Aseem Baranwal <aseem.baranwal@uwaterloo.ca>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] Other types of graph convolution exist, for simplicity we focus on averaging since it's one of the most popular.

of nodes, p and q the intra- and inter-class edge probabilities respectively, and $D \approx n(p+q)/2$ the expected degree of a node. In our analysis we find the following:

- If the means of the mixture model are at most $O(1/\sqrt{d})$ apart, then the data from the mixture model is not linearly separable and the minimal value of the binary cross entropy loss on the sphere of radius R is bounded away from 0 in a way that depends quantitatively on this distance and the sizes of the labeled data-sets uniformly over $R > 0$.[2]
- If the means are at least $\tilde{\omega}(1/\sqrt{d \cdot D})$ apart, the graph is not too sparse ($p, q = \tilde{\omega}(\log^2(n)/n)$), and the noise level is not too large ($(p-q)/(p+q) = \Omega(1)$), then the graph convolution of the data is linearly separable with high probability.
- Furthermore, if these conditions hold, then the minimizer of the training loss achieves exponentially small binary-cross entropy even for out-of-sample data with high probability.
- On the other hand, if the means are $O(1/\sqrt{dD})$, then the convolved data is not linearly separable as well and one obtains the same lower bound on the loss for fixed radius as in the non-convolved setting.

In particular, we see that if the average degree is large enough, then there is a substantial gain in the scale on which the corresponding graph convolution can perform well as compared to logistic regression. On the other hand, it is important to note that if the noise level of the graph is very high and the noise level of the data is small then the graph convolution can be disadvantageous. This is also shown empirically in our experiments in Subsections 5.3 and 5.4.

The rest of the paper is organized as follows: we review recent literature on graph convolutions and (contextual) stochastic block models in Section 2. In Section 3, we give a precise definition of the semi-supervised contextual stochastic block model. In Section 4 we present our results along with a discussion. Finally, in Section 5 we present extensive experiments which illustrate our results.

2. Previous Work

Computer scientists and statisticians have taken a fresh perspective on semi-supervised classification by coupling the graph structure and node attributes, see, e.g., (Scarselli et al., 2009; Cheng et al., 2011; Gilbert et al., 2012; Dang & Viennet, 2012; Günnemann et al., 2013; Yang et al., 2013; Hamilton et al., 2017; Jin et al., 2019; Mehta et al., 2019; Klicpera et al., 2019). These papers focus largely on practical aspects of these problems and new graph-based machine learning models.

On the other hand, there is a vast body of theoretical work on unsupervised learning for stochastic block models, see, e.g., (Decelle et al., 2011; Massoulié, 2014; Mossel et al., 2018; 2015; Abbe & Sandon, 2015; Abbe et al., 2015; Bordenave et al., 2015; Deshpande et al., 2015; Montanari & Sen, 2016; Banks et al., 2016; Abbe & Sandon, 2018; Li et al., 2019; Kloumann et al., 2017), as well as the recent surveys (Abbe, 2018; Moore, 2017). More recently, there has been work on the related problem of unsupervised classification using the contextual stochastic block model (Binkiewicz et al., 2017; Deshpande et al., 2018). In their work, (Deshpande et al., 2018; Lu & Sen, 2020) explore the fundamental thresholds for correctly classifying a macroscopic fraction of the nodes in the regime of linear sample complexity and large but finite degree. Furthermore, they establish a conjecture for the sharp threshold and characterize the threshold for detection and weak recovery, showing that the average degree need not be large, and the results hold for any degree larger than 1. Their study, however, is largely focused on the fundamental limits of unsupervised learning whereas the work here is focused on understanding the relative merits of graph convolutions over traditional learning methods for semi-supervised learning.

Another line of work has been studying the power of graph convolution models to distinguish graphs (Xu et al., 2019; Garg et al., 2020; Loukas, 2020a), and the universality of models that use graph convolution (Loukas, 2020b). In this last paper and the references therein, the authors study the expressive power of graph neural networks, i.e., the ability to learn a hypothesis set. This, however, does not guarantee generalization for unseen data. Another relevant work that also studies semi-supervised classification using the graphs generated by the SBM is (Chen et al., 2019). There, the authors show that all local minima of cross entropy are approximately global minima if the graphs follow an SBM distribution. Their work, however, does not provide theoretical evidence for the learning benefits of graph convolution in improving linear separability of data, neither do they show generalization bounds for out-of-distribution data. In (Chien et al., 2020; Zhu et al., 2020), the authors show that Multi-Layer Perceptrons (MLPs) outperform standard GNNs on heterophilic graphs. Our results agree with this observation, and we provide theoretical results that characterize this relationship precisely for the contextual stochastic block models, along with a generalization bound for out-of-distribution settings.

[2] It is easy to see that this is essentially sharp, that is, if the means are $\omega(\sqrt{\log d/d})$ apart then the data is linearly separable.

3. The Model

In this section we describe the CSBM (Deshpande et al., 2018), which is a simple coupling of a stochastic block model with a Gaussian mixture model.

