Mixture-of-Experts-Ensemble Meta-Learning for Physics-Informed Neural Networks

Rafael Bischof[1] and Michael A. Kraus[2]
[1]Swiss Data Science Center, Turnerstrasse 1, 8092 Zürich, Switzerland
[2]Design++ and Chair of Concrete Structures and Bridge Design (IBK), ETH Zürich, Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland
E-mail(s): rafael.bischof@sdsc.ethz.ch, kraus@ibk.baug.ethz.ch

Abstract: Partial Differential Equations (PDEs) arise in natural and engineering sciences to model reality and allow for numerical assessment of these phenomena. Classical numerical solutions for PDEs rely on discretisations such as e.g. the finite or spectral element method (FEM/SEM). Physics Informed Neural Networks (PINN) leverage physical laws by including PDE together with a respective set of boundary and initial conditions (BC / IC) as penalty terms into their loss function during training without the need for discretisation. However, as the computational domain or the nonlinearity of the PDE becomes more complex, finding PDE solutions via FEM/SEM as well as PINNs becomes hard. Mixture of Experts (MoE) is a probabilistic model, consisting of local experts weighted by a gating network. As in ensemble methods, MoE involves decomposing predictive modeling tasks into sub-tasks, i.e. training an expert model on partitions of the input space, and devising a gating model that learns to weight the experts' predictions conditioned on the input. In this work, we combine the two methods to form Physics Informed Mixture of Neural Network Experts (PIMNNE) / Mixture of Experts Physics Informed Neural Networks (MoE-PINNs) and investigate the learning and predictive capabilities by the example of a 2-dimensional Poisson PDE over a L-shaped domain with homogeneous Dirichlet BC. Our simulation studies show, that all MoE-PINNs setups approximate the SEM reference solution to a satisfactory high degree. A hyperparameter study revealed that an increase in the number of PINN experts reduces the approximation error significantly even if regularised aggressively for sparsity. A second interesting finding is that a resolute regularisation - effectively driving weak learners importance towards zero - is to be preferred over a uniform penalty. We also found the most popularly used *tanh* activation function for PINNs being consistently discarded by the gating network, while PINNs with *sine* activations provided an additional boost of performance. Our proposed Moe-PINNs in the future serve as differentiable computational performance predictors for computational mechanics or fluid dynamics applications in design and verification of structures.

Keywords: Physics-Informed Machine Learning, Multi-Objective Optimisation (MOO), Ensemble Methods, Computational Mechanics

1 Introduction and Related Work

During the design and verification of engineering structures, PDEs are employed to describe and model the mechanical or fluid behavior in order to allow for a computational assessment, design and verification. Solving partial differential equations (PDEs), which arise from various first-order principles in natural and engineering sciences, by using neural networks was first introduced by Lagaris et al. [1] and revived in 2017 by Raissi et al. [2], who coined the term Physics-Informed Neural Networks (PINNs). By leveraging physical laws and the power of automatic differentiation, readily implemented in all major deep learning libraries, they require just a few lines of code to be implemented and trained. Further boosted by industry scale software packages like "Modulus" by NVIDIA [3], PINNs have so far been successfully applied to various linear and nonlinear PDEs from e.g. computational mechanics , fluid dynamics [2] and more. Often however, a number of pathologies have been observed at training time, impeding proper training and ultimately leading to unsatisfactory solutions. Addressing these pathologies in PINNs is an active area of research and a number of extensions have recently been proposed, such as building specialised architectures that favor backpropagation [4], or improving the network initialisation by resorting to meta learning [5].

Analogously to major established numerical solution methods such as the finite element method (FEM), which split (i.e. discretise) the domain prior to optimisation, cPINNS or XPINNs [6] apply different networks to slightly overlapping partitions of the domain in a divide and conquer fashion. While having several advantages over vanilla PINNs, such as larger representation capabilities and the possibility of distributed computing, selecting the optimal number of networks to distribute over the domain and defining the boundaries between partitions are sensitive hyperparameters that require considerable labour to tune. Instead, Stiller et al. proposed GatedPINNs [7], a framework leveraging multiple networks as an ensemble and an additional learner acting as the gate, which appoints the various models to partitions of the domain. Such frameworks have been successfully applied in different fields of ML and are known as Mixtures of Experts (MoE) [8]. It allows the individual learners to specialise on distinct subsets of the data, while also providing the possibility to average the results of several experts in challenging areas of the input space [9]. MoE have been of particular interest in multitask learning problems, as devising different portions of the model for different tasks while keeping some shared layers constitutes an effective way of reducing the bias-variance trade-off. PINNs are similar to multitask learning problems in that they have to optimise multiple different terms representing the governing equation as well as one or more IC and/or BC.

