ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230273

Resource-Constrained Knowledge Diffusion Processes Inspired by Human Peer Learning

Ehsan Beikihassan, Amy K. Hoover, Ioannis Koutis [*], Ali Parviz and Niloofar Aghaieabiane

Ying Wu College of Computing
New Jersey Institute of Technology
Newark, NJ 07102, USA
{eb283, ahoover, ikoutis, ap2248, na396}@njit.edu

Abstract. We consider a setting where a population of artificial learners is given, and the objective is to optimize aggregate measures of performance, under constraints on training resources. The problem is motivated by the study of peer learning in human educational systems. In this context, we study *natural* knowledge diffusion processes in networks of interacting artificial learners. By 'natural', we mean processes that reflect human peer learning where the students' internal state and learning process is mostly opaque, and the main degree of freedom lies in the formation of peer learning groups by a coordinator who can potentially evaluate the learners before assigning them to peer groups. Among else, we empirically show that such processes indeed make effective use of the training resources, and enable the design of modular neural models that have the capacity to generalize without being prone to overfitting noisy labels.

1 Introduction

A core issue in human educational systems is how best to use the existing resources, or more broadly how to diffuse knowledge over networks of human interactions that reflect pragmatic constraints [11]. In particular, in educational settings, humans learn in peer groups. Recent literature has introduced quantitative models and knowledge diffusion processes for peer learning. While these models are simple, they appear to give insights that may have practical implications for peer learning [41]. This provides motivation for a study of analogous knowledge diffusion processes in the context of machine learning.

In human educational systems, the objective is, broadly speaking, to train multiple individuals in finite time horizons and under resource constraints, e.g. the scarcity of teachers and their bounded individual teaching capacity. In the context of machine learning, we can informally define a proxy problem as follows: We are given a population of parametric models, a finite budget of accesses to the training set labels, and a finite time budget, and the objective is to maximize measures of average performance over the population.

1.1 Natural Knowledge Diffusion

Towards addressing the above problem, we introduce our *natural Knowledge Diffusion* framework (NKDIFF), that models human peer learning processes, following the paradigm developed in [41].

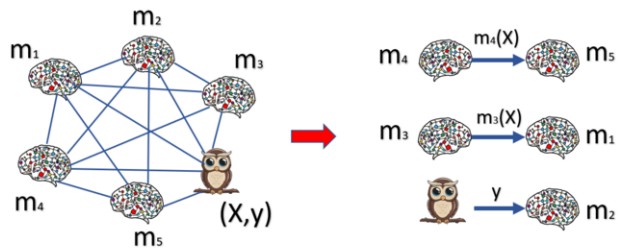

Figure 1: Groups during one round of training

NKDIFF implicitly defines a set of admissible natural training processes for populations of learners. As illustrated in Figure 1, we are given a population of identical-architecture neural models that communicate with each other in *rounds*. We consider only the special case of a fully connected network of learners, indicating that all pairwise communications are possible. The network includes an *Oracle* agent that holds the true labels y. The training points X are available to all models. The goal is to diffuse the Oracle's knowledge over the network and train all models with the objective of optimizing measures of aggregate performance. Towards the goal, we enable models to also operate as teachers by providing to other parts their *pseudolabels* $m_i(X)$. [1] We also impose the following constraints on the training resources of the process:

(i) *Training Capacity Constraint*: We set a bound on how many learners can interact with a teacher in each round of learning.

(ii) *Time Constraint*: We consider finite time budgets, typically smaller than what is required for 'convergence' to a feasible optimal. In other words, we favor *fast diffusion*.

1.2 Significant Observations

The value of initiating and pursuing a study of natural peer learning processes in the context of machine learning is validated by the remarkable generalization properties of the various NKDIFF processes we consider in this work (see Section 2).

[*] E. Beikihassan, A.K. Hoover, and I. Koutis contributed equally to this work. Corresponding author: *ikoutis+ecai23@njit.edu*

[1] Note that participant models learn only through standard backpropagation training with labels they receive from other participants. Participants are oblivious of their own and the other participants' parameters, backpropagation gradients, and training losses, i.e. their learning processes are handled by individual black-box optimizers.

Please contact the corresponding author for any appendices or supplementary material mentioned in the paper.

These are our main questions and significant observations:

A. *Are partially trained teachers detrimental or useful?*

It is not a priori clear to what extent employing partially trained teachers can be detrimental to population training. Surprisingly, the top accuracy reached by NKDIFF processes *at convergence*, is comparable to that of baselines that do not employ partially trained teachers. When looking at finite time budgets (i.e. in pre-convergent states), partially trained teachers are actually *useful*. NKDIFF reaches higher accuracy significantly faster than standard population training algorithms with respect to the number of accesses to the training set. In other words, NKDIFF makes more efficient use of the constrained training resources, which is precisely its main objective. In effect, NKDIFF trades off accesses to the training set with parameters that are distributed over independent learners [2] (Section 3.2)

B. *What is the impact of grouping policies?*

As elaborated in section 2.1, the main degree of freedom in NKDIFF is in deciding the peer groups in each training round. We find that the choice of a specific mechanism or policy for NKDIFF has a clear impact on performance measures, largely corroborating the results obtained via simple analytical models in [41]. (Sections 2.2 and 3.3)

C. *Does training diversity have positive effects?*

A population of models trained with NKDIFF consists of individuals that have undergone very diverse training processes due to their interactions with different teachers whose pseudolabels define constantly evolving loss functions. It is then natural to ask what the population-level effect of this diversity is. We find that when the population of learners is construed as an *ensemble* model, NKDIFF prevents the ensemble from memorizing random training labels *despite* the individual capacity of its members to do so [44]. We also find that the population can still generalize in noisy label settings, without overfitting to the noisy labels. Combining these observations we arrive at the conclusion that, as a training framework, NKDIFF allows the composition of simple learning modules into a single model that has the capacity for generalization, but lacks the capacity for overfitting. (Section 3.4)

These and other observations point to various topics and previously observed phenomenal in machine learning that may offer partial justifications for NKDIFF properties. We provide a related discussion in Section 1.4

