Adaptive Mixing of Auxiliary Losses in Supervised Learning

Durga Sivasubramanian[1,2*]**, Ayush Maheshwari**[1]**, Prathosh AP**[3]**,**
Pradeep Shenoy[2]**, Ganesh Ramakrishnan**[1]

[1]Indian Institute of Technology Bombay
[2]Google Research, India
[3]Indian Institute of Science, Bengaluru
{durgas, ayusham, ganesh}@cse.iitb.ac.in, prathosh@iisc.ac.in, shenoypradeep@google.com

Abstract

In several supervised learning scenarios, *auxiliary losses* are used in order to introduce additional information or constraints into the supervised learning objective. For instance, knowledge distillation aims to mimic outputs of a powerful *teacher model*; similarly, in rule-based approaches, weak labeling information is provided by *labeling functions* which may be noisy rule-based approximations to true labels. We tackle the problem of learning to combine these losses in a principled manner. Our proposal, AMAL, uses a bi-level optimization criterion on validation data to learn optimal mixing weights, at an instance-level, over the training data. We describe a meta-learning approach towards solving this bi-level objective and show how it can be applied to different scenarios in supervised learning. Experiments in a number of knowledge distillation and rule denoising domains show that AMAL provides noticeable gains over competitive baselines in those domains. We empirically analyze our method and share insights into the mechanisms through which it provides performance gains. The code for AMAL is at: https://github.com/durgas16/AMAL.git.

1 Introduction

Deep learning techniques have shown significant impact in a wide range of machine learning applications, driven primarily by the availability of large amounts of reliable labeled data (Sun et al. 2017). Despite this progress, supervised learning faces certain challenges: first, the time and effort needed to obtain large, reliable labeled datasets, and second, the limited information contained in human-annotated labels. Several approaches aim to improve generalization and sample efficiency of supervised learning by incorporating additional sources of information, or learning constraints, into the supervised learning paradigm. For instance, rule-denoising techniques (Ratner et al. 2016) use simple, approximate labeling rules (labeling functions) that provide weak supervision and reduce dependence on data annotation. Other work has combined learning from labeling functions with supervised learning from limited human-annotated data (Maheshwari et al. 2021)–these approaches leverage the supervised learning objective to offset the noisy labels from labeling functions. A challenge here is how to optimally combine these complementary objectives.

Equally, cardinal labels do not capture the richness of information contained in the input data–*e.g.*, object category labels for images of natural scenes. Some of this imprecision can be mitigated by using more nuanced 'soft labels', or distributions over labels, as the target for supervision instead of the cardinal labels. Knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015)) proposes using the inherent uncertainty of a supervised model trained on cardinal labels (the 'teacher model') to generate these soft labels for training, in combination with the conventional supervision loss. Indeed, recent work (Menon et al. 2021) formalizes this process from a Bayesian perspective, showing that when one-hot labels are an imperfect representation of the true probability distribution, KD reduces the variance associated with probability estimates in a student model. Other work examines, from an empirical perspective, when and how distillation may improve upon training from scratch on the labels alone. For instance, an overtrained teacher will likely achieve low/zero error rates *w.r.t.* the (incomplete) label loss simply by overfitting on random noise in the dataset; in these circumstances, the probabilities output by the teacher do not accurately represent the underlying uncertainty, and students may be led astray.

We propose AMAL, an adaptive loss mixing technique for addressing the challenge of optimally combining supervised learning objectives with these varied auxiliary objectives. Our proposal is driven by the following key insight: the mixing of primary and auxiliary objectives greatly benefits by being regulated on a *sample-by-sample basis*. This draws from substantial literature showing the promise of instance-reweighting, for example in handling noisy labels or outliers (Castells, Weinzaepfel, and Revaud 2020; Ren et al. 2018). We therefore propose to learn instance-specific mixing parameters that combine complementary learning objectives. We devise a meta-learning algorithm, based on a separate *validation metric*, to estimate these instance-specific parameters in an unbiased manner. We demonstrate how our method yields more accurate models when rule-based losses are mixed with limited supervision losses (Maheshwari et al. 2021) as well as in a knowledge distillation setting (KD) (Hinton, Vinyals, and Dean 2015).

Motivation for our work: We present motivation for our
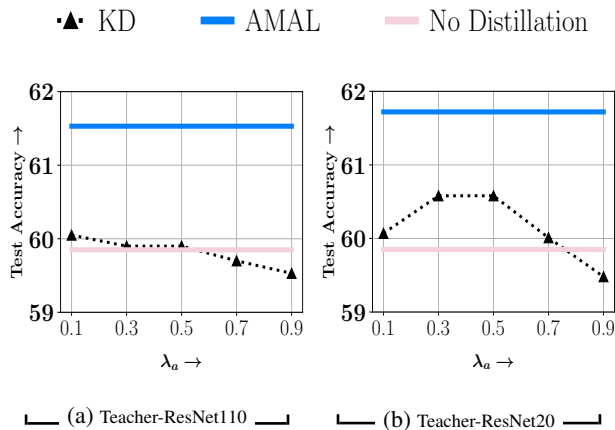
(a) Teacher-ResNet110   (b) Teacher-ResNet20

Figure 1: Knowledge distillation (KD) performed on CI-FAR100, with ResNet 8 as a student model. Subfigure (a) uses ResNet110 as teacher whereas subfigure (b) uses ResNet20 as teacher. KD performed with uniformly weighted (λ_a) performs poorly as the gap between the learning capacities of the teacher and student models increases. In both the cases, AMAL with the weights learned performs the best.