2 Contribution and Methods

We introduce a number of extensions to the original formulation of GatedPINNs and demonstrate their effectiveness on a new benchmark for ensembles of PINNs. We will hereafter be referring to GatedPINNs including our extensions as MoE-PINNs.

- We add a regularisation term to the loss function, which in a data-driven fashion incentives the model to utilise as few PINNs as possible.
- We show that this regularisation can act as an automated and differentiable architecture search by initialising a large bag of PINNs with varying depth, width and activation functions, and letting the gating network select the most suitable subset of models at training time.
- We apply MoE-PINNs to a new benchmark for ensembles of PINNs, the Poisson equation on a L-shaped domain (as investigated initially in [6], used as validation reference here).

2.1 Physics-Informed Neural Networks (PINNs)

Consider the following abstract parameterised and nonlinear PDE problem:

$$\begin{aligned} \text{PDE} &: \mathcal{F}\left(\hat{\mathbf{u}}, \partial_t\hat{\mathbf{u}}, \partial_x\hat{\mathbf{u}}, \cdots; \mu\right) = 0, \quad \mathbf{x} \in \Omega, t \in [0, T] \\ \text{B.C.} &: \mathcal{B}\left(\hat{\mathbf{u}}, \partial_x\hat{\mathbf{u}}, \partial_x^2\hat{\mathbf{u}}, \cdots\right) = 0, \quad \mathbf{x} \in \Gamma \\ \text{I.C.} &: \mathcal{C}\left(\hat{\mathbf{u}}, \partial_t\hat{\mathbf{u}}, \partial_t^2\hat{\mathbf{u}}, \cdots\right) = 0, \quad t \in \Upsilon \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the spatial coordinate and t is the time; $\mathcal{F}$ denotes the residual of the PDE, containing the differential operators (i.e. $\partial_{\mathbf{x}}\hat{\mathbf{u}}, \partial_t\hat{\mathbf{u}}, ...$); $\mu = [\mu_1, \mu_2, ...]$ are the PDE parameters; $\hat{\mathbf{u}}(\mathbf{x}, t)$ is the solution of the PDE with initial condition $\mathcal{C}$ and boundary condition $\mathcal{B}$ (which can be Dirichlet, Neumann or mixed); Ω, Γ and Υ represent the spatial domain resp. boundaries. In order to find the unknown function $\hat{\mathbf{u}}(\mathbf{x}, t)$, satisfying the residual $\mathcal{F}$ as well as the boundary- and initial conditions, $\mathcal{B}$ and $\mathcal{C}$, Raissi et al. [2] proposed to use a neural network $U(\mathbf{x}, t; \theta)$ as solution to the nonlinear PDE problem of eq. 1, and train it to optimally approximate the conditions $\mathcal{F} = 0$, $\mathcal{B} = 0$ and $\mathcal{C} = 0$. This is achieved by taking the necessary derivatives of U through automatic differentiation (AD), inserting these expressions into $\mathcal{F}$, $\mathcal{B}$ and $\mathcal{C}$, applying the mean squared error and training the network as a scalarised multi-objective optimisation, where the several terms are aggregated using weights [10].