1.3 Related Work

Distributed Optimization. The NKDIFF algorithms we consider in this work can be viewed as distributed optimization algorithms. There is a large related literature on distributed optimization [26, 35] and federated learning [43, 18]. Our work is more closely related to recent works on (decentralized) learning of personalized models through interactions of learners [40, 3, 39]. All these works ascribe algorithmic intent to the learners and consider algorithms for jointly optimizing them, thus requiring access to their internal state (e.g., gradients, or parameters) that may be exchanged with their neighbors on the network. They are also motivated by practical problems where the learners hold different data with true labels, so the main problem is to arrive at better personalized models (and/or a global model) that combine local data. In sharp contrast, NKDIFF algorithms do not 'open the box' of parameters, gradients, or training losses, and rely on oblivious learners that simply see and passively trust the pseudolabels provided by their designated teachers. In that sense, our work derives from the well-researched topic of information diffusion in social networks (e.g. see [11, 22]. Many of these works on information diffusion fix simple mechanisms of information broadcasting and examine the impact of the network structure. On the other hand, the work on peer learning mechanisms in [41] fixes a simple network and considers the effect of broadcasting mechanisms. Our work is inspired by the latter paradigm.

Ensemble Training. A population of trained models can be naturally viewed as a single ensemble classifier. The focus of ensemble learning algorithms is on maximizing ensemble accuracy, and the number of the ensemble constitutes is a hyperparameter that frequently can be picked to be a relatively small number [7]. It is clear that a population of learners can have high ensemble accuracy with a low average accuracy; to see that, it suffices to think of a population containing only a few trained individuals while the rest return random outputs. Thus our objective of training a *given* number of learners for average accuracy is considerably different. Nevertheless, the time resource is also of interest in the context of ensemble learning as well. Boosting algorithms such as Adaboost [15, 16] and Gradient Boost [8, 17] sequentially add classifiers to the ensemble over time, hence they are not time-aware. Bagging algorithms on the other hand train the ensemble's constituent models in parallel, improving accuracy under time constraints. Our results indicate that when viewed as ensemble training, knowledge diffusion has markedly different properties relative to plain bagging baselines. However, a deeper study in this direction is not in this paper's main scope.

Education-inspired ML. This paper adds to the broad literature on machine learning methods inspired by human education. We have in particular drawn inspiration from Curriculum Learning (CL) [5, 42, 37] and Knowledge Distillation [21]. The main premise of CL is that *single* learners can reach higher levels of generalization (and possibly earlier in the training process) by carefully planning the way the training set is presented to the (passive) model undergoing the training. Related to CL is Active Learning (AL) [4, 34] where control to generalization and training speed is achieved through the active choice of training points by a single learner. CL, and AL are somewhat related to our work, in the sense that they are resource-aware methods, but they aim at different objectives, they are orthogonal to our knowledge diffusion mechanisms and they can be even combined with them. We thus do not consider them further in this paper.

Empirical Phenomena in Deep Learning. Our work is closely related to research on empirical phenomena in deep learning [33]. In particular, in Section 1.4, we discuss how some of our findings may be partly explained by phenomena related to the disagreement of randomly initialized models [23, 28] and the simplicity bias [24, 1]. We are also inspired by the influential work of [44] on generalization, and the subsequent work on learning from noisy labels [36]. Notably, the idea of using a model's pseudolabels to train a learner has appeared in Co-Teaching [19], a training method involving two models. In Co-Teaching, each model takes into account training losses and based on them 'cherry-picks' the training points/pseudolabels that are fed to the other model. In contrast, in our framework, the models do not have access to training/validation losses, and they are not even designed to be 'aware' of when they interact with the Oracle, i.e. when they see true labels. Furthermore, teachers feed their pseudolabels indiscriminately to the learners. Thus, robustness to noise emerges as a byproduct of a natural process, unlike Co-Teaching and other training algorithms [36] that are explicitly designed to deal with noisy labels.

[2] For example, similar levels of test accuracy can be reached by using 9x parameters but accessing the training set 9x less frequently.

1.4 NKDIFF: *Intuition and justification*

NKDIFF mechanisms employ multiple partially trained teachers and one Oracle agent that holds the training set. These teachers provide pseudolabels to their students without checking the quality of the information they emit, hence diffusing false information. Moreover, each learner interacts with the Oracle for only a fraction of the time. Despite that, we find that partially trained teachers are useful as they enable a more efficient utilization of the constrained training resources. We view this phenomenon as a manifestation of overparameterization[3] used in tandem with first-order training methods. In particular, we argue that 'excess' parameters may be what causes the natural emergence of population-level learning, which in turn naturally prevents overfitting to noisy labels.

Overparameterization for diverse learners. Large models can be pruned to sparse 'lottery' models that can reach or exceed the test performance of the original dense models [14]. Finding these sparse models requires elaborate initialization or pruning techniques of the fully dense models [38]. In other words, sparsity makes initialization a very delicate issue.

On the flip side, a remarkable empirically observed property of (dense) models trained with gradient descent is that two randomly initialized models tend to have very similar test accuracy, but they also have a high rate of disagreement which is almost equal to their test error [23, 29].

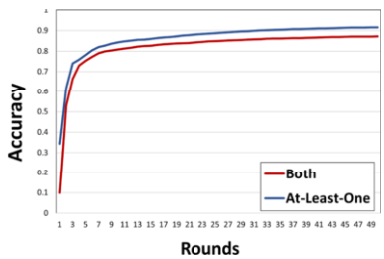

Figure 2: Training two randomly initialized LeNet models and tracking the number of test points that are correctly classified by *both* models vs the number of test points that are correctly classified by *at least one* model. The difference is more pronounced on bigger ResNet models.

Figure 2 depicts an instance of this phenomenon, over the entire training period. The disagreement is captured by the gap between the two curves. We observe that the relative disagreement of the two models tends to be higher in the beginning of training when, according to previous works, the networks learn simpler functions [24]. Intuitively this hints at a *spatial lottery*: the two models learn different parts of the data 'manifold' at different rates, due to randomness in initialization. Then, when one model is used as a teacher to the other, the student model can still learn information that it has not previously learned. In other words, the excess parameters not only make random initialization much easier, but render it a natural resource.