work in a knowledge distillation (KD) setup on the standard CIFAR100 dataset (Krizhevsky 2009)– the student model is set to ResNet8 and the teacher model to ResNet110 in subfigure (a). Since the capacity difference between the student and teacher models is large, mimicking the teachers outputs maybe harmful rather than helpful for the training of the ResNet8 model (Cho and Hariharan 2019). This is illustrated in Figure 1 where we present the performance of KD obtained with different values of λ_a (parameter controlling the influence of KD loss - *c.f.* Section 4.1). Here we set $\lambda_p = 1 - \lambda_a$ as in equation (10). We compare this against learning with only standard hard labels (no KD, *i.e.*, $\lambda_a = 0$) and AMAL with learned λs. In Figure 1 (a) we observe that KD performs almost similar to or worse than No-Distillation baseline. To understand the effect of capacity difference between the teacher and student models, in Figure 1 (b) we perform distillation with ResNet20 as teacher and ResNet8 as student. Due to this reduction in the capacity difference, for some values of λ, KD performs better than No-Distillation, but cannot bridge the gap to AMAL's performance using optimal loss mixing.

To further motivate instance-wise mixing, we apply AMAL to a KD setup with 40% label noise injected into the CIFAR100 dataset. Here, too, we use ResNet110 as teacher and ResNet8 as student. We examine the difference between the weights associated with the distillation loss and the supervision loss (λ_a and λ_p respectively). Figure 2 shows this difference as a histogram over instances separated into clean and noisy labels. AMAL favors supervision loss for clean data points (*i.e.*, negative range of $\lambda_a - \lambda_p$), as intended from an optimal mixing perspective. This is consistent with the observation we made on Figure 1, where the student learns better when the KD loss is assigned lower weightage. In a
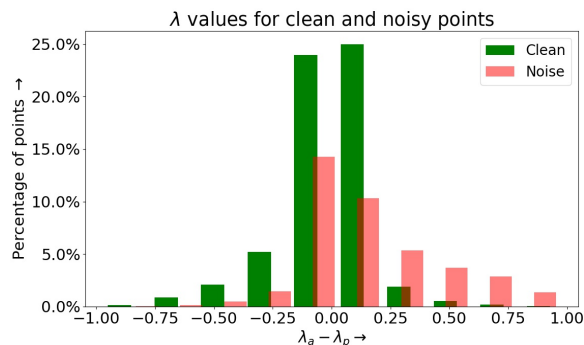

Figure 2: Distribution of difference between λ_a (weight associated with the KD loss) and λ_p (weight associated with the CE loss), obtained using AMAL while performing knowledge distillation with Resnet8 as the student model and Resnet110 the teacher model on CIFAR100 dataset with 40% label noise.

similar fashion, AMAL emphasizes KD loss for noisy points, correctly identifying that the teacher model is more informative for those points than their misleading hard labels.

Our Contributions: Our key contributions are as follows:
1) We propose a general formulation for *instance-specific* mixing of auxiliary objectives in supervised learning. This is, to our knowledge, the first proposal of its kind (*c.f.* Section 3).
2) **AMAL in KD settings:** We explore a range of settings in Knowledge Distillation (KD), including vanilla KD, multi-teacher KD, and early-stopping, showing significant gains over and above SOTA KD approaches in these settings (*c.f.* Section 4.1).
3) **AMAL in Rule-Denoising Setting with Limited Supervision:** We show how the problem of semi-supervised data programming can benefit from AMAL and report gains of 2-5% on various datasets (*c.f.* Section 4.2).

2 Related Work

Knowledge Distillation (KD) KD (Hinton, Vinyals, and Dean 2015) in a supervised learning setting trains a 'student' model to mimic the outputs of a larger, pre-trained 'teacher' model instead of directly training on the supervised signal. The efficacy of KD can be limited by teacher accuracy (see (Menon et al. 2021) for some theoretical results), and student representational capacity, among other factors. Interestingly, early stopped teacher models aid better in training the student models (Cho and Hariharan 2019); however, identifying the best possible teacher requires repeating the distillation process multiple times on the student model. To bridge the representational gap between the teacher and the students, Teacher Assistants (TA) or intermediate models were introduced (Mirzadeh et al. 2020), and were improved by a stochastic approach (DGKD (Son et al. 2021)) for simultaneously training all intermediate models with occasional model dropout. In (Liu, Zhang, and Wang 2020), multiple teacher networks are used with an intermedi-

ate knowledge transfer step using latent distillation. *All these works attempt to improve KD efficacy in cases in which there is a large gap between the teacher and student model as in the case presented by us in Figure 1. However these methods require us to independently train additional models, in contrast to our work wherein we strategically mix loss components.*

Instance-Specific Learning: A significant amount of past literature has explored instance-specific learning, for instance instance-specific temperature parameters in supervised learning (Saxena, Tuzel, and DeCoste 2019). Other closely related work (Algan and Ulusoy 2021; Vyas, Saxena, and Voice 2020) learns a per-instance *label uncertainty* parameter to account for potential label noise. In the distillation setting, too, Zhao et al. (2021) demonstrate the benefits of learning an instance-level sequence (or curriculum) on training samples. Castells, Weinzaepfel, and Revaud (2020) propose a task-agnostic per-sample loss-function representing the reliability of each prediction. Other recent works such as (Ren et al. 2018; Shu et al. 2019; Raghu et al. 2020), use validation set based meta learning to learn instance-specific weights to improve robustness. *The novelty of our work is that we seek task-agnostic, per-sample, loss mixing coefficients, specifically for effective learning over multiple losses.*

Bi-level Optimization and Meta-Learning: Prior work (Jenni and Favaro 2018; Bengio 2000; Domke 2012) has explored learning network hyper-parameters via solving a two-level optimization problem–one on the base-task and another on an external model-selection or meta-task, often on validation data. These algorithms are similar in spirit to the *learning to learn* literature, typically in multi-task contexts (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018; Hospedales et al. 2020; Vyas, Saxena, and Voice 2020). Typical approaches aim to learn a "meta-"algorithm which can generalize across tasks by mimicking the test dynamics (sampling test tasks, in addition to test data, for measuring and optimizing loss) during training (Hospedales et al. 2020). Although this literature, too, employs nested optimization objectives, *it differs from our work in that we wish to improve generalization within a single task, rather than across tasks.*