2.2 Ensemble Learners and Mixture-of-Experts

Mixture of Experts (MoE) is an ensemble learning technique implementing a training of experts on subtasks (e.g. a particular data region) of a predictive modeling problem [11]. The MoE model works in a four-step fashion: (i) division of a task into subtasks, e.g. by decomposition of the input space, (ii) develop an expert for each subtask, (iii) use a gating model to decide which expert to use, and (iv) pool expert outputs and gating model output to make a prediction. The subtasks may or may not overlap, and experts from similar or related subproblems may be able to contribute to the examples that are technically outside of their expertise. The MoE regressor is defined as

$$g(x) = \sum_i^m P(i|\mathbf{x}, \theta_{\mathbf{p}}) f(\mathbf{x}, \theta_{\mathbf{i}}) \tag{2}$$

with a local expert regressor $f(\mathbf{x}, \theta_i)$ and associated model parameters θ_i of expert i and a gating function P conditioned on the input $\mathbf{x}$ as well as its parameters θ_p delivering the probability of association $P(i|x, \theta_p)$ between input and expert i (i.e. the degree to which the expert contributes to the total output).

3 Mixture of Experts of Physics Informed Neural Networks (MoE-PINNs)

Employing PINNs as expert models to assemble the PDE solution and defining $\mathbf{x} = (\mathbf{x}, t)$, with $\mathbf{x}$ and t being the spacial and / or time variables, it follows that $f(\mathbf{x}, \theta_i) = U(\mathbf{x}, t; \theta_i)$ (cf. Fig. 1). As the gating network represents the conditional probability $P(i|\mathbf{x}, \theta_i)$ of datum $\mathbf{x}$ given by expert m_i, a proper gate is formulated via the *softmax* function:

$$\lambda(\mathbf{x})_i = \frac{\exp(P(i|\mathbf{x}, \theta_i))}{\sum_{j=1}^{m} \exp(P(j|\mathbf{x}, \theta_j))} \tag{3}$$

where $\lambda(\mathbf{x})_i$ is called "importance" of expert i.

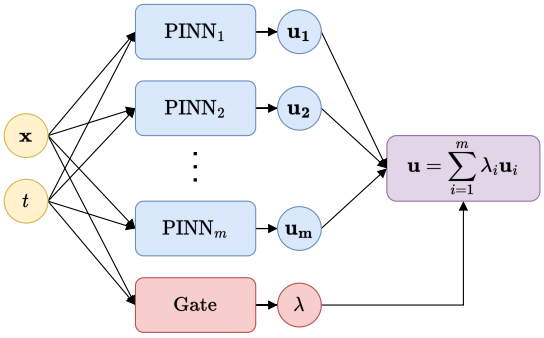

Figure 1: Mixture-of-Experts PINN.

As previously established, the number of experts in the ensemble m constitutes an additional, sensitive hyperparameter. In order to reduce the laborious process of hyperparameter-tuning, we introduce a sparsity inducing regularisation to act as data-driven selection of the amount of experts m.

$$\mathcal{L}_{sp} = \left| \frac{1}{|B|} \sum_{i}^{m} \sum_{\mathbf{x} \in B} \lambda_i(\mathbf{x}) \right|^p \tag{4}$$

where B is a batch of collocation points (x, t) and p is a hyperparameter. We are particularly interested in selecting $0 < p \leq 1$, as these values yield a regularisation term that penalises small values more than larger ones, and which will thus lead to more sparsity in the importances across experts.

4 Results: Poisson's Equation on L-shaped Domain

The Poisson equation is a second-order, elliptic partial differential equation and has many applications in natural or engineering sciences, e.g. the elastostatic problem of a rod under a torsion load, the Helmholtz equation in dynamically excited continua without bending stiffness or the equilibrium displacement of a stretched membrane subjected to a distributed force. Given the brevity of this paper, we investigate the solution efficiency and quality of the MoE-PINNs approach for solving the Poisson's problem, defined over a 2-dimensional L-shaped domain Ω with homogeneous Dirichlet boundary conditions on Γ:

$$\begin{aligned} -\Delta u(x, y) &= 1, \quad (x, y) \in \Omega = [-1, 1]^2 \setminus [0, 1]^2 \\ u(x, y) &= 0, \quad (x, y) \in \Gamma \end{aligned} \tag{5}$$

The training data for MoE-PINNs is sampled at the collocations points within the domain Ω as $\{x_i, \mathcal{F}(x_i)\}_{i=1}^{N_\Omega=683}$ and on the boundary $\{x_{\Gamma,i}, \mathcal{B}(x_{\Gamma,i})\}_{i=1}^{N_\Gamma=341}$.

