SUPPLEMENTARY MATERIAL: UNDERSTANDING CALIBRATION TRANSFER IN KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

CONTENTS

1 INTRODUCTION

We complement our main text with supplementary materials encompassing the following components:

1. **Theoretical Insights**: This section contains main theoretical results, along with an explanation of the rationale behind choosing Padé approximants over more commonly used Taylor approximation, as discussed in Lemma 1.
2. **Rationale for Enhanced Performance**: This section elucidates the superior performance of the KD(C) framework, attributing it to three key factors: (a) insights from penultimate visualizations, (b) considerations of inter-class semantic similarities, and (c) the careful design of calibrators for the teacher model.
3. **Illustration of Generality**: Included is Fig. S5, which provides a visual demonstration of KD(C)'s versatility by comparing direct calibration with the KD(C) framework. It also presents an example featuring the Hebbalaguppe et al. (2022) regularizer.
4. **Expanded Experimental Scope**: We strengthen the KD(C) methodology with additional experiments, covering various scenarios, including large-to-small, small-to-large, self-distillation, and iterative self-distillation. These experiments involve different descriptors and datasets.
5. **Additional Results**: We provide supplementary results that encompass calibration performance in the presence of dataset drift and reliability diagrams featuring confidence histograms, as elaborated in Sec. 5.1.
6. **Hyperparameter Analysis**: A detailed study explores how calibration and accuracy are influenced by various hyperparameters in the KD(C) framework, as depicted in Fig. S8.
7. **Source Code**: The supplementary materials include the source code along with a `readme.md` file, enclosed within the provided zip file.
8. **Training and Compute Details**: We furnish comprehensive information on the specifics of training and compute resources employed in our experiments.
9. **Limitations and Broader Impact**: This section delves into the limitations of our research and contemplates its broader impact on the field.

These supplementary materials serve to enrich and provide a deeper understanding of our main findings and contributions.

2 THEORETICAL SUPPORT: ADDITIONAL DETAILS

2.1 PROOF OF THEOREM 1

The proof of Theorem 1 is contingent on several essential Lemmas, which will be introduced beforehand. Lemma 1 and Lemma 2 capture the effect of quadratic temperature scaling in the KD loss function, $\mathcal{L}_{\text{KD}}$. In particular, it is shown that the partial derivative of $\mathcal{L}_{\text{KD}}$ w.r.t. student's logit for a given sample is equal to the difference in predicted probabilities of the student and teacher classifiers for that sample. These results are leveraged to characterize the first-order condition of optimality for the total loss function $\mathcal{L}_{\text{tot}}$ w.r.t. parameters of the student classifier.

Lemma 1. *Let* $z_{i,s} := \mathbf{W}_s^\top \mathbf{x}_i$ *and* $z_{i,t} := \mathbf{W}_t^\top \mathbf{x}_i$ *with* $\tilde{p}_{i,s}$, $\tilde{p}_{i,t}$ *be defined as above. Then* $\lim_{T\to\infty} T(\tilde{p}_{i,s} - \tilde{p}_{i,t}) \approx p_{i,s} - p_{i,t}$.

Proof. The result follows as a consequence of Padé approximation. Recall that from definition,

$$\tilde{p}_{i,s} - \tilde{p}_{i,t} = \frac{1}{1+e^{-z_{i,s}/T}} - \frac{1}{1+e^{-z_{i,t}/T}} \approx \frac{1}{1+\dfrac{1-\frac{z_{i,s}}{2T}}{1+\frac{z_{i,s}}{2T}}} - \frac{1}{1+\dfrac{1-\frac{z_{i,t}}{2T}}{1+\frac{z_{i,t}}{2T}}}, \tag{S1}$$

where the last approximation follows from Padé approximation of the exponential when T is large.