Training with Auxiliary Tasks: Information from auxiliary tasks are used to improve the main task in methods like (Lin et al. 2019; Navon et al. 2021) learn to reweigh auxiliary tasks to improve performance on the main task. Guo et al. (2018) construct a dynamic curriculum by weighing individual auxiliary tasks. Similarly, Shi et al. (2020) weigh auxiliary tasks to perform learning in a limited labeled data setting. The aforementioned approaches focus on unifying several losses into a single coherent loss *whereas our focus is on instance-wise contribution of the loss components.*

3 AMAL: Adaptive Mixing of Auxiliary Losses

We consider the scenarios in which there are two or more loss terms participating in a supervised learning setting. The loss functions we consider adhere to the form specified in

---

Algorithm 1: Algorithm for learning λs via meta learning

Require: Training data $\mathcal{U}$, Validation data $\mathcal{V}$, $\theta^{(0)}$ model parameters initialization, τ Temperature, η: learning rate, η_λ: learning rate for updating λ.

Require: LL_{CE} Primary Supervised Loss, L_a auxiliary loss, max iterations T

1: Initialize model parameters $\theta^{(0)}$ and $\lambda_p^0, \lambda_{a_1}^0, \lambda_{a_2}^0, \cdots, \lambda_{a_K}^0$.
2: **for** $t \in \{0, \ldots, T\}$ **do**
3:     Update θ^{t+1} by Eq. (7).
4:     **if** t % L == 0 **then**
5:         $\boldsymbol{x}^{train}, y^{train} \leftarrow \texttt{SampleMiniBatch}(\mathcal{U})$
6:         $\boldsymbol{x}^{val}, y^{val} \leftarrow \texttt{SampleMiniBatch}(\mathcal{V})$
7:         Compute one step update for model parameters as function of $\lambda_p^{\lfloor \frac{t}{L} \rfloor}, \lambda_{a_1}^{\lfloor \frac{t}{L} \rfloor}, \lambda_{a_2}^{\lfloor \frac{t}{L} \rfloor}, \cdots, \lambda_{a_K}^{\lfloor \frac{t}{L} \rfloor}$ by Eq.(4).
8:         Update $\Lambda^{\lfloor \frac{t}{L} \rfloor}$ by Eq.(6).
9:     **end if**
10: **end for**

Eq. 1, where there is a primary objective and K auxiliary objectives.

$$\mathcal{L} = \lambda_p * \mathcal{L}_p + \sum_{k=1}^{K} \lambda_{a_k} \mathcal{L}_{a_k} \quad (1)$$

Here, $\mathcal{L}_p$ and $\mathcal{L}_{a_i}$ respectively are the primary and auxiliary loss objectives. While this formulation is general, in this paper, we explicate the formulation in two different settings – knowledge distillation (Section 4.1), and rule-denoising (Section 4.2). In these settings, we begin with a labeled dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ with instances $\boldsymbol{x}_i$ and categorical labels y_i and an unlabelled dataset $\mathcal{U} = \{(\boldsymbol{x}_i)\}_{i=N}^{N+M}$ only with instances $\boldsymbol{x}_i$. Note that, in the knowledge distillation setting, $\mathcal{U}$ will be empty and in the case of rule-based denoising setting $N << M$. Our main proposal is to modify the objective in Eq. (1) so that loss-mixing coefficients (Λ) are instance-specific. Formally, we modify the loss function in Eq. (1) as follows:

$$\mathcal{L}(\theta, \Lambda) = \sum_i \left(\lambda_{p_i} \mathcal{L}_p(y_i, \boldsymbol{x}_i | \theta) + \sum_{k=1}^{K} \lambda_{a_{k,i}} \mathcal{L}_{a_{k,i}} \right) \quad (2)$$

Note that formulation in Eq. (2) is a generalization of Eq. (1), with an instance-specific value of mixing parameters $\Lambda = \{\lambda_p, \lambda_{a_1}, \lambda_{a_2}, \cdots, \lambda_{a_K}\}$ corresponding to the ith training instance $\boldsymbol{x}_i$. Jointly optimizing the objective in Eq. (2) with respect to both sets of parameters θ, Λ on the training dataset alone can lead to severe overfitting. To mitigate this risk, we instead attempt to solve the bi-level minimization problem in Eq. (3) using a meta-learning procedure:

$$\overbrace{\operatorname*{argmin}_{\Lambda} \mathcal{L}_{CE}\big(\underbrace{\operatorname*{argmin}_{\theta} \mathcal{L}(\theta, \Lambda)}_{inner-level}, \mathcal{V}\big)}^{outer-level} \quad (3)$$
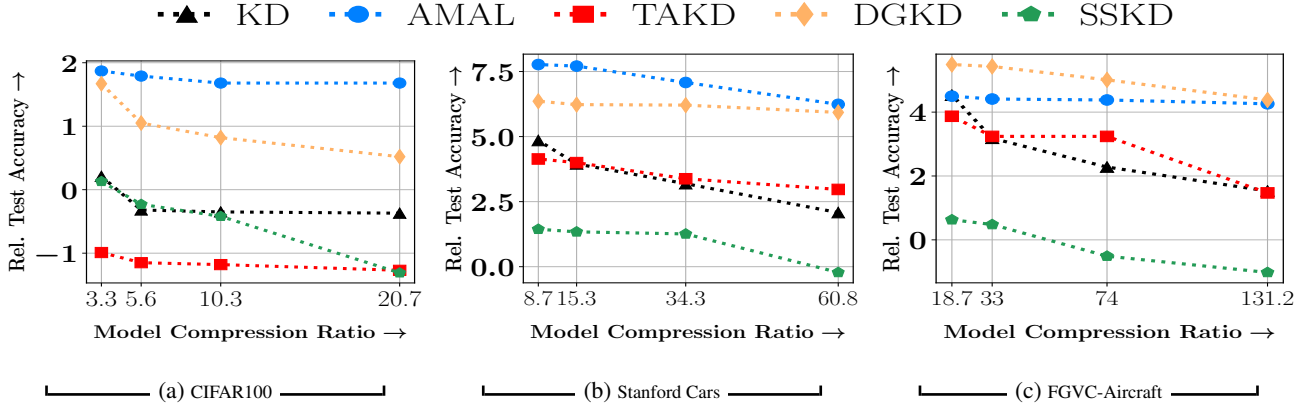
Figure 3: KD performed with ResNet (8,20,32,56,110) models on CIFAR100 in subfigure (a), Wide Residual Networks (WRN-16-1, WRN-16-3,WRN-16-4,WRN-16-6,WRN-16-8) on Stanford Cars in subfigure (b) and with Wide Residual Networks (WRN-16-3,WRN-16-4,WRN-16-6,WRN-16-8) as teachers and Resnet8 as student on FGVC-Aircraft in subfigure (c). AMAL consistently outperforms various state-of-the-art methods when the model compression ratio is higher.

By solving the inner level minimization, we wish to obtain model parameters θ that minimise the objective in Eq. (2). The outer minimization yields λs such that the standard cross-entropy loss is minimised on the validation set $\mathcal{V}$. This problem is a bi-level optimisation problem since model parameters θ are dependent on Λ and computation of Λ is dependent on model parameters θ as shown in Eq.(3).

Since the inner optimisation problem cannot be solved in a closed form in Eq. (3), we need to make some approximations in order to solve the optimization problem efficiently. We take an iterative approach, simultaneously updating the optimal model parameters θ and appropriate Λ in alternating steps as described in the Algorithm1. We first update the model parameters by sampling a mini-batch with n instances from the training set, and simulating a one step look-ahead SGD update for the loss in Eq.(2) on model parameters (θ^t) as a function of Λ^t, resulting in Eq. (4), with L being a hyperparameter governing how often the lambda values are updated.

$$\hat{\theta}^t\left(\Lambda^{\lfloor \frac{t}{L} \rfloor}\right) = \theta^t - \frac{\eta}{n}\sum_{i=1}^{n}\nabla_{\theta^t}\mathcal{L}_i(\theta^t(\Lambda_i^{\lfloor \frac{t}{L} \rfloor})) \qquad (4)$$

Using the approximate model parameters obtained using the one step look-ahead SGD update, the outer optimization problem is solved as,

$$\nabla_{\Lambda_i^{\lfloor \frac{t}{L} \rfloor}}\mathcal{L}_{CE}(\hat{\theta}^t, \mathcal{V}) \qquad (5)$$
$$= -\frac{\eta}{n}.\nabla_{\hat{\theta}^t}\mathcal{L}_{CE}(\hat{\theta}^t, \mathcal{V}).\nabla_{\Lambda_i^{\lfloor \frac{t}{L} \rfloor}}\nabla_{\theta^t}\mathcal{L}_i^T$$

Using the meta-gradient in Eq.(5) we update the λs for each of the training samples using the first order gradient update rule as,

$$\Lambda_i^{\lfloor \frac{t}{L} \rfloor+1} = \Lambda_i^{\lfloor \frac{t}{L} \rfloor} - \eta_\lambda\nabla_{\Lambda_i^{\lfloor \frac{t}{L} \rfloor}}\mathcal{L}_{CE}(\hat{\theta}^t, \mathcal{V}) \qquad (6)$$

Here, η_Λ is the learning rate for mixing parameters. We update Λ values every L epochs. The updated $\Lambda_i^{\lfloor \frac{t}{L} \rfloor+1}$ values are then used to update the model parameters as,

$$\theta^{t+1} = \theta^t - \frac{\eta}{n}\sum_{i=1}^{n}\nabla_{\theta^t}\mathcal{L}_i(\theta^t(\Lambda_i^{\lfloor \frac{t}{L} \rfloor+1})) \qquad (7)$$

Our method converges to the optima of both the validation and training loss functions under some mild conditions.

3.1 Speeding Up AMAL

We borrow two important implementation schemes from few of the recent subset selection techniques (Killamsetty et al. 2021b,a) to streamline mixing parameter updates in AMAL. Firstly, instead of using the complete high dimensional loss gradient associated with modern deep neural networks we only consider last-layer gradient of a network. This helps in reducing both computation time and memory in both the one step update (Eq. (4)) and computation of the meta-gradient (Eq. (5)). Similarly, the proposal to update Λ only after L epochs also reduces the computation time significantly. Bi-level optimisation solved with these tricks has been shown to yield significant speedup (Killamsetty et al. 2021b) and with minimal loss in performance. Thus, training with AMAL introduces negligible overhead.

4 Two Application Scenarios for AMAL

In this Section, we present two application scenarios for AMAL, described in the previous Section 3, *viz.*, knowledge distillation 4.1 and learning with limited supervision and rule-denoising in Subsection 4.2.