Table 1: Performance of MoE-PINNs with different configurations and *tanh* activation function. MSE u denotes the mean squared error between the prediction and the reference solution computed using the spectral element method (SEM).

# of experts m	Depth	Width	Parameters	Order p	MSE u	MSE u GatedPINN
1	3	128	33'537	none	$8.32 \cdot 10^{-5}$	
1	3	256	132'609	none	$8.73 \cdot 10^{-5}$	
3	3	128	105'164	none	$9.42 \cdot 10^{-5}$	$9.06 \cdot 10^{-4}$
3	3	128	105'164	0.25	$5.93 \cdot 10^{-5}$	
3	3	128	105'164	0.5	$8.27 \cdot 10^{-5}$	
3	3	128	105'164	2	$1.08 \cdot 10^{-4}$	
4	3	128	138'767	none	$7.78 \cdot 10^{-5}$	$1.38 \cdot 10^{-3}$
4	3	128	138'767	0.25	$6.38 \cdot 10^{-5}$	
4	3	128	138'767	0.5	$6.89 \cdot 10^{-5}$	
4	3	128	138'767	2	$7.69 \cdot 10^{-5}$	
5	3	128	172'370	none	$7.70 \cdot 10^{-5}$	$1.71 \cdot 10^{-3}$
5	3	128	172'370	0.25	$8.28 \cdot 10^{-5}$	
5	3	128	172'370	0.5	$7.75 \cdot 10^{-5}$	
5	3	128	172'370	2	$8.56 \cdot 10^{-5}$	

For ease of interpretation of the results, we manually varied the values of only a selection of the most important hyperparameters. More specifically, we first compared ensembles containing $m \in \{1, 3, 4, 5\}$ experts, each with a depth of 3 layers, a width of 128 nodes each and the hyperbolic tangent as activation function. In a second step, the experiments involving the automated architecture search were conducted on a more diverse set of architectures. We initialised the PINN experts in the ensemble with the activation functions *tanh*, *sigmoid*, *sine*, *softplus* or *swish*, varied the depth between 1 and 4 layers and selected the width from the set 32, 64, 128, 256. Since the experiments were all conducted on the Poisson PDE with L-shaped domain, we kept the hyperparameters of the gating network fixed at depth 2, width 64 and used the hyperbolic tangent as activation function. In addition to the losses from the PINNs and the gating NN, a penalisation of the number of experts is added by setting a sparsity order p. The contributions of the terms in the objective function are balanced using ReLoBRaLo [10]. Furthermore, we initialised the learning rate of the Adam optimiser at 0.001 and decreased it by a multiplicative factor of 0.1 whenever the optimisation stopped making progress for over 3'000 optimisation steps and finally used early stopping in the event of 9'000 steps without improvement, followed by an additional 15'000 iterations of L-BFGS. The validation data (ground truth $\bar{u}$) are taken from [12] and were obtained using the spectral element method (SEM). All computations in the scope of this study were performed on a NVIDIA A5000 GPU. Table 1 summarises the performances of MoE-PINNs with different configurations and compares them to GatedPINNs with similar configurations. Note that the largest improvement over the baseline GatedPINNs stems from replacing the *ReLU*