Thus, Eq. (S1) can be re-written as:

$$\tilde{p}_{i,s} - \tilde{p}_{i,t} = \frac{1 + z_{i,s}/2T}{2} - \frac{1 + z_{i,t}/2T}{2} \implies T(\tilde{y}_{i,s} - \tilde{y}_{i,t}) = \frac{z_{i,s} - z_{i,t}}{4}. \tag{S2}$$

On the other hand, a similar analysis following the Padé approximation yields:

$$p_{i,s} - p_{i,t} \approx (z_{i,s} - z_{i,t})/4. \tag{S3}$$

Thus, Lemma 1 follows directly from Eq. (S2) and Eq. (S3). □

Remark: Padé approximants have a wider range of convergence than the corresponding Taylor series, and can even converge where the Taylor series does not. For a detailed exposition, please refer to Sec. 2.3 and Fig. S1.

The following result shows that the quadratic temperature scaling in the KD loss function ensures that the gradients used to update the network weights are independent of the smoothed labels.

Lemma 2 (Quadratic temperature scaling)**.** *Let $\mathcal{L}_{KD}$ be defined as in Eq. (2). Then,*

$$\lim_{T\to\infty} \frac{\partial \mathcal{L}_{KD}}{\partial z_{i,s}} = p_{i,s} - p_{i,t}.$$

Proof. Recall that by definition $\tilde{p}_{i,s} = \frac{1}{1 + e^{-z_{i,s}/T}}$. The partial derivative of $\tilde{p}_{i,s}$ w.r.t. $z_{i,s}$ reads:

$$\frac{\partial \tilde{p}_{i,s}}{\partial z_{i,s}} = \frac{1}{T}\tilde{p}_{i,s}(1 - \tilde{p}_{i,s}). \tag{S4}$$

On the other hand,

$$\frac{\partial \mathcal{L}_{\text{KD}}}{\partial z_{i,s}} = -T^2\left(\frac{\tilde{p}_{i,t}}{\tilde{p}_{i,s}} - \frac{1 - \tilde{p}_{i,t}}{1 - \tilde{p}_{i,s}}\right)\frac{\partial \tilde{p}_{i,s}}{\partial z_{i,s}} = T^2\frac{(\tilde{p}_{i,s} - \tilde{p}_{i,t})}{\tilde{p}_{i,s}(1 - \tilde{p}_{i,s})}\frac{\partial \tilde{p}_{i,s}}{\partial z_{i,s}}. \tag{S5}$$

Thus, from Eq. (S4), Lemma 1 and for large T, Eq. (S5) reduces to:

$$\lim_{T\to\infty} \frac{\partial \mathcal{L}_{\text{KD}}}{\partial z_{i,s}} = p_{i,s} - p_{i,t},$$

which completes the proof. □

Lemma 3. *The derivative of the total loss function $\mathcal{L}_{tot}$ w.r.t. the parameters $\mathbf{W}_s$ of the student network lies in the span of $\mathbf{X}$, and is given by:*

$$\frac{\partial \mathcal{L}_{tot}}{\partial \mathbf{W}_s} = \sum_{i=1}^{N} \left(p_{i,s} - \{\alpha p_{i,t} + (1-\alpha) y_i\}\right)\mathbf{x}_i.$$

Proof. The proof follows directly from Lemma 2. □

Theorem 1. *Let $\mathbf{X} \in \mathbb{R}^{d\times N}$ be a data matrix satisfying Assumption 1, and $\mathbf{W}_s$ and $\mathbf{W}_t$ represent the parameters of the student and the teacher networks, respectively. Then, under Assumption 2 and using the gradient-descent algorithm, the parameters $\mathbf{W}_s$ of the student network converge to:*

$$\mathbf{W}_s \approx \alpha\mathbf{W}_t + 4(1-\alpha)\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{Y}_{1/2},$$

where $\mathbf{Y}_{1/2} \coloneqq \left[y_i - \frac{1}{2}\right]_{i=1}^{N}$ is an N-dimensional vector.

Proof. First observe that the minimum value of the total loss function in Eq. (2) is finite. Moreover, the total loss function is convex in the parameters of the student network. Thus, any gradient-based descent algorithm with suitable step-size will converge to the optimizer asymptotically fast.

