Unified Wisdom: Harnessing Collaborative Learning to Improve Efficacy of Knowledge Distillation

Anonymous authors
Paper under double-blind review

Abstract

Knowledge distillation (KD), which involves training a smaller *student* model to approximate the predictions of a larger *teacher* model is useful in striking a balance between model accuracy and computational constraints. However, KD has been found to be ineffective when the teacher and student models have a significant capacity gap. In this work, we address this issue via "meta-collaborative distillation" (MC-Distil), where students of varying capacities collaborate during distillation. Using a "coordinator" network (C-Net), MC-Distil enables mutual learning among students as a meta-learning task. Our insight is that C-Net learns from each student's performance and training instance characteristics, allowing students of different capacities to improve together. Our method enhances student accuracy for all students, surpassing state-of-the-art baselines, including multi-step distillation, consensus enforcement, and teacher re-training. We achieve average gains of 2.5% on CIFAR100 and 2% on TinyImageNet datasets, consistently across diverse student sizes, teacher sizes, and architectures. Notably, larger students benefiting through meta-collaboration with smaller students is a novel idea. MC-Distil excels in training superior student models under real-world conditions such as label noise and domain adaptation.

1 Introduction

Although modern deep learning methods can effectively learn from large datasets with high-dimensional features, their associated compute and memory requirements often exceed the constraints faced by practical applications. To address this, Knowledge distillation (KD (Hinton et al., 2015)) trains a smaller student model to approximate the higher quality predictions of a larger teacher model. These distilled student models often outperform supervised models of similar capacity; however, such gains are limited by the choice of teacher and student. In particular, larger teacher-student capacity gaps are observed to produce poorer quality student models (Cho & Hariharan, 2019).

Previous work on this challenge use an altered teacher (Cho & Hariharan, 2019; Li & Jin, 2022) or introduce additional "assistant" models of intermediate learning capacities between teacher and student model (Mirzadeh et al., 2020; Son et al., 2021). An interesting idea in this literature is the use of multiple teachers for a given student, providing the student with a multi-grain view of the relationship between instance input and label. However, the use of inferior or intermediate-quality teacher models in these approaches potentially limit the quality of information available to the student. We believe, instead, that students should have access to high-quality teacher signals, but focus more on instances that are "learnable" given their model capacity (see e.g., Mindermann et al. (2022) for supervised learning). To this, we add the following insight: we could obtain a finer, multi-grain picture of instance hardness by observing the performance of *students of different capacities* (as opposed to generating and utilizing multiple teachers), which could then be used for better distillation. These observations lead us to the following **problem statement:** Can students of different capacities learn from each other in a collaborative distillation framework, thereby understanding how to prioritize relevant "learnable" information and improving distillation outcomes?

Approach: We build MC-Distil – *meta-collaborative distillation* – where students of different capacities are codistilled from a single teacher. We propose to use an auxiliary meta-network for aggregating information

across students, and using it to modulate each student's learning on a per-instance basis. We term this meta-learning approach towards joint multi-student distillation as meta-collaboration. As shown in Figure 1, the auxiliary network C-Net modulates the training loss of each student through an instance-dependent reweighting of the teacher and cross-entropy losses. C-Net removes the need for intermediate teachers, and students have access to high-quality teacher signals that they can emulate on a per-instance basis. We set C-Net's objective as the pooled student accuracy on a separate, held-out dataset, setting up a nested optimization that we approximate using an efficient alternating update algorithm (see section 3 for details). A key advantage of MC-Distil is that it produces a spectrum of student models with varying capacities all with improved generalization capabilities, which can be deployed contextually at test time to suit specific application requirements.

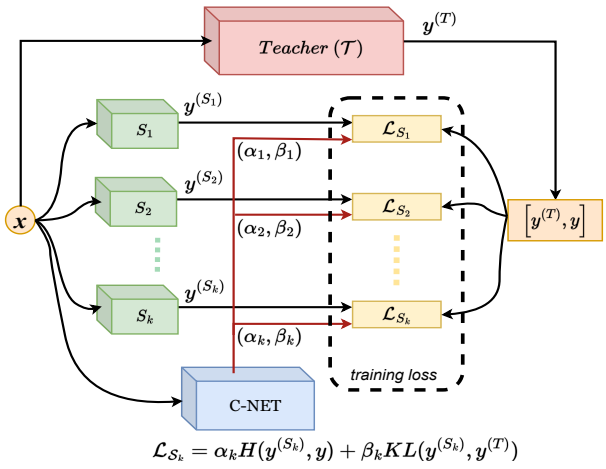


Figure 1: MC-Distil Training Workflow. Losses are weighted according to the parameters obtained from C-Net which are then used to train the individual students.

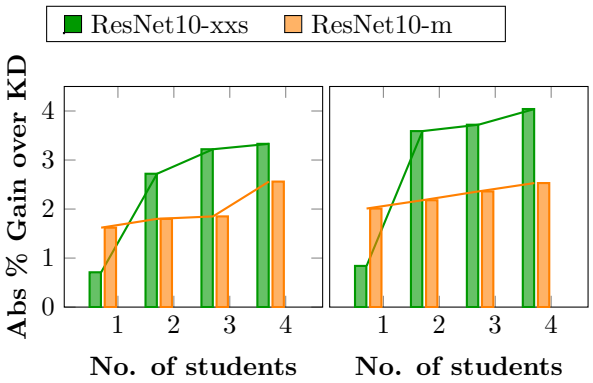
Pooling information across students: Figure 2 shows accuracy gains for 2 student models using MC-Distil compared to vanilla KD, as a function of total number of students distilled together. As more students are added, the C-Net is better able to understand data instances, and deliver better quality student models. The student pool was subsets of { ResNet10-xxs, ResNet10-xs, ResNet10-s and ResNet10-m } He et al. (2016); Kag et al. (2023). For ResNet10-xxs results, models of higher capacity are added one by one to the cohort, whereas for ResNet10-m, models of lower capacity are added sequentially. Experiments were conducted on CIFAR100 dataset Krizhevsky (2009), with Resnet10 and Resnet 18 as teacher (figure 2a,2b respectively). In appendix C.1 we present the complete result for all the students.

Modulating distillation at a (student,instance) grain: Figure 3 shows the relative weight assigned in MC-Distil to training instances as a function of instance hardness (x-axis), and student model capacity (different curves). C-Net implicitly learns instance hardness, and encourages weaker students to focus more on the easier examples, while stronger students learn from a wider range of data. Experiment details are as above; we used label likelihood gap between teacher and largest student as a measure of instance hardness (x-axis), and total weight assigned to loss components ($\alpha + \beta$) as instance weight (y-axis). Since instance hardness isn't an explicit input to C-Net, it has learned an implicit map between input instance and difficulty through the MC-Distil training process. We further inspect how choice of choosing loss component-wise weights α and β (see equation 2) helps MC-Distil in improving KD in section 4.5

We present extensive evaluation of MC-Distil on a wide range of student & teacher architectures and model sizes (section 4.3), quantifying gains against SOTA benchmarks. MC-Distil improves the performance of each student in the ensemble, highlighting the interesting finding that smaller student models can help improve larger students in collaborative distillation. Finally, we show that MC-Distil achieves better generalization and robust models (section 4.7).

2 Related works

Knowledge Distillation (KD) In supervised learning, Knowledge Distillation (KD) (Hinton et al., 2015) is a valuable method where a 'student' model learns by mimicking a pre-trained 'teacher' model, rather than solely relying on labeled data. The success of KD depends on factors like teacher model accuracy and student model capacity (Menon et al., 2021). Recent research (Harutyunyan et al., 2023) explores



(a) ResNet10 teacher (b) ResNet18 teacher

Figure 2: Influence of student cohort size on distillation gains using meta-collaboration. See text for details.

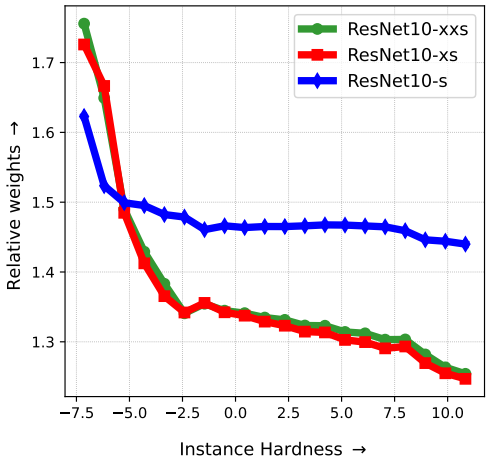


Figure 3: C-Net prioritizes training data based on student capacity and instance hardness. See text for details.

its efficacy in relation to supervision complexity. Using early-stopped teacher models has shown promise in improving student training (Cho & Hariharan, 2019), though it requires iterative distillation. Another approach (Liu et al., 2020) employs multiple teacher networks with intermediate knowledge transfer. The strategic blending of loss components (Sivasubramanian et al., 2023) has recently improved KD, particularly in scenarios with significant representation gaps between teacher and student models. The challenge of suboptimal KD performance caused by significant capacity disparities between student and teacher models was addressed by strategically choosing points to learn from the teacher model in Kag et al. (2023). Other innovations such as 'Teacher Assistants' (TAs) or intermediate models have been introduced (Mirzadeh et al., 2020), with further enhancements achieved through stochastic techniques such as Dense Gradient Knowledge Distillation (DGKD) (Son et al., 2021), which involves the simultaneous training of intermediates with occasional model dropout. Recently Li & Jin (2022) showed that the gap between online and offline KD could be bridged using a shadow head on top of the backbone of the teacher model. The shadow head is trained to perform bidirectional distillation. *These works indicate knowledge transfer to the smaller models, resulting in their improvement by the presence of the intermediate bigger models; however the generalization of the larger models has also been shown to improve from the knowledge of smaller models (Mindermann et al., 2022) and bidirectional learning to improve KD (Li & Jin, 2022). Therefore, inspired by these bidirectional signals, we present an approach to train multiple student models simultaneously and communicate vital information via a coordinator network.*

Online knowledge distillation : In the absence of a strong pre-trained teacher, online knowledge distillation often involves collaborative model efforts to acquire soft labels. (Chen et al., 2020) aggregate predictions from peer models using an attention-based mechanism to obtain soft targets. (Wu & Gong, 2021) employ an m-branch model treating each branch as a peer model, constructing soft distillation targets from weighted logits of these branches and an exponentially moving-averaged m-branch model. (Guo et al., 2020) propose using pooled student logits with varying learning capacities as soft distillation targets. Additionally, Du et al. (2023) creates a curriculum using variance in predictions of multiple students with varying sparsity levels as a measure of task complexity. In Zhang et al. (2018), uses KL-divergence loss between every student pair to improve consensus among student models. *The primary focus in these works lies either in constructing logits due to the absence of a pre-trained teacher model or explicit synchronization of the student models. In contrast, our approach centers on influencing the learning patterns of the models participating in the collaborative process via signals from peer models.*

Instance-Specific and Meta-Learning : Instance-specific learning has been greatly explored to improve training in the presence of noise. (Saxena et al., 2019; Algan & Ulusoy, 2021; Vyas et al., 2020) propose

to learn instance-specific temperature or smoothing parameters to account for potential label noise. In the context of knowledge distillation, (Zhao et al., 2021) have demonstrated the advantages of learning instance-level sequences (or curricula) for training samples. A similar instance-wise weighing scheme has been proposed to improve distillation in semi-supervised settings (Iliopoulos et al., 2022). Recent contributions such as that in (Ren et al., 2018; Shu et al., 2019a; Raghu et al., 2020; Jain & Shenoy, 2022) employ meta-learning based on validation sets to acquire instance-specific weights, enhancing robustness. *We introduce a novel approach, MC-Distil, that involves the utilization of meta-learning based on validation sets to facilitate a collaborative learning process among multiple models. To the best of our knowledge, this has not been previously explored.*

3 MC-Distil: Meta-collaborative distillation

3.1 Standard Knowledge Distillation

In supervised learning, one is concerned with training a classifier $y = f_\theta(x)$ using data $D = (x_i, y_i) \mid i \in (1, \cdots, n)$. Here, $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ denote pairings of inputs x_i along with their corresponding labels y_i and θ denote the model parameters. Typically, the labels y are cardinal, and practitioners have found that such labels are often inadequate in capturing nuances in the data. A popular mitigation is to incorporate more nuanced 'soft labels', or distributions over labels, as the target for supervision instead of cardinal labels. In particular, Knowledge distillation (KD) (Hinton et al., 2015) uses the logits from a pre-trained model (the 'teacher model') as soft labels for training a classifier, in addition to the standard cardinal labels.

Suppose a pre-trained model (teacher) generates logits denoted by $y^{(T)} = \mathcal{T}(\boldsymbol{x})$. Then a new model (student) could be trained using a "teacher matching" objective that involves minimizing the KL-divergence between $y^{(T)}$ and the student's logits $y^{(S)}$. The KD objective is as follows:

$$\mathcal{L}_s = \sum_D \Big((1-\lambda) l_{ce} + \lambda \big(\tau^2 KL\big(y^{(S)}, y^{(T)}\big)\big)\Big) \tag{1}$$

Here, $l_{ce} = H\big(y^{(S)}, y\big)$ is a supervised learning loss matching $y^{(S)}$ to the true labels y, and $l_{kd} = \tau^2 KL\big(y^{(S)}, y^{(T)}\big)$ is the teacher-matching loss. Typically, H is the standard cross-entropy loss. The hyperparameters τ, λ control the softening of the KL-divergence term, and the relative contributions of the two loss components.

3.2 MC-Distil: Multi Student Knowledge Distillation

Successful application of KD often depends on the quality of the pre-trained teacher model (see *e.g.*, (Menon et al., 2021)), and the representational gap between the teacher and student models. To address the capacity gap issue, often, several Teacher Assistants (TA) or intermediate models are introduced (Mirzadeh et al., 2020; Son et al., 2021). The success of the such multi-teacher approaches suggests that supervisory inputs from teacher models of different sizes enrich the information available to the student. We leverage this insight in a completely different setup where a single teacher is simultaneously distilled into multiple cooperating student models. Our primary thesis is that the diversity in student capacities and learning behaviors can be leveraged to tailor the distillation process more effectively for each student.