Let $(\varepsilon_k)_{k\in[n]}$ be i.i.d. $\mathrm{Ber}(\frac{1}{2})$ random variables. Corresponding to these, consider a stochastic block model consisting of two classes $C_0 = \{i \in [n] : \varepsilon_i = 0\}$ and $C_1 = C_0^{\mathsf{c}}$ with inter-class edge probability q and intra-class edge probability p with no self-loops. In particular, conditionally on (ε_k) the adjacency matrix $A = (a_{ij})$ is Bernoulli with $a_{ij} \sim \mathrm{Ber}(p)$ if i, j are in the same class and $a_{ij} \sim \mathrm{Ber}(q)$ if they are in distinct classes. Along with this, consider $X \in \mathbb{R}^{n\times d}$ to be the feature matrix such that each row X_i is an independent d-dimensional Gaussian random vector with $X_i \sim N(\boldsymbol{\mu}, \frac{1}{d}I)$ if $i \in C_0$ and $X_i \sim N(\boldsymbol{\nu}, \frac{1}{d}I)$ if $i \in C_1$. Here $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^d$ are fixed vectors with $\|\boldsymbol{\mu}\|_2, \|\boldsymbol{\nu}\|_2 \le 1$ and I is the identity matrix. Denote by $\mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu})$ the coupling of a stochastic block model with a two component Gaussian mixture model with means $\boldsymbol{\mu}, \boldsymbol{\nu}$ and covariance $\frac{1}{d}I$ as described above and we denote a sample by $(A, X) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu})$.[3] Observe that the marginal distribution for A is a stochastic block model and that the marginal distribution for X is a two-component Gaussian mixture model. In the rest of the paper, we denote $\tilde{A} = (\tilde{a}_{ij})$ to be the matrix $A + I$ and D to be the diagonal degree matrix for $\tilde{A}$, so that $D_{ii} = \sum_{j\in[n]} \tilde{a}_{ij}$ for all $i \in [n]$. Then the graph convolution of some data X is given by $\tilde{X} = D^{-1}\tilde{A}X$.

For parameters $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, the label predictions are given by $\hat{\mathbf{y}} = \sigma(D^{-1}\tilde{A}X\mathbf{w} + b\mathbf{1})$, where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function applied element-wise in the usual sense. Note that we work in the semi-supervised setting where only a fraction of the labels are available. In particular, we will assume that for some fixed $0 < \beta_0, \beta_1 \le \frac{1}{2}$, the number of labels available for class C_0 is $\beta_0 n$ and for class C_1 is $\beta_1 n$. Let $S = \{i : y_i \text{ is available}\}$ so that $|S| = (\beta_0 + \beta_1)n$. The loss function we use is the binary cross entropy,

$$L(A, X, (\mathbf{w}, b)) = -\frac{1}{|S|}\sum_{i\in S} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i), \tag{1}$$

where y_i is the given label of node i, and $\hat{y}_i$ is the predicted label of node i (also, the i-th component of vector $\hat{\mathbf{y}}$). Observe that the binary cross-entropy loss used in Logistic regression can be written as $L(I, X, (\mathbf{w}, b))$.

4. Results

In this paper we have two main results. Our first result is regarding the relative performance of the graph convolution as compared to classical logistic regression. Here, there are two types of questions to ask. The first is geometric in nature, namely when is the data linearly separable with high probability? This is a statement about the fundamental limit of logistic regression for this data. The second is about the output of the corresponding optimization procedure, the minimizer of (1), namely whether or not it performs well in classifying out-of-sample data.

Note that the objective function, while convex, is non-coercive when the data are linearly separable. Therefore, we introduce a norm-ball constraint and consider the following problem:

$$\mathrm{OPT}_d(A, X, R) = \min_{\substack{\|\mathbf{w}\|\le R,\\ b\in\mathbb{R}}} L(A, X, (\mathbf{w}, b)), \tag{2}$$

where $\|\cdot\|$ is the ℓ_2-norm. The analogous optimization problem in the setting without graph structure, i.e., logistic regression, is then $\mathrm{OPT}_d(I, X, R)$. We find that graph convolutions can dramatically improve the separability of a dataset and thus the performance of the regression. In particular, we find that by adding the graph structure to a dataset and using the corresponding convolution, i.e., working with AX as opposed to simply X, can make a dataset linearly separable when it was not previously.

Our second result is about the related question of generalization on out-of-distribution data. Here we take the optimizer, $(\mathbf{w}^*, b^*)$, of problem (2) and we are interested in how well it classifies data coming from a CSBM with the same means but with a different number of nodes, n', and different intra- and inter-class edge probabilities, p' and q' respectively. We find that $(\mathbf{w}^*, b^*)$ performs nearly optimally, even when the values of n', p', and q' are substantially different from those in the training set.

Let us now state our results more precisely. Given a sample $(A, X) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu})$, we say that $(X_i)_{i=1}^n$ is *linearly separable* if there is some unit vector $\mathbf{v}$ and scalar b such that $\langle X_i, \mathbf{v}\rangle + b < 0$ for all $i \in C_0$ and $\langle X_i, \mathbf{v}\rangle + b > 0$ for all $i \in C_1$, i.e., there is some half-space which correctly classifies the data. We say that $(\tilde{X}_i)_{i=1}^n$ is linearly separable if the same holds for $\tilde{X}$. Let us now define the scaling assumptions under which we work. Define the following quantity:

$$\Gamma(p, q) = \frac{p - q}{p + q}.$$

Assumption 1. *We say that n satisfies* Assumption 1 *if*

$$\omega(d \log d) \le n \le O(\mathrm{poly}(d)).$$

Assumption 2. *We say that (p, q) satisfies* Assumption 2 *if*

$$p, q = \omega(\log^2(n)/n) \quad \textit{and} \quad \Gamma(p, q) = \Omega(1).$$

[3] We note here that, we could also have considered $\sigma^2 I$ instead of I/d, in which case all of our results still hold after rescaling the thresholds appropriately. For example, if we took $\sigma^2 = 1$, then the relevant critical thresholds for linear separability become $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| \sim 1$ and $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| \sim 1/\sqrt{D}$ for the mixture model and the CSBM respectively.