Learner diversity prevents 'unlearning'. In the presence of noisy labels, single models first undergo a phase of learning and reach high test accuracy. That is followed by an 'unlearning' phase of overfitting to the noisy labels which negatively impacts test performance [1]. In the case of NKDIFF all learners undergo the first phase and reach high test accuracy. However, the unlearning phase does not take place because the learners interact with the (noisy) Oracle only a fraction of the time, while seeing more consistent labels from their peers.

[3] By 'overparameterization' we mean –somewhat loosely– a number of parameters that suffices for reaching high training accuracy on noisy labels.

2 NKDIFF: Formulation and Mechanisms

This section details and justifies the various NKDIFF mechanisms we consider in this paper.

2.1 *Definitions and Problem Formulation*

Learners, Trainers and the Oracle Model. We are given a *population* of N-1 classifier models, and we want to train them with a training set X with categorical labels y. The models are artificial neural networks (ANNs) that have identical architectures. They are trained in *rounds*.

During a round of training, each model acts as a learner or a teacher. When acting as a *teacher* a model provides its (partially correct) predictions $h_i(X)$ for the training set X and it does not update its parameters. When acting as a *learner*, a model undergoes training with the training set provided by its teacher, and accordingly updates its parameters through standard iterations of forward and back-propagation operations. This is illustrated in Figure 3.

The *Oracle model* is the N^{th} model, h_N; it always teaches the correct labels $y = h_N(X)$ when queried, and consequently ignores learning the predictions of other models.

Figure 3: (Left:) Model h_i acts as teacher to model h_j by providing its predictions on the training dataset X as labels for a training sessions with h_j. (Right:) A schematic abbreviation.

Prior Knowledge and Learner Initialization Schemes. In an educational system, at any point of time, there are students that have different degrees of prior exposure to knowledge. Teachers also have varying levels of expertise. For that reason we allow the models to start the peer learning process after undergoing partial pre-training with the true labels. The type and amount of pre-training can be viewed as a hyperparameter of the peer learning mechanism. We call that the *learner initialization scheme*. Of course, the case of no prior knowledge is also of interest. In such case the *N-1* models are randomly initialized.

Coordinator and Groups. Training the *N-1* classifiers takes place in rounds and it may include a *Coordinator*. The Coordinator implements *grouping policies*. More specifically, before each round the Coordinator can: (i) Evaluate the performance of the *N-1* models (ii) Define groups of models and designate a single teacher for each group, for the upcoming round. Any algorithmic process that determines these groups is considered a grouping policy.

Learning Sessions. During a round, in each group, the teacher has an independent *session* with each of its assigned learners. In this paper, the session between a learner and teacher-model h_i consists of an epoch over the entire training data X with labels $h_i(X)$.

Training Capacity Constraint. We impose a capacity bound C on the size of the groups in each round of training. That implies that any teacher can teach up to C-1 learners per round. We will also be denoting by k the number of groups per round. An illustration of training capacity constraints can be seen in Figure 4.

Performance Metrics. In our setting we are typically interested in general 'educational welfare' objectives [41]. For that reason, we use metrics of aggregate generalization performance, e.g. the average test

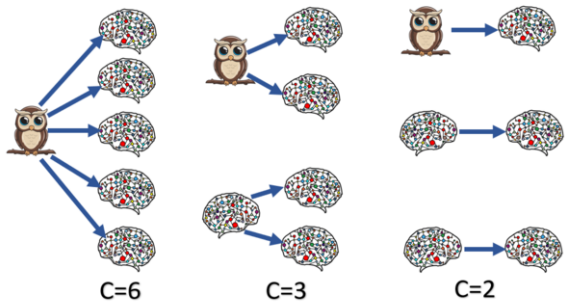

Figure 4: Without a teaching capacity constraint (C=6), the Oracle can teach all learners in each round, defaulting to plain ensemble training. In our framework, any model can act as a teacher, and our convention is to consider teaching groups of equal size; here (C=3) and (C=2) are depicted.

accuracy of the learners. However, a trained population can naturally be viewed as an ensemble, and thus ensemble test accuracy remains of interest in our context as well. To make things more concrete we introduce some definitions.

Definition 1 (Average Learner Accuracy)**.** Let $\mathcal{E}$ be an ensemble of m classifiers, $h_1, \ldots, h_m$. We define the average learner accuracy of $\mathcal{E}$ on a dataset (X, y) by

$$alacc_{\mathcal{E}}(X, y) = \left(\sum_{i=1}^{m} acc_{h_i}(X, y) \right) /m$$

Definition 2 (Ensemble Output)**.** Let $\mathcal{E}$ be an ensemble of identical-architecture classifiers, and further assume that the output of each classifier $h_i \in \mathcal{E}$ on an input point x is a probability distribution $p_i \in \mathbf{R}^K$, where $p_i[j]$ denotes the probability assigned to class j. The classification of x by $\mathcal{E}$ is given by:

$$\mathcal{E}(x) = argmax_j \sum_i \log p_i[j].$$

Definition 3 (Ensemble Accuracy)**.** The accuracy of a classifier $\mathcal{C}$ over a dataset (X, y), denoted by $acc_{\mathcal{C}}(X, y)$, is the ratio of points in X that are classified correctly by $\mathcal{C}$. Viewing $\mathcal{E}$ as a classifier, we refer to $acc_{\mathcal{E}}(X, y)$ as the ensemble accuracy.

2.2 *Peer learning group policies*

In this section we review the policies included in our empirical study and provide some additional background and justification for them.

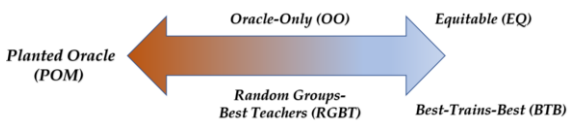

Figure 5: **Grouping policies in the spectrum of coordination**. POM is a decentralized and uncoordinated mechanism where the Oracle is concealed as a participant. OO and RGBT are moderately coordinated. In particular OO is a baseline that does not make use of partially trained teachers. EQ and BTB are fully coordinated policies aiming at different types of 'social objectives'.