Specifically, we propose to learn adaptive, instance-wise distillation losses using a coordinator network, denoted as g_ϕ (referred to as C-Net). This network modulates the contribution of each training sample to the student models based on their individual learning dynamics and their relationship with the teacher. Formally, we define a set of student models $\mathcal{S} = S_j \mid j \in 1, \cdots, k$, a single pre-trained teacher model $\mathcal{T}$, and a shared coordinator network g_ϕ. The overall training objective for MC-Distil is then defined as:

$$\overbrace{\underset{\phi}{\operatorname{argmin}} \sum_{j=1}^{k} \mathcal{L}_{\text{C-NET}}\big(\underbrace{\underset{\theta_j}{\operatorname{argmin}}\, \mathcal{L}_{s_j}(\theta_j, \alpha_j, \beta_j)}_{inner-level}, \mathcal{V}\big)}^{outer-level} \tag{2}$$

$$\text{where } \langle \alpha_{1i}, \ldots, \alpha_{ki} \rangle, \langle \beta_{1i}, \ldots, \beta_{ki} \rangle = g_\phi(x_i), \alpha_j = \langle \alpha_{j1}, \cdots, \alpha_{jn} \rangle, \beta_j = \langle \beta_{j1}, \cdots, \beta_{jn} \rangle$$

$$\text{and } \mathcal{L}_{s_j} = \sum_{(x_i, y_i) \in D} \Big(\alpha_{ji} H(x_i, y_i) + \beta_{ji} \tau^2 KL\big(y_i^{(S)}, y_i^{(T)}\big) \Big)$$

3.2.1 Coordinator network g_ϕ (C-Net):

Our key innovation in equation 2 is to use an additional "coordinator network", which we call C-NET (Figure 1), to address two key challenges, *viz.*, a) reducing the number of learnable parameters and b) introducing interaction among the student models. The C-NET is a learnt function $(\mathrm{A_i}, \mathrm{B_i}) = g_\phi(x_i)$, where $\mathrm{A_i} = [\alpha_{1i}, \ldots, \alpha_{ki}], \mathrm{B_i} = [\beta_{1i}, \ldots, \beta_{ki}]$ represent the loss mixing parameters for the k students on the input x_i.Therefore, the combined (A, B) for the entire dataset would result in a matrix of size $n \times k$, which can incur a significant memory overhead as the training dataset grows. Thus to compactly represent the loss mixing parameters we use C-NET parameterized by $\phi \in \Phi$. Moreover, as the C-NETis a centralized mechanism, it acts as a shared interface across students and training instances, enabling the coordinator to incorporate the relative learning dynamics of models with different capacities. This allows C-NET to assign more informed, adaptive weights to each student based on instance-specific difficulty and model behavior. The training procedure for the C-NET is described in the following section.

Training the C-Net We train the g_ϕ (C-NET) using a separate validation set of data $\mathcal{V} = (x_i^v, y_i^v) \mid i \in (1, \cdots, m)$. The objective is to learn ϕ such that the resulting student models generalize well—i.e., they achieve low prediction error on the validation data. Specifically, we minimize the average cross-entropy loss across all k student models on the validation set:

$$\mathcal{L}_{\text{C-NET}} = \frac{1}{km} \sum_{j=1}^{k} \sum_{i=1}^{m} H\big(y_i^{v(S_j)}, y_i^v\big) = \frac{1}{km} \sum_{j=1}^{k} \sum_{i=1}^{m} H\big(f_{\theta_j}(x_i^v), y_i^v\big) \tag{3}$$

Here, $H(\cdot, \cdot)$ denotes the cross-entropy loss between the predicted label from student j and the ground truth. Crucially, $\mathcal{L}_{\text{C-NET}}$ depends on ϕ only implicitly — via the student parameters θ_j, which themselves are optimized using the loss mixing weights generated by $g_\phi(\cdot)$ (see Eq. 2). In other words, our proposal defines a *bi-level optimization* that encompasses both the C-NET and classifier parameters. Thus, the optimal θ values depend on the optimal choice of ϕ and vice versa. For most modern deep learning models, solving the inner optimization in Eq. equation 2 in closed form is intractable due to the non-convexity and scale of the models. To address this, we adopt an iterative strategy to approximate the solution to the bilevel optimization problem. Rather than fully optimizing the inner objective before updating the outer parameters, we employ an alternating stochastic gradient descent (SGD) approach. This involves interleaving updates of the student model parameters and the coordinator network. Specifically, we perform a few gradient steps on the student parameters using the training set and the weights predicted by the coordinator, followed by an update of the coordinator parameters based on validation loss(equation 5). This alternating scheme provides a practical and efficient means of optimizing the overall objective. The update steps are summarized as follows:

$$\theta_j^{t+1} = \theta_j^t - \frac{\eta_1^j}{n} \sum_{i=1}^{n} g^{\alpha}_{\phi^t i}[j] * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big) + g^{\beta}_{\phi^t i}[j] * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \quad \forall j \in \{1, \cdots, k\} \tag{4}$$

$$\phi^{t+1} = \phi^t - \frac{\eta_2}{m} \sum_{i=1}^{m} \nabla_{\phi^t} \mathcal{L}_{\text{C-NET}}(x_i^v, y_i^v, \Theta^{t+1}) = \phi^t - \frac{\eta_2}{km} \sum_{j=1}^{k} \sum_{i=1}^{m} \nabla_{\theta_j^{t+1}} \mathcal{L}_{\text{C-NET}}(x_i^v, y_i^v, \Theta^{t+1}) * \nabla_{\phi^t} \theta_j^{t+1} \tag{5}$$

Here, we denote $\Theta^{t+1} = \langle \theta^{t+1}1, \cdots, \theta^{t+1}k \rangle$ as the collection of updated student model parameters. The learning rates for the student models and the C-NET are represented by $\eta_1^1, \cdots, \eta_1^k$ and η_2, respectively. We

Algorithm 1 The MC-Distil approach: learning student $\mathcal{S}_1, \cdots \mathcal{S}_k$, Training data $\mathcal{D}$, Validation data $\mathcal{V}$, teacher $\mathcal{T}$ and C-Net g_ϕ.

Hyperparameters: τ Temperature, $\eta_1^1 \cdots \eta_1^k$: learning rates for k students, η_2 learning rate for C-Net, L epoch interval for C-Net update

Initialize student model parameters with $\theta_1^0 \cdots \theta_k^0$ and C-Net with g_ϕ^0
for $t \in \{0, \ldots, T\}$ **do**
  for $j \in \{1, \ldots, k\}$ **do**
    Update θ_j^{t+1} by Equation 4.
  end for
  if $t\%L == 0$ **then**
    $\{x^v, y^v\} \leftarrow$ `SampleMiniBatch`$(\mathcal{V})$
    Compute $\mathcal{L}_{\text{C-Net}}$ using $\{x^v, y^v\}$ and (θ_j^{t+1}) as described in Equation 3.
    Update $\phi^{\lfloor \frac{t}{L} \rfloor + 1}$ by Eq. Equation 5.
  end if
end for

use the notations $g^{\alpha}_{\phi^t i}[j] = \alpha^i_j$ and $g^{\beta}_{\phi^t i}[j] = \beta^i_j$ to remind our readers that αs and βs are output from a model ($g(.)$) parameterised by ϕ. The update step for the C-Net is similar to the standard meta-learning objectives as it uses the updated student model parameters. We present the complete algorithm of MC-Distil in Algorithm 1. Since training C-Net adds to the cost of training, we propose to update C-Net only after L epochs. In Appendix A, we show theoretically that our method converges to the optima of both the validation and training loss functions under some mild conditions.

4 Experiments

4.1 Model Architecture and Training

To demonstrate the utility of our method across groups of different model sizes we experiment with a group of ResNet (He et al., 2016) models and several recent larger models. We use the ResNet32 model as C-Net with the classification head changed to output weighting parameters. We present results of varying C-Net size in Appendix sec. C.2. We use ResNet10-xxxs, ResNet10-xxs, ResNet10-xs, ResNet10-s and ResNet10-m (Kag et al., 2023) models as the student models. These models are simultaneously trained with either ResNet-10L, ResNet-10, ResNet-18 or ResNet-34 models as a teacher model. In Section B.2 we present details of these models. We present experiment results on these combinations in Table 1. To illustrate the utility of our method in larger vision models we perform knowledge distillation with ResNet-32x4 as the teacher and ResNet-8x4, ShuffleNet-V2 (Ma et al., 2018), WideResNet-16x2 (Zagoruyko & Komodakis, 2016) and MobileNet-V2x2 (Sandler et al., 2018) as the group of student models. We also perform knowledge distillation with WideResNet-40x2 as a teacher model and ResNet-8x4, ShuffleNet-V2, WideResNet-40x1, and MobileNet-V2x2 as a student model group in the large vision model setting. The results of these experiments are presented in Table 2.

We train the student models for 500 epochs and update C-Net every 20 epoch (*i.e.*, L = 20). Other training-related details are presented in Section B.3. We perform all our experiments on the **CIFAR-100** (Krizhevsky, 2009), **Tiny-ImageNet** (Le & Yang, 2015) and **ImageNet** (Russakovsky et al., 2015) datasets. Details of train-val-test splits, input dimensions, and the augmentations used on the input to model are presented in Appendix B.1.

4.2 Baselines

In our comparative analysis, we assess the performance of our method against a selection of recent works in the field of knowledge distillation, alongside standard knowledge distillation and Empirical Risk Minimization (ERM) or Cross-Entropy based training (CE). Specifically, we compare against TAKD (Mirzadeh et al., 2020)

CIFAR100 Test Accuracies											
Teacher		**Student**	**CE**	**KD**	**TAKD**	**DGKD**	**RMC**	**DML**	**SHAKE**	**MetaDistil**	**Ours**
ResNet10-l	72.2	ResNet10-xxs	31.85	33.45	34.39	35.34	34.07	33.41	33.56	34.92	**36.19**
		ResNet10-xs	42.75	44.87	44.97	47.11	45.18	44.16	42.12	46.01	**47.57**
		ResNet10-s	52.48	55.38	56.16	57.02	53.74	55.13	56.5	57.2	**58.36**
		ResNet10-m	64.28	66.93	-	-	66.66	66.06	68.8	68.28	**69.42**
ResNet10	75.18	ResNet10-xxs	31.85	33.95	34.98	34.85	33.64	33.54	33.7	34.66	**35.97**
		ResNet10-xs	42.75	44.87	45.64	46.68	42.45	44.7	44.2	46.32	**47.53**
		ResNet10-s	52.48	55.56	56.51	56.84	53.64	55.29	56.59	57.78	**58.1**
		ResNet10-m	64.28	67.27	-	-	66.58	66.25	68.63	68.89	**69.21**
ResNet18	76.99	ResNet10-xxs	31.85	33.56	34.26	34.26	33.77	34.02	32.32	34.4	**35.94**
		ResNet10-xs	42.75	45.02	45.27	47.33	45.14	44.1	43.82	46.24	**46.99**
		ResNet10-s	52.48	55.73	55.41	56.7	54. 03	54.82	56.82	57.4	**57.5**
		ResNet10-m	64.28	66.42	-	-	66.04	65.94	68.12	68.43	**68.45**
ResNet34	79.47	ResNet10-xxs	31.85	33.32	34.46	35.64	34.46	33.68	33.2	33.76	**36.1**
		ResNet10-xs	42.75	44.94	45.92	47.21	42.78	44.45	45.8	46.43	**47.12**
		ResNet10-s	52.48	54.73	56.17	57.12	53.58	55.28	56.2	56.91	**57.67**
		ResNet10-m	64.28	66.52	-	-	65.58	66.88	68.96	68.09	**68.24**
Tiny-ImageNet Test Accuracies											
ResNet10-l	41.25	ResNet10-xxs	13.76	13.53	13.81	14.34	13.69	13.95	14.7	14.78	**15.48**
		ResNet10-xs	18.56	19.19	19.22	20.54	19.04	19.57	19.92	20.13	**21.68**
		ResNet10-s	24.56	25.95	26.35	27.24	25.86	26.2	27.0	27.06	**29.74**
		ResNet10-m	33.47	34.63	-	-	33.72	35.34	36.66	36.02	**38.26**
ResNet10	44.04	ResNet10-xxs	13.76	13.8	14.01	14.52	13.83	14.01	14.2	14.19	**15.36**
		ResNet10-xs	18.56	19.48	19.09	21.21	19.28	19.78	19.87	20.13	**21.78**
		ResNet10-s	24.56	26.95	25.58	26.99	26.18	27.04	27.79	27.06	**29.82**
		ResNet10-m	33.47	35.5	-	-	34.78	35.79	38.01	36.02	**38.54**
ResNet18	47.94	ResNet10-xxs	13.76	14.12	14.53	13.87	14.08	14.28	14.58	14.24	**15.26**
		ResNet10-xs	18.56	19.78	19.35	19.54	19.75	18.24	19.85	19.96	**21.36**
		ResNet10-s	24.56	26.3	26.17	27.42	25.08	25.92	27.92	27.32	**30.36**
		ResNet10-m	33.47	35.08	-	-	33.37	36.77	38.6	36.08	**39.11**
ResNet34	50.1	ResNet10-xxs	13.76	14.43	13.47	14.58	13.78	14.3	14.67	13.96	**15.24**
		ResNet10-xs	18.56	19.72	18.33	20.84	19.28	19.29	19.76	20.93	**21.82**
		ResNet10-s	24.56	27.05	24.96	27.89	25.99	26.81	28.05	27.64	**29.59**
		ResNet10-m	33.47	35.94	-	-	33.58	35.88	37.97	36.88	**38.34**

