LaplaceNet: A Hybrid Graph-Energy Neural Network for Deep Semisupervised Classification

Philip Sellars®, Angelica I. Aviles-Rivero®, Member, IEEE, and Carola-Bibiane Schönlieb®

Abstract—Semisupervised learning (SSL) has received a lot of recent attention as it alleviates the need for large amounts of labeled data which can often be expensive, requires expert knowledge, and be time consuming to collect. Recent developments in deep semisupervised classification have reached unprecedented performance and the gap between supervised and SSL is ever-decreasing. This improvement in performance has been based on the inclusion of numerous technical tricks, strong augmentation techniques, and costly optimization schemes with multiterm loss functions. We propose a new framework, LaplaceNet, for deep semisupervised classification that has a greatly reduced model complexity. We utilize a hybrid approach where pseudolabels are produced by minimizing the Laplacian energy on a graph. These pseudolabels are then used to iteratively train a neural-network backbone. Our model outperforms state-of-the-art methods for deep semisupervised classification, over several benchmark datasets. Furthermore, we consider the application of strong augmentations to neural networks theoretically and justify the use of a multisampling approach for SSL. We demonstrate, through rigorous experimentation, that a multisampling augmentation approach improves generalization and reduces the sensitivity of the network to augmentation.

Index Terms—Data augmentation, deep learning, graph-based methods, image classification, pseudolabeling, semisupervised learning (SSL).

I. INTRODUCTION

THE advent of deep learning has been key in achieving outstanding performance in several computer vision tasks, including image classification [1], [2], [3], [4], [5], object detection, e.g., [6], [7], [8], and image segmentation [9], [10], [11]. Training deep learning models often relies upon access to large amounts of labeled training data. In real-world scenarios, we often find that labels are scarce, expensive to collect, prone to errors (high uncertainty), and might require expert knowledge. Therefore, relying on a well-representative dataset to achieve good performance is a major limitation for the practical deployment of machine-learned methods. These issues have motivated the development of techniques that are less reliant on labeled data.

Semisupervised learning (SSL) aims to extract information from unlabeled data, in combination with a small amount of label data, and produce results comparable to fully supervised approaches. In recent years, developments in deep learning have motivated new directions in SSL image classification. The major benefit of these new deep approaches is the ability to learn feature representations, rather than rely upon handcrafted features. In the last few years, deep SSL papers have reached unprecedented performance, e.g., [12], [13], and the gap between supervised and semisupervised models is much smaller now than it was even five years ago, with semisupervised methods surpassing certain supervised techniques.

What techniques have been crucial to the improved performance of deep semisupervised methods? Although there is no universal answer, there are several shared commonalities between the current state-of-the-art (SOTA). The works of [12], [13], and [14] demonstrated that a key factor for improving performance is the use of strong augmentations strategies, such as AutoAugment [15], RandAugment [16], cutout [17], and CTAugment [14]. Additionally, the use of confidence thresholding [12], [18] and temperature sharpening [13], [19] are thought to be vital in improving performance for pseudolabeling methods. Other papers [19], [20], [21] have shown great improvement by using interpolating techniques, such as MixUp [22]. Several SOTA has also promoted large batch sizes [12] with a large ratio of unlabeled to labeled data per batch.

Recent approaches in SSL have proposed costly optimization schemes involving multiterm loss functions to improve the generalization of their models [14], [20]. Some approaches [12] use separate loss terms for unlabeled and labeled data, while consistency regularization approaches, such as [13], use a standard supervised loss in combination with a specialized consistency loss. Other approaches go even further [14], [20] and use three or more loss terms, which promote entropy minimization, class balancing, or simultaneously minimize several consistency losses.
Over-costly computational approaches and unnecessary complexity make it hard to directly say what tools or approaches are important for improved generalization and make it difficult to use SSL methods in realistic settings. Furthermore, despite the significant improvements found in using augmentations, there has been little effort in the field of SSL to investigate how best to include strong augmentations techniques in the learning framework. With these points in mind, in this work, we introduce a new deep SSL framework for image classification which offers SOTA performance with massively reduced model complexity. Our main contributions are as follows.

1) We propose a graph-based pseudolabel approach for semisupervised image classification which we name LaplaceNet. We demonstrate through extensive testing, that our approach produces SOTA results on benchmark datasets CIFAR-10, CIFAR-100, and Mini-ImageNet. We do so with vastly reduced model complexity compared with the current SOTA. We show that a single loss, the classic supervised loss, is all that is required for a fantastic performance in the SSL domain.

2) We show that using an energy-based graphical model for pseudolabel generation produces more accurate pseudolabels, with a small computational overhead, than using the network’s predictions directly. Furthermore, we demonstrate that energy-based pseudolabel approaches can produce SOTA results without the techniques (temperature sharpening, confidence thresholding, and soft labels) that are currently thought to be essential for pseudolabel methods.

3) Instead, we offer further evidence that strong augmentation is by far and away the most important tool for improving the performance of semisupervised models in the natural image domain. With this in mind, we propose, theoretically justify and experimentally demonstrate that a multisample averaging approach to strong augmentation can not only improve generalization but also reduce the sensitivity of the model’s output to data augmentation.

II. RELATED WORK

The problem of improving image classification performance using SSL has been extensively investigated from the classic perspective, e.g., [23], [24], [25], [26], [27], [28], in which one seeks to minimize a given energy functional that exploits the assumed relationship between labeled and unlabeled data [29]. However, classical approaches tended to rely on hand-crafted features that limited their performance and generalization capabilities. With the popularization of deep learning and its ability to learn generalizable feature representations, many techniques have incorporated neural networks to mitigate problems of generalization. These modern SOTA methods are dominated by two approaches, consistency regularization and pseudolabeling, which differ in how they incorporate unlabeled data into the loss function.