4.1 Knowledge Distillation

Any (student) model having output logits as $a^{(S)} = \texttt{StudentModel}(\boldsymbol{x})$, is traditionally trained by optimizing a cross-entropy based loss $\mathcal{L}_s$ defined as follows:

$$\mathcal{L}_s = L_{CE}\big(\texttt{softmax}(a^{(S)}), y\big) \qquad (8)$$

Let us say we have access to a pretrained teacher model (typically of higher learning capacity) which outputs the logits $a^{(T)} = \texttt{TeacherModel}(\boldsymbol{x})$. We can frame a *teacher*

9858

matching objective for the student as a KL-divergence between the predictions of the student and the teacher:

$$\mathcal{L}_{KD} = \tau^2 KL\big(y^{(S)}, y^{(T)}\big) \tag{9}$$

Then the training of the student model can be performed using both the *teacher matching* objective and the traditional cross entropy loss as,

$$\mathcal{L}_{student}(\theta, \lambda) = \sum_i (1 - \lambda)\mathcal{L}_s\left(y_i, \boldsymbol{x}_i | \theta\right) + \lambda \mathcal{L}_{KD}\left(y_i^{(S)}, y_i^{(T)}\right) \tag{10}$$

This is the standard knowledge distillation loss, in which a temperature parameter τ is typically used to control the softening of the KD loss in Eq. (9); therefore we have $y^{(S)} = \mathtt{softmax}\big(\frac{a^{(S)}}{\tau}\big)$ and $y^{(T)} = \mathtt{softmax}\big(\frac{a^{(T)}}{\tau}\big)$. We change this objective to match AMAL's objective as,

$$\mathcal{L}_{student}(\theta, \boldsymbol{\Lambda}) = \sum_i \lambda_{p_i}\mathcal{L}_s\left(y_i, \boldsymbol{x}_i | \theta\right) + \lambda_{a1_i}\mathcal{L}_{KD}\left(y_i^{(S)}, y_i^{(T)}\right) \tag{11}$$

Clearly, here $\mathcal{L}_p$ would be $\mathcal{L}_s$ and $\mathcal{L}_a$ would be $\mathcal{L}_{KD}$. We present the results of applying AMAL to adaptively mix these losses in Section 5.1. AMAL can be extended to settings where distillation is performed with multiple teachers such as DGKD (Son et al. 2021).

4.2 Learning with Limited Supervision and Rule-Denoising

Several rule-denoising approaches (Maheshwari et al. 2022, 2021; Awasthi et al. 2020; Ratner et al. 2017) encode multiple heuristics in the form of rules (or labeling functions) to weakly associate labels with instances. These weak labels are aggregated to determine the probability of the correct labels using generative models (Chatterjee, Ramakrishnan, and Sarawagi 2020; Ratner et al. 2017) without requiring labeled data. In contrast, recent approaches (Maheshwari et al. 2021; Karamanolakis et al. 2021; Awasthi et al. 2020; Ren et al. 2020, 2018) assume that a small labeled dataset is available in conjunction with the noisy rules. Motivated by the success of rule denoising approaches, we propose adaptive loss mixing to leverage a small labeled set while being trained in a joint manner. We directly adopt the model and loss formulations from the most recent of these approaches (Maheshwari et al. 2021), since it performs consistently better than the previous ones (Maclaurin, Duvenaud, and Adams 2015; Awasthi et al. 2020; Ren et al. 2020, 2018) (see Section 5.3).

Our setting borrowed from SPEAR (Maheshwari et al. 2021) is as follows: In addition to the setting described in Section 3, we also have access to m rules or labelling functions (LF) lf_1 to lf_m. We modify $\mathcal{D}$ to be $\mathcal{D}' = \{(\boldsymbol{x}_i, y_i, l_i)\}_{i=1}^N$ and $\mathcal{U}$ to be $\mathcal{U}' = \{(\boldsymbol{x}_i, l_i)\}_{i=N}^{N+M}$, where $l_i = (l_{i1}, l_{i2}, \cdot, l_{im})$ is a boolean vector with $l_{ij} = 1$ if the corresponding j^{th} LF is activated on example $\boldsymbol{x}_i$ and $l_{ij} = 0$ otherwise. Exactly as per (Maheshwari et al. 2021), our model is a blend of the feature-based classification model $P_\theta^f(\mathbf{x})$ and the rule-based model $P_\phi(\mathbf{l}_i, y)$. We have two types of supervision in our joint objective. First, we have

access to y for the labeled instances $\mathcal{D}'$ and to l_{ij} for all the labeled as well as unlabeled instances $\mathcal{D}' \bigcup \mathcal{U}'$. We measure the loss of P_θ^f and P_ϕ on the small labeled set $\mathcal{D}'$ through standard cross-entropy. Second, we model interaction between P_θ^f and P_ϕ on the union of labeled and unlabeled sets. Intuitively, the rule denoising model P_ϕ learns with respect to the clean labeled set $\mathcal{D}'$ and simultaneously provides labels over $\mathcal{U}$ that can be used to train the feature model $P_\theta^f(y|x)$. We want both the models to agree in their predictions over the union $\mathcal{D}' \bigcup \mathcal{U}'$ (Please refer to Supplementary Section E for details about individual loss components.)

5 Results

In this section, we present results for the two application scenarios for AMAL as outlined in Section 4.

5.1 Results with Knowledge Distillation

In this section, we report a range of experimental results from the knowledge distillation (KD) scenario as described in Section 4.1. We performed a range of experiments comparing AMAL against several SOTA knowledge distillation approaches on several real-world datasets, with a special focus on those settings wherein we found the gap between the teacher and student models to be large.

Datasets The datasets in our experiments include CIFAR100 (Krizhevsky 2009), Stanford Cars (Krause et al. 2013)and FGVC-Aircraft (Maji et al. 2013). For the CIFAR datasets we used the standard RGB images of size 32×32, whereas for the other datasets we used RGB images of size 96×96.