Table 1: Comprehensive comparison of methods across datasets. For each teacher model, we perform knowledge distillation with a group of student models. MC-Distil(Ours, last column) substantially improves the average accuracy on unseen test data. compared to other distillation baselines especially the ones designed to take advantage of a multi-student setup. The highest accuracies are highlighted in bold.

and DGKD (Son et al., 2021) baselines involving multiple students or intermediate models, RMC(Du et al., 2023) uses variance in predictions of students of levels of sparsity as a measure of task complexity for each instance, DML (Zhang et al., 2018) forces collaboration among the models by introducing a KL-divergence loss across different models and SHAKE (Li & Jin, 2022) which introduces a pseudo teacher that allows bidirectional learning during KD. We also compare MC-Distil against one more baseline that involves distilling knowledge to each of the students independently using a network architecturally similar to C-Net. We refer to this baseline as the **MetaDistil**. This baseline is similar to AMAL (Sivasubramanian et al., 2023); the strategic mixing loss components are achieved via the C-Net optimization. Further, in instance dependent label-noise setting we compare against L2R (Ren et al., 2018) and MWN (Shu et al., 2019b) reweighing techniques created for eliminating label noise by using bi-level optimization and MCD (Gal &

CIFAR100 Test Accuracies											
Teacher		**Student**	**CE**	**KD**	**TAKD**	**DGKD**	**RMC**	**DML**	**SHAKE**	**MetaDistil**	**Ours**
ResNet-32x4	80.1	ResNet-8x4	71.12	72.62	74.26	74.45	73.89	73.37	73.96	73.12	**75.38**
		MobileNetV2x2	69.52	71.78	72.07	72.27	71.23	72.46	72.17	72.53	**73.05**
		WideResNet16x2	72.79	73.34	73.72	74.12	75.19	73.87	74.22	74.1	**75.52**
		ShuffleNet-V2	73.73	75.33	-	-	76.52	75.98	76. 12	76.74	**77.97**
WideResNet40x2	77.6	ResNet-8x4	71.12	72.77	73.82	74.63	73.84	74.12	74.31	74.49	**76.73**
		MobileNetV2x2	69.52	72.69	72.69	71.61	70.90	73.97	**74.68**	74.20	74.36
		WideResNet40x1	72.90	73.01	73.72	74.67	75.44	74.26	74.41	75.38	**75.85**
		ShuffleNet-V2	73.73	75.85	-	-	76.72	77.87	78.08	**78.17**	77.94
Tiny-ImageNet Test Accuracies											
ResNet-32x4	50.2	ResNet-8x4	37.16	37.23	38.83	39.52	36.76	38.86	38.12	40.46	**41.89**
		MobileNetV2x2	47.68	49.89	49.89	48.21	48.07	48.24	49.79	49.85	**49.95**
		WideResNet16x2	39.11	39.47	41.66	41.77	39.77	41.35	41.98	42.12	**43.88**
		ShuffleNet-V2	47.76	50.44	-	-	49.46	49.24	49.46	50.62	**52.38**

Table 2: Comprehensive comparison of methods when training models with larger learning capacity. Here again MC-Distil(Ours, last column) substantially improves the test accuracy compared to other distillation baselines even in this setting.

Ghahramani, 2016) which uses dropout to model uncertainties. Please refer to Sec. B.4 for more information on these methods.

4.3 Improving efficacy of Knowledge Distillation

In Table 1, we present results from experiments conducted on CIFAR100 and TinyImagenet datasets, exploring scenarios with a significant capacity gap between teacher and student models. We start with ResNet10-l as the teacher and go on to continue the increasing learning capacity of the teacher model and perform knowledge distillation with ResNet10, ResNet18, and ResNet34. On both the datasets, MC-Distil, by virtue of meta-collaboration, achieves the best performance among the baselines showing accuracy gains of up to 4% on both the datasets compared to KD. The gains are much more pronounced on the larger models in the student pool for the TinyImagenet dataset owing to the increased difficulty in classifying it whereas for CIFAR100 the gains are pretty uniform across the student models. These gains are consistent across a wide range of student and teacher capacities. MC-Distil improves all of the student model's performances as compared to the baselines, thereby showing that joint distillation of knowledge to a student set is beneficial for both smaller and larger students. We analyze the effect of introducing student models one by one in the presence of C-Net in Appendix sec.C.1.

MC-Distil remains competitive even in scenarios with a small capacity difference. As illustrated in Table 2, MC-Distil maintains its competitive advantage over KD, even when the student model closely matches the size of the teachers, achieving gains of up to 3% relative to KD. These improvements can be attributed to two key factors: (i) the reweighing of loss terms and (ii) meta-collaboration. The reweighting strategy aligns the learning focus with the model's capacity, emphasizing examples that are more suitable for the student. This targeted weighting helps explain the performance improvement observed with MetaDistil over standard KD, particularly for students like 'ResNet-m'. However, just this reweighing is not sufficient for students such as 'ResNet-xxs' and 'Resnet-s' in the case of larger teachers. This is where MC-Distil's ability to leverage C-Net as a communication channel among student models is useful in enhancing knowledge transfer from the teacher model. While the benefits of information flow from intermediate models to smaller ones to improve final performance, as demonstrated in previous studies (Son et al., 2021; Mirzadeh et al., 2020), are well-established, the reverse scenario has been under-explored. The gains reported in Tables 1 and 2 indicate that larger models can also benefit from information exchange with smaller models, akin to standard supervised settings (Mindermann et al., 2022).

Teacher		Student	CE	KD	DGKD	SHAKE	MetaDistill	MC-Distil
ResNet101	81.68	ResNet14	22.28	19	20.92	21.15	22.84	**23.79**
		ResNet20	26.87	22.1	24.31	24.48	26.44	**27.94**
		ResNet32	31.04	26.83	29.34	29.26	31.56	**33.07**
		ResNet56	37.56	33.38	-	34.82	37.9	**39.73**

Table 3: Comprehensive comparison ImageNet test accuracies of student models distilled from a ResNet101 teacher using different knowledge distillation techniques. The proposed method (MC-Distil) consistently outperforms baseline approaches across all student architectures.

MC-Distil outperforms bidirectional learning methods in KD. We compare against two KD methods viz. SHAKE's (Li & Jin, 2022) and DML (Zhang et al., 2018), both aim to enhance teacher outputs to improve KD's effectiveness. SHAKE learns a pseudo-teacher (a few additional learnable layers on the teacher's backbone) to adapt according to the student model. This performs relatively well for larger students but struggles to accommodate the specific needs of smaller students in Table 1. In contrast, MC-Distil carefully controls the knowledge transfer from the teacher model, proving more beneficial for smaller students. On the other hand, DML's forceful explicit collaboration among student cohorts results in inferior performances. This is because aligning outputs while stops the larger models from learning to their full capacities, burdens smaller models to match their outputs to their larger cohorts (like in traditional KD). In Appendix 4.9 we show that such collaboration can be added to our setup and if carefully reweighted can lead to improved performances.

4.4 MC-Distil in extreme teacher student gap settings

To evaluate the resilience of MC-Distil in scenarios with a significant teacher-student capacity gap, we conduct knowledge distillation using ResNet101 as the teacher model and ResNet14, ResNet20, ResNet32, and ResNet56 as student models on the ImageNet dataset. Due to computational constraints, we compare only against the most competitive baselines: DGKD, SHAKE, and MetaDistill.

Given ImageNet's complexity and the large teacher-student disparity, KD (Knowledge Distillation) underperforms even compared to standard Cross-Entropy (CE) training, aligning with prior findings from Cho & Hariharan (2019). The alternative methods designed to mitigate poor transfer (such as DGKD and SHAKE) provide some improvements over KD, they still fail to surpass CE, with the exception of MetaDistill, which only matches CE performance. Despite these challenges, MC-Distil achieves a consistent 1-2% improvement over CE, demonstrating its effectiveness even in extreme teacher-student gap scenarios.

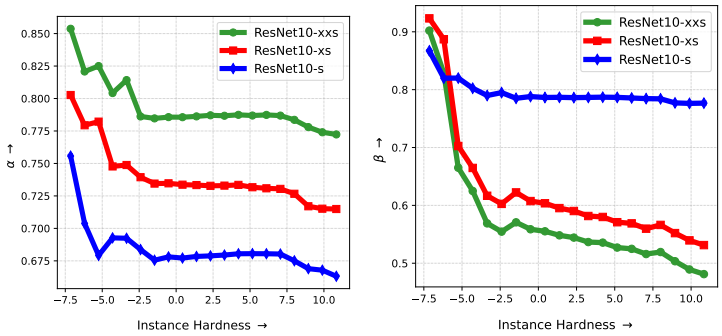
4.5 Why MC-Distil uses instance and loss component specific weights

We expand on the analysis presented in Figure 3 to understand how C-Net assigns weights to individual loss components. Figure 4a shows how αs, parameter controlling learning from the hard label and Figure 4b shows how βs, parameter controlling learning from the teacher model (see equation 2) changes with instance hardness for models with different learning capacity. For the smaller student models, learning majorly happens via hard labels, whereas larger models prefer learning from the teacher's nuanced soft labels. KD improves the label student model's performance by offering nuanced soft labels as targets. However, when the student model cannot encode these nuanced soft labels, their introduction can worsen the student model's performance Cho & Hariharan (2019). Therefore, choosing a favorable ground truth is also important along with focusing on the "learnable" points. MC-Distil achieves this by using two sets of weighing parameters- αs and βs and carefully turning it according to student model capacity. For the smaller models, MC-Distil carefully introduces soft teacher labels of only the easy points, and thus MC-Distil improves the smaller model's performance over traditional KD.

Teacher		Student	CE	KD	DGKD	DML	SHAKE	Meta-Distill	MC-Distil
RN10-l	89.05	RN10-xxs	75.4	75.92	76.17	76.12	76.72	76.85	**76.96**
		RN10-xs	77.3	77.37	77.86	77.5	78.3	78.27	**78.82**
		RN10-s	79.97	80.42	80.3	80.51	81.79	81.94	**82.8**
		RN10-m	84.77	84.94	-	85.80	86.91	86.52	**87.35**
RN10	91.86	RN10-xxs	75.4	75.85	76.2	76.17	76.25	76.82	**76.94**
		RN10-xs	77.3	77.45	77.65	77.72	78.38	78.04	**78.97**
		RN10-s	79.97	80.49	80.58	80.61	82.19	81.73	**82.75**
		RN10-m	84.77	85.07	-	85.28	86.34	86.44	**87.49**

Table 4: Results of performing KD with challenging iWildCam Dataset (Beery et al., 2020). Notably, MC-Distil demonstrates significant improvements in test accuracy compared to other distillation baselines.

4.6 MC-Distil trains models with better generalisation



(a) α vs Instance hardness (b) β vs Instance hardness

Figure 4: Plot between student model errors and weighting parameters while KD is performed using ResNet-18 as the teacher model on CIFAR-100 dataset.

We perform knowledge distillation on the challenging iWildCam dataset (Beery et al., 2020). The challenge arises from variations in the capturing environment, resulting in a demanding test set (see appendix B.1). Table 4 shows the performance of various knowledge distillation methods based on multi-student setups, including MC-Distil. Across both weak (RN10-l) and strong (RN10) teachers, and for a wide range of student model capacities—from extremely small (RN10-xxs) to moderately sized (RN10-m)—MC-Distil consistently achieves superior test accuracies. The consistent performance of MC-Distil highlights its robustness in leveraging instance-level importance through soft selection, enabling better generalization in complex, real-world scenarios such as iWildCam. This reinforces the effectiveness of our method not only in standard benchmarks but also under real-world distribution shifts and limited model capacities.

Using models trained with Resnet-18 as the teacher on the TinyImagenet dataset (reported in Table 1), we classify TinyImageNet-C (Hendrycks & Dietterich, 2019a). Table 6 presents the accuracies obtained by models trained with different KD methods across three levels of corruption (details in B.1). In Appendix appendix C.3 we present the complete table with results obtained across all five corruption levels. Notably, MC-Distil produces models that demonstrate robustness and superior generalization, even when the test data distribution differs from the training set.

4.7 KD in Instance Dependent Label-Noise

We now evaluate MC-Distil under a setting where instance-dependent label noise is present in the labels. We illustrate this experiment on two datasets, namely. *instance CIFAR-100* (Xia et al., 2020a) and *Clothing-1M* Xiao et al. (2015b) where the labels inherit noise due to the data generation process as against randomly flipping the labels. The results are reported in Table 5. Since MC-Distil employs a reweighting scheme, C-Net learns weights to reduce the impact of the noise. For the inputs with high noise, it is able to assign low weight to CE loss and rely more heavily on the distillation term, thereby outperforming other methods.

Student	CE	KD	MCD	MWN	L2R	Meta-Distil	MC-Distil
Inst. CIFAR-100 Accuracies							
RN10xxxs	23.69	23.67	24.33	24.14	24.88	26.48	**27.07**
RN10xxs	28.97	28.55	29.81	28.66	29.04	29.59	**31.88**
RN10xs	38.04	38.19	38.88	37.34	39.39	**42.57**	42.18
RN10s	48.4	49.22	49.71	45.84	48.14	52.67	**53.26**
RN10m	59.42	60.14	61.62	55.74	60.9	60.64	**61.61**
Clothing-1M Accuracies							
RN10xxxs	43.56	44.58	45.65	41.72	45.72	46.08	**46.43**
RN10xxs	51.67	51.28	52.22	48.74	51.6	**52.88**	52.58
RN10xs	57.14	57.16	57.51	53.49	56.7	57.06	**58.28**
RN10s	61	62.02	61.59	48.74	60.34	61.31	**62.27**
RN10m	64.83	66.55	64.86	61.21	63.45	63.64	**64.92**

Table 5: Assessment of performances of de-noising techniques and MC-Distil on a diverse set of student models, employing ResNet-18 as the teacher model on datasets with instance-specific label noise.