A. Consistency Regularization Techniques

One of the fundamental assumptions that allow SSL to help performance is the cluster assumption, which states that points in the same cluster are likely to be in the same class. This can be seen to be equivalent to the low-density assumption which states that the decision boundaries of the model should lie in low-density regions of the data distribution. Following from the above-mentioned assumptions, if we have access to some labeled data $Z_l = \{(x_i, y_i)\}_{i=1}^n$ and a large amount of unlabeled data $Z_u = \{x_i\}_{i=n+1}^{n+\nu}$, we should seek to move our decision boundaries to be in low density regions of the joint labeled and unlabeled data distributions.

Consistency regularization seeks to implement the low-density assumption by encouraging the model $f$ to be invariant to perturbations $\delta$ to the data $x$. As a result, the decision boundaries are pushed to low-density regions. Mathematically, given some data perturbing function $u: \mathcal{X} \rightarrow \mathcal{X}$, such that $u(x) = x + \delta$, consistency-based approaches seek to minimize some consistency loss $L_{con}$ in the general form of

$$L_{con} = ||f(u(x)) - f(x)||^2_2. \quad (1)$$

A large number of papers have applied this idea to SSL, including the $\prod -$model and temporal ensembling [30], virtual adversarial training (VAT) [31], mean teacher (MT) [32], the interpolation consistency training (ICT) [21], MixMatch [14], and MixMatch [19]. The downside of consistency regularization techniques is the vagueness in choosing an appropriate $\delta$. This vagueness is reflected in the wide range of perturbations, which have been used in the field. VAT uses adversarial training to learn an effective $\delta$ for each point. MT [32] decided to apply a perturbation to the model itself and replaces $f(u(x))$ with an exponential moving average of the model $f_{EMA}(x)$. ICT [21] seeks to train the model to provide consistent predictions at interpolations of unlabeled points. Xie et al. [13] demonstrated that by replacing simple noising perturbations with stronger augmentation perturbations (e.g., RandAugment [16] or CTAugment [14]) leads to a substantial performance improvements. Furthermore, from this work, Gong et al. [33] proposed optimization improvements for consistency methods using alpha-divergences and an expectation minimization-like algorithm.

Although these techniques have demonstrated great performance, it is unclear how best to set the perturbations $\delta$ and how best to incorporate them in learning frameworks. In our work, we avoid using model-based perturbations and instead focus on the application of strong data augmentation.

B. Pseudolabeling Techniques

Another family of methods, termed pseudolabel approaches, focus on estimating labels for the unlabeled data points and then using them in a modified loss function. Forcing the network to make predictions on unlabeled points minimizes the entropy of the unlabeled predictions [29] and moves the decision boundaries to low-density regions. Additionally, dependent on the accuracy of the pseudolabels, we increase the amount of labeled data the model has access to and reduce overfitting to the initially small label set. There are many ways to incorporate unlabeled data/pseudolabel pairs into the loss function but the most common ways are to either create a specific loss term for the unlabeled data pseudolabel pairs [12], [18], [34], [35] or by using composite batches containing
both labeled and unlabeled data and keeping the standard supervised classification loss [20], [36].

The first application of this idea to the deep learning setting was presented by Lee [37]. Viewing the output of the neural network \( f(x) \) as a discrete probability distribution, Lee [37] assigned a hard pseudolabel \( \hat{y} \) for each unlabeled data point according to its most likely prediction \( \hat{y}_i = \arg \max_{y} f(x_i) \).

These pseudolabels were then used in a two-termed loss function of the form

\[
\hat{L}_{\text{ssl}} = \frac{1}{n_l} \sum_{i=1}^{n_l} l_s(f(x), y) + \eta \frac{1}{n_u} \sum_{i=1}^{n_u} l_s(f(x), \hat{y})
\]

where \( l_s \) is some loss function and \( \eta \) is a weighting parameter. The pseudolabels are recalculated every time the unlabeled data are passed through the network. As an alternative to hard labels, [19] used the full output probability distribution of the network as a soft label for each point. However, it was found that sharpening this distribution helped ensure the model’s prediction entropy was minimized.

As pointed out by Arazo et al. [20] there is a potential pitfall in this style of approach. Networks are often wrong and the neural network can overfit to its own incorrectly guessed pseudolabels in a process termed confirmation bias. Arazo et al. proposed using MixUp [22], soft labels, and a minimum ratio of labeled to unlabeled data to reduce confirmation bias. An alternative method to reduce confirmation bias is to use uncertainty quantification for the produced pseudolabels. These methods calculate a confidence score \( r_i \) for each pseudolabel \( \hat{y}_i \). The works of [12], [36], and [38] used the entropy of the probability distribution to give \( r_i \), while [39] used the distance between unlabeled points and labeled points in feature space. One can then either weight the loss terms by \( r_i \) or exclude pseudolabels whose \( r_i \) is below some threshold \( \tau \) in an attempt to prevent the network learning from low confidence predictions. Zhang et al. [40] sought to improve upon a fixed threshold value by having a separate threshold for each class. The threshold was lower for classes with a greater learning difficulty, according to the model’s entropy scores.

This style of approach is based upon the idea that the neural network is well calibrated, i.e., that the model’s softmax score is a good indicator of the actual likelihood of a correct prediction. However, recent research has suggested that modern neural networks are not as well calibrated as may be intuitively thought [41]. In our work, we demonstrate that, while an intuitive solution, uncertainty quantification is not needed for our pseudolabel approach.

In a completely different direction to network predictions, it has been shown from a classical perspective [25] that energy-based models, such as graphs, are well suited to the task of label propagation. Therefore, several works [36], [42], [43] have shown good performance by iteratively feeding the feature representation of a neural network to a graph, performing pseudolabel generation on the graph, and then using those labels to train the network. However, graphical approaches have yet to show that they can produce SOTA results compared with model-based approaches, such as [12] and [13]. In our work, we present a graphical approach, which surpasses the performance of model-based approaches, demonstrating that graphical approaches have a lot of promise for practical applications. In Fig. 1, we display the advantage of graphical pseudolabels using as a case study, the Two Moon dataset. From Fig. 1, we observe that the graphical approach offers a clear advantage over the network pseudolabel approach.