Assumption 1 states that we have at least quasilinearly many samples (i.e., nodes) and at most polynomially many such samples in the dimension of the data. The need for the $poly(d)$ upper bound is, heuristically, for the following simple reasons: if $n \sim \exp(Cd)$ for C sufficiently large then the dataset will hit essentially any point in the support of the two Gaussians, even large deviation regions. As such since there should be a large number of points from either community which will lie on the "wrong" side of any linear classifier. In particular, our arguments will apply if we relax this assumption to taking n to be subexponential in d. Assumption 2 states that the CSBM is not too sparse but such that there is a notable difference between the amount of edges within a class as opposed to between different classes. Assumptions of this latter type are similar to those in the stochastic block model literature, see, e.g., (Abbe, 2018).

Finally, let $\mathbb{B}^d = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$ denote the unit ball, let $\Phi(x)$ denote the cumulative distribution function of a standard Gaussian. We then have the following.

Theorem 1. *Suppose that n satisfies Assumption 1 and that (p, q) satisfies Assumption 2. Fix $0 < \beta_0, \beta_1 \leq 1/2$ and let $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{B}^d$. For any $(A, X) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu})$, we have the following:*

1. *For any $K \geq 0$ if $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| \leq K/\sqrt{d}$, then there are some $C, c > 0$ such that for $d \geq 1$*

$$\mathbb{P}((X_i)_{i\in S} \textit{ is linearly separable}) \leq C\exp(-cd).$$

Furthermore, for any $t > 0$ there is a $c > 0$ such that for every $R > 0$,

$$\mathrm{OPT}_d(I, X, R) \geq 2(\beta_0 \wedge \beta_1)\Phi(-\frac{K}{2}(1 + t))\log(2)$$

with probability $1 - \exp(-cd)$.

2. *If $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| = \omega(\frac{\log n}{\sqrt{dn(p+q)/2}})$, then*

$$\mathbb{P}((\tilde{X}_i)_{i\in S} \textit{ is linearly separable}) = 1 - o_d(1),$$

where $o_d(1)$ denotes a quantity that converges to 0 as $d \to \infty$. Furthermore, with probability $1 - o_d(1)$, we have for all $R > 0$

$$\mathrm{OPT}_d(A, X, R) \leq \exp\big(-\frac{R}{2}\Gamma(p, q)\|\boldsymbol{\mu} - \boldsymbol{\nu}\|(1 - o_d(1))\big).$$

3. *For any $K \geq 0$ if $\|\boldsymbol{\mu} - \boldsymbol{\nu}\| \leq K/\sqrt{dn(p + q)/2}$, then*

$$\mathbb{P}((\tilde{X}_i)_{i\in S} \textit{ is linearly separable}) = o_d(1).$$

Furthermore, for any $t > 0$ with probability $1 - o_d(1)$, for all $R > 0$

$$\mathrm{OPT}_d(A, X, R) \geq 2(\beta_0\wedge\beta_1)\Phi\Big(-\frac{K}{2}(1+t)\Big)\log(2).$$

Let us briefly discuss the meaning of Theorem 1. The first part of this theorem shows that if we consider a two-component mixture of Gaussians in $\mathbb{R}^d$ with the same variances but different means, then if the means are $O(1/\sqrt{d})$ apart, it is highly unlikely to linearly separate the data and the minimal loss is order 1 with high probability. For the second part we find that the convolved data, $\tilde{X} = D^{-1}\tilde{A}X$, is linearly separable provided the means are a bit more than $\Omega(1/\sqrt{d(n(p + q)/2)})$ apart and furthermore, on this scale the loss decays exponentially in $R\|\boldsymbol{\mu} - \boldsymbol{\nu}\|\Gamma$. Consequently, as $n(p + q)/2$ is diverging this regime contains the regime in which the data (X_i) is not linearly separable and logistic regression fails to classify well. We note here that our arguments show that this bound is essentially sharp, provided R is chosen to be at least $\Omega(\sqrt{d(n(p + q))/2})$. Finally the third part shows that, analogously, the convolved data is not linearly separable below the $1/\sqrt{dn(p + q)/2}$ threshold.

We note here that these results hold here under Assumption 2, and in particular, under the assumption of $\Gamma(p, q) = \Omega(1)$. This is to be compared to the work on community detection for stochastic block models and CSBMs (Abbe et al., 2015; Mossel et al., 2015; Massoulié, 2014; Mossel et al., 2018; Deshpande et al., 2018) where the sharp threshold is at $(p - q)\Gamma(p, q) = 1$. Those works, however, are for the (presumably) harder problems of unsupervised learning and hold in a much sparser regime.