Student	CE	KD	MCD	DGKD	DML	SKAKE	Meta-Distill	MC-Distil
Corruption level 1 Accuracies								
RN10-xxs	11.25	11.38	10.61	10.77	9.99	11.47	11.02	**12.05**
RN10-xs	14.68	15.82	14.67	15.43	15.51	16.56	16.02	**16.73**
RN10-s	20.88	21.97	21.94	23.18	20.29	22.58	23.97	**24.88**
RN10-m	26.91	28.56	27.08	29.56	28.09	30.40	30.43	**31**
Corruption level 2 Accuracies								
RN10-xxs	10.65	10.79	9.54	10.42	9.44	10.88	10.52	**11.3**
RN10-xs	13.75	14.6	13.72	14.60	14.56	15.42	14.78	**15.52**
RN10-s	19.28	20.64	20.46	21.26	18.74	20.74	21.93	**23.04**
RN10-m	24.76	25.87	24.36	26.97	26.20	27.94	27.39	**28.67**

Table 6: Comprehensive table presenting an assessment of model (trained with TinyImageNet in Table1) robustness by conducting inference on the TinyImagenet-C dataset (Hendrycks & Dietterich, 2019a) across different degrees of image corruption, alongside results for all baseline models. The columns indicate the method employed for training the model on the TinyImagenet dataset, utilizing Resnet-18 as the teacher model.

Furthermore, the joint distillation (by communication via C-Net) of different-sized student models provides additional valuable information to combat noise effectively.

4.8 Ablation: Changing size of the validation data

	Validation set size		
Student	2%	5%	10%
ResNet10-xxs	36.05	36.22	36.54
ResNet10-xs	47.23	47.53	47.72
ResNet10-s	57.73	57.91	58.00
ResNet10-m	68.93	68.95	69.01

Table 7: Performance of MC-Distil with different validation dataset sizes. Here KD is performed using ResNet-18 as the teacher model on CIFAR-100 dataset.

We allocate 10% of the training data as a validation set, as detailed in appendix B.1. However, to demonstrate that MC-Distil is not heavily dependent on large validation sets, we conduct additional experiments with reduced validation set sizes—specifically 2% and 5% of the training data. The corresponding test accuracies are reported in Table 7, where ResNet-18 is used as the teacher model for distillation on the CIFAR-100 dataset.

As shown, MC-Distil maintains stable and competitive performance across all student models even with significantly smaller validation sets. This indicates that the method is robust and can effectively learn

meaningful instance-level weights without requiring requiring supervision from large validation corpora. The performance gains reported in Tables 1 and 2 can therefore be achieved with minimal validation overhead, making MC-Distil practical and scalable for real-world applications with limited validation data.

4.9 Incorporating online distillation techniques in MC-Distil

Consensus across students: To further encourage information sharing across students, we propose an additional term in the loss function eq. (2). This term minimizes the KL divergence between each student's logits and a *consensus* representation of logits across students (a representation we call the "Pooled Student"). For a training instance $(x, y) \in D$ such that the true label is c, *i.e.*, $y[c] = 1$, we first define the PooledStudent logits for each label l:

$$y^{(PS)}[l] = \begin{cases} \max\left(y^{(S_1)}[l], \cdots y^{(S_k)}[l]\right), & \text{if } l = c \\ \min\left(y^{(S_1)}[l], \cdots y^{(S_k)}[l]\right), & \text{otherwise} \end{cases} \tag{6}$$

This is similar to MinLogit introduced in (Guo et al., 2020). Using the PooledStudent logits, the loss function in eq. (2) is updated to:

$$\mathcal{L}_{s_j} = \frac{1}{n}\sum_{i=1}^{n} \alpha_{ij} H\left(y_i^{(S_j)}, y_i\right) + \beta_{ij}\tau^2 KL\left(y_i^{(S_j)}, y_i^{(T)}\right) + \gamma_{ij}\tau^2 KL\left(y_i^{(S_j)}, y_i^{(PS)}\right) \quad \forall j \in \{1, \cdots, k\} \tag{7}$$

Teacher	Student	MC-Distil	MC-Distil-PS
ResNet10-l	ResNet10-xxs	36.29	**36.44**
	ResNet10-xs	47.14	**47.65**
	ResNet10-s	57.37	**57.55**
	ResNet10-m	68.47	**68.51**
ResNet10	ResNet10-xxs	36.57	**36.78**
	ResNet10-xs	47.55	**48.03**
	ResNet10-s	57.59	**58.2**
	ResNet10-m	69.03	**69.5**

Table 8: We present the analysis of the effect of adding a loss component based on the pooled student to MC-Distil. Here MC-Distil-PS represents the test result obtained with model trained with pooled student-based loss component.

Correspondingly, the C-Net outputs are extended to be $(\mathrm{A}, \mathrm{B}, \Gamma) = g_\phi(x)$. Putting it all together, MC-Distil minimizes the total loss over all students, *i.e.*, $\mathcal{L}_s = \sum_j \mathcal{L}_{s_j}$. We investigate the impact of incorporating a loss component derived from the pooled student, as introduced in equation 6, in Table 8. The test results labeled MC-Distil-PS correspond to the model trained with the pooled student-based loss component. It is noteworthy that integrating this new logit proves beneficial in enhancing the performance gains achieved through meta-collaborative learning when applied in a controlled manner, as suggested in equation 6. This approach doesn't enforce explicit collaboration among the students; rather, it serves as an additional source of knowledge. Some students, particularly those finding learning directly from the teacher challenging, may leverage this supplementary knowledge, contributing to improved performance.

5 Conclusion

We introduced MC-Distil, a novel knowledge distillation framework based on meta-collaboration, or learning-to-collaborate. The collaborative learning among a group of student models is facilitated by a coordinating network C-Net. This allows models to benefit from the learning dynamics of their cohorts without any explicit compulsion to mimic each other, while also leveraging teacher model signals. Therefore MC-Distil produces a plethora of superior models that could be used according to deployment constraints and needs. Extensive experiments with various teacher-student combinations and datasets show MC-Distil consistently outperforms state-of-the-art distillation methods, even with noisy datasets or when there are distributional shifts between the training and test data.

References

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

Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020.

Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3430–3437, Apr. 2020. doi: 10.1609/aaai.v34i04.5746. URL https://ojs.aaai.org/index.php/AAAI/article/view/5746.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4794–4802, 2019.

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1766–1778, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.129. URL https://aclanthology.org/2023.eacl-main.129.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/gal16.html.

Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11020–11029, 2020.

Hrayr Harutyunyan, Ankit Singh Rawat, Aditya Krishna Menon, Seungyeon Kim, and Sanjiv Kumar. Supervision complexity and its role in knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8jU7wy7N7mA.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019a. URL https://openreview.net/forum?id=HJz6tiCqYm.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019b.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Fotis Iliopoulos, Vasilis Kontonis, Cenk Baykal, Gaurav Menghani, Khoa Trinh, and Erik Vee. Weighted distillation with unlabeled examples. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=M34VHvEU4NZ.

Nishant Jain and Pradeep Shenoy. Selective classification using a robust meta-learning approach, 2022.

Anil Kag, Durmus Alp Emre Acar, Aditya Gangrade, and Venkatesh Saligrama. Scaffolding a student to instill knowledge. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=N4K5ck-BTT.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lujun Li and Zhe Jin. Shadow knowledge distillation: Bridging offline and online knowledge transfer. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=prQT0gN81oG`.

Yuang Liu, Wei Zhang, and Jun Wang. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *NeurIPS*, 2013.

Aditya K Menon, Ankit Singh Rawat, Sashank Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7632–7642. PMLR, 18–24 Jul 2021.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pp. 15630–15649. PMLR, 2022.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Aniruddh Raghu, Maithra Raghu, Simon Kornblith, David Duvenaud, and Geoffrey Hinton. Teaching with commentaries. In *International Conference on Learning Representations*, 2020.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. *NeurIPS*, 2019.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019a.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019b.

Durga Sivasubramanian, Ayush Maheshwari, AP Prathosh, Pradeep Shenoy, and Ganesh Ramakrishnan. Adaptive mixing of auxiliary losses in supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9855–9863, 2023.

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

Nidhi Vyas, Shreyas Saxena, and Thomas Voice. Learning soft labels via meta learning, 2020.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pp. 196–202. Springer, 1992.

Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10302–10310, May 2021. doi: 10.1609/aaai.v35i12.17234. URL `https://ojs.aaai.org/index.php/AAAI/article/view/17234`.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020a.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise, 2020b.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2691–2699, 2015a. doi: 10.1109/CVPR.2015.7298885.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2691–2699, 2015b.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4320–4328, 2018.

Haoran Zhao, Xin Sun, Junyu Dong, Zihe Dong, and Qiong Li. Knowledge distillation via instance-level sequence learning. *Knowledge-Based Systems*, 233, 2021.

Appendix

A Convergence

In this section, we lay out the complete proof of theorems. Our proof technique is inspired partly by the prior literature (Shu et al., 2019a).

Lemma 1 *Suppose the meta loss function is Lipschitz smooth with constant L, and C-Net$(\cdot)$ is differential with a δ-bounded gradient and twice differential with its Hessian bounded by $\mathcal{B}$, and the loss function $\mathcal{L}$ have ρ-bounded gradients with respect to training/metadata. Then the gradient of ϕ with respect to meta loss is Lipschitz continuous.*

Proof From equation 3

$$\mathcal{L}_{\text{C-Net}} = \frac{1}{km}\sum_{j=1}^{k}\sum_{l=1}^{m} H\big(y_l^{v(S_j)}, y_l^v\big) = \frac{1}{km}\sum_{j=1}^{k}\sum_{l=1}^{m} H\big(f_{\theta_j}(x_l^v), y_l^v\big) \tag{8}$$

The gradient of ϕ with respect to meta loss can be written as:

$$\begin{aligned}\nabla_\phi \mathcal{L}_{\text{C-NET}_l} &= \frac{1}{k}\sum_{j=1}^{k} \nabla_{\theta_j} H\big(f_{\theta_j}(x_l^v), y_l^v\big) * \nabla_\phi \theta_j \\ &= \frac{1}{k}\sum_{j=1}^{k} -\frac{\eta_1^j}{n}\sum_{i=1}^{n} \nabla_{\theta_j} H\big(f_{\theta_j}(x_l^v), y_l^v\big) \\ &* \nabla_\phi\Bigg(g_{\phi^t}^{\alpha}[j] * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big) + g_{\phi^t}^{\beta}[j] * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Bigg)\end{aligned} \tag{9}$$

Let $G_{li}^H = \nabla_{\theta_j} H\big(f_{\theta_j}(x_l^v), y_l^v\big) * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big)$ and $G_{li}^{KL} = \nabla_{\theta_j} H\big(f_{\theta_j}(x_l^v), y_l^v\big) * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big)$.Then again taking gradients on both sides w.r.t ϕ,

$$\nabla_\phi^2 \mathcal{L}_{\text{C-NET}_l} = \frac{1}{k}\sum_{j=1}^{k} -\frac{\eta_1^j}{n}\sum_{i=1}^{n}\Bigg[\nabla_\phi G_{li}^H * \nabla_\phi g_\phi^\alpha + G_{li}^H * \nabla_\phi^2 g_\phi^\alpha \nabla_\phi + \nabla_\phi G_{li}^{KL} * \nabla_\phi g_\phi^\beta + G_{li}^{KL} * \nabla_\phi^2 g_\phi^\beta\Bigg] \tag{10}$$

For the first term on the right-hand side, we have that,

$$\begin{aligned}&\Big\|\nabla_\phi G_{li}^H * \nabla_\phi g_\phi^\alpha\Big\| \le \delta\Big\|\nabla_\phi \nabla_{\theta_j} H\big(f_{\theta_j}(x_l^v), y_l^v\big) * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big)\Big\| \\ &= \delta\Bigg\| - \frac{\eta_1^j}{n}\sum_{i=1}^{n} \nabla_{\theta_j}^2 H\big(f_{\theta_j}(x_l^v), y_l^v\big) \\ &* \nabla_\phi\Bigg(g_{\phi^t}^{\alpha}[j] * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big) + g_{\phi^t}^{\beta}[j] * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Bigg) * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big)\Bigg\|\end{aligned} \tag{11}$$

since $\Big\|\nabla_{\theta_j}^2 H\big(f_{\theta_j}(x_l^v), y_l^v\big)\Big\| \le L, \Big\|\nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big)\Big\| \le \rho, \Big\|\nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Big\| \le \rho, \Big\|\nabla_\phi g_\phi^\alpha\Big\| \le \delta$ and $\Big\|\nabla_\phi g_{\phi^t}^\beta\Big\| \le \delta$,

$$\begin{aligned}&\delta\Bigg\| - \frac{\eta_1^j}{n}\sum_{i=1}^{n} \nabla_{\theta_j}^2 H\big(f_{\theta_j}(x_l^v), y_l^v\big) * \nabla_\phi\Bigg(g_{\phi^t}^{\alpha}[j] * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big) + g_{\phi^t}^{\beta}[j] * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Bigg) \\ &* \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big)\Bigg\| \le \delta\Bigg\| - \frac{\eta_1^j}{n}\sum_{i=1}^{n} L * \Big(\delta * \rho + \delta * \rho * \tau^2\Big) * \rho\Bigg\| \le \eta_1^{tj} L\rho^2\delta^2(1+\tau^2)\end{aligned} \tag{12}$$

Similarly,

$$\Big\|\nabla_\phi G_{li}^{KL} * \nabla_\phi g_\phi^\beta\Big\| \le \eta_1^{tj} L\rho^2\delta^2(1+\tau^2) \tag{13}$$

And for the second term on the right-hand side in Equation equation 10 we have,

$$\|G_{li}^H * \nabla_\phi^2 g_\phi^\alpha \nabla_\phi\| \le \mathcal{B} * \delta^2 \tag{14}$$