Previous works on analytical models for peer human learning consider a number of grouping policies that reflect a long-standing debate on class formation in the realm of the social sciences and education policy making (e.g. see [12, 31, 6, 30]).Underlying the debate is the common-ground intuition that public policies can have an impact on the educational welfare of the population. In our study we pursue an 'artificialization' of this question and consider a spectrum of policies with various degrees of coordination and underlying social intent, as illustrated in Figure 5. More specifically, we consider these policies:

OO: *Oracle-Only*. In each round, *C-1* models are selected at random and they are trained by the Oracle. We view this policy as a special case of our framework, and we include it as a baseline.

POM: *Planted Oracle Mechanism.* The POM is our only fully decentralized policy, testing whether knowledge can be diffused without any evaluation of the learners or any other type of external coordination. In each round of the training process, the N models are randomly split into $N/2$ pairs. Suppose that models h_i and h_j form a pair. Then two sessions take place, one where h_i is trained with $(X, h_j(X))$ and one where h_j is trained with $(X, h_i(X))$. In this case, the only possible training capacity is *C=2*. Observe that this mechanism does not evaluate the models at any point of the process and that the Oracle model conceals itself as a participant in the training process.

The remaining grouping policies are based on measuring the validation accuracy of the learners $h_1, \ldots, h_{N-1}$ before each round. These accuracies are communicated to the coordinator who then decides the grouping for the next round. While communication complexity is not our main concern, we note that if the learners hold copies of the validation set, then the complexity of this process is small; the coordinator needs to only receive a single number (validation accuracy) from each learner, and send them back the identity of their teacher for the next round. In what follows, we let v_i be the validation accuracy of h_i, and m_i denote the model whose validation accuracy is the i^{th} lowest in the list $\{v_1, \ldots, v_N\}$.

RGBT: *Random-Groups Best-Teachers.* The models are randomly split into *k= N/C* groups. Then the model with the highest validation accuracy in its group is designated as the teacher. In this approach, there is no coordination in the selection of groups, but only in the selection of teachers.

BTB: *Best-Trains-Best.* The policy uses the current-round best learners as teachers. It furthermore greedily assigns better students to better teachers. The trainers of the k groups are models $m_N, \ldots, m_{N-k+1}$, i.e. the models with the highest validation accuracy. The rest of the ordered list $m_{N-k}, \ldots, m_1$ is split into $k-1$ contiguous buckets that are assigned in order to $m_N, \ldots, m_{N-k+1}$. This corresponds to situations where better students are grouped together in classes, and they learn from better teachers.

EQ: *Equitable.* The policy uses the current-round best learners as teachers. It furthermore greedily assigns students in order to create 'balanced' groups, in terms of the students' ability. The policy attempts to be fair (some weak students will be assigned to good teachers), while still giving a slight edge to better students (e.g., the weakest student will be matched with the weakest teacher). More concretely, the trainers of the k groups are models $m_N, \ldots, m_{N-k+1}$. The rest of the models in the ordered list $m_{N-k}, \ldots, m_1$ are assigned in a round-robin fashion to $m_N, \ldots, m_{N-k+1}$. In this grouping each teacher is assigned to learners at all levels of accuracy.

3 The empirical study: Questions and Findings

In this section we present the main findings of our extensive empirical study. After reviewing our experimental setting in section 3.1, we discuss the effectiveness of NKDIFF in section 3.2, the performance difference among policies in section 3.3, and the generalization benefits of learner diversity in section 3.4.

3.1 Experimental setting

Architectures and Datasets. We perform experiments with three different types of architectures and two different types of datasets. In particular, we use 'toy' versions of LeNet (5 layers with *CrossEntropyLoss* function and *Stochastic Gradient Descent* optimizer with learning rate of 0.9, where we have a total number of 61.7K parameters) and ResNet (18 layers with *CrossEntropyLoss* function and *Adam* optimizer with learning rate of 3e-4, where we have a total number of 11.1M parameters) [27, 20]. These networks are used on Fashion-MNIST dataset which consists of 50,000 training, 10,000 validation, and 10,000 test images of fashion and clothing items, taken from 10 classes, where each image is a standardized 28×28 size in grayscale. We also use a Graph Convolutional Network (GCN) (consisting of 3 layers, 32 hidden channels and dropout of 0.5, with *CrossEntropyLoss* function, *Adam* optimizer with learning rate of 0.01, where we have a total number of 8.9K parameters) [25, 46]. This is employed for a transductive classification problem. We use the ogbn-arxiv dataset which is a un-directed graph representing the citation network between all Computer Science (CS) arXiv papers, where each node is an paper with a 128-dimensional feature vector which consists of 90K training, 48K validation and 29K testing points.

Peer Framework Settings. In our study, we look at populations of size *N=10*. We explored two settings for C, groups of size two (*C=2*) and groups of size five (*C=5*). Thus, we have $k = 5$ and $k = 2$ groups respectively. We refer to these settings as split-in-five and split-in-two. We include experiments without pre-training and with pre-training. In our learner initialization scheme, model i has been trained for i rounds with the true labels.

Number of Random Experiments. In each experiment we first randomly initialize each model. Then on the *same* initialized population, we try each combination (*Network*, *Pretraining/No-pretraining*, *C*, *Policy*). In the experiments of Sections 3.2 and 3.3 we report *averages* of our metrics, taken over 100 random experiments. In the experiments of Section 3.4 we report averages over the following numbers of random experiments: 10 for LeNet, 20 for ResNet, and 10 for GCN. The number of random experiments has been picked in order to derive 95% confidence intervals, shown in cases when the difference among policies was not extremely clear cut. Confidence levels are calculated in a standard way. [4]

Justification. For the image classification experiments we fix a single dataset, which is difficult for the LeNet model, but relatively easy for the much larger and fundamentally more expressive ResNet model. With this choice we want to test how NKDIFF works in different 'hardness' and parameterization regimes, especially under the light of the discussion in Section 1.4 that identifies overparameterization and model disagreement as a cause underlying the effectiveness of NKDIFF. The GCN experiment is meant to test NKDIFF in a transductive setting, with a fundamentally different type of problem and architecture, and relatively fewer trainable parameters.