Let us now turn to the related question of generalization. Here we are interested in the performance of the optimizer of (2), call it $(\mathbf{w}^*, b^*)$ on out-of-distribution data and, in particular, we are interested in an upper bound on the loss achieved with respect to new data (A', X'). We find that the graph convolution performs well on any out-of-distribution example. In particular, given that the attributes of the test example are drawn from the same distribution as the attributes of the training sample, the graph convolution makes accurate predictions with high probability even when the graph is sampled from a different distribution. More precisely, we have the following theorem.

Theorem 2. *Suppose that n and n' satisfy Assumption 1. Suppose furthermore that the pairs (p, q) and (p', q') satisfy Assumption 2. Fix $0 < \beta_1, \beta_2 \leq 1/2$ and $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{B}^d$. Let $(A, X) \sim \mathrm{CSBM}(n, p, q, \boldsymbol{\mu}, \boldsymbol{\nu})$. Let $\theta^* = \theta^*(R) = (\mathbf{w}^*(R), b^*(R))$ be the optimizer of (2). Then for any sample $(A', X') \sim \mathrm{CSBM}(n', p', q', \boldsymbol{\mu}, \boldsymbol{\nu})$ independent of (A, X), there is a $C > 0$ such that with probability $1 - o_d(1)$ we have that for all $R > 0$*

$$L(A', X', \theta^*) \leq C\exp\Big(-\frac{R}{2}\|\boldsymbol{\mu} - \boldsymbol{\nu}\|\Gamma(p', q')(1 - o(1))\Big)$$

where the loss (1) is with respect to the full test set $S = [n']$.

Let us end by noting here that while we have stated our result for generalization in terms of the binary-cross entropy, our arguments immediately yield that the number of nodes misclassified by the half-space classifier defined by $(\mathbf{w}^*, b^*)$ must vanish with probability tending to 1.

4.1. Proof sketch for Theorems 1 and 2

We now briefly sketch the main ideas of the proof of Theorems 1 and 2. Let us start with the first.

To show that the data $(X_i)_{i=1}^n$ is not linearly separable, we observe that we can decompose the data in the form

$$X_i = (1-\varepsilon_i)\boldsymbol{\mu} + \varepsilon_i \boldsymbol{\nu} + \frac{Z_i}{\sqrt{d}},$$

where $Z_i \sim N(\mathbf{0}, I)$ are iid. The key observation is that when the means are $O(1/\sqrt{d})$ apart then the intersection of the high probability regions of the two components of the mixture is most of the mass of both, so that no plane can separate the high probability regions. To make this precise, consider the Gaussian processes, $g_i(\mathbf{v}) = \langle Z_i, \mathbf{v}\rangle$. Linear separability can be reduced to showing that for some unit vector $\mathbf{v}$, either the maximum of $g_i(\mathbf{v})$ for $i \in S_0$ or the minimum of $g_i(v)$ for $i \in S_1$ is bounded above or below respectively by an order 1 quantity over the entire sphere. This is exponentially unlikely by direct calculation using standard concentration arguments via an $\epsilon-$net argument. In fact, this calculation also shows that for $0 < t < \Phi(-K/2)$, every hyperplane misclassifies at least nt of the data points from each class with high probability, which yields the corresponding loss lower bound.

For the convolved data, the key observation is that

$$\tilde{X}_i \approx \begin{cases} \frac{p\boldsymbol{\mu}+q\boldsymbol{\nu}}{p+q} + \frac{Z_i}{\sqrt{dD_{ii}}} & i \in C_0 \\ \frac{q\boldsymbol{\mu}+p\boldsymbol{\nu}}{p+q} + \frac{Z_i}{\sqrt{dD_{ii}}} & i \in C_1. \end{cases}$$

From this we see that, while the means move closer to each other by a factor of $(p-q)/(p+q)$, the variance has reduced by a factor of $D_{ii} \approx (n(p+q)/2)^{-1}$. This lowers the threshold for non-separability by the same factor. Consequently, if the distance between the means is a bit larger than $1/\sqrt{dn(p+q)/2}$ apart then we can separate the data by the plane through the mid-point of the two means whose normal vector is the direction vector from $\boldsymbol{\mu}$ to $\boldsymbol{\nu}$ with overwhelming probability. More precisely, it suffices to take as ansatz $(\tilde{\mathbf{w}}, \tilde{b})$ given by

$$\tilde{\mathbf{w}} \propto \boldsymbol{\nu} - \boldsymbol{\mu} \qquad \tilde{b} = \langle \boldsymbol{\mu} + \boldsymbol{\nu}, \tilde{\mathbf{w}}\rangle/2.$$

To obtain a training loss upper bound, it suffices to evaluate $L(A, X, \tilde{\mathbf{w}}, \tilde{b})$. A direct calculation shows that this decays exponentially fast with rate $-R\Gamma\|\boldsymbol{\mu} - \boldsymbol{\nu}\|/2$.

Let us now turn to Theorem 2. The key point here is to observe that the preceding argument in fact shows two things. Firstly, the optimizer of the training loss, $\mathbf{w}^*$, must be close to this ansatz and the corresponding b^* must be such that the pair $(\mathbf{w}^*, b^*)$ separates the data better than the ansatz. Secondly, the ansatz we chose does not depend on the particular values of p and q. As such, it can be shown that $(\tilde{\mathbf{w}}, \tilde{b})$ performs well on out-of-distribution data corresponding to different values of $p' > q'$. Combining these two observations then shows that $(\mathbf{w}^*, b^*)$ also performs well on the out-of-distribution data.