### III. PROPOSED TECHNIQUE

This section details our proposed semisupervised method. We cover the generation of pseudolabels, and the optimization of the model alongside a full algorithm, and we explore our multisample augmentation approach.

#### A. Problem Statement

From a joint distribution \( Z = (X, Y) \), we have a dataset \( Z \) of size \( n = n_l + n_u \) comprised of a labeled part of joint samples \( Z_l = \{x_l, y_l\}_{l=1}^{n_l} \) and an unlabeled part \( Z_u = \{x_l\}_{l=n_l+1}^{n} \) of single samples on \( X \). The labels come from a discrete set of size \( C \) \( y \in \{1, 2, \ldots, C\} \). Our task is to train a classifier \( f \), modeled by a neural network with parameter vector \( \theta \), which can accurately predict the labels of unseen data samples from the same distribution \( X \). The classifier \( f \) can be viewed as the composition of two functions \( z \) and \( g \) such that \( f(x) = g(z(x)) \). \( z : X \rightarrow \mathbb{R}^d \) is the embedding function mapping our data input to some \( d_p \) dimensional feature space and \( g : \mathbb{R}^d \rightarrow \mathbb{R}^C \) projects from the feature space to the classification space.
As a pseudolabel-based approach, we iteratively assign a pseudolabel $\hat{y}$ to all data points in $Z_u$ once per epoch. In this work, we generate hard pseudolabels using a graph-based approach first proposed by Zhou et al. [26] and first adapted to deep networks by Iscen et al. [36], which has been thoroughly studied in the classical machine learning literature. We give a visual overview of our approach in Fig. 2.

We first extract the feature representation of the dataset $V$ by using the embedding function of the neural network $z$ so that $V = \{z(x_1), \ldots, z(x_n)\}$. Unlike other works, we do not apply augmentation to the data while producing the pseudolabels. Using $V$ and a similarity metric $d$, we use $d(v_i, v_j) = (v_i, v_j)$, we construct a symmetric weighted adjacency matrix $W \in \mathbb{R}^{n \times n}$. The elements $w_{ij} \in W$ are given by $W_{ij} = d(v_i, v_j)$ and represent the pairwise similarities between data points. We then sparsify $W$ using the following nearest neighbor approach, which reads:

$$W_{ij} = \begin{cases} d(v_i, v_j), & \text{if } i \text{ is one of the } k \text{ nearest neighbors of } j \\ 0, & \text{otherwise.} \end{cases}$$

We then construct the degree matrix $D := \text{diag}(W)$. We use this to normalize the affinity matrix $W = D^{-1/2}WD^{-1/2}$, which prevents nodes with high degree having a large global impact. Finally, we use the initial label information to create the labeled matrix $Y \in \mathbb{R}^{n \times C}$

$$Y_{ij} = \begin{cases} 1, & \text{if } y_i = j \\ 0, & \text{otherwise.} \end{cases}$$

We can then propagate the information contained in $Y$ across the graph structure $W$ by minimizing the graphical Laplacian of the prediction matrix $F \in \mathbb{R}^{n \times C}$ plus a fidelity term to the supplied labeled data

$$Q(F) = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \left\| \frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}} \right\|^2 + \frac{\mu}{2} \sum_{i=1}^{n} \|F_i - Y_i\|^2$$

$$\text{for } i = \{1, \ldots, n\} \text{ and } j \in \{1, \ldots, n\} \text{ for all } F_i \in \mathbb{R}^C$$

where $\mu$ is a scalar weight. The first term enforces points that are close according to the metric $d$ to share a similar label, while the second term encourages initially labeled points to keep their label. To side-step the computationally infeasible closed form solution, we use the conjugate gradient approach to solve the linear system $(1 - \gamma W)F = Y$, where $\gamma (1 + \mu) = 1$. Using $F$, the pseudolabels $\hat{y}$ are given by

$$\hat{y}_j = \text{arg max}_{\gamma \in [0, C]} F_{ij}.$$  

A common problem in label propagation is that the pseudolabels produced by the graph may be unbalanced over the classes and Iscen et al. [36] attempted to weight the optimization problem to avoid this possibility. We found that the weighting approach of Iscen et al. [36] actually made the performance of the model worse than leaving the predictions as is. An alternate approach to counter class imbalances is distribution alignment [14], which enforces the distribution of the pseudolabel predictions to match some given prior distribution. The implementation of this idea by ReMixMatch focused on applying this idea to the network predictions and was not optimal for a graph-based framework.

Instead, we propose a novel smoother version of distribution alignment, which can be applied during or just after the conjugate gradient approach. We give a full algorithm for this in Algorithm 1. An algorithm is an iterative approach that smoothly deforms the pseudolabel predictions $F$ by the ratio $R$ between the prior distribution $D$ and the pseudolabel distribution of the unlabeled points $D_U$, thereby promoting the prediction of underrepresented classes and vice versa. To ensure the deformation is smooth, we clip the range of $R$ values to be close to one. We show in the experimental section that this approach improves the performance of the model.

$$C. \text{ Semisupervised Loss}$$

In the deep semisupervised setting, particularly in the current SOTA [12], [19], several works seek to minimize a semisupervised loss $L_{ssl}$ composed of two or more terms, one each for the labeled and unlabeled data points and potentially...
Algorithm 1 Smooth Distribution Alignment

1: **Input:** Pseudo-label Prediction $F \in \mathbb{R}^{n \times C}$, Prior Distribution $D \in \mathbb{R}^C$, labeled and unlabeled indexes $L = \{l_i\}_{i=1}^n$ and $U = \{u_i\}_{i=1}^m$, and max iteration $T$
2: **Output:** Adjusted Pseudo-label Prediction $F \in \mathbb{R}^{n \times C}$
3: for $t_i = 1, t_i++, \text{while } t_i < T$ do
4: $D_U \in \mathbb{R}^C \leftarrow \text{Initialize with zeros}$
5: Get the pseudo-label distribution:
6: for $u_i \in U$ do
7: $D_U[\arg \max_j F[u_i]] += \frac{1}{n_u}$
8: end for
9: $R = D/D_u$
10: **Clip values for smooth deformation:**
11: $R[R > 1.01] = 1.01$ and $R[R < 0.99] = 0.99$ # Deform the current predictions:
12: for $c_i = 1, c_i++, \text{while } c_i < C$ do
13: $F[U, c_i] == R[c_i]$
14: end for
15: Row normalize $F$ to give valid distributions.
16: end for