Code. The code can be accessed here:
https://github.com/peer-ai-njit/ecai23

[4] We believe that actual confidence levels are much tighter than reported, but we stray away from such statistical arguments.

3.2 The effectiveness of NKDIFF

• *The price of teacher scarcity at convergence.* We take a look at the highest levels of performance the learners were able to reach without any time constraint. [5] The results are shown in Table 1; here we use the BTB policy which performed best among NKDIFF policies.

		C = 10	C = 5	C = 2
ResNet	*Avg*	90.4 ± 0.002	90.4 ± 0.001	90.4 ± 0.002
	Ens	91.6 ± 0.004	91.0 ± 0.004	90.1 ± 0.007
LeNet	*Avg*	89.5 ± 0.002	89.3 ± 0.003	86.9 ± 0.001
	Ens	89.8 ± 0.003	89.3 ± 0.003	86.6 ± 0.002
GCN	*Avg*	69.1 ± 0.003	67.4 ± 0.002	64.7 ± 0.005
	Ens	69.7 ± 0.004	68.1 ± 0.004	67.4 ± 0.004

Table 1: Impact of training capacity on the maximum performance attained by the population. *Avg* denotes average test accuracy and *Ens* ensemble test accuracy. The case *C=10* is when all models are trained in parallel by the Oracle model, as in Figure 4(a). Standard deviation is over multiple random experiments (see Section 3.1).

When partially trained teachers are used i.e. when *C=2* and *C=5*, the Oracle teacher is respectively accessed only 55.5% and 11.1% of the times accessed relative to *C=10*, and its true labels are replaced by false/inconsistent labels. We thus expect to see an impact in test performance. We find though that this impact is relatively small. A potentially interesting fact is that the smallest impact is observed for ResNet, the most 'overparameterized' of these architectures.

• *The usefulness of partially trained teachers.* Recall that the primary reason behind our NKDIFF study is the efficient utilization of the teaching resources. We thus want to consider the test performance of the population as a function of the number of accesses of the true labels. The results are shown in Figure 6.[6].

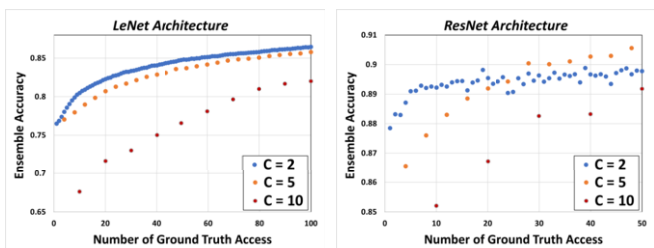

Figure 6: Ensemble accuracy as a function of the number of oracle sessions, for *C=2*, *C=5* and *C=10*. GCN behavior is qualitatively similar to LeNet. Also all other policies have similar behavior to *BTB*.

We observe that a *smaller* training capacity C leads to *higher* performance, for any given small budget of Oracle sessions. This implies that partially trained teachers are indeed helping in extracting knowledge from the Oracle more efficiently, and by a large margin in pre-convergent epochs! We also find that *smaller* classes (*C=2*) are better. Interestingly, smaller classes employ more and thus weaker teachers. Overall the 'natural' necessity of using more teachers, in tandem with overparameterization at the population level, lead to faster mining of the ground truth.

[5] In this experiment ResNet was trained for 50 rounds, and LeNet and GCN were trained for 100 rounds. There was no significant change in their validation accuracy in the last 20% of their epochs, so we consider them converged.

[6] Results are similar for GCNs. Average learner accuracy as a function of the number of ground truth accesses is also similar to that of Figure 6. The plots can be found in the Supplementary Material.

• *The economics of class size.* In an educational setting, employing weaker teachers is, of course, not free. We thus study an alternative cost model where we measure performance as a function of the number of training sessions (or equivalently, forward operations). The result is shown in Figure 7.

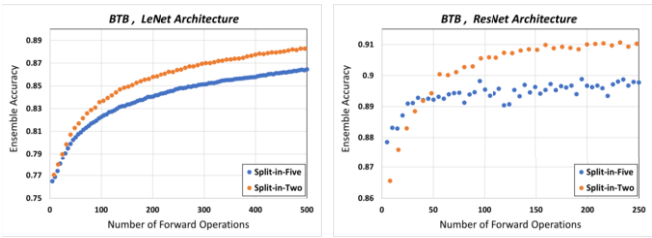

Figure 7: Ensemble test accuracy $acc_{\mathcal{E}}$ for LeNet and ResNet as a function of number of *Forward Operations*.

Here we see larger classes (C=5) have a higher knowledge extraction rate relative to the smaller classes (C=2), which is not unexpected given the fact that larger classes employ better teachers.

3.3 Policy Effect

• *The Planted Oracle Mechanism.* In the *POM*, all models act both as teachers and learners in every round. Models are not evaluated at any point of the training process, and the Oracle model conceals itself as a participant in the process. The Coordinator has the very limited role of simply timing the rounds. Thus POM corresponds to an extreme case of a completely unorganized population, with random interactions among its members who just exchange information.

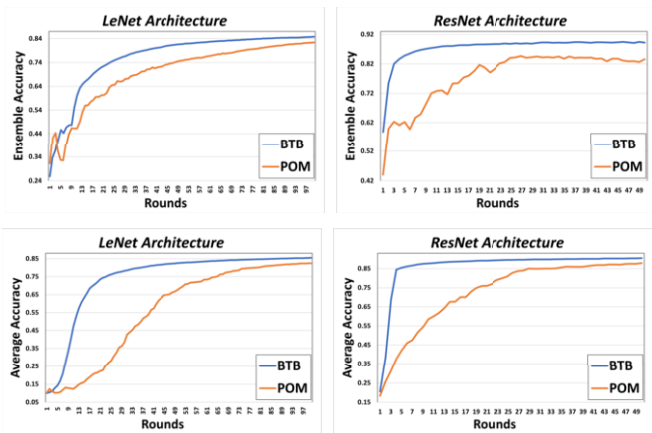

Figure 8: Comparing *POM* with *BTB* on ensemble accuracy (upper part), and average learner accuracy (lower part).