5. Experiments

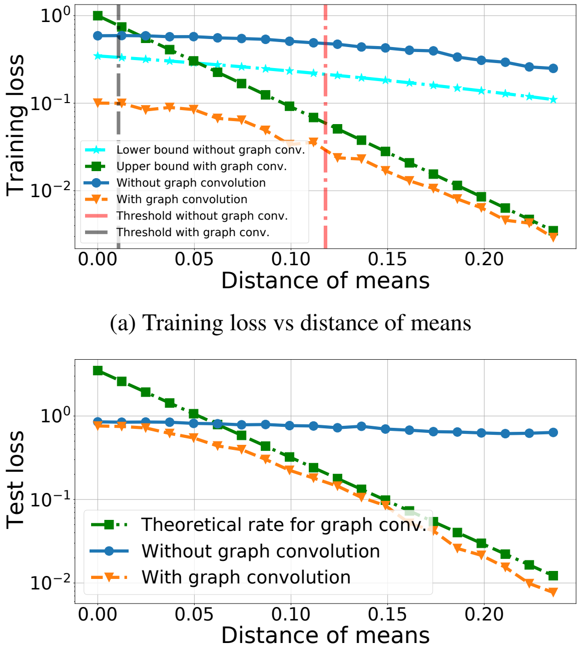
In this section we provide experiments to demonstrate our theoretical results in Section 4. To solve problem (2) we used CVX, a package for specifying and solving convex programs (Grant & Boyd, 2013; Blondel et al., 2008). Throughout the section we set $R = d$ in (2) for all our experiments.

5.1. Training and test loss against distance of means

In our first experiment we illustrate how the training and test losses scale as the distance between the means increases from nearly zero to $2/\sqrt{d}$. Note that according to Part 1 and Part 3 of Theorem 1, $1/\sqrt{0.5dn(p+q)}$ and $1/\sqrt{d}$ are the thresholds for the distance between the means, below which the data with and without graph convolution are not linearly separable with high probability, respectively. For this experiment we train and test on a CSBM with $p = 0.5$, $q = 0.1$, $d = 60$, and $n = 400$ which is roughly equal to $0.85 \cdot d^{3/2}$, and each class has 200 nodes. We present results averaged over 10 trials for the training data and 10 trials for the test data. This means that for each value of the distance between the means we have 100 combinations of train and test data. The results for training loss are shown in Figure 1a and the results of the test loss are shown in Figure 1b. We observe that graph convolution results in smaller training and test loss when the distance of the means is larger than $\log n/\sqrt{dn} \approx 0.035$, which is the threshold such that graph convolution is able to linearly separate the data (Part 2 of Theorem 1).

5.2. Training and test loss against density of graph

In our second experiment, we illustrate how the training and test losses scale as the density of the graph increases while maintaining the same signal to noise ratio for the graph. By density we mean the value of the intra- and inter-class edge probabilities p and q, since they both control the average degree of each node in the graph. It is important to note that our theoretical results are based on Assumption 2, which states lower bounds for p, q and $\Gamma(p, q)$. For this experiment we train and test on a CSBM with $q = 0.2p$ where p varies from $1/n$ to 0.5 and $\Gamma(p, q) \approx 0.6$, $d = 60$, and $n = 400$ which is roughly equal to $0.85 \cdot d^{3/2}$, and each class has 200 nodes. For this experiment we set the distance between the means to $2/\sqrt{d}$. The results for training loss are shown



(a) Training loss vs distance of means

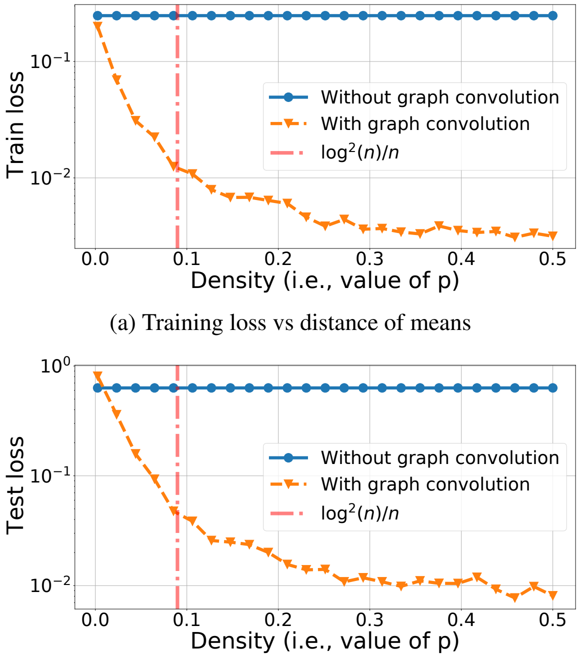
(b) Test loss vs distance of means

Figure 1. Training and test loss with/without graph convolution for increasing distance between the means. The vertical dashed red and black lines correspond to the separability thresholds from Parts 1 and 3 of Theorem 1, respectively. The green dashed line with square markers illustrates the theoretical rate from Theorem 2. The cyan dashed line with star markers corresponds to the lower bound from Part 1 of Theorem 1. We train and test on a CSBM with $p = 0.5$, $q = 0.1$, $n = 400$ and $d = 60$. The y-axis is in log-scale.

in Figure 2a and the results of the test loss are shown in Figure 2b. In these figures we observe that the performance of graph convolution improves as density increases. We also observe that for $p, q \leq \log^2 n/n$, the performance of graph convolution is as poor as that of standard logistic regression.