Similarly,

$$\|G_{li}^{KL} * \nabla_{\phi}^{2} g_{\phi}^{\beta}\| \leq \mathcal{B} * \delta^{2} \tag{15}$$

Let $\eta_1 = \max\{\eta_1^1, \cdots, \eta_1^k\}$. Thus combining equations 12,13,14 and 15,

$$\|\nabla_{\phi}^{2} \mathcal{L}_{\text{C-NET}_l}\| \leq 2 * \eta_1 \rho^2 (L\delta^2(1+\tau^2) + \mathcal{B}) \tag{16}$$

Define $L_V = 2 * \eta_1 \rho^2 (L\delta^2(1+\tau^2) + \mathcal{B})$, based on Lagrange mean value theorem, we have:

$$\|\nabla \mathcal{L}_{\text{C-NET}}(\phi_1) - \nabla \mathcal{L}_{\text{C-NET}}(\phi_2)\| \leq L_V \|\phi_1 - \phi_2\|, \quad for\ all \quad \phi_1, \phi_2 \tag{17}$$

■

Theorem 1 *Suppose the loss function ℓ is Lipschitz smooth with constant L, and C-NET$(\cdot)$ is differential with a δ-bounded gradient and twice differential with its Hessian bounded by $\mathcal{B}$, and the loss function $\mathcal{L}$ have ρ-bounded gradients concerning training/metadata. Let the learning rates $\eta_1^{1t}, \cdots, \eta_1^{kt}$ satisfies $\eta_1^{jt} = \min\{1, \frac{a}{T}\} \forall j \in \{1, \cdots, k\}$, for some $a > 0$, such that $\frac{a}{T} < 1$, and $\eta_2^t, 1 \leq t \leq N$ is a monotone descent sequence, $\eta_2^t = \min\{\frac{1}{L}, \frac{c}{\sigma\sqrt{T}}\}$ for some $c > 0$, such that $\frac{\sigma\sqrt{T}}{c} \geq L$ and $\sum_{t=1}^{\infty} \eta_2^t \leq \infty, \sum_{t=1}^{\infty} (\eta_2^t)^2 \leq \infty$. Then MC-DISTIL can achieve $\mathbb{E}[\|\nabla_{\phi} \mathcal{L}_{C\text{-}NET}(\Theta^t(\phi^t))\|_2^2] \leq \epsilon$ in $\mathcal{O}(1/\epsilon^2)$ steps. More specifically,*

$$\min_{0 \leq t \leq T} \mathbb{E}[\|\nabla_{\phi} \mathcal{L}_{C\text{-}NET}(\Theta^t(\phi^t))\|_2^2] \leq \mathcal{O}(\frac{C}{\sqrt{T}}), \tag{18}$$

where C is some constant independent of the convergence process, σ is the variance of drawing uniformly mini-batch samples at random.

Proof

From Equation equation 5

$$\phi^{t+1} = \phi^t - \frac{\eta_2^t}{m} \sum_{i=1}^{m} \nabla_{\phi^t} \mathcal{L}_{\text{C-NET}}(x_i^v, y_i^v, \Theta^{t+1}) = \phi^t - \eta_2^t \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) \tag{19}$$

This can be written as,

$$\phi^{t+1} = \phi^t - \eta_2^t \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\big|_{\Xi_t} \tag{20}$$

Since the mini-batch Ξ_t is drawn uniformly from the entire data set, we can rewrite the update equation as:

$$\phi^{t+1} = \phi^t - \eta_2^t [\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}] \tag{21}$$

where $\xi^{(t)} = \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\big|_{\Xi_t} - \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))$. Note that $\xi^{(t)}$ are i.i.d random variables with finite variance since Ξ_t are drawn *i.i.d* with a finite number of samples. Furthermore, $\mathbb{E}[\xi^{(t)}] = 0$, since samples are drawn uniformly at random. Observe that

$$\mathcal{L}_{\text{C-NET}}(\Theta^{t+2}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})) = \Big[\mathcal{L}_{\text{C-NET}}(\Theta^{t+2}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1}))\Big] + \Big[\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t}))\Big] \tag{22}$$

By Lipschitz smoothness of meta loss function, we have

$$\mathcal{L}_{\text{C-NET}}(\Theta^{t+2}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1})) \leq \Big\langle \nabla\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1})), \Theta^{t+2}(\phi^{t+1}) - \Theta^{t+1}(\phi^{t+1}) \Big\rangle + \frac{L}{2}\Big\|\Theta^{t+2}(\phi^{t+1}) - \Theta^{t+1}(\phi^{t+1})\Big\|_2^2 \tag{23}$$

Here, $\Theta^{t+2}(\phi^{t+1}) - \Theta^{t+1}(\phi^{t+1}) = \Big\langle \frac{\eta_1^{1t}}{n}\sum_{i=1}^{n} g^{\alpha}_{\phi^t i}[1] * \nabla_{\theta_1^{t+1}} H\big(y_i^{(S_1)}, y_i\big) + g^{\beta}_{\phi^t i}[1] * \tau^2 * \nabla_{\theta_1^{t+1}} KL\big(y_i^{(S_1)}, y_i^{(T)}\big), \cdots, \frac{\eta_1^{kt}}{n}\sum_{i=1}^{n} g^{\alpha}_{\phi^t i}[k] * \nabla_{\theta_k^{t+1}} H\big(y_i^{(S_k)}, y_i\big) + g^{\beta}_{\phi^t i}[k] * \tau^2 * \nabla_{\theta_k^{t+1}} KL\big(y_i^{(S_k)}, y_i^{(T)}\big)\Big\rangle$ according to Eq.(4) and $\Big\|\nabla_{\theta_j^{t+1}} H\big(y_i^{(S_j)}, y_i\big)\Big\| \leq \rho, \Big\|\nabla_{\theta_j^{t+1}} KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Big\| \leq \rho, \big\|\nabla H\big(f_{\theta_j}(x_i^v), y_i^v\big)\big\| \leq \rho, g_{\phi}(\cdot) \leq 1$ and $\eta_1 = \max\{\eta_1^1, \cdots, \eta_1^k\}$, we have

$$\begin{aligned}\|\mathcal{L}_{\text{C-NET}}(\Theta^{t+2}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1}))\| &\leq k\eta_1^t\rho^2(1+\tau^2) + \frac{kL(\eta_1^t(1+\tau^2))^2}{2}\rho^2 \\ &= k\eta_1^t\rho^2(1+\tau^2)\Big(1 + \frac{\eta_1^t L(1+\tau^2)}{2}\Big)\end{aligned} \tag{24}$$

By Lipschitz continuity of $\nabla\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi))$ according to Lemma 1, we can obtain the following:

$$\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})) \leq \Big\langle \nabla\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})), \phi^{t+1} - \phi^t \Big\rangle + \frac{L}{2}\Big\|\phi^{t+1} - \phi^t\Big\|_2^2 \tag{25}$$

From Eq. 21,

$$\begin{aligned}\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})) &\leq \Big\langle \nabla\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})), -\eta_2^t[\nabla_{\phi}\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})) + \xi^{(t)}] \Big\rangle \\ &+ \frac{L(\eta_2^t)^2}{2}\Big\|\nabla_{\phi}\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})) + \xi^{(t)}\Big\|_2^2 = -\Big(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\Big)\big\|\nabla_{\phi}\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t}))\big\| \\ &+ \frac{L(\eta_2^t)^2}{2}\|\xi^{(t)}\|_2^2 - (\eta_2^t - L(\eta_2^t)^2)\big\langle \nabla\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})), \xi^{(t)}\big\rangle\end{aligned} \tag{26}$$

Thus Eq. 22 satifies,

$$\begin{aligned}\mathcal{L}_{\text{C-NET}}(\Theta^{t+2}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})) &\leq k\eta_1^t\rho^2(1+\tau^2)\Big(1 + \frac{\eta_1^t L(1+\tau^2)}{2}\Big) \\ &- \Big(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\Big)\big\|\nabla_{\phi}\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t}))\big\|_2^2 \\ &+ \frac{L(\eta_2^t)^2}{2}\|\xi^{(t)}\|_2^2 - (\eta_2^t - L(\eta_2^t)^2)\big\langle \nabla\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^{t})), \xi^{(t)}\big\rangle\end{aligned} \tag{27}$$

Rearranging the terms, we obtain

$$\left(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\right)\left\|\nabla_\phi \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\right\|_2^2 \leq \mathcal{L}_{\text{C-NET}}(\Theta^{t+2}(\phi^{t+1})) - \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + k\eta_1^t\rho^2(1+\tau^2)\left(1+\frac{\eta_1^t L(1+\tau^2)}{2}\right) + \frac{L(\eta_2^t)^2}{2}\|\xi^{(t)}\|_2^2 - (\eta_2^t - L(\eta_2^t)^2)\left\langle \nabla \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)}\right\rangle \tag{28}$$

Summing up the above inequalities and rearranging the terms, we obtain

$$\begin{aligned}\sum_{t=1}^{T}\left(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\right)\left\|\nabla_\phi \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\right\|_2^2 &\leq \mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1)) - \mathcal{L}_{\text{C-NET}}(\Theta^{T+1}(\phi^T)) \\ &+ \sum_{t=1}^{T} k\eta_1^t\rho^2(1+\tau^2)\left(1+\frac{\eta_1^t L(1+\tau^2)}{2}\right) + \sum_{t=1}^{T}\frac{L(\eta_2^t)^2}{2}\|\xi^{(t)}\|_2^2 \\ &- \sum_{t=1}^{T}(\eta_2^t - L(\eta_2^t)^2)\left\langle \nabla \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)}\right\rangle \\ &\leq \mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1)) + \sum_{t=1}^{T} k\eta_1^t\rho^2(1+\tau^2)\left(1+\frac{\eta_1^t L(1+\tau^2)}{2}\right) + \sum_{t=1}^{T}\frac{L(\eta_2^t)^2}{2}\|\xi^{(t)}\|_2^2 \\ &- \sum_{t=1}^{T}(\eta_2^t - L(\eta_2^t)^2)\left\langle \nabla \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)}\right\rangle \end{aligned} \tag{29}$$

Taking expectations with respect to $\xi^{(N)}$ on both sides, we obtain:

$$\sum_{t=1}^{T}\left(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\right)\mathbb{E}_{\xi^{(N)}}\left\|\nabla_\phi \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\right\|_2^2 \leq \mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1)) + \sum_{t=1}^{T} k\eta_1^t\rho^2(1+\tau^2)\left(1+\frac{\eta_1^t L(1+\tau^2)}{2}\right) + \sum_{t=1}^{T}\frac{L(\eta_2^t)^2}{2}\|\xi^{(t)}\|_2^2 \tag{30}$$

since $\mathbb{E}_{\xi^{(N)}}\langle \nabla \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)}\rangle = 0$ and $\mathbb{E}[\|\xi^{(t)}\|_2^2] \leq \sigma^2$, where σ^2 is the variance of $\xi^{(t)}$. Furthermore, we can deduce that

$$
\begin{aligned}
\min_t \mathbb{E}\left\|\nabla_\phi \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\right\|_2^2 &\leq \frac{\sum_{t=1}^T \left(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\right)\mathbb{E}_{\xi^{(N)}}\left\|\nabla_\phi \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\right\|_2^2}{\sum_{t=1}^T \left(\eta_2^t - \frac{L(\eta_2^t)^2}{2}\right)} \\
&\leq \frac{1}{\sum_{t=1}^T (2\eta_2^t - L(\eta_2^t)^2)}\Bigg[2\mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1)) + \sum_{t=1}^T k\eta_1^t\rho^2(1+\tau^2)(2+\eta_1^t L(1+\tau^2)) \\
&+ L\sigma^2\sum_{t=1}^T(\eta_2^t)^2\Bigg] \\
&\leq \frac{1}{\sum_{t=1}^T \eta_2^t}\left[2\mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1)) + \sum_{t=1}^T k\eta_1^t\rho^2(1+\tau^2)(2+\eta_1^t L(1+\tau^2)) + L\sigma^2\sum_{t=1}^T(\eta_2^t)^2\right] \\
&\leq \frac{1}{T\eta_2^t}\left[2\mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1)) + Tk\eta_1^1\rho^2(1+\tau^2)(2+\eta_1^1 L(1+\tau^2)) + L\sigma^2\sum_{t=1}^T(\eta_2^t)^2\right] \\
&\leq \frac{2\mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1))}{T\eta_2^t} + \frac{k\eta_1^1\rho^2(1+\tau^2)(2+L(1+\tau^2))}{\eta_2^t} + L\sigma^2\eta_2^t \\
&= \frac{\mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1))}{T}\max\{L, \frac{\sigma\sqrt{T}}{c}\} \\
&+ \min\{1, \frac{a}{T}\}\max\{L, \frac{\sigma\sqrt{T}}{c}\}k\rho^2(1+\tau^2)(2+L(1+\tau^2)) + L\sigma^2\min\{\frac{1}{L}, \frac{c}{\sigma\sqrt{T}}\} \\
&\leq \frac{\mathcal{L}_{\text{C-NET}}(\Theta^1(\phi^1))}{c\sqrt{T}} + \frac{a\sigma k\rho^2(1+\tau^2)(2+L(1+\tau^2))}{c\sqrt{T}} + \frac{L\sigma c}{\sqrt{T}} = \mathcal{O}(\frac{1}{\sqrt{T}}).
\end{aligned} \tag{31}
$$

The third inequlity holds for $\sum_{t=1}^T(2\eta_2^t - L(\eta_2^t)^2) \leq \sum_{t=1}^T \eta_2^t$.Therefore, we can conclude that our algorithm can always achieve $\min_{0\leq t\leq T}\mathbb{E}[\|\nabla\mathcal{L}_{\text{C-NET}}(\Theta^1)\|_2^2] \leq \mathcal{O}(\frac{1}{\sqrt{T}})$ in T steps, and this finishes our proof of Theorem 1.