Model Architecture and Experimental Setup We explored two families of models, *viz.*, (i) Wide Residual Networks (WRN-16-1, WRN-16-3,WRN-16-4,WRN-16-6,WRN-16-8) (Zagoruyko and Komodakis 2016), and (ii) ResNet (8,20,32,56,110) models (He et al. 2016) to show the effectiveness of our method across the different model families. We also perform a distillation on Resnet8 with WRN-16-3,WRN-16-4,WRN-16-6 and WRN-16-8 as teachers to show the effect of our technique in the cross-model distillation.

For datasets without pre-specified validation sets, we split the original training set into new train (90%) and validation sets (10%). Training consisted of SGD optimization with an initial learning rate of 0.05, momentum of 0.9, and weight decay of 5e-4. We divided the learning rate by 0.1 on epochs 150, 180 and 210 and trained for a total of 240 epochs.

Effect of Optimal λs on Knowledge Distillation In the first experiment, we examine effective transfer of learned knowledge from various teachers to a student model which has fewer parameters. We compares test accuracies obtained with KD, AMAL , TAKD (Mirzadeh et al. 2020) and DGKD (Son et al. 2021) and SSKD (Xu et al. 2020). TAKD takes taking multiple KD training hops, with each step reducing the model complexity from teacher to student by a small amount. DGKD introduces all the intermediate teachers in a single KD training step using a single λ, across all training instances, for each teacher. In addition, stochastic

DGKD was proposed where a subset of teachers is introduced at each training step, determined by a binomial (hyperparamter) variable.

Additional Experimental Setup We perform KD with ResNet (20,32,56,110) as teacher and ResNet8 as student models on CIFAR100, Wide Residual Networks (WRN-16-3,WRN-16-4,WRN-16-6,WRN-16-8) as teacher and WRN-16-1 as student models on Stanford Cars and with Wide Residual Networks (WRN-16-3,WRN-16-4,WRN-16-6,WRN-16-8) as teachers and Resnet8 as student on FGVC-Aircraft. For TAKD and DGKD we use ResNet14 for CIFAR100 and WRN-16-2 for Stanford Cars and FGVC-Aircraft as teaching assistant models. In all our knowledge distillation experiments we use temperature $\tau = 4$ and $\lambda_a = 0.9$ (weights associated with KD loss) except in case of AMAL. For DGKD we use set the binomial variable to be 0.75, best reported in the paper.

Figure 3 shows that AMAL consistently outperforms other techniques when a much smaller model learns from large teacher model (CIFAR100, Stanford Cars) and is comparable to DGKD in FGVC-Aircraft dataset. The figure shows plot relative test accuracies (w.r.t. non-KD students) vs model compression ratio[1].Interestingly, methods such as KD, SSKD and TAKD actually perform worse than training a student model with standard cross entropy loss. This observation is consistent with (Cho and Hariharan 2019), where authors claim KD may fail if the student is too weak. This problem gets worse when techniques such as SSKD bring even more additional information for the student model to learn. TAKD tries to address this issue by bring in teaching assistant model, which have already gone through with knowledge distillation from the teacher model. However, this also transfer errors from the higher level to the lower level models (Son et al. 2021). It is important to note that AMAL doesn't require any additional intermediate model to be trained like TAKD and DGKD and therefore has a lesser memory footprint and training time.

Knowledge Distillation in Presence of Noise As AMAL performs instance wise mixing of loss components, noise filtering in knowledge distillation (with two loss components) is an appropriate use case. We perform knowledge distillation with CIFAR100 dataset with n% labels randomly changed to a wrong label. We continue using the ResNet (8,20,32,56,110) model with ResNet8 being the student model. We present test accuracies obtained while training with 40% and 60% label noise in Figure 4. We compare against two loss agnostic robust learning techniques *viz.* (i) **Superloss** (Castells, Weinzaepfel, and Revaud 2020): It is curriculum learning based approach which dynamically assigns weights to each instance to perform robust learning.(ii) **CRUST** (Mirzasoleiman, Bilmes, and Leskovec 2020): It selects a noise free subset of data points which approximates the low-rank Jacobian matrix.

Figure 4 we see that AMAL achieves best performance which could be explained by the mixing parameters'(Λ)
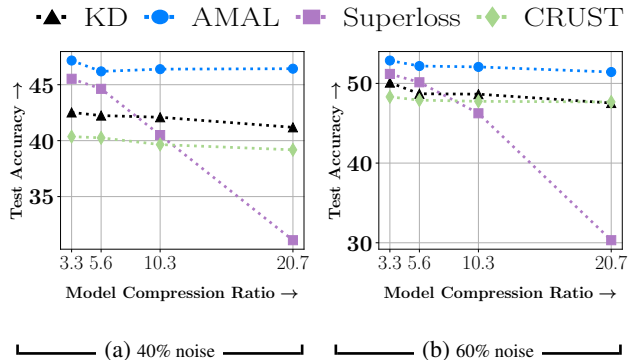
Figure 4: Test performance obtained after performing knowledge distillation with ResNet8 as the student and Resnet (20,32,56,110) as the teachers with CIFAR100 dataset corrupted with 40% and 60% label noise.

Method	Test Accuracy
Complete data (skyline)	66.43
Random	44.92
Sampled according to $\lambda_p^2 + \lambda_a^2$	45.5
Sampled according to $\lvert \lambda_p - \lambda_a \rvert$	46.28
Sampled according to $\frac{\lambda_a}{\lambda_p}$	46.31

Table 1: Test accuracies obtained after training with 20% subset obtained using various strategies using the WRN-16-1 model on the CIFAR100 dataset. We perform training only with the CE loss.

distribution presented in Figure 2. AMAL identifies importance of learning form cross entropy based loss for the clean points and learning from KD loss for noisy points. However, CRUST as it selects a subset selection it can't take advantage of both the losses. Superloss, on the other enjoys performance improvement over KD for smaller model compression ratios. However, superloss's performance drops significantly for higher compression ratios as it doesn't perform any kind of mixing.