5.3. Out-of-distribution generalization

In this experiment we test the performance of the trained classifier on out-of-distribution datasets. We perform this experiment for two different distances between the means, $16/\sqrt{d}$ and $2/\sqrt{d}$. We train on a CSBM with $p_{train} = 0.5$, $q_{train} = 0.1$, $n = 400$ and $d = 60$, and we test on CSBMs with $n = 400$, $d = 60$ and varying p_{test} and q_{test} while $p_{test} > q_{test}$. The results are shown in Figure 3[4]. In this figure we observe what was studied in Theorem 2 that is, out-of-distribution generalization to CSBMs with the same means but different p and q pairs. In particular, for small



(a) Training loss vs distance of means

(b) Test loss vs distance of means

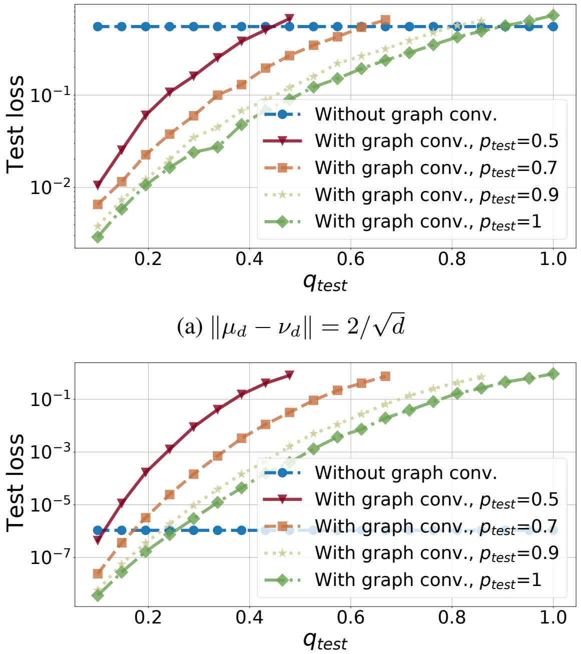
Figure 2. Training and test loss with/without graph convolution for increasing density. The vertical dashed red line corresponds to the lower bound of p and q from Assumption 2. See the main text for a detailed description of the experiment's parameters. The y-axis is in log-scale.

distance between the means, i.e., $2/\sqrt{d}$, where the data are close to being not linearly separable with high probability (Part 1 Theorem 1), Figure 3a shows that graph convolution results in much lower test error than not using the graph. This happens even when q_{test} is close to p_{test} in the figure, i.e., $\Gamma(p_{test}, q_{test})$ from the bound in Theorem 2 is small. Furthermore, in Figure 3b, we observe that for large distance between the means, i.e., $16/\sqrt{d}$, where the data are linearly separable with high probability (Part 1 Theorem 1), and q_{test} is much smaller than p_{test} (i.e., $\Gamma(p_{test}, q_{test})$ is large), then graph convolution has low test error, and this error is lower than that obtained without using the graph. On the other hand, in this regime for the means, as q_{test} approaches p_{test} (i.e, as $\Gamma(p_{test}, q_{test})$ decreases), the test error increases and eventually it becomes larger than without the graph.

In summary, we observe that in the difficult regime where the data are close to linearly inseparable, i.e., the means are close but larger than $1/\sqrt{d}$, then graph convolution can be very beneficial. However, if the data are linearly separable and their means are far apart, then we get good performance without the graph. Furthermore, if $\Gamma(p_{test}, q_{test})$ is small then the graph convolution can actually result in worse training and test errors than logistic regression on the data alone. In the supplementary material, we provide similar plots for various training pairs p_{test} and q_{test}. We observe similar

[4] Note that the x-axis is q. Another option that is more aligned with Theorem 2 is $\Gamma(p_{test}, q_{test})$, however, the log-scale collapses all lines to one and the result is less visually informative.

trends in those experiments.



(a) $\|\mu_d - \nu_d\| = 2/\sqrt{d}$

(b) $\|\mu_d - \nu_d\| = 16/\sqrt{d}$

Figure 3. Out-of-distribution generalization. We train on a CSBM with $p_{train} = 0.5$, $q_{train} = 0.1$, $n = 400$ and $d = 60$. We test on CSBMs with $n = 400$, $d = 60$ and varying p_{test} and q_{test} while $p_{test} > q_{test}$ and fixed means. The y-axis is in log-scale.

5.4. Out-of-distribution generalization on real data

In this experiment we illustrate the generalization performance on real data for the linear classifier obtained by solving 2. In particular, we use the partially labelled real data to train two linear classifiers, with and without graph convolution. We generate new graphs by adding inter-class edges uniformly at random. Then we test the performance of the trained classifiers on the noisy graphs with the original attributes. Therefore, the only thing that changes in the new unseen data are the graphs, the attributes remain the same. Note that our goal in this experiment is not to beat current baselines, but rather to demonstrate out-of-distribution generalization for real data when we use graph convolution.