Others covering entropy minimization and so on, which has the following form:

$$L_{ssl} = \frac{1}{n_l} \sum_{i=1}^{n_l} l_s(f(x), y) + \eta \frac{1}{n_u} \sum_{i=1}^{n_u} l_s(f(x), \hat{y}) + \cdots \quad (7)$$

where $\eta$ is a balancing parameter. For our approach, we wanted to strip away as much complexity from the loss function as possible in an effort to see what elements are required for good performance. We move away from using a composite loss and instead only use the standard supervised loss, which has worked so well in supervised image classification. To include our unlabeled data, we use composite batches of size $b$ that is made up of $b_l$ labeled samples and $b_u$ unlabeled samples to which we have assigned a pseudolabel $\hat{y}$. Our semisupervised loss, $L_{ssl}$, is given by

$$L_{ssl} = \frac{1}{b} \sum_{i=1}^{b} l_s(f(x_i), y_i). \quad (8)$$

Note that in (8), we use $y_i$ to refer to both ground truth labels and pseudolabels for brevity. What is remarkable about this loss is its simplicity. There is no confidence thresholding of the pseudolabels, additional weighting parameters, no consistency-based terms, or other regularizations. Instead, we rely upon the strength of the combination of an energy-based graphical approach to pseudolabels estimation and the clever use of strong augmentation to increase generalization.

D. Training the Model

For initialization purposes, we quickly extract some baseline knowledge from the dataset by minimizing a supervised loss $L_{sup}$, for one hundred passes through $Z_i$. This supervised loss reads

$$L_{sup} = \frac{1}{b} \sum_{i=1}^{b} l_s(f(x_i), y_i) \quad (9)$$

where $b$ is the batch size and $l_s$ is the cross entropy loss.

We emphasize that (9) uses only the tiny labeled set $Z_l$ and is performed once before the main semisupervised optimization. We then begin our main learning loop which alternates between updating the pseudolabel predictions and minimizing the semisupervised loss $L_{ssl}$ for one epoch, where we define one epoch to be one pass through the unlabeled data $Z_u$. This cycle then runs for a total of $S$ optimization steps and the fully trained model is then tested on the relevant testing set. Note that we do use Mixup [22] on both $L_{sup}$ and $L_{ssl}$ with a beta distribution parameters $\alpha$. In Algorithm 2, we give an overview of training our model for $S$ optimization steps.

Algorithm 2 Training Scheme for LaplaceNet

1: **Input:** labeled data $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$, unlabeled data $Z_u = \{x_i\}_{i=1}^{n_{unl}}$, unlabeled model $f$ with trainable parameters $\theta$ and embedding function $z$. Hyper-parameters: Number of optimization steps $S$
2: # Initialization:
3: for $i = 1 \rightarrow 100$ do
4: optimize $L_{sup}$ over $Z_l$
5: Set current step to zero $s_i = 0$
6: # Main Optimization Process:
7: while $s_i < S$ do
8: Extract features: $V = \{z(x_i)\}_{i=1}^{n_l}$
9: Construct Graph Matrix $W$
10: Degree Normalization $W = D^{\frac{1}{2}}WD^{\frac{1}{2}}$
11: Propagate Information via $Q(F)$
12: Distributed Alignment on $F$
13: $\hat{y}_i = \arg \max F \forall n_l + 1 \leq i \leq n$
14: for $i = 1 \rightarrow \lfloor \frac{n_u}{b} \rfloor$ do
15: $B_k = \{x_i, y_i\}_{i=1}^{b} \subset Z_l$, $B_U = \{x_i, \hat{y}_i\}_{i=1}^{b} \subset Z_u$
16: Composite Batch $B = B_k \cup B_U$
17: Optimize $L_{ssl}$, $s_i = s_i + 1$
18: end while

E. Multisampling Augmentation

Since the work of [13], several approaches have implemented the use of strong augmentations [12], [14], [18] to the problem of SSL, with each work having a different way of including augmentation to their framework. Very recent works [14], [18] have begun using multiple augmented versions of the same unlabeled image. As yet there is no motivation for why this multiple sampling idea is preferable to alternatives, such as larger batch sizes or running the code for more steps. In this section, we offer a theoretical motivation for why multisampling improves generalization along with a mathematically bound on its performance gain.
With this knowledge in mind, we provide a simple method for including augmentation averaging into our SSL framework and demonstrate this approach increases accuracy and reduces the sensitivity of the model to data augmentation.

We view an augmentation strategy \( A \) as a set \( T \) of transformations \( t : X \rightarrow X \) and denote it as \( T = \{t_1, t_2, \ldots, t_n\} \). The current standard approach, which the majority of existing techniques follow, is to simply sample \( t \sim T \) once for each data point and compute some augmented loss \( L_{\text{Aug}} \)

\[
L_{\text{Aug}} = \frac{1}{n} \sum_{i=1}^{n} l_i(f(t(x_i)), y_i). \tag{10}
\]

However, we argue that such a simple implementation, might not extract the full information present in the augmentation. If we want to encourage our model output to be more resistant to data augmentations from \( T \), we should optimize to a lower of both early and later ones. This result explains prior behaviors reported but not reasoned in past papers, such as [14]. When using an \( n_a \) sample, the computational complexity increases as \( O(n_a) \), but as there should be diminishing returns for increasing \( n_a \), it should only be necessary to use \( n_a \) values slightly above one. As we have shown that a multisample approach should offer generic performance increases for suitable \( T \), we change (8) and (9) to a multisample version. For (8), this becomes

\[
L_{\text{ssl}} = \frac{1}{b} \frac{1}{n_a} \sum_{i=1}^{n_a} \frac{1}{b} \sum_{j=1}^{b} l_i(f(t_j(x_i)), y_i) \tag{14}
\]

where the index \( j \) represents repeated samples from \( T \) and again \( y_i \) refers to both ground truth and pseudolabels. In the ablation section, we perform a thorough experimental evaluation to test the theoretical predictions we have made in this section.