5.2 Connection to a Coreset

Since, AMAL controls the contribution of each of the instances in training a model by weighting each of the points loss functions. We try to understand the significance of the weights associated with each data point with a coreset based experiment. Coreset selection has become popular in recent times where a subset of training points are used to train a model from scratch. Based on the final λ_p(weighted associated with the CE loss) and λ_a(weighted associated with the KD loss) values while training WRN-16-1 model using WRN-16-8 as a teacher model on CIFAR100 dataset, we derive a probability of selection p_i for each point i in the training set as,

1. $p_i \propto \lambda_{p_i}^2 + \lambda_{a_i}^2$, here we pick points with maximum weights as they would contribute maximum to the model training

2. $p_i \propto |\lambda_{p_i} - \lambda_{a_i}|$, here we pick points that should be preferably learnt with only one of the losses

3. $p_i \propto \frac{\lambda_{a_i}}{\lambda_{p_i}}$, here we pick points that should be preferably learnt with only KD loss

In Table 1 we present the test accuracies obtained on training WRN-16-1 with the coresets obtained when sampled using the corresponding probabilities. We also present the result of training the same model with randomly (sampled with uniform distribution) obtained subset. We train with subsets of 20% size of the original training data and train with only the CE loss. Clearly, the points that have higher weights have maximum information. More, specifically the points that require a teacher model's assistance and cannot be learned using the ground truth seem to have the most information and therefore coreset formed using 3 performs the best.

We tried techniques such as Platt-scaling (Guo et al. 2017) etc. which strengthens baselines with the use of validation data, but all those efforts either weakened or did not add any value to the existing baselines.

5.3 AMAL with Limited Supervision and Rule-Denoising

	SMS	IMDB	YouTube
Only-L	91.45 (1.3)	77.35 (1.5)	89.60 (2.9)
Imply Loss	+0.25 (1)	-1.47 (1.8)	+2.70 (0.8)
L2R	-0.20 (1.3)	-2.18 (1.4)	+3.40 (1.2)
MWN	-0.10 (1.2)	-1.53 (1.7)	+3.70 (1.5)
SPEAR	-0.76 (1.4)	-0.04 (1)	+4 (1)
AMAL	**+1.53** (0.9)	**+1.67** (1.6)	**+4.70** (0.8)

Table 2: Performance of our AMAL approach with rule-based approaches Imply Loss, SPEAR, L2R and MWN. AMAL with fixed $\lambda = 1$ corresponds to SPEAR. All numbers reported are gains over the baseline method (Only-L). All results are averaged over 5 random seed runs having different $\mathcal{L}$ and $\mathcal{U}$ set in each run. Numbers in brackets '()' represent standard deviation of the original score.

In this section, we report our experimental results for the scenario of limited supervision combined with weak supervision from labeling functions (also referred to as semi-supervised data programming (Maheshwari et al. 2021)), as summarized in Section 4.2. **Datasets** We used three dataset in our experiments, namely, YouTube, SMS and IMDB. **YouTube** (Alberto, Lochter, and Almeida 2015) is a spam classification task over YouTube comments; **SMS** (Almeida, Hidalgo, and Yamakami 2011) is a binary spam classification containing 5574 documents; **IMDB** is a movie plot genre binary classification dataset.

In Table 2, we compare our approach with the following approaches: (1) **Only-$\mathcal{L}$** : We train the classifier $P_\theta(y|x)$ only on the labeled data. To ensure fair comparison, we use the same classifier model for different datasets as mentioned in (Maheshwari et al. 2021). We choose this as a baseline and report gains over it. (2) **L2R** (Ren et al. 2018): This is an online reweighting algorithm that leverage validation set to assign weights to examples based on gradient directions. It learns to re-weigh weak labels from domain specific rules and learn instance-specific weights via meta-learning. (3) **Meta-Weight-Net(MWN)** (Shu et al. 2019) Trains a neural network assigns instantaneous weights. Neural network is trained to minimise validation set loss. However, weights are not learnt to mix losses in L2R and MWN. (4) **Imply Loss** (Awasthi et al. 2020): This is a rule-exemplar approach that jointly trains a rule denoising network and leverages *exemplar*-based supervision for learning instance-specific and rule-specific weights. In addition, it also learns a classification model with a soft implication loss in a joint objective. (5) **SPEAR** (Maheshwari et al. 2021): Finally, we compare with another rule-denoising approach that uses same objective as AMAL and is trained on both feature-based classifier and rule-classifier using a small labeled set. AMAL with all λs fixed to 1 (and not trainable) corresponds to SPEAR.

Our approach outperforms both rule-based and reweighing approaches on all datasets. MWN, L2R and SPEAR perform worse than the baseline method (only-L) on SMS and IMDB dataset whereas Imply-Loss is marginally better on SMS. All approaches achieve better performance over the baseline method on YouTube dataset. However, AMAL consistently reports highest gains. Recall that SPEAR has the same objective as AMAL but without trainable λs and all λs fixed to 1. AMAL tries to identify instance-wise weighted combination of loss components so that the trained feature classification model performs better. Instance wise mixing is useful to identify the loss component from which a data point could be learned better and use of fixed weights prevents from understanding nuance of each data point.

6 Conclusion

In this paper we present two setting viz. rule-denoising setting with limited supervision and knowledge distillation (KD), where Adaptive Loss Mixing is useful. We present AMAL which via adaptive loss mixing extracts useful information from the limited supervision to prevent degradation of model learnt due to the presence of noisy rule. In knowledge distillation (KD) setting it titrates the teacher knowledge and ground truth label information through an instance-specific combination of teacher-matching and ground supervision objectives to learn student models that are more accurate. Our iterative approach is pivoted on solving a bi-level optimization problem in which the instance weights are learnt to minimize the CE loss on a held-out validation set whereas the model parameters are themselves estimated to minimize the weight-combined loss on the training dataset. Through extensive experiments on real-world datasets, we present how AMAL yields accuracy improvement and better generalization on a range of datasets in both the settings.