We use the popular real data Cora, PubMed and WikipediaNetwork. These data are publicly available and can be downloaded from (Fey & Lenssen, 2019). The datasets come with multiple classes, however, for each of our experiments we do a one-v.s.-all classification for a single class. WikipediaNetwork comes with multiple masks for the labels, in our experiments we use the first mask. Moreover, this is a semi-supervised problem, meaning that only a fraction of the training nodes have labels. Details about the datasets are given in Table 1.

Table 1. Information about the datasets, β_0 and β_1 are defined in Section 3. Note that for each dataset we only consider classes A and B and we perform linear classification in a one-v.s.-all fashion. Here, A and B refer to the original classes of the dataset. Results for other classes are given in the supplementary material.

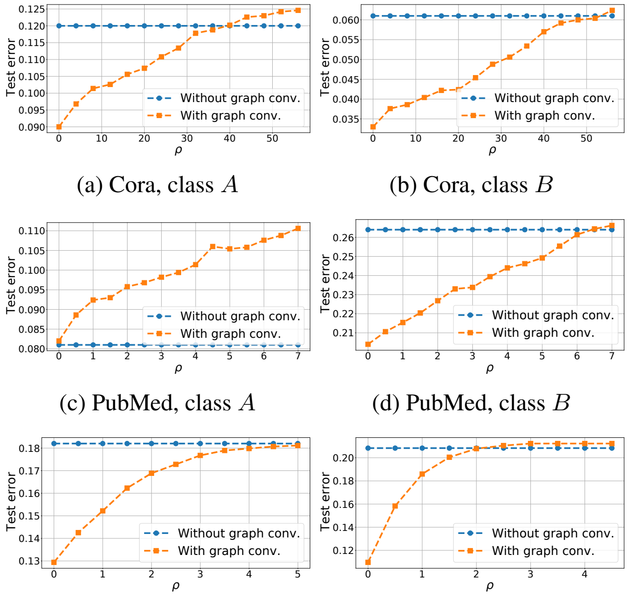
Info./Dataset	Cora	PubMed	Wiki.Net.
# nodes	2708	19717	2277
# attributes	1433	500	2325
β_0, class A	5.0e−2	2.5e−3	4.7e−1
β_1, class A	5.6e−2	4.8e−3	4.9e−1
β_0, class B	4.8e−2	3.3e−3	4.7e−1
β_1, class B	9.2e−2	2.5e−3	4.7e−1
$\|\boldsymbol{\mu} - \boldsymbol{\nu}\|$, class A	7.0e−1	1.0e−1	3.6e−1
$\|\boldsymbol{\mu} - \boldsymbol{\nu}\|$, class B	9.4e−1	7.2e−2	3.0e−1

The results for this experiments are presented in Figure 4. We present results for classes A and B for each dataset. This set of experiments is enough to demonstrate good and bad performance when using graph convolution. The results for the rest of the classes are presented in the supplementary material. The performance for other classes is similar. Note in the plots that in this figure the y-axis (Test error) measures the number of misclassified nodes[5] over the number of nodes in the graph. In all sub-figures in Figure 4 except for Figure 4c we observe that graph convolution has lower test error than without the graph convolution. However, as we add inter-class edges (noise increases), then graph convolution can be disadvantageous. Also, there can be cases like in Figure 4c where graph convolution is disadvantageous for any level of noise. Interestingly, in the experiment in Figure 4c the test errors with and without graph convolution are low (roughly ~ 0.080). This seems to imply that the dataset is close to being linearly separable with respect to the given labels. However, the dataset seems to be nearly non-separable after the graph convolution, since adding noise to the graph results in larger test error.

6. Conclusion and Future Work

In this work we study the benefits of graph convolution for the problem of semi-supervised classification of data. Using the contextual stochastic block model we show that graph convolution can transform data which is not linearly separable into data which is linearly separable. However, we also show empirically that graph convolution can be disadvantageous if the intra-class edge probability is close

[5] We do not plot the loss on the y-axis because the test loss does not differ much between using and not using graph convolution. However, the number of misclassified nodes differs significantly as shown in Figure 4. As noted after Theorem 2, our argument for the bound on the loss immediately yields a bound on the number of misclassified nodes.



(a) Cora, class A

(b) Cora, class B

(c) PubMed, class A

(d) PubMed, class B

(e) WikipediaNetwork, class A

(f) WikipediaNetwork, class B

Figure 4. Test loss as the number of nodes increases. The test error measures the number of misclassified nodes over the number of nodes in the graph. Moreover, ρ denotes the ratio of added inter-class edges over the number of inter-class edges of the original graph. The y-axis is in log-scale.

to the inter-class edge probability. Furthermore, we show that a classifier trained on the convolved data can generalize to out-of-distribution data which have different intra- and inter-class edge probabilities.

Our work is only the first step in understanding the effects of graph convolution for semi-supervised classification. There is still a lot of future work to be done. Below we indicate two questions that need to be addressed.

1. Graph neural networks (Hamilton, 2020) have recently dominated practical aspects of relational machine learning. A lot of these models utilize graph convolution in the same way that we do in this paper. However, the key point of these models is to utilize more than 1 layers in the graph neural network. It is still an open question to understand the benefits of graph convolution for these highly non-linear models for semi-supervised node classification.