■

Lemma 2 *(Lemma A.5 in (Mairal, 2013)) Let $(a_n)_{n\leq 1}, (b_n)_{n\leq 1}$ be two non-negative real sequences such that the series $\sum_{i=1}^\infty a_n$ diverges, the series $\sum_{i=1}^\infty a_n b_n$ converges, and there exists $K > 0$ such that $|b_{n+1} - b_n| \leq Ka_n$. Then the sequences $(b_n)_{n\leq 1}$ converges to 0.*

Theorem 2 *Suppose the loss function $\mathcal{L}$ is Lipschitz smooth with constant L, and C-NET$(\cdot)$ is differential with a δ-bounded gradient and twice differential with its Hessian bounded by $\mathcal{B}$, and the loss function $\mathcal{L}$ have ρ-bounded gradients with respect to training/meta data. Let the learning rates $\eta_1^{1t}, \cdots, \eta_1^{kt}$ satisfies $\eta_1^{jt} = \min\{1, \frac{a}{T}\}\forall j \in \{1, \cdots, k\}$, for some $a > 0$, such that $\frac{a}{T} < 1$, and $\eta_2^t, 1 \leq t \leq N$ is a monotone descent sequence, $\eta_2^t = \min\{\frac{1}{L}, \frac{c}{\sigma\sqrt{T}}\}$ for some $c > 0$, such that $\frac{\sigma\sqrt{T}}{c} \geq L$ and $\sum_{t=1}^\infty \eta_2^t \leq \infty, \sum_{t=1}^\infty(\eta_2^t)^2 \leq \infty$.Then*

$$
\lim_{t\to\infty}\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t, \phi^t)\|_2^2] = 0. \tag{32}
$$

Proof

From Eq. **??** we have,

$$
\mathcal{L}_{S_j}(\theta_j^t, \phi^t) = \frac{1}{n}\sum_{i=1}^n g_{\phi^t i}^{\alpha}[j] * H\big(y_i^{(S_j)}, y_i\big) + g_{\phi^t i}^{\beta}[j] * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \quad \forall j \in \{1, \cdots, k\} \tag{33}
$$

It is easy to conclude that $\eta j t_1$ satisfy $\sum_{t=0}^{\infty} \eta j t_1 = \infty, \sum_{t=0}^{\infty} (\eta j t_1)^2 < \infty$. From eq. 4

$$\theta_j^{t+1} = \theta_j^t - \frac{\eta_1^j}{n} \sum_{i=1}^{n} g^{\alpha}_{\phi^t i}[j] * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big) + g^{\beta}_{\phi^t i}[j] * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \quad \forall j \in \{1, \cdots, k\} \tag{34}$$

Since the mini-batch Ψ_t is drawn uniformly at random, we can rewrite the update equation as:

$$\begin{aligned}
\theta_j^{t+1} &= \theta_j^t - \frac{\eta_1^j}{n} \sum_{i=1}^{n} \Big[g^{\alpha}_{\phi^t i}[j] * \nabla_{\theta_j^t} H\big(y_i^{(S_j)}, y_i\big) \\
&\quad + g^{\beta}_{\phi^t i}[j] * \tau^2 * \nabla_{\theta_j^t} KL\big(y_i^{(S_j)}, y_i^{(T)}\big) + \Psi_t \Big] \quad \forall j \in \{1, \cdots, k\} \\
&= \theta_j^t - \eta_1^{tj} [\nabla \mathcal{L}_{S_j}(\theta_j^t, \phi^t) + \Psi_t]
\end{aligned} \tag{35}$$

where $\psi^{(t)} = \nabla \mathcal{L}_{S_j}(\theta_j^t, \phi^t)|_{\Psi_t} - \mathcal{L}_{S_j}(\theta_j^t, \phi^t)$. Note that $\psi^{(t)}$ is i.i.d. random variable with finite variance, since Ψ_t are drawn i.i.d. with a finite number of samples. Furthermore, $\mathbb{E}[\psi^{(t)}] = 0$, since samples are drawn uniformly at random, and $\mathbb{E}[\|\psi^{(t)}\|_2^2] \leq \sigma^2$.

The objective function $\mathcal{L}_{S_j}(\theta_j^t, \phi^t)$ can be easily checked to be Lipschitz-smooth with constant L, and have ρ-bounded gradients with respect to training data. Observe that,

$$\mathcal{L}_{S_j}(\theta_j^{t+1}, \phi^{t+1}) - \mathcal{L}_{S_j}(\theta_j^t, \phi^t) = \Big\{ \mathcal{L}_{S_j}(\theta_j^{t+1}, \phi^{t+1}) - \mathcal{L}_{S_j}(\theta_j^{t+1}, \phi^t) \Big\} + \Big\{ \mathcal{L}_{S_j}(\theta_j^{t+1}, \phi^t) - \mathcal{L}_{S_j}(\theta_j^t, \phi^t) \Big\} \tag{36}$$

For the first term,

$$\begin{aligned}
&\mathcal{L}_{S_j}(\theta_j^{t+1}, \phi^{t+1}) - \mathcal{L}_{S_j}(\theta_j^{t+1}, \phi^t) = \frac{1}{n} \sum_{i=1}^{n} (g^{\alpha}_{\phi^{t+1} i}[j] - g^{\alpha}_{\phi^t i}[j]) * H\big(y_i^{(S_j)}, y_i\big) \\
&+ (g^{\beta}_{\phi^{t+1} i}[j] - g^{\beta}_{\phi^t i}[j]) * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \Big\{ \langle \nabla_{\phi} g^{\alpha}_{\phi_i^t}[j], \phi^{t+1} - \phi^t \rangle + \frac{\delta(\eta_2^t)^2}{2} \|\phi^{t+1} - \phi^t\|_2^2 \Big\} * H\big(y_i^{(S_j)}, y_i\big) \\
&+ (g^{\beta}_{\phi^{t+1} i}[j] - g^{\beta}_{\phi^t i}[j]) * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \\
&= \frac{1}{n} \sum_{i=1}^{n} \Big\{ \langle \nabla_{\phi} g^{\alpha}_{\phi_i^t}[j], -\eta_2^t [\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}] \rangle \\
&+ \frac{\delta(\eta_2^t)^2}{2} \|\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}\|_2^2 \Big\} * H\big(y_i^{(S_j)}, y_i\big) + (g^{\beta}_{\phi^{t+1} i}[j] - g^{\beta}_{\phi^t i}[j]) * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \\
&= \frac{1}{n} \sum_{i=1}^{n} \Big\{ \langle \nabla_{\phi} g^{\alpha}_{\phi_i^t}[j], -\eta_2^t [\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}] \rangle + \frac{\delta(\eta_2^t)^2}{2} \Big(\|\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2 \\
&+ 2 \langle \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)} \rangle \Big) \Big\} * H\big(y_i^{(S_j)}, y_i\big) + (g^{\beta}_{\phi^{t+1} i}[j] - g^{\beta}_{\phi^t i}[j]) * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big) \\
&\leq \frac{1}{n} \sum_{i=1}^{n} \Big\{ \langle \nabla_{\phi} g^{\alpha}_{\phi_i^t}[j], -\eta_2^t [\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}] \rangle + \frac{\delta(\eta_2^t)^2}{2} \Big(\|\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2 \\
&+ 2 \langle \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)} \rangle \Big) \Big\} * H\big(y_i^{(S_j)}, y_i\big) + \frac{1}{n} \sum_{i=1}^{n} \Big\{ \langle \nabla_{\phi} g^{\beta}_{\phi_i^t}[j], -\eta_2^t [\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}] \rangle \\
&+ \frac{\delta(\eta_2^t)^2}{2} \Big(\|\nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2 + 2 \langle \nabla_{\phi} \mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)} \rangle \Big) \Big\} * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big)
\end{aligned} \tag{37}$$

For the second term from eq. 36,

$$\begin{aligned}
\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^t) - \mathcal{L}_{S_j}(\theta_j^t,\phi^t) &\leq \langle \nabla_{\theta_j}\mathcal{L}_{S_j}(\theta_j^t,\phi^t), \theta_j^{t+1}-\theta_j^t\rangle + \frac{L(\eta_1^j)^2}{2}\|\theta_j^{t+1}-\theta_j^t\|_2^2 \\
&= \langle \nabla_{\theta_j}\mathcal{L}_{S_j}(\theta_j^t,\phi^t), -\eta_1^{tj}[\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)+\Psi_t]\rangle + \frac{L(\eta_1^j)^2}{2}\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)+\Psi_t\|_2^2 \\
&= -\left(\eta_1^{tj} - \frac{L(\eta_1^j)^2}{2}\right)\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2 + \frac{L(\eta_1^j)^2}{2}\|\psi^{(t)}\|_2^2 \\
&- (\eta_1^{tj} - L(\eta_1^j)^2)\langle\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t), \psi^{(t)}\rangle
\end{aligned} \tag{38}$$

Combining Eq. 37 and Eq. 38 we can rewrite Eq. 36 as,

$$\begin{aligned}
&\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1}) - \mathcal{L}_{S_j}(\theta_j^t,\phi^t) \leq \frac{1}{n}\sum_{i=1}^{n}\Bigg\{\langle\nabla_\phi g^{\alpha}_{\phi_i^t}[j], -\eta_2^t[\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}]\rangle \\
&+ \frac{\delta(\eta_2^t)^2}{2}\left(\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2 + 2\langle\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)}\rangle\right) * H\big(y_i^{(S_j)}, y_i\big) \\
&+ \langle\nabla_\phi g^{\beta}_{\phi_i^t}[j], -\eta_2^t[\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)) + \xi^{(t)}]\rangle \\
&+ \frac{\delta(\eta_2^t)^2}{2}\left(\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2 + 2\langle\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t)), \xi^{(t)}\rangle\right) * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Bigg\} \\
&- \left(\eta_1^{tj} - \frac{L(\eta_1^j)^2}{2}\right)\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2 + \frac{L(\eta_1^j)^2}{2}\|\psi^{(t)}\|_2^2 - (\eta_1^{tj} - L(\eta_1^j)^2)\langle\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t), \psi^{(t)}\rangle
\end{aligned} \tag{39}$$

Taking expectation on both sides and since $\mathbb{E}[\xi^{(t)}] = 0, \mathbb{E}[\psi^{(t)}] = 0$, we have

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})] - \mathbb{E}[\mathcal{L}_{S_j}(\theta_j^t,\phi^t)] \leq \mathbb{E}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\Bigg\{\langle\nabla_\phi g^{\alpha}_{\phi_i^t}[j], -\eta_2^t\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\rangle \\
&+ \frac{\delta(\eta_2^t)^2}{2}\left(\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2\right)\Bigg\} * H\big(y_i^{(S_j)}, y_i\big)\Bigg] \\
&+ \mathbb{E}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\Bigg\{\langle\nabla_\phi g^{\beta}_{\phi_i^t}[j], -\eta_2^t\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\rangle \\
&+ \frac{\delta(\eta_2^t)^2}{2}\left(\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \|\xi^{(t)}\|_2^2 + \right) * \tau^2 * KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\Bigg\}\Bigg] \\
&- \eta_1^j\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] + \frac{L(\eta_1^j)^2}{2}\left(\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] + \mathbb{E}[\|\psi^{(t)}\|_2^2]\right)
\end{aligned} \tag{40}$$

Summing up the above inequalities over $t = 1, ..., \infty$ in both sides, we obtain,

$$\begin{aligned}
&\sum_{t=1}^{\infty}\eta_1^{tj}\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] + \sum_{t=1}^{\infty}\eta_2^t\mathbb{E}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\|H\big(y_i^{(S_j)}, y_i\big)\|\|\nabla_\phi g^{\alpha}_{\phi_i^t}[j]\|.\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|\Bigg] \\
&+ \sum_{t=1}^{\infty}\eta_2^t\tau^2\mathbb{E}\Bigg[\frac{1}{n}\sum_{i=1}^{n}\|KL\big(y_i^{(S_j)}, y_i^{(T)}\|\|\nabla_\phi g^{\beta}_{\phi_i^t}[j]\|.\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|\Bigg] \\
&\leq \sum_{t=1}^{\infty}\frac{L(\eta_1^j)^2}{2}\left(\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] + \mathbb{E}[\|\psi^{(t)}\|_2^2]\right) + \mathbb{E}[\mathcal{L}_{S_j}(\theta_j^1,\phi^1)] - \lim_{T\to\infty}\mathbb{E}[\mathcal{L}_{S_j}(\theta_j^t,\phi^t)] \\
&+ \sum_{t=1}^{\infty}\frac{\delta(\eta_2^t)^2}{2}\Bigg\{\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2 + \mathbb{E}\|\xi^{(t)}\|_2^2\right)\left(\|H\big(y_i^{(S_j)}, y_i\big)\| + \tau^2\|KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\|\right)\Bigg\} \\
&\leq \sum_{t=1}^{\infty}\frac{L(\eta_1^j)^2}{2}[\rho^2+\sigma^2] + \mathbb{E}[\mathcal{L}_{S_j}(\theta_j^1,\phi^1)] + \sum_{t=1}^{\infty}\frac{\delta(\eta_2^t)^2}{2}\left\{2\tau^2 * M(\rho^2+\sigma^2)\right\} \leq \infty
\end{aligned} \tag{41}$$