**Augmentation Implementation:** Similar to other approaches, we use two different augmentation strategies: one for labeled data and another for unlabeled data. However, we apply strong augmentations to both labeled and unlabeled data, unlike past approaches [12], which reported divergences using strong augmentations to both labeled and unlabeled data. In this section, we detail the implementation of LaplaceNet, including hyperparameter values and training schemes, and the evaluation protocol we used to measure our model’s performance and compare it against the current SOTA.

**A. Dataset Description**

We use three image classification datasets: CIFAR-10 and CIFAR-100 [44] and Mini-ImageNet [45]. Following standard protocol, we evaluate our method’s performance on differing amounts of labeled data for each dataset.

1) **CIFAR-10**. CIFAR-100 Containing 50k training images and 10k test images, these datasets contain ten and 100 classes, respectively. The image size is small at 32 by 32 pixels. We perform experiments using 500, 1k, 2k, and 4k labels for CIFAR-10 and 4k and 10k labels for CIFAR-100.

2) **Mini-ImageNet**. A subset of the popular ImageNet dataset, containing 100 classes each with 500 training and 100 test images. The resolution of the images is 84 × 84 pixels and represents a much harder challenge.

### Table 1

| Labeled Transform | Unlabeled Transform |
|-------------------|---------------------|
| Random Horizontal Flip | Random Crop and Pad |
| RandAugment Sample | RandAugment Sample |
| CutOut Normalisation |                   |

*UNLABELED DATA FOR NORMALISATION, WE USE THE OFFICIAL CHANNEL.*

IV. IMPLEMENTATION AND EVALUATION

In this section, we detail the implementation of LaplaceNet, including hyperparameter values and training schemes, and the evaluation protocol we used to measure our model’s performance and compare it against the current SOTA.

**A. Dataset Description**

We use three image classification datasets: CIFAR-10 and CIFAR-100 [44] and Mini-ImageNet [45]. Following standard protocol, we evaluate our method’s performance on differing amounts of labeled data for each dataset.

1) **CIFAR-10**. CIFAR-100 Containing 50k training images and 10k test images, these datasets contain ten and 100 classes, respectively. The image size is small at 32 by 32 pixels. We perform experiments using 500, 1k, 2k, and 4k labels for CIFAR-10 and 4k and 10k labels for CIFAR-100.

2) **Mini-ImageNet**. A subset of the popular ImageNet dataset, containing 100 classes each with 500 training and 100 test images. The resolution of the images is 84 × 84 pixels and represents a much harder challenge.

### Table 1

| Labeled Transform | Unlabeled Transform |
|-------------------|---------------------|
| Random Horizontal Flip | Random Crop and Pad |
| RandAugment Sample | RandAugment Sample |
| CutOut Normalisation |                   |

*UNLABELED DATA FOR NORMALISATION, WE USE THE OFFICIAL CHANNEL.*
We use an initial learning rate of \( \alpha \). We pare it against the current SOTA models for SSL. For ease of CIFAR-10/CIFAR-100 and Mini-Imagenet datasets and comparison with other works, we use the WideResNet (WRN) 28-2 and a WRN-28-8 [46] architecture. We additionally use a ResNet-18 [5] for Mini-Imagenet. For all models, we set the dropout rate to 0. For the “13-CNN,” we add a l2 normalization layer to the embedding function. Infrastructure: For all experiments, we use 1-2 NVIDIA P100 GPUs. Training Details: We train with stochastic gradient descent (SGD) using Nesterov momentum \( n_m \) with value 0.9 and weight decay \( \omega \) with value 0.0005. We use an initial learning rate of \( l_r = 0.3 \) and use \( S = 250000 \) optimization steps in total. We utilize a cosine learning rate decay such that the learning rate decays to zero after 255000 steps. We do not make use of any EMA model averaging.

2) Parameters: We list the parameter values used in Table II. Most parameter values are common parameter settings from the deep learning field and are not fine-tuned to our application. Being able to work with reasonably generic parameters is well-suited to the task of SSL where using fine-tuning over validation sets is often impossible in practical applications.

C. Evaluation Protocol

We evaluate the performance of LaplaceNet on the CIFAR-10/CIFAR-100 and Mini-Imagenet datasets and compare it against the current SOTA models for SSL. For ease of comparison, we split the current SOTA into two groups.

1) Methods which used the “13-CNN” architecture [32]: Pi-Model [30], MT [32], VAT [31], label propagation for deep SSL (LP) [36], smooth neighbors on teacher graphs (SNTG) [47], stochastic weight averaging (SWA) [48], ICT [21], dual student [49], transductive semisupervised deep learning (TSSDL) [39], density-aware graphs (DAGs) [43], and pseudolabel mixup [20]. Unfortunately, due to the natural progress in the field, each paper has different implementation choices which are not standardized. Despite this, comparisons with this group are still useful as a barometer for model performance.

2) Recent methods which used the WRN [46] (MixMatch [19], FixMatch (RandAugment variant) [12], and unsupervised data augmentation (UDA) [13]). To guarantee a fair comparison with these techniques, and as suggested by [50], we used a shared code-base for UDA and FixMatch which reimplemented the original baselines. Additionally, we then ensured UDA and FixMatch used the same model code, the same optimizer with the same parameters, the same number of optimization steps, and the same RandAugment implementation as our approach.

1) Evaluation Protocol: For each dataset, we use the official train/test partition and use the Top-1 error rate as the evaluation metric. For each result, we give the mean and standard deviation over five label splits.