In Figure 8 we compare the *POM* with *BTB* which is the best coordinated policy. It can be seen that *POM* has a much slower rate of learning relative to the coordinated policy, even in the case of ResNet, where learning is very fast for *BTB*. Nevertheless, *POM* eventually converges to much higher performance that nearly approaches that of the coordinated policy. Notably, even the average learner accuracy is high. This implies that knowledge diffuses from the Oracle to all individuals. The fact that true knowledge eventually dominates the false and inconsistent information circulating in the population is of independent interest and possibly worthy of further exploration in the context of machine learning.

• *Coordinated, evaluation-based policies*. We first note that the choice of policy has a negligible on the highest performance levels reached *at convergence*. But it is worth noting that even by a small margin, the highest performance was achieved by the *BTB* policy in 4 out of the 6 cases, corroborating the insights in [41].

To our main point in this part, recall that NKDIFF is motivated as a framework for training under teaching *and* time constraints, favoring faster diffusion, i.e. better test performance within limited time frames. For that reason we now focus on the first 10 epochs of training and compare the performance of the different policies. Here we use models that have been partially pre-trained at different degrees, as discussed in Section 2.1. The results are shown in Figure 9.

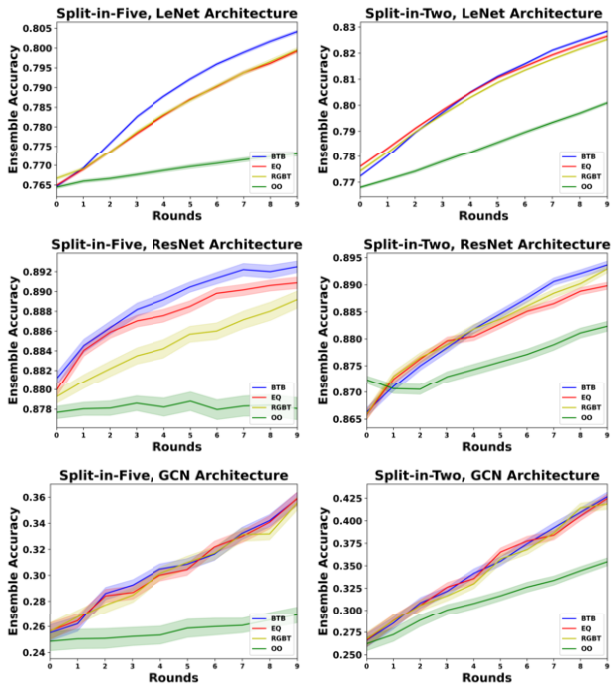

Figure 9: Ensemble test accuracy for 10 rounds/epochs with pre-training, for C=2 (left) and C=5 (right). Note that the reported metrics are averages over multiple random experiments, as discussed in Section 3.1. *BTB* is clearly better on LeNet/ResNet, especially when C=2.

The following points summarize our observations from Figure 9 and other results deferred to the Supplementary Material due to space.[7]

(a) In all cases the average learner accuracy of the *Oracle-Only* policy is significantly smaller than all other policies. This is also the case for ensemble accuracy, with the exception of LeNet in the no-pretraining case. Recall that *OO* makes no use of peer sessions. This indicates that genuine learning takes place in peer sessions. Thus, if general 'educational welfare' metrics are of interest, then peer learning mechanisms appear to be a necessity.

(b) In the case when pre-training is used, there are clear differences between policies. In LeNet and GCN, the fully coordinated policies *BTB, EQ* are clearly better than the moderately coordinated *RGBT*. The picture is more mixed for ResNet, where in average accuracy, *RGBT* does better. This may be due to different properties of the ResNet architecture, or simply to the fact that ResNet learns much more quickly, as noted earlier.

[7] The Supplementary Material includes a comprehensive set of experiments. In particular, on the topic of policy effect, the differences between policies are less significant without pre-training, although full coordination still appears to be better than moderate coordination.

3.4 Memorization and Robustness to Noisy Labels

• *Memorization.* The surprising observation that NKDIFF works even in the very restricted setting of the Planted Oracle Model leads us to a contrarian question inspired by the work in [45]: Can NKDIFF train the population to memorize *random* labels?

For background, Zhang et al. [45] observe that the training accuracy of large deep networks reaches 100% on training sets with random labels, i.e. the models have the capacity to fully memorize the data. In our case we use smaller models that still have a significant capacity for memorization after sufficient training. For example, as illustrated in Figure 10-(a), training the LeNet model on a set (X, y_r) where y_r are random labels of the points in X reaches an accuracy of roughly 40%, much higher than the random expectation of 10%.

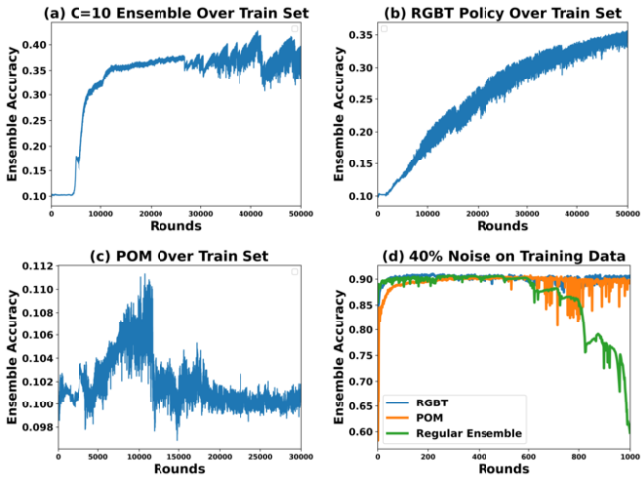

Figure 10: (a,b,c) Training accuracy of LeNet on randomly labeled dataset with 10 classes, for three different NKDIFF policies. (d) Test accuracy of ResNet on 40% noisy training data.