The last inequality holds since $\sum_{t=0}^{\infty}(\eta_1^j)^2 < \infty, \sum_{t=0}^{\infty}(\eta_2^t)^2 < \infty$, $1+\tau^2 \approx \tau^2$and $\frac{1}{n}\sum_{i=1}^{n}\|H\big(y_i^{(S_j)}, y_i\big)\| \leq M$ and $\frac{1}{n}\sum_{i=1}^{n}\|KL\big(y_i^{(S_j)}, y_i^{(T)}\big)\| \leq M$ for limited number of samples' loss is bounded. Thus we have

$$\sum_{t=1}^{\infty}\eta_1^{tj}\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] + \sum_{t=1}^{\infty}\eta_2^t\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|H\big(y_i^{(S_j)}, y_i\big)\|\|\nabla_\phi g^{\alpha}_{\phi_i^t}[j]\|.\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|\Big] + \sum_{t=1}^{\infty}\eta_2^t\tau^2\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|KL\big(y_i^{(S_j)}, y_i^{(T)}\|\|\nabla_\phi g^{\beta}_{\phi_i^t}[j]\|.\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|\Big] \leq \infty \tag{42}$$

Since

$$\sum_{t=1}^{\infty}\eta_2^t\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|H\big(y_i^{(S_j)}, y_i\big)\|\|\nabla_\phi g^{\alpha}_{\phi_i^t}[j]\|.\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|\Big] \leq M\delta\rho\sum_{t=1}^{\infty}\eta_2^t \leq \infty \tag{43}$$

and

$$\sum_{t=1}^{\infty}\eta_2^t\tau^2\mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^{n}\|KL\big(y_i^{(S_j)}, y_i^{(T)}\|\|\nabla_\phi g^{\beta}_{\phi_i^t}[j]\|.\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|\Big] \leq \tau^2 M\delta\rho\sum_{t=1}^{\infty}\eta_2^t \leq \infty \tag{44}$$

This implies that $\sum_{t=1}^{\infty}\eta_1^{tj}\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] < \infty$. By Lemma 2, to substantiate $\lim_{t\to\infty}\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2] = 0$ since $\sum_{t=0}^{\infty}\eta_1^{tj} = \infty$, therefore we only need to show,

$$\big|\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2^2 - \mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2]]\big| \leq C\eta_1^j \tag{45}$$

for some constant C. Based on the inequality:

$$|(\|a\|+\|b\|)(\|a\|-\|b\|)| \leq \|a+b\|\|a-b\|, \tag{46}$$

we then have:

$$
\begin{aligned}
&\left|\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2^2-\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2^2]]\right|=\left|(\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2\right.\\
&\left.-\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2]])(\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2]+\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2])\right|\\
&\leq\mathbb{E}\left[\left|\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2-\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2\right|\left|\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2+\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2\right|\right]\\
&\leq\mathbb{E}\left[\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})-\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})+\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2\right]\\
&\leq\mathbb{E}\left[(\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|_2-\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2)(\|\nabla\mathcal{L}_{S_j}(\theta_j^{t+1},\phi^{t+1})\|+\|\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2)\right]\\
&\leq 2L\rho\mathbb{E}\big[\|(\theta_j^{t+1},\phi^{t+1})-(\theta_j^{t},\phi^{t})\|_2\big]\\
&\leq 2L\rho\eta_2^t\eta_1^{tj}\mathbb{E}\left[\|(\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})+\Psi_t,\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))+\xi^{(t)})\|_2\right]\\
&\leq 2L\rho\eta_2^t\eta_1^{tj}\mathbb{E}\left[\sqrt{\|(\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})+\Psi_t\|_2^2}+\sqrt{\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))+\xi^{(t)})\|_2^2}\right]\\
&\leq 2L\rho\eta_2^t\eta_1^{tj}\sqrt{\mathbb{E}[\|(\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})+\Psi_t\|_2^2]+\mathbb{E}[\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))+\xi^{(t)})\|_2^2]}\\
&\leq 2L\rho\eta_2^t\eta_1^{tj}\sqrt{\mathbb{E}[\|(\nabla\mathcal{L}_{S_j}(\theta_j^{t},\phi^{t})\|_2^2+\mathbb{E}[\Psi_t\|_2^2]+\mathbb{E}[\|\nabla_\phi\mathcal{L}_{\text{C-NET}}(\Theta^{t+1}(\phi^t))\|_2^2]+\mathbb{E}[\xi^{(t)})\|_2^2]}\\
&\leq 2L\rho\eta_2^t\eta_1^{tj}\sqrt{2\sigma^2+2\rho^2}
\end{aligned}
\tag{47}
$$

According to the above inequality, we can conclude that our algorithm can achieve

$$
\lim_{t\to\infty}\mathbb{E}[\|\nabla\mathcal{L}_{S_j}(\theta_j^t,\phi^t)\|_2^2]=0 \tag{48}
$$

The proof is completed.

■

B Training Details

B.1 Dataset Details

We conduct experiments using the following real-world datasets to showcase the effectiveness of our approach,

CIFAR-100 (Krizhevsky, 2009). The dataset consists of a total of 60K examples, distributed across 100 distinct classes. Each example in this dataset comprises images with a resolution of $32 \times 32 \times 3$. Specifically, the training set encompasses 50,000 examples, while the remaining 10K serve as the testing set. In our experimental setup, approximately 5K examples are allocated for use as a validation set for our C-NET. For the other baseline models, these validation examples are utilized for hyper-parameter tuning.

ImageNet (Russakovsky et al., 2015).

Tiny-ImageNet (Le & Yang, 2015). This dataset is derived from the extensive ImageNet-1K dataset (Russakovsky et al., 2015). This dataset encompasses a total of 100K images, which have been downsampled to a 64×64 resolution, representing a subset of 200 classes mirroring those in the ImageNet-1K dataset, with each class containing precisely 500 images. As part of our experimental protocol, we have set aside an independent validation set comprising 10K examples, which is utilized for both our proposed method and the baseline models.

Clothing-1M (Xiao et al., 2015a). This dataset consists of 10,00,000 noisy label training examples spread across 14 class labels of clothing items taken from various online shopping websites. Each example consists of an image with resolution $224 \times 224 \times 3$. The validation and test set consists of 14,313 and 10,526 examples respectively. The dataset also contains 47,560 clean examples, but we do not use that portion for our experiments. It is a dataset with noisy labels, since the data is collected from several online shopping websites and include many mislabelled samples.

Inst. CIFAR-100 (Xia et al., 2020b) This dataset contains manually corrupted version (with an error rate τ) of the CIFAR-100 (Krizhevsky, 2009) dataset. This is done by first constructing a transition matrix $\mathcal{T}(\mathbf{x})$, where $\mathcal{T}_{ij}(\mathbf{x})$ is the probability that the corrupted label is j, given the true label is i; while keeping the overall error rate as τ. Then, the labels are flipped according to the defined transition probabilties. In our experiments, the training uses 50K examples with noisy labels, corrupted by the above method, while the remaining 10K examples, which serve as testing set have no label noise. Approximately 5K noisy examples are allocated for use as a validation set for our C-NET. The overall label flip rate τ, used in our experiments was 0.2.

iWildCam-2020 (Beery et al., 2020) The dataset is set of collection of images collected using the heat trap and motion activated cameras for the better understanding of wildlife and biodiversity. There is so wide diversity in lighting, camera angle, backdrop, vegetation, color, and relative animal frequencies between different camera traps, and also the number of animals of a particular species varies a lot, hence the data about different spices also vary similarly. In our experimentation we trained the models using approx 100k training, 12k validation and 12k test images across 216 classes.

Tiny-ImageNet-C (Hendrycks & Dietterich, 2019b). This dataset is a downsampled version of the ImageNet-C dataset, which consists of 15 diverse corruption types applied to the ImageNet-1K (Russakovsky et al., 2015) validation dataset. These corruptions have been drawn from 4 main categories - noise, blur, weather, and digital. Each corruption has 5 severity levels, 1 being the least severe and 5 being the most severe. In our experiments, the models trained on Tiny-ImageNet (Le & Yang, 2015) dataset, with train-validation split of 90K and 10K examples respectively, are tested on Tiny-ImageNet-C for testing robustness gains.

For all the datasets, we use the data augmentation methods, using the torchvision's transforms module. We use RandomCrop, RandomResizedCrop, RandomSizedCrop, RandomHorizontalFlip, Normalize, and ColorJitter for our purpose. Augmentation methods were applied on both datasets, over all experiments, baselines, overall models.

B.2 Model Details

In Section 4.1, we introduce several smaller ResNet models, specifically ResNet10-xxxs, ResNet10-xxs, ResNet10-xs, ResNet10-s, ResNet10-m, and ResNet10-l. These models are adopted from Kag et al. (2023), and in Table 9, we compare their architectural details with well-established ResNet models like ResNet10, ResNet18, and ResNet34. Similar to the standard ResNet models, these newer, more compact versions also employ the traditional 'BasicBlock' as their fundamental building block. The architectural structure consists of a convolutional block, four stages of residual blocks, an adaptive average pooling layer, a convolutional block, and a classifier layer. The sole variation among the various capacity variants within this experimental setup is the number of filters in each stage and the residual block. Additionally, Table 9 provides information about the number of parameters and multiply-addition (MAC) operations for each model for the datasets CIFAR-100 and Tiny-ImageNet.

B.3 Hyper-Parameters

For all the KD baselines listed in Section 4.2 and MC-Distil, we use temperature $\tau = 2$, employ the SGD optimizer to train the student models and ADAM optimizer (Kingma & Ba, 2014) to train C-Net. We use a batch size of 400 for all datasets. We train the student models for 300 epochs on CIFAR-100, Inst. CIFAR-100, Clothing-1M, Tiny-ImageNet and Tiny-ImageNet-C datasets; and for 100 epochs on iWildCam

Architecture	Filters	Basic block Repeats	CIFAR-100		Tiny-Imagenet	
			MACs	Params	MACs	Params
ResNet10-xxs	[8, 8, 16, 16]	[1, 1, 1, 1]	2 M	13 K	8 M	15 K
ResNet10-xs	[8, 16, 16, 32]	[1, 1, 1, 1]	3 M	28 K	12 M	31 K
ResNet10-s	[8, 16, 32, 64]	[1, 1, 1, 1]	4 M	84 K	16 M	90 K
ResNet10-m	[16, 32, 64, 128]	[1, 1, 1, 1]	16 M	320 K	64 M	333 K
ResNet10-l	[32, 64, 128, 256]	[1, 1, 1, 1]	64 M	1.25 M	255 M	1.28 M
ResNet10	[64, 128, 256, 512]	[1, 1, 1, 1]	253 M	4.92 M	1013 M	5 M
ResNet18	[64, 128, 256, 512]	[2, 2, 2, 2]	555 M	11.22 M	2221 M	11.27 M
ResNet34	[64, 128, 256, 512]	[3, 4, 6, 3]	1159 M	21.32 M	4637 M	21.38 M

Table 9: Comparision of newly introduced smaller ResNet with the standard ResNet models

dataset, while updating C-Net every 20 epoch (*i.e.*, L = 20). We use cosine annealing (Loshchilov & Hutter, 2017) as the learning rate schedule for training the student models. We warm start each student model by first training it using the cross-entropy loss without using the teacher model for all KD baselines and MC-Distil. For all datasets, we perform a grid search on $0.1, 0.05$ for the learning rate, on $1, e-5, 5e-5, 1e-4$ for the weight decay, and $0.65, 0.75, 0.85, 0.95$ for the momentum. For the C-Net training we use a learning rate of $1e-3$ and set weight decay to $1e-4$.

B.4 Baselines

We compare MC-Distil with the following SOTA Knowledge distillation baselines:

Teacher Assistant Knowledge Distillation (TAKD) (Mirzadeh et al., 2020) This approach introduces a multi-step distillation process that leverages intermediate-level teachers to facilitate the efficient transfer of knowledge from a large pre-trained teacher network to a more compact student model. To realize this, we employ a teacher network identical to our own and enlist fellow student models with higher learning capacities to serve as intermediate models in this knowledge distillation process.

Densely Guided Knowledge Distillation (DGKD) (Son et al., 2021) Much like TAKD, this approach employs several intermediate models; however, it distinguishes itself by training the final student model through a single distillation step. In addition to the teacher KL divergence loss, the training objective for the final student model incorporates the KL divergence loss obtained from the pre-trained intermediate models.

Robust Model Compression (RMC) (Du et al., 2023). It uses multiple students with various levels of sparsity and interprets the variance in their predictions for each instance as a measure of task complexity. Subsequently, it refines the teacher predictions based on this complexity metric, resulting in a more robust knowledge distillation process. In our experiments, we employ students with diverse learning capacities as a substitute for models with different levels of sparsity, achieving similar benefits.

Deep mutual information (DML)(Zhang et al., 2018) It is a collaborative learning paradigm that leverages the interactions among multiple neural networks to enhance their collective performance. In DML, a group of neural networks, often referred to as student models, collaboratively learns by exchanging information with each other during training. This is achieved by using the predictions or feature representations of one model to guide the learning process of others. The collaborative nature of DML allows the models to exploit the diverse knowledge learned by their peers, fostering a mutual learning dynamic.

SHAdow KnowlEdge transfer framework (SHAKE) (Li & Jin, 2022) It is a novel knowledge distillation framework designed to enhance the transfer of knowledge from a teacher model to a student model in deep learning. Introduced to address the challenges associated with large capacity gaps between teacher and student models, SHAKE employs a pseudo-teacher concept. This pseudo-teacher consists of a set of additional

learnable layers attached to the backbone of the teacher model. What sets SHAKE apart is its ability to dynamically adapt to the student model during training.

In experiments where we train with noisy datasets, we compare against additional two bilevel optimisation-based reweighting schemes,

Learning to Reweight(L2R) (Ren et al., 2018) It is a machine learning paradigm that focuses on training models to dynamically assign weights to different instances in a dataset during the learning process. The fundamental idea behind L2R is to adaptively adjust the significance of individual data points, emphasizing more informative instances and de-emphasizing noisy or less relevant ones. This approach is particularly valuable in scenarios where the training data may be imbalanced or subject to noise, helping models prioritize the most crucial samples for better generalization.