V. RESULTS AND DISCUSSION

In this section, we discuss the experiments we performed to evaluate and compare our model against the SOTA. Additionally, we detail several ablation experiments, which explore the benefits of graph-based pseudolabels, the effect of augmentation averaging, and evaluate the importance of individual components.

A. Comparison With SOTA

First, we test our model on the less complex CIFAR-10 and CIFAR-100 datasets. In Table III, we compare LaplaceNet against the first group of methods using the 13-CNN network. Our approach, by some margin, produces the best results on CIFAR-10 and CIFAR-100 and represents a new SOTA for pseudolabel methods. We obtain a lower error rate using 500 labels than the recent work of Arazo et al. [20] obtain using 4000 labels. For CIFAR-100, LaplaceNet is a full 6% more accurate than any other approach and the first method to achieve an error rate below 30% on CIFAR-100 using 10k labels. In Table IV, we compare against the second group of methods using the WRN-28-2 network. LaplaceNet is again the best performing method, outperforming the works of UDA [13] and FixMatch [12]. In particular, we find a significant increase in performance on the more complex CIFAR-100 dataset and beat the other considered methods by more than 3% with 10k labels.

To test the performance of LaplaceNet on a more complex dataset, we evaluate our model on the Mini-ImageNet dataset, which is a subset of the well-known ImageNet dataset and in Table V, we compare our results against all other methods which have used this dataset. Once again, we find our method performs very well, producing an error rate of 10% and 7% better than any other method on 4k and 10k labeled images, respectively. Demonstrating our approach can be applied to complex problems in the field. Additionally, we are more than 20% more accurate than the nearest graphical approach (LP).

To test the effect of increasing network size on our performance we also ran our model on CIFAR-10/100 using a WRN-28-8 (26 million parameters) architecture and compared to the WRN-28-2 (1.6 million parameters) architecture.

| PARAMETER | CIFAR-10 | CIFAR-100 | MINI-IMAGENET |
|-----------|----------|-----------|---------------|
| \( \alpha \) | 1.0 | 0.5 | 0.5 |
| \( \mu \) | 0.01 | 0.01 | 0.01 |
| \( k \) | 50 | 50 | 50 |
| \( S \) | \( 2.5 \times 10^5 \) | \( 2.5 \times 10^5 \) | \( 2.5 \times 10^5 \) |
| \( b \) | 300 | 100 | 100 |
| \( b_t \) | 48 | 50 | 50 |
| \( l_r \) | 0.03 | 0.03 | 0.1 |
| \( n_m \) | 0.9 | 0.9 | 0.9 |
| \( \omega \) | \( 5 \times 10^{-4} \) | \( 5 \times 10^{-4} \) | \( 5 \times 10^{-4} \) |
TABLE III

| DATASET | CIFAR-10 | CIFAR-100 |
|---------|----------|-----------|
| Method  | 500      | 1000      | 2000      | 4000      | 10000 |
| Supervised Baseline | 37.12 ± 0.89 | 26.60 ± 0.22 | 19.53 ± 0.12 | 14.02 ± 0.10 | 53.10 ± 0.34 | 36.59 ± 0.47 |
| II-Model | 12.36 ± 0.31 | - | 39.19 ± 0.36 | - | - |
| MT | 27.45 ± 2.64 | 21.55 ± 1.48 | 15.73 ± 0.31 | 12.31 ± 0.20 | 45.36 ± 0.49 | 36.08 ± 0.51 |
| VAT | - | - | 11.36 ± 0.34 | - | - |
| MT-LP | 24.02 ± 2.44 | 16.93 ± 0.70 | 13.22 ± 0.29 | 10.61 ± 0.28 | 43.73 ± 0.20 | 35.92 ± 0.47 |
| SNTG | - | 18.41 ± 0.52 | 13.64 ± 0.32 | 9.89 ± 0.34 | - | 37.97 ± 0.29 |
| MT-fast-SWA | - | 15.38 ± 0.12 | 11.02 ± 0.12 | 9.05 ± 0.21 | - | 34.10 ± 0.31 |
| MT-ICT | - | 15.48 ± 0.78 | 9.26 ± 0.09 | 7.29 ± 0.02 | - | - |
| Dual Student | - | 14.17 ± 0.38 | 10.72 ± 0.19 | 8.89 ± 0.09 | - | 32.77 ± 0.24 |

TABLE IV

| DATASET | CIFAR-10 | CIFAR-100 |
|---------|----------|-----------|
| Method  | 500      | 1000      | 2000      | 4000      | 10000 |
| MixMatch | 9.65 ± 0.94 | 7.05 ± 0.15 | 6.34 ± 0.06 | - | - |
| SAME CODEBASE | | | | | |
| UDA | 6.88 ± 0.74 | 5.61 ± 0.16 | 5.40 ± 0.19 | 36.19 ± 0.39 | 31.49 ± 0.19 | 34.87 ± 0.17 | 30.89 ± 0.18 |
| FixMatch(RA) | 5.92 ± 0.11 | 5.42 ± 0.11 | 5.30 ± 0.08 | 33.16 ± 0.22 | 27.49 ± 0.22 |
| LaplaceNet | 5.57 ± 0.60 | 4.71 ± 0.05 | 4.35 ± 0.10 | - | - |

TABLE V

| METHOD | 4000 | 10000 |
|--------|------|-------|
| Supervised Baseline | 66.04 ± 0.32 | 52.89 ± 0.33 |
| Consistency Regularisation Methods | | |
| MT | 72.51 ± 0.22 | 57.55 ± 1.11 |
| MT-LP | 72.78 ± 0.15 | 57.35 ± 1.66 |
| Pseudo-Label Methods | | |
| LP | 70.29 ± 0.81 | 57.58 ± 1.47 |
| Pseudo-Label Mixup | 56.49 ± 0.51 | 46.08 ± 0.13 |
| LaplaceNet | 46.32 ± 0.27 | 39.43 ± 0.09 |

TABLE VI

| DATASET | CIFAR-10 | CIFAR-100 |
|---------|----------|-----------|
| Model  | 500      | 1000      | 2000      | 4000      | 10000 |
| WRN-28-2 | 5.57 ± 0.60 | 4.35 ± 0.10 | 33.16 ± 0.22 | 27.49 ± 0.22 |
| WRN-28-8 | 3.81 ± 0.37 | 2.87 ± 0.18 | 26.61 ± 0.10 | 22.11 ± 0.23 |

in Table VI. Unsurprisingly, we achieved a large performance improvement using a WRN-28-8 on both CIFAR-10 and CIFAR-100, with a 2.87 error rate on CIFAR-10 using 4k labels and a 22.11% error rate on CIFAR-100 using 10k labels.