2. Our analysis holds for graphs with average number of neighbors at least $\omega(\log^2 n)$. Since a lot of large-scale data consist of sparse graphs it is still an open question to extend our results to sparser graphs where the average number of neighbors per node is $O(1)$.

We end by noting here that while we only study the two class setting, we expect that our arguments extend to the k-class setting with $k = O(1)$ with only minor modifications under natural assumptions.

Acknowledgements. K.F. would like to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [RGPIN-2019-04067, DGECR-2019-00147]. A.J. acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [RGPIN-2020-04597, DGECR-2020-00199].

References

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

Abbe, E. and Sandon, C. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 670–688, 2015. doi: 10.1109/FOCS.2015.47.

Abbe, E. and Sandon, C. Proof of the achievability conjectures for the general stochastic block model. *Communications on Pure and Applied Mathematics*, 71(7): 1334–1406, 2018.

Abbe, E., Bandeira, A. S., and Hall, G. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2015.

Banks, J., Moore, C., Neeman, J., and Netrapalli, P. Information-theoretic thresholds for community detection in sparse networks. In *Conference on Learning Theory*, pp. 383–416. PMLR, 2016.

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., and Kavukcuoglu, K. Interaction Networks for Learning about Objects, Relations and Physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Binkiewicz, N., Vogelstein, J. T., and Rohe, K. Covariate-assisted spectral clustering. *Biometrika*, 104:361–377, 2017.

Blondel, V., Boyd, S., and Kimura, H. Graph implementations for nonsmooth convex programs, recent advances in learning and control (a tribute to M. Vidyasagar). *Lecture Notes in Control and Information Sciences, Springer*, pp. 95–110, 2008.

Bordenave, C., Lelarge, M., and Massoulié, L. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 1347–1357. IEEE, 2015.

Chen, Z., Li, L., and Bruna, J. Supervised community detection with line graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Cheng, H., Zhou, Y., and Yu, J. X. Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Transactions on Knowledge Discovery from Data*, 12, 2011.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Joint adaptive feature smoothing and topology extraction via generalized pagerank gnns. *arXiv preprint arXiv:2006.07988*, 2020.

Dang, T. A. and Viennet, E. Community detection based on structural and attribute similarities. In *The Sixth International Conference on Digital Society (ICDS)*, 2012.

Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

Deshpande, Y., Abbe, E., and Montanari, A. Asymptotic mutual information for the two-groups stochastic block model. *ArXiv*, 2015. arXiv:1507.08685.

Deshpande, Y., S. Sen, A. M., and Mossel, E. Contextual stochastic block models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Garg, V., Jegelka, S., and Jaakkola, T. Generalization and representational limits of graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 119, pp. 3419–3430, 2020.

Gilbert, J., Valveny, E., and Bunke, H. Graph embedding in vector spaces by node attribute statistics. *Pattern Recognition*, 45(9):3072–3083, 2012.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, 2013.

Günnemann, S., Färber, I., Raubach, S., and Seidl, T. Spectral subspace clustering for graphs with feature vectors. In *IEEE 13th International Conference on Data Mining*, 2013.

Hamilton, L. W. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2017.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Jin, D., Liu, Z., Li, W., He, D., and Zhang, W. Graph convolutional networks meet markov random fields: Semi-supervised community detection in attribute networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 3(1):152–159, 2019.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Klicpera, J., Bojchevski, A., and Günnemann, S. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations (ICLR)*, 2019.

Kloumann, I. M., Ugander, J., and Kleinberg, J. Block models and personalized pagerank. *Proceedings of the National Academy of Sciences*, 114(1):33–38, 2017.

Li, P., Chien, I. E., and Milenkovic, O. Optimizing generalized pagerank methods for seed-expansion community detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11705–11716, 2019.

Loukas, A. How hard is to distinguish graphs with graph neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Loukas, A. What graph neural networks cannot learn: Depth vs width. In *International Conference on Learning Representations (ICLR)*, 2020b.

Lu, C. and Sen, S. Contextual stochastic block model: Sharp thresholds and contiguity. *ArXiv*, 2020. arXiv:2011.09841.

Massoulié, L. Community detection thresholds and the weak ramanujan property. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 694–703, 2014.

Mehta, N., Duke, C. L., and Rai, P. Stochastic blockmodels meet graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 4466–4474, 2019.

Montanari, A. and Sen, S. Semidefinite programs on sparse random graphs and their application to community detection. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 814–827, 2016.

Moore, C. The computer science and physics of community detection: Landscapes, phase transitions, and hardness. *Bulletin of The European Association for Theoretical Computer Science*, 1(121), 2017.

Mossel, E., Neeman, J., and Sly, A. Consistency thresholds for the planted bisection model. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 69–75, 2015.

Mossel, E., Neeman, J., and Sly, A. A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708, 2018.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 2009.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *In International Conference on Learning Representations (ICLR)*, 2019.

Yang, J., McAuley, J., and Leskovec, J. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pp. 1151–1156, 2013.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, 2018.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Generalizing graph neural networks beyond homophily. *arXiv preprint arXiv:2006.11468*, 2020.