Meta-Weight-Net (MWN) (Shu et al., 2019b) Like L2R, MWN also learns weights assigned to individual samples based on their specific characteristics or importance. However, this explicit mapping of sample weights is learned through a meta-learning framework. Meta-Weight-Net aims to address challenges related to imbalanced or noisy datasets by explicitly learning a mapping function that adapts sample weights during the training process. By doing so, the model can assign higher importance to informative or challenging samples and reduce the impact of less relevant ones.

B.5 Compute Resources

We our experiments on a mixture of GPUS viz. A100s and RTX 2080 as our experiments don't need any sophisticated modern GPUs. MC-Distil memory requirements are slightly higher that traditional KD as we train student co-horts simultaneously. The additional memory requirements depend the number of student co-horts used and size of co-horts.

C Additional Experiments

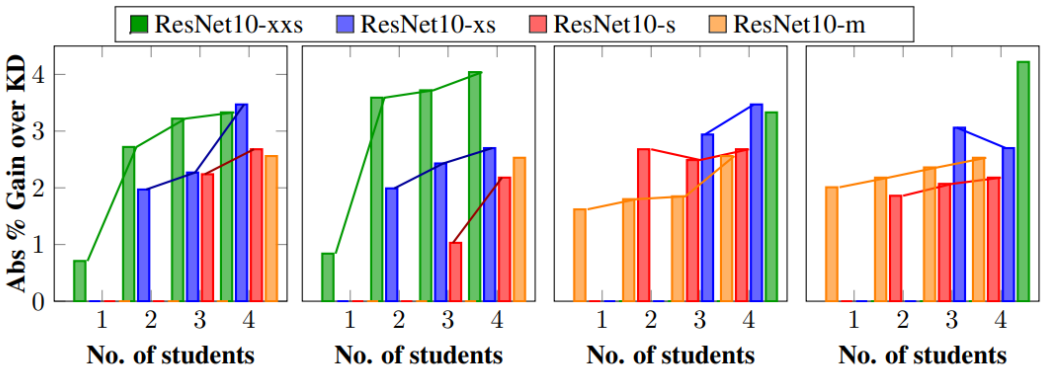
We have released anonymized code at the URL: `https://anonymous.4open.science/r/Multinet-E01D`.

C.1 Ablations: Changing no. of student cohorts

To investigate the impact of introducing additional students in the presence of C-Net, we conducted an experiment in which we incrementally introduced student models, one at a time. We present the results of these experiments in Figure 5. This experiment was conducted in two distinct settings: one in which each subsequent addition involved a student with a larger learning capacity. This is presented in Figure 5a and 5b. The other setting is in which each additional introduced student is of a smaller learning capacity as shown in Figure 5c and 5d. In both settings, we perform knowledge distillation with two teachers *viz.*, ResNet10 and Resnet18. We note that across different teachers, introducing additional students improves the performances of all the students participating in the training process. This is much more pronounced in the setting where larger students are added.

C.2 Ablations: Varying the size of C-Net

In table 10 presents an analysis of test accuracies on the CIFAR100 dataset for different student architectures under the influence of various sizes of the coordinating network (C-Net).We observe that, in the majority of instances, there are negligible alterations in the performance of the student model when it is trained with coordinating networks (C-Net) of varying sizes. The term "performance" here refers to the model's accuracy, and the lack of significant changes suggests that the choice of C-Net size has minimal impact on the student model's ability to capture and generalize from the teacher's knowledge. This stability across different C-Net sizes indicates robustness in the knowledge distillation process, emphasizing that the coordination mechanism implemented by C-Net is effective across a range of architectural scales.



(a) Adding larger students with ResNet10 model as teacher

(b) Adding larger students with ResNet18 model as teacher

(c) Adding smaller students with ResNet10 model as teacher

(d) Adding smaller students with ResNet18 model as teacher

Figure 5: Analysis of the effect of introducing student model cohorts for different learning capacities in meta-collaboration setting.

Teacher		**Student**	**C-Net**			
			ResNet10-L	ResNet10	ResNet18	ResNet32
ResNet10	75.18	ResNet10-xxs	36.13	34.95	36	36.38
		ResNet10-xs	47.4	46.88	47.19	47.69
		ResNet10-s	56.93	57.48	57.7	58.24
		ResNet10-m	68.65	67.95	67.91	69.11
ResNet18	76.99	ResNet10-xxs	36.12	34.48	35.28	36.14
		ResNet10-xs	47.68	46.44	46.98	47.68
		ResNet10-s	57.12	57.82	57.65	58.21
		ResNet10-m	69.72	69.72	68.96	69.62
ResNet34	79.47	ResNet10-xxs	35.93	34.71	35.65	36.13
		ResNet10-xs	47.31	46.44	46.42	47.52
		ResNet10-s	57.64	57.3	56.89	57.7
		ResNet10-m	69.28	68.2	67.54	68.49
ResNet10-l	41.25	ResNet10-xxs	35.88	34.35	35.76	35.9
		ResNet10-xs	47.28	45.8	46.59	47.42
		ResNet10-s	57.51	56.61	56.49	57.74
		ResNet10-m	68.86	67.94	67.49	68.54

Table 10: The table showcases the test accuracies for different student-teacher-coordinator network combinations, providing insights into the impact of varying C-Net sizes on model performance. The table serves as a comprehensive analysis of how different combinations influence the final model accuracies in the CIFAR100 dataset.

C.3 Robustness of MC-Distil trained models

For the models trained with Resnet-18 serving as the teacher model on the TinyImagenet dataset in Table 1, we present their performance on TinyImageNet-C (Hendrycks & Dietterich, 2019a) (see appendix B.1) in Table 11 incorporating all baseline models and diverse corruption levels. We assess the resilience of models whose perfromance was reported in Table 1 when confronted with data distribution mismatches.

Student	CE	KD	MCD	DGKD	DML	SKAKE	Meta-Distill	MC-Distil
Corruption level 1 Accuracies								
RN10-xxs	11.25	11.38	10.61	10.77	9.99	11.47	11.02	**12.05**
RN10-xs	14.68	15.82	14.67	15.43	15.51	16.56	16.02	**16.73**
RN10-s	20.88	21.97	21.94	23.18	20.29	22.58	23.97	**24.88**
RN10-m	26.91	28.56	27.08	29.56	28.09	30.40	30.43	**31**
Corruption level 2 Accuracies								
RN10-xxs	10.65	10.79	9.54	10.42	9.44	10.88	10.52	**11.3**
RN10-xs	13.75	14.6	13.72	14.60	14.56	15.42	14.78	**15.52**
RN10-s	19.28	20.64	20.46	21.26	18.74	20.74	21.93	**23.04**
RN10-m	24.76	25.87	24.36	26.97	26.20	27.94	27.39	**28.67**
Corruption level 3 Accuracies								
RN10-xxs	8.66	8.78	7.54	8.93	8.46	9.29	8.58	**9.42**
RN10-xs	10.94	11.54	11.08	11.82	11.96	12.19	11.78	**12.58**
RN10-s	15.68	16.85	16.71	16.95	14.76	16.95	17.65	**18.79**
RN10-m	19.7	21.07	18.79	22.20	21.45	22.23	20.83	**22.63**
Corruption level 4 Accuracies								
RN10-xxs	4.92	4.66	4.38	5.52	5.05	5.57	4.6	**5.87**
RN10-xs	6.77	6.96	5.88	**7.54**	6.89	6.57	7.35	7.39
RN10-s	9.41	9.92	10.21	9.42	8.35	10.29	9.77	**10.8**
RN10-m	11.12	12.28	10.74	12.74	**12.87**	12.44	11.54	12.03
Corruption level 5 Accuracies								
RN10-xxs	3.58	3.15	3.44	3.98	3.66	4.29	3.33	**4.4**
RN10-xs	5.14	5.18	4.34	**6.23**	5.15	5.05	5.47	5.53
RN10-s	6.94	7.24	7.33	7.07	6.19	**7.93**	7.15	7.38
RN10-m	8.16	9.09	7.86	9.73	9.46	**9.48**	8.44	8.89

Table 11: Comprehensive table presenting an assessment of model (trained with TinyImageNet in Table1) robustness by conducting inference on the TinyImagenet-C dataset (Hendrycks & Dietterich, 2019a) across different degrees of image corruption, alongside results for all baseline models. The columns indicate the method employed for training the model on the TinyImagenet dataset, utilizing Resnet-18 as the teacher model.

Notably, across various corruption levels and teacher-studentś combinations, MC-Distil demonstrates robust performance, even though it was not explicitly tailored for handling distributional shifts between training and test datasets. Specifically, at lower corruption levels, models trained with MC-Distil exhibit a performance advantage of 1-2% compared to other baselines. Even at higher corruption levels, MC-Distil maintains competitiveness with the baselines.

C.4 Standard deviation and statistical significance results:

In Table 12, we show the p-values of one-tailed Wilcoxon signed-rank test Wilcoxon (1992) performed on every single possible pair of knowledge distillation strategies to determine whether there is a significant statistical difference between the strategies in each pair, across all datasets. Our null hypothesis is that there is no difference between the knowledge distillation strategies pair. From the results, it is evident that MC-Distil significantly outperforms other baselines at $p < 0.05$.

Table 13 shows the standard deviation results over three training runs on CIFAR100 and tiny-Imagenet datasets. The results show that the MC-Distil has the least standard deviation compared two of the most competitive baselines viz. DGKD and MetaDistil. We note that DGKD has large standard deviations owing to not having a principled way of introducing student cohort models.

	CE	KD	TAKD	DGKD	RMC	DML	SHAKE	MetaDistil
KD	1.53e-05							
TAKD	1.53e-05	0.0005						
DGKD	1.53e-05	1.53e-05	0.0006					
RMC	4.78e-05	0.99	0.9992	1				
DML	1.53e-05	0.978	1	1	0.1156			
SHAKE	4.78e-05	0.115	0.64714	0.9855	0.0222	0.0416		
MetaDistil	1.53e-05	1.53e-05	0.0005	0.5896	4.58e-05	1.53e-05	0.001	
MC-Distil	1.53e-05	1.53e-05	1.53e-05	7.63e-05	1.53e-05	1.53e-05	0.00015	1.53e-05

Table 12: Pairwise significance p-values using Wilcoxon signed rank test

CIFAR100 Test Accuracies							
Teacher		**Student**	**CE**	**KD**	**DGKD**	**MetaDistil**	**MC-Distil**
ResNet10	75.18	ResNet10-xxs	32.41 ± 0.517	35.23 ± 1.132	34.57 ± 0.453	35.47 ± 0.737	36.23 ± 0.247
		ResNet10-xs	42.42 ± 0.724	45.62 ± 0.722	44.89 ± 1.775	46.86 ± 0.836	47.74 ± 0.244
		ResNet10-s	52.7 ± 0.384	55.64 ± 0.106	56.25 ± 0.5445	57.25 ± 0.535	58.42 ± 0.275
		ResNet10-m	64.19 ± 0.142	67.05 ± 0.191	67.27 ± 0.577	68.75 ± 0.185	69.45 ± 0.427
ResNet18	76.99	ResNet10-xxs	32.41 ± 0.517	35.19 ± 1.415	34.14 ± 1.259	35.26 ± 0.745	36.03 ± 0.442
		ResNet10-xs	42.42 ± 0.724	45.49 ± 0.412	46.18 ± 1.055	46.23 ± 0.42	47.24 ± 0.459
		ResNet10-s	52.7 ± 0.384	55.49 ± 0.315	56.07 ± 0.578	57.26 ± 0.131	57.63 ± 0.176
		ResNet10-m	64.19 ± 0.142	66.42 ± 0.095	66.73 ± 0.548	68.02 ± 0.49	68.52 ± 0.237
Tiny-ImageNet Test Accuracies							
ResNet10	44.04	ResNet10-xxs	13.68 ± 0.46	14.4 ± 0.526	14.2 ± 0.475	14.47 ± 0.597	15.33 ± 0.521
		ResNet10-xs	18.78 ± 0.231	20.2 ± 0.624	20.90 ± 0.351	20.850.692	22.07 ± 0.285
		ResNet10-s	24.86 ± 0.362	27.83 ± 0.779	27.59 ± 0.605	28.07 ± 0.89	29.62 ± 0.263
		ResNet10-m	33.93 ± 0.493	35.86 ± 0.332	35.82 ± 0.398	37.64 ± 1.452	38.46 ± 0.068
ResNet18	47.94	ResNet10-xxs	13.68 ± 0.46	14.52 ± 0.368	14.42 ± 0.481	14.68 ± 0.595	15.26 ± 0.105
		ResNet10-xs	18.78 ± 0.231	20.28 ± 0.5	20.58 ± 0.931	20.95 ± 1.038	21.36 ± 0.36
		ResNet10-s	24.96 ± 0.362	27.26 ± 0.874	27.82 ± 0.378	28.45 ± 0.983	30.36 ± 0.006
		ResNet10-m	33.93 ± 0.493	35.9 ± 0.724	35.96 ± 0.612	37.82 ± 1.562	39.11 ± 0.295

Table 13: Standard deviation results for CIFAR100 and Tiny-Imagenet datasets for 3 runs. We compare MC-Distil with two most competitive baselines.

D Limitations & Social Impact

We have presented a novel *metacollaborative distillation*–where students of different capacities are codistilled from a single teacher. While the initial results wide range of student & teacher architectures and model sizes are encouraging, more work is needed to understand under what conditions such adaptive mixing optimizations for metacollaborative distillation can provide gains. Our experiments are proof-of-concept on widely used benchmarking datasets; future work will explore whether similar gains can be realized in real-world applications as well. Our proposal involves a bi-level optimization objective (for which we provide a cheap approximate meta-learning preocedure) – the tradeoff of increased training time vs increased accuracy may depend on the specific application context.

Our work is a fairly general-purpose optimization procedure for improving knowledge transfer from a teacher model to a very small student model and therefore tremendously improving the inference time. This could

yield incremental improvements to a broad range of ML applications, and as a result benefit those applications. We do not anticipate any inherent risk of negative social impact from our work, over and above the risk inherent in the ML application itself.