From these results, we underline the strength of our technique. First, we show that our proposed method has a performance advantage over a range of consistency and pseudolabel-based methods, including other related graphical models, on both simpler and more complex tasks. Second, we also showed this performance behavior to be true over a range of model architectures. Thus demonstrating that improving the pseudolabels that are fed back to the neural network by exploiting classical energy models can greatly improve the final test accuracy of the network.

B. Graph-Based Pseudolabels

Many pseudolabel-based techniques [12], [20] have produced SOTA results using pseudolabels generated directly by the network, rather than using an energy-based approach, such as label propagation on a constructed graph, which is computationally more complex. Therefore, in this section, we examine the advantage of using a graph-based approach. To test the importance of graph-based pseudolabels, we created two variants of LaplaceNet, both without distribution alignment and with $n_i = 1$.

1) The pseudolabels are generated directly from the network predictions: $\hat{y}_i = \arg\max x_i \forall i > l$.
2) The pseudolabels are generated from the graph, as in (6), $\hat{y}_i = \arg\max_{j} F_{ij} \forall i > l$.

We then compared the Top-1 error rate of these two variants on the CIFAR-100 dataset, see Fig. 3. The graph-variant greatly outperformed the direct prediction variant, emphasizing the clear advantage that graphically produced
pseudolabels have. What is contributing to this advantage? As an energy-based approach, propagation on the graph incorporates information on the global structure of the data, while the network is making a purely local decision at each point. Arazo et al. [20] showed that the naive network-based pseudolabel approach could not generate an accurate solution for the “Two Moons” toy dataset, despite the fact that this problem has been solved by graphical methods for some time [25]. Thus demonstrating that purely local decisions are detrimental to accuracy when the global structure of data is not taken into account.

C. Augmentation Averaging

In this article, we justify a multiple augmentation approach to further improve semisupervised models. In this section, we present the experimental verification of our theoretical predictions about augmentation averaging as well as compare its effect to potential alternative techniques. To test the effect of augmentation averaging, we ran our approach on the CIFAR-100 dataset using the 13-CNN network for a range of values \( n_a = [1, 3, 5] \). Additionally, we compared the changes caused by augmentation averaging to the more common approaches of scaling the batch size \( b \) and labeled batch size \( b_l \) by \( [1, 3, 5] \) and scaling the number of optimization steps \( S \) by \( [1, 3, 5] \).

To quantify the effect of a given change, we use two measures: the augmentation invariance of the classifier, which we define in this article, and the Top-1 error. Augmentation invariance measures the extent to which the classifier’s performance changes under data augmentation. Given an augmentation function \( u : \mathcal{X} \rightarrow \mathcal{X} \) and a classifier \( f_\theta \) the augmentation invariance \( V \) with respect to a dataset \( Z = \{x_i, y_i\}_{i=1}^n \) is given by

\[
V_Z = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\arg \max f_\theta(u(x_i)) = y_i} - \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{\arg \max f_\theta(x_i) = y_i},
\]

which can be viewed as the performance ratio with and without data augmentation. We consider both the augmentation invariance of our model with respect to the fully labeled training and test data in order to give a full picture of the model’s invariance, but we still only use a subset of the labeled data for training.

In Fig. 4, we present our findings. We found that naively scaling the number of optimization steps without changing the hyperparameters led to the model diverging as we spent too many epochs at a high learning rate. Therefore, we provide results for the other two considered techniques which can be directly compared. As theorized in Section III, we find that increasing the number of augmentation samples decreased the sensitivity of the model’s predictions to augmentation on both the training and test data. An almost identical effect was found by scaling the batch size. However, the major difference between the two is their effect on the Top-1 error rate. We found scaling the batch size offered no improvement to Top-1 error; in fact, the largest batch size offered the worst outcome while increasing the number of augmentation samples noticeably improved the model’s accuracy. Additionally, as theorized in Section III, we see that the gain in performance from \( n_a = 1 \rightarrow 3 \) is much greater than \( n_a = 3 \rightarrow 5 \), supporting our statements regarding the exponential bound in probability.

To further test the limits of augmentation averaging, we ran additional experiments with \( n_a = [7, 10] \) with 4k labels on the CIFAR-10 dataset and present these results in Table VII. We see that increasing \( n_a \) above-mentioned five led to small

![Experimental comparison of the effect of using pseudolabels produced in a graphical framework versus pseudolabels generated by the neural network on the Top-1 error rate on the CIFAR-100 dataset.](image)

**Table VII**

| Dataset | CIFAR-100 |
|---------|-----------|
| \( n_a \) | Top-1 Error | Test-Data Aug-invariance |
| 1       | 32.44 ± 0.26 | 93.97 ± 0.01 |
| 3       | 31.45 ± 0.18 | 94.84 ± 0.01 |
| 5       | 31.15 ± 0.38 | 95.11 ± 0.04 |
| 7       | 31.02 ± 0.28 | 95.09 ± 0.03 |
| 10      | 30.95 ± 0.25 | 95.18 ± 0.06 |
Fig. 4: Comparison on the effect of increasing batch size versus increasing the number of augmentation samples on Top-1 error rate, test data augmentation invariance, and training data augmentation invariance for the CIFAR-100 dataset. Increasing the amount of augmentation averaging decreased the error rate while also decreasing the sensitivity of the model’s output predictions to augmented data. Increasing the batch size had a similar effect on the model’s sensitivity, but it offered no improvement to model accuracy.