We next try a moderately coordinated, evaluation-based policy, *RGBT*, for *C=2*. In Figure 10-(b) we observe that memorization does take place, albeit at a much slower rate, reaching a lower value of 35%. However, when we use the *uncoordinated POM*, we see in Figure 10-(c) that training accuracy stays between 10% and 11% throughout training. In combination with the results in Section 3.3, this suggests that *POM* is able to learn a very effective classifier, which however has no capacity of memorizing training points. In particular, it appears that the somewhat reduced convergence rate and test performance of *POM* comes with the benefit of memorization resistance.[8]

• *Robustness to noisy labels.* The next logical step is to consider the more general case when *a fraction* of the labels are corrupted randomly, where overfitting can become a serious issue. Figure 10-(d) depicts a single experiment with ResNet, over 1000 epochs. In this case, 40% noise was added to training data. In our NKDIFF framework, *POM* and *RGBT* are *robust* against the noisy training data and very little overfitting happens. In contrast, when all models learn from the (noisy) Oracle, test accuracy dropped almost 30% due to overfitting which started after 600 epochs. It is also interesting while overfitting episodes happen after 600 epoch for *POM* and *RGBT*, NKDIFF exhibits self-correcting behavior, presumably due to the 'healty' learners that overpower the noisy Oracle.

[8] Up to our knowledge, none of the existing methods for learning from noisy labels [36] have been tested at the extreme of fully random labels. Thus, NKDIFF with *POM* may be the only known training algorithm that is resistant to memorization. However, this requires further investigation.

4 Conclusion

We have initiated a study of *natural* Knowledge Diffusion processes for training populations of artificial learners. The work is motivated by the problem of peer group formation in human educational systems, a sensitive and widely debated issue in social psychology and policy making. While the analytical modeling or 'artificialization' of similar questions may be an interesting line of work that can help researchers develop real-world insights [41], we wish to emphasize that it is not our intention to take a stance, make policy recommendations, or claim any impact outside the field of artificial intelligence.

In the context of machine learning, we did not aim to solve an existing problem, but to explore generalization phenomena under natural resource constraints. As discussed in section 1.3, many recent works aim to design sophisticated algorithms for addressing pitfalls of overparameterization, such as model complexity and overfitting to noisy data. In our work, dealing with natural resource constraints led us to explore the possibility that overparameterization, in synergy with stochasticity, is a *natural resource* that enables populations of learners to mine faster, more efficiently, and without overfitting, the ground truth knowledge. Interestingly, that emerges with a higher-order overparameterization at the population level; *more* trainable parameters are now distributed to multiple individuals who communicate via simple diffusion processes.

While we envisioned NKDIFF as a population-level process, our individuals are simple identical-architecture neural models that can be alternatively conceived as a single ensemble model with modular characteristics [32]. The notion and role of modularity have been extensively investigated in biology, in particular in connection with the architecture of the mind [13, 9, 10]. We find it interesting that the study of generalization under *natural* constraints led us to the design of a modular system that as a whole does not have the capacity to memorize despite the ability of its constituent parts to do so, possibly suggesting a mechanism that Natural Intelligence uses to generalize without memorizing.

Multiple questions arise, including the following:

(a). We used training rounds that coincide with standard epochs over existing training benchmark datasets. A more detailed picture may emerge for rounds of finer granularity. Related to that is the effect of population size. In our experiments, we have found that even when learners interact with the true labels only 11% of their learning time, the population (and -on average- the individual learners) reach high test accuracy. Increasing the size N of the population would enable situations with a lower rate of access to the Oracle. It is then interesting to study whether and under what conditions, interaction with inconsistent labels becomes potentially catastrophic for the performance of peer-trained populations.

(b). Robustness to noisy labels emerged as a byproduct of our study. This phenomenon is worth of further study. An open question is whether knowledge diffusion mechanisms can yield any advantages over existing algorithms for dealing with noisy labels. It would also be interesting to explore whether populations trained with NKDIFF are robust to other types of noise and their adaptability to distribution shifts.

More generally, our study revealed various intriguing phenomena. We believe that this framework where *'Machine Learning meets Knowledge Diffusion'* opens up multiple directions for future research.

Acknowledgements. Preliminary work was presented at the ICLR 2022 Workshop on Agent Learning in Open-Endedness [2]. This work was supported by and a Faculty Seed Grant from NJIT and NSF Grant #997421.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien, 'A closer look at memorization in deep networks', in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, eds., Doina Precup and Yee Whye Teh, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, (2017).

[2] Ehsan Beikihassan, Ali Parviz, Amy K. Hoover, and Ioannis Koutis, 'Ensemble learning as a peer process', in *ICLR 2022 Workshop ALOE*, (2022).

[3] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi, 'Personalized and private peer-to-peer machine learning', in *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, eds., Amos J. Storkey and Fernando Pérez-Cruz, volume 84 of *Proceedings of Machine Learning Research*, pp. 473–481. PMLR, (2018).

[4] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler, 'The power of ensembles for active learning in image classification', in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 9368–9377, Salt Lake City, UT, United States, (June 2018). Computer Vision Foundation / IEEE Computer Society.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, 'Curriculum learning', in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pp. 1–8, Quebec, Canada, (2009). ACM Press.

[6] Jo Boaler, Dylan Wiliam, and Margaret Brown, 'Students experiences of ability grouping—disaffection, polarisation and the construction of failure1', *British Educational Research Journal*, **26**(5), 631–648, (2000).

[7] Hamed R. Bonab and Fazli Can, 'A theoretical framework on the ideal number of classifiers for online ensembles in data streams', in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*, pp. 2053–2056, Indiana, United States, (October 2016). ACM.

[8] Leo Breiman, 'Arcing the edge'. Technical Report 486, Statistics Department, University of California at Berkeley, (June 1997).