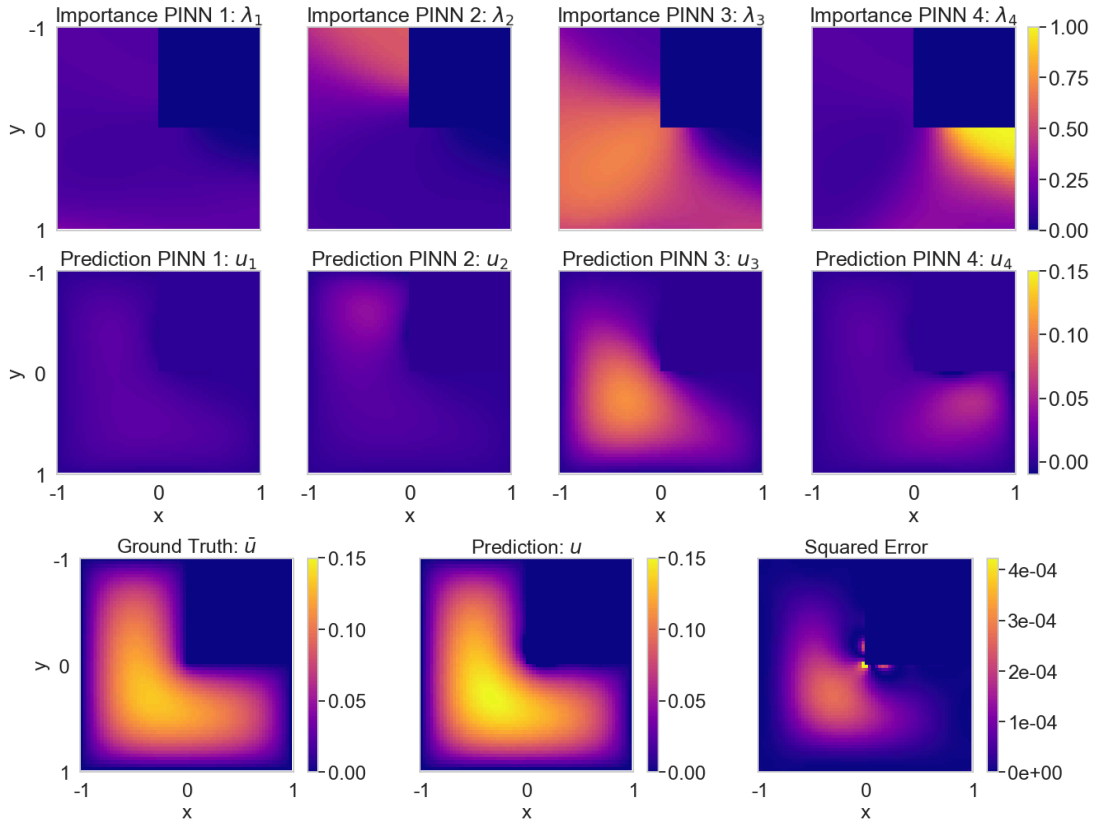

Figure 2: Results for a MoE-PINN containing 4 experts with depth 3, width 128, *tanh* activation function and sparsity order p = 0.5. Top row: distribution and magnitude of the importances for each PINN in the ensemble. Center row: output of each individual PINN. Bottom row: ground truth and prediction of the entire ensemble.

activation function in the gate with the infinitely, continuously differentiable function *tanh*. Finally, the results for the automated and differentiable architecture search with MoE-PINNs for different activation functions and varying number of experts is presented in Fig. 3.

5 Discussion and Conclusions

In general, we can show that all MoE-PINNs setups approximate the reference solution to a satisfactory degree. It can be observed that an ensemble of three learners noticeably improves the performance over the baseline model with a comparable number of parameters. However, the accuracy deteriorates when increasing the number of experts beyond 3. Thus, we conjecture that the sparsity regularisation is not enough for the MoE-PINNs to automatically tend towards the optimal number of three experts and dropping any additional networks in the ensemble. However, note that choosing the sparsity regularisation p = 0.5 results in a lower error than the less aggressive p = 2 across all conducted experiments. This indicates that a resolute regularisation, which effectively drives weak learners towards zero importance, is to be preferred over a more uniform division of the domain. Future research will therefore evolve around model pruning, where experts falling below a certain threshold of

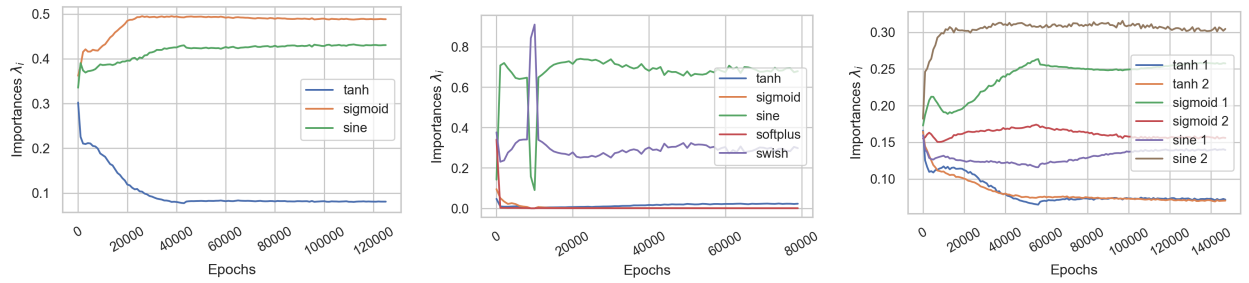

Figure 3: Evolution of the importances λ_i averaged over the entire batch of random collocation points per expert PINN with different activation functions with a total number of experts equal to the number of lines in each plot.

importance are removed from the ensemble and the model subsequently fine-tuned solely with the remaining experts.