Acknowledgements

References

Alberto, T. C.; Lochter, J. V.; and Almeida, T. A. 2015. Tubespam: Comment spam filtering on youtube. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 138–143. IEEE.

Algan, G.; and Ulusoy, I. 2021. Meta soft label generation for noisy labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 7142–7148. IEEE.

Almeida, T. A.; Hidalgo, J. M. G.; and Yamakami, A. 2011. Contributions to the study of SMS spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering*, 259–262.

Awasthi, A.; Ghosh, S.; Goyal, R.; and Sarawagi, S. 2020. Learning from Rules Generalizing Labeled Exemplars. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Bengio, Y. 2000. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8): 1889–1900.

Castells, T.; Weinzaepfel, P.; and Revaud, J. 2020. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems*, 33: 4308–4319.

Chatterjee, O.; Ramakrishnan, G.; and Sarawagi, S. 2020. Robust data programming with precision-guided labeling functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3397–3404.

Cho, J. H.; and Hariharan, B. 2019. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4794–4802.

Domke, J. 2012. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, 318–326. PMLR.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1321–1330. PMLR.

Guo, M.; Haque, A.; Huang, D.-A.; Yeung, S.; and Fei-Fei, L. 2018. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, 270–287.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Jenni, S.; and Favaro, P. 2018. Deep bilevel learning. In *Proceedings of the European conference on computer vision (ECCV)*, 618–633.

Karamanolakis, G.; Mukherjee, S.; Zheng, G.; and Hassan, A. 2021. Self-Training with Weak Supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 845–863.

Killamsetty, K.; Sivasubramanian, D.; Ramakrishnan, G.; De, A.; and Iyer, R. 2021a. GRAD-MATCH: Gradient Matching based Data Subset Selection for Efficient Deep Model Training. In *International Conference on Machine Learning*.

Killamsetty, K.; Sivasubramanian, D.; Ramakrishnan, G.; and Iyer, R. 2021b. GLISTER: Generalization based Data Subset Selection for Efficient and Robust Learning. *In AAAI*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3D Object Representations for Fine-Grained Categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Tront*.

Lin, X.; Baweja, H.; Kantor, G.; and Held, D. 2019. Adaptive auxiliary task weighting for reinforcement learning. *Advances in neural information processing systems*, 32.

Liu, Y.; Zhang, W.; and Wang, J. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415: 106–113.

Maclaurin, D.; Duvenaud, D.; and Adams, R. 2015. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, 2113–2122. PMLR.

Maheshwari, A.; Chatterjee, O.; Killamsetty, K.; Ramakrishnan, G.; and Iyer, R. 2021. Semi-Supervised Data Programming with Subset Selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4640–4651. Online: Association for Computational Linguistics.

Maheshwari, A.; Killamsetty, K.; Ramakrishnan, G.; Iyer, R.; Danilevsky, M.; and Popa, L. 2022. Learning to Robustly Aggregate Labeling Functions for Semi-supervised Data Programming. In *Findings of the Association for Computational Linguistics: ACL 2022*, 1188–1202.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Menon, A. K.; Rawat, A. S.; Reddi, S.; Kim, S.; and Kumar, S. 2021. A statistical perspective on distillation. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 7632–7642. PMLR.

Mirzadeh, S. I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; and Ghasemzadeh, H. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5191–5198.

Mirzasoleiman, B.; Bilmes, J.; and Leskovec, J. 2020. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, 6950–6960. PMLR.

Navon, A.; Achituve, I.; Maron, H.; Chechik, G.; and Fetaya, E. 2021. Auxiliary Learning by Implicit Differentiation. In *International Conference on Learning Representations*.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Raghu, A.; Raghu, M.; Kornblith, S.; Duvenaud, D.; and Hinton, G. 2020. Teaching with Commentaries. In *International Conference on Learning Representations*.

Ratner, A.; Bach, S. H.; Ehrenberg, H.; Fries, J.; Wu, S.; and Ré, C. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, 269. NIH Public Access.

Ratner, A. J.; De Sa, C. M.; Wu, S.; Selsam, D.; and Ré, C. 2016. Data Programming: Creating Large Training Sets, Quickly. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to Reweight Examples for Robust Deep Learning. In *International Conference on Machine Learning*, 4334–4343.

Ren, W.; Li, Y.; Su, H.; Kartchner, D.; Mitchell, C.; and Zhang, C. 2020. Denoising Multi-Source Weak Supervision for Neural Text Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3739–3754.

Saxena, S.; Tuzel, O.; and DeCoste, D. 2019. Data parameters: A new family of parameters for learning a differentiable curriculum. *NeurIPS*.

Shi, B.; Hoffman, J.; Saenko, K.; Darrell, T.; and Xu, H. 2020. Auxiliary Task Reweighting for Minimum-data Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 7148–7160. Curran Associates, Inc.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Son, W.; Na, J.; Choi, J.; and Hwang, W. 2021. Densely guided knowledge distillation using multiple teacher assistants. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9395–9404.

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, 843–852.

Vyas, N.; Saxena, S.; and Voice, T. 2020. Learning Soft Labels via Meta Learning. arXiv:2009.09496.

Xu, G.; Liu, Z.; Li, X.; and Loy, C. C. 2020. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, 588–604. Springer.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *BMVC*.

Zhao, H.; Sun, X.; Dong, J.; Dong, Z.; and Li, Q. 2021. Knowledge Distillation via Instance-level Sequence Learning. *Knowledge-Based Systems*, 233.