[9] David J. Buller, 'Get over: Massive modularity', *Biology and Philosophy*, **20**(4), 881–891, (2005).

[10] Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson, 'The evolutionary origins of modularity', *CoRR*, **abs/1207.2743**, (2012).

[11] Robin Cowan and Nicolas Jonard, 'Network structure and the diffusion of knowledge', *Journal of Economic Dynamics and Control*, **28**(8), 1557–1575, (2004).

[12] Dominick Esposito, 'Homogeneous and heterogeneous ability grouping: Principal findings and implications for evaluating and designing more effective educational environments', *Review of Educational Research*, **43**(2), 162–179, (June 1973).

[13] Jerry A. Fodor, *The Modularity of Mind: An Essay on Faculty Psychology*, MIT Press, 1983.

[14] Jonathan Frankle and Michael Carbin, 'The lottery ticket hypothesis: Finding sparse, trainable neural networks', in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, (2019).

[15] Yoav Freund and Robert E. Schapire, 'A decision-theoretic generalization of on-line learning and an application to boosting', in *Computational Learning Theory, Second European Conference, EuroCOLT*, volume 904 of *Lecture Notes in Computer Science*, pp. 23–37, Barcelona, Spain, (March 1995). Springer.

[16] Yoav Freund and Robert E. Schapire, 'A short introduction to boosting', *Journal of Japanese Society for Artificial Intelligence*, **14**(5), 771–780, (September 1999).

[17] Jerome H. Friedman, 'Greedy function approximation: A gradient boosting machine', *The Annals of Statistics*, **29**(5), 1189–1232, (October 2001).

[18] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran, 'An efficient framework for clustered federated learning', in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, eds., Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, (2020).

[19] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama, 'Co-teaching: Robust training of deep neural networks with extremely noisy labels', in *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., (2018).

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, Nevada, United States, (June 2016). IEEE.

[21] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean, 'Distilling the knowledge in a neural network', in *NIPS Deep Learning and Representation Learning Workshop*, (2015).

[22] Mahdi Jalili and Matjaž Perc, 'Information cascades in complex networks', *Journal of Complex Networks*, **5**(5), 665–693, (07 2017).

[23] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J. Zico Kolter, 'Assessing generalization of SGD via disagreement', in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, (2022).

[24] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin L. Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang, 'SGD on neural networks learns functions of increasing complexity', in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, eds., Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, pp. 3491–3501, (2019).

[25] Thomas N. Kipf and Max Welling, 'Semi-supervised classification with graph convolutional networks', in *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, (April 2017).

[26] Jakub Konečný, Brendan McMahan, and Daniel Ramage, 'Federated optimization: Distributed optimization beyond the datacenter', *CoRR*, **abs/1511.03575**, (2015).

[27] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, **86**(11), 2278–2324, (November 1998).

[28] Preetum Nakkiran and Yamini Bansal, 'Distributional generalization: A new kind of generalization', *CoRR*, **abs/2009.08092**, (2020).

[29] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever, 'Deep double descent: Where bigger models and more data hurt', in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, (2020).

[30] Asma Ounnas, Hugh C. Davis, and David E. Millard, 'A framework for semantic group formation in education', *J. Educ. Technol. Soc.*, **12**(4), 43–55, (2009).

[31] Stephen Richer, 'Reference-group theory and ability grouping: A convergence of sociological theory and educational research', *Sociology of Education*, **49**(1), 65–71, (January 1976).

[32] Philip Robbins, 'Modularity of mind', in *Stanford Encyclopedia of Philosophy*, ed., Edward N. Zalta, (2009).

[33] Hanie Sedghi, Samy Bengio, Kenji Hata, Aleksander Madry, Ari Morcos, Behnam Neyshabur, Maithra Raghu, Ali Rahimi, Ludwig Schmidt, and Ying Xiao. Identifying and understanding deep learning phenomena. ICML Workshop, 2019.

[34] Burr Settles, *Active Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012.

[35] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takác, Michael I. Jordan, and Martin Jaggi, 'Cocoa: A general framework for communication-efficient distributed optimization', *J. Mach. Learn. Res.*, **18**, 230:1–230:49, (2017).

[36] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee, 'Learning from noisy labels with deep neural networks: A survey', *IEEE Transactions on Neural Networks and Learning Systems*, 1–19, (2022).

[37] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe, 'Curriculum learning: A survey', *CoRR*, **arXiv:2101.10382**, (January 2021).

[38] Kartik Sreenivasan, Jy-yong Sohn, Liu Yang, Matthew Grinde, Alliot Nagle, Hongyi Wang, Kangwook Lee, and Dimitris S. Papailiopoulos, 'Rare gems: Finding lottery tickets at initialization', *CoRR*, **abs/2202.12002**, (2022).

[39] Yi Sui, Junfeng Wen, Yenson Lau, Brendan Leigh Ross, and Jesse C. Cresswell, 'Find your friends: Personalized federated learning with the right collaborators', *CoRR*, **abs/2210.06597**, (2022).

[40] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi, 'Decentralized collaborative learning of personalized models over networks', in

Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, eds., Aarti Singh and Xiaojin (Jerry) Zhu, volume 54 of *Proceedings of Machine Learning Research*, pp. 509–517. PMLR, (2017).

[41] Dong Wei, Ioannis Koutis, and Senjuti Basu-Roy, 'Peer learning through targeted dynamic groups formation', in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 121–132, Chania, Greece, (April 2021). IEEE.

[42] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur, 'When do curricula work?', in *9th International Conference on Learning Representations, ICLR 2021*, Austria, (May 2021). OpenReview.net.

[43] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong, 'Federated machine learning: Concept and applications', *ACM Trans. Intell. Syst. Technol.*, **10**(2), 12:1–12:19, (2019).

[44] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, 'Understanding deep learning requires rethinking generalization', in *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France, (April 2017). OpenReview.net.

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, 'Understanding deep learning requires rethinking generalization', in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, (2017).

[46] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun, 'Graph neural networks: A review of methods and applications', *AI Open*, **1**, 57–81, (2020).