Fig. 2 illustrates the results of a MoE-PINN with four equivalent experts. It is noticeable that the gating network opted for almost completely dropping one expert under the sparsity regularisation, and divided the domain symmetrically amongst the remaining PINNs by assigning one dominant expert to each of the three quadrants.

When inspecting the importances in ensembles of very diverse experts, i.e. consisting of a variable number of layers and nodes, as well as different activation functions, it was surprising to observe that the gating consistently discarded the networks with *tanh* activations (c.f. Fig. 3), as the *tanh* is one of the most common choices in the literature of PINNs. On the other hand, MoE-PINNs containing the *sine* activation performed exceedingly well compared to the case where a single PINN was initialised with this same activation function. There have been several attempts at making use of the *sine* function due to its well-behaved derivatives, but it has so far not been able to depose *tanh* or *swish* as the most popular choices. The observations in Figure 3 suggest that an ensemble of networks with *sine* activations may provide this function with an additional boost of performance. Finally, the notion of a linear combination of *sine* waves ties in nicely with the popular signal decomposition as used for example in the Fourier Transform.

Future research is concerned with inspection of performance, efficiency, robustness and scalability of MoE-PINNs to other PDE families arising in civil engineering.

Acknowledgements

The authors would like to thankfully acknowledge the facilities of Design++ at ETH Zürich and the funding through ETH Foundation grant No. 2020-HS-388 (provided by Kollbrunner/Rodio) as well as the SDSC Project "Domain-Aware AI-augmented Design of Bridges (DAAAD Bridges)".

References

[1] I. E. Lagaris, A. Likas, and D. I. Fotiadis, “Artificial neural networks for solving ordinary and partial differential equations”, *IEEE transactions on neural networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[2] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations”, *arXiv*, 2017.

[3] *Nvidia modulus*, 2022. [Online]. Available: https://developer.nvidia.com/modulus.

[4] S. Wang, Y. Teng, and P. Perdikaris, “Understanding and mitigating gradient pathologies in physics-informed neural networks”, *arXiv e-prints*, arXiv:2001.04536, arXiv:2001.04536, Jan. 2020. arXiv: 2001.04536 [cs.LG].

[5] X. Liu, X. Zhang, W. Peng, W. Zhou, and W. Yao, “A novel meta-learning initialization method for physics-informed neural networks”, *arXiv preprint arXiv:2107.10991*, 2021.

[6] A. Jagtap and G. Karniadakis, “Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations”, *Communications in Computational Physics*, vol. 28, pp. 2002–2041, Nov. 2020.

[7] P. Stiller, F. Bethke, M. Böhme, *et al.*, “Large-scale Neural Solvers for Partial Differential Equations”, *arXiv e-prints*, arXiv:2009.03730, arXiv:2009.03730, Sep. 2020. arXiv: 2009.03730 [cs.LG].

[8] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, “Adaptive mixture of local expert”, *Neural Computation*, vol. 3, pp. 78–88, Feb. 1991.

[9] S. Masoudnia and R. Ebrahimpour, “Mixture of experts: A literature survey”, *Artificial Intelligence Review*, vol. 42, no. 2, pp. 275–293, 2014.

[10] R. Bischof and M. Kraus, “Multi-Objective Loss Balancing for Physics-Informed Deep Learning”, *arXiv*, Oct. 2021.

[11] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.

[12] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, “DeepXDE: A deep learning library for solving differential equations”, *arXiv e-prints*, arXiv:1907.04502, arXiv:1907.04502, Jul. 2019. arXiv: 1907.04502 [cs.LG].