**TABLE VIII**

| EFFECT OF REMOVING INDIVIDUAL COMPONENTS FROM THE BASELINE MODEL ON TOP-1 ERROR RATE FOR CIFAR-100 ON THE 13-CNN NETWORK |
|---|
| DATASET | CIFAR-100 |
| MODEL | 4k | 10k |
| Baseline | 32.41 ± 0.25 | 27.37 ± 0.20 |
| COMPONENT REMOVED | 4k | 10k |
| RandAugment | 44.43 ± 0.66 | 34.75 ± 0.23 |
| Distribution Alignment | 33.26 ± 0.24 | 29.07 ± 0.07 |
| MixUp | 33.74 ± 0.26 | 28.02 ± 0.20 |

Together these results suggest that scaling the number of augmentation samples could be a great option for semisupervised models using suitable strong augmentations.

**D. Component Evaluation**

As LaplaceNet combines several different techniques, we tested the importance of strong augmentation, distribution...
alignment, and MixUp to the overall accuracy of the model. We created a baseline model \((n_a=1)\) and then remove each component one at a time and tested the performance on the CIFAR-100 dataset, see Table VIII. While the removal of each component decreased the performance of the model, it is clear the most crucial component to model performance is a strong augmentation and removing it drastically reduces model accuracy. However, unlike other works [20], we find that while MixUp [22] offers a small advantage is it not critical to composite batch pseudolabel approaches. This may be due to the advantages of graph-based approaches overcoming the flaws of naïve neural network predictions.

VI. CONCLUSION

We propose a new graph-based pseudolabel approach for semisupervised image classification, LaplaceNet, that outperforms the current SOTA on several datasets while having a much lower model complexity. Our model utilizes a simple single-term loss function without the need for additional complexity while additionally avoiding the need for confidence thresholding or temperature sharpening which was thought to be essential for SOTA performance. We instead generate accurate pseudolabels through a graph-based technique with distribution alignment. We also explore the role that augmentation plays in SSL and justify a multisampling approach to augmentation, which we demonstrate through rigorous experimentation improves both the generalization of the network as well as the model’s sensitivity to augmented data.

APPENDIX A
COMPUTATIONAL TIME

To give clarity on how long our code takes to run we provide the computational run times of LaplaceNet on the CIFAR-100 dataset using the 13-CNN model for a variety of settings, see Table IX. Each experiment was run on one P100 NVIDIA GPU. From Table IX, we see that the time increase caused by increasing the batch size or increasing the number of samples is very similar. Componentwise, removing strong augmentation gives the largest decrease in computational time while removing the graphical propagation saved just over an hour on CIFAR-100. This represents a very small time tradeoff given the advantages present in using graphical pseudolabels.

APPENDIX B
AUGMENTATION POOL

In this work, we use RandAugment [16], rather than a learned augmentation strategy, such as AutoAugment [15], which has a large computational cost. In Table X, we detail the augmentation pool used. Additionally, we apply CutOut [17] augmentation after RandAugment sampling.

We use two different augmentation strategies in our work: one for labeled data and one for unlabeled data. We use “strong” augmentations, RandAugment and CutOut, on both labeled and unlabeled data with the only difference being that we sample once from RandAugment for labeled data and twice for unlabeled data. Given a preselected list of transformations, RandAugment randomly samples from the list with each transformation having a magnitude parameter. Rather than optimizing this parameter on a validation set, which may not exist in semisupervised applications, we sample a random magnitude from a preset range. This is the same as is done in FixMatch [12] and UDA [13]. We list the transformations for RandAugment in Table X.

REFERENCES

[1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1097–1105.
[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
Philip Sellars received the Ph.D. degree from the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, U.K., in 2021.

He is currently a Machine Learning Researcher with Darktrace, Cambridge, a cyber-security company. His research was supported by the Engineering and Physical Sciences Research Council, Swindon, U.K., and the National Physical Laboratory, Teddington, U.K. His research interests include the applications of semisupervised graph-based learning to the problem of image classification, including the medical, natural, and hyperspectral imaging domains, computer vision, self-supervised and semisupervised learning, graphical architectures, and the training and deployment of large-scale natural language modeling for understanding cybersecurity data.

Angelica I. Aviles-Rivero (Member, IEEE) received the Ph.D. degree from the Polytechnic University of Catalonia, Barcelona, Spain, in 2017.

She is currently a Senior Research Associate with the Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, U.K. She has widely explored how to develop hybrid models, where explicit mathematical modeling and machine learning are considered. This line of research, (Math + ML + X), has allowed her to gain expertise in a wide range of applications, (X), including medical imaging, computational photography, computer graphics, and remote sensing to name a few. Her research interests include the intersection of computational mathematics, computer vision, machine learning for applications to large-scale real-world problems, inverse problems, machine learning, and graph-based techniques for imaging and nonimaging applications.

Dr. Aviles-Rivero is an Society for Industrial and Applied Mathematics (SIAM), Computer Vision Foundation (CVF), and Association for Computing Machinery (ACM) member.

Carola-Bibiane Schönlieb received the B.S. degree from the Institute for Mathematics, University of Salzburg, Salzburg, Austria, in 2004, and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 2009.

From 2004 to 2005, she held a teaching position in Salzburg. She held a post-doctoral position at the University of Göttingen, Göttingen, Germany, for one year. She became a Lecturer with the Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, in 2010, where she was promoted to a Reader in 2015 and a Professor in 2018. Since 2011, she has been a fellow of the Jesus College, University of Cambridge, where she is currently a Professor of applied mathematics with DAMTP, the Head of the Cambridge Image Analysis Group, the Director of the Cantab Capital Institute for Mathematics of Information, and the Director of the Engineering and Physical Sciences Research Council Center for Mathematical and Statistical Analysis of Multimodal Clinical Imaging. Her research interests include variational methods, partial differential equations, machine learning for image analysis, image processing, and inverse imaging problems.