We now characterize the set of optimizers. Recall that the first-order condition of optimality implies:

$$\frac{\partial \mathcal{L}_{\text{tot}}}{\partial \mathbf{W}_s} = 0 \implies \sum\nolimits_{i=1}^{N} \left(p_{i,s} - \{\alpha p_{i,t} + (1-\alpha) y_i\}\right)\mathbf{x}_i = 0,$$

where the last equality follows from Lemma 3. Since the vectors $\{\mathbf{x}_i\}$ are linearly independent (see Remark 1), the above equality holds if:

$$p_{i,s} - \{\alpha p_{i,t} + (1-\alpha)y_i\} = 0, \;\; \forall i \in \{1, \dots, N\}. \tag{S6}$$

Expanding Eq. (S6) in terms of logits $z_{i,s}$ leads to:

$$\frac{1}{1+e^{-z_{i,s}}} = \alpha \frac{1}{1+e^{-z_{i,t}}} + (1-\alpha)y_i \implies \frac{1}{1+\frac{1-\frac{z_{i,s}}{2}}{1+\frac{z_{i,s}}{2}}} \approx \alpha \frac{1}{1+\frac{1-\frac{z_{i,t}}{2}}{1+\frac{z_{i,t}}{2}}} + (1-\alpha)y_i, \tag{S7}$$

where the last equation follows from Padé approximation. Rearranging the terms in Eq. (S7), and using the fact that $z_{i,s} = \mathbf{W}_s^\top \mathbf{x}_i$ and $z_{i,t} = \mathbf{W}_t^\top \mathbf{x}_i$, one obtains:

$$\left(\mathbf{W}_s - \alpha \mathbf{W}_t\right)^\top \mathbf{x}_i = 4(1-\alpha)\left(y_i - 1/2\right).$$

Since the above condition holds for every $i \in \{1, \dots, N\}$, the vector form of it can be written as:

$$\mathbf{X}^\top (\mathbf{W}_s - \alpha \mathbf{W}_t) = 4(1-\alpha)\mathbf{Y}_{1/2}, \tag{S8}$$

which is an underdetermined system of linear equations whose least-norm solution is given by:

$$\mathbf{W}_s = \alpha \mathbf{W}_t + 4(1-\alpha)\mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{Y}_{1/2}, \tag{S9}$$

which completes the proof. □

2.2 PROOF OF THEOREM 2

Theorem 2. *Let Assumptions 1-2 hold. Let t_c and t_{uc} be two teacher classifiers with output probabilities $\{p_{i,t_c}\}$ and $\{p_{i,t_{uc}}\}$, respectively. Also, let s_c, s_{uc} depict two student classifiers trained independently from the corresponding teacher classifiers t_c and t_{uc} through KD, with output probabilities $\{p_{i,s_c}\}$ and $\{p_{i,s_{uc}}\}$, respectively. Furthermore, assume that the teacher classifier t_c is well calibrated, then the student classifier s_c is also well calibrated. Conversely, if the teacher classifier t_{uc} is uncalibrated, the corresponding student classifier s_{uc} mimics a similar behavior, i.e.,*

$$\sum\nolimits_{i=1}^{N} p_{i,s_c} = \sum\nolimits_{i=1}^{N} y_i, \;\; \textit{and} \;\; \sum\nolimits_{i=1}^{N} p_{i,s_{uc}} \neq \sum\nolimits_{i=1}^{N} y_i.$$

Proof. From Eq. (S6), the first-order condition for optimality for a student s trained from a teacher t through KD reads:

$$\sum\nolimits_{i=1}^{N} p_{i,s} = \alpha \sum\nolimits_{i=1}^{N} p_{i,t} + (1-\alpha) \sum\nolimits_{i=1}^{N} y_i,$$

which can be rewritten as

$$\sum_{i=1}^{N} (p_{i,s} - y_i) = \alpha \sum_{i=1}^{N} (p_{i,t} - y_i).$$

Thus for the same value of $\alpha \in (0,1)$, if the teacher classifier s_c is well calibrated, then

$$\sum_{i=1}^{N} (p_{i,s_c} - y_i) = \alpha \left(\sum_{i=1}^{N} (p_{i,t_c} - y_i) \right) = 0,$$

where, the last equality follows from well calibration of the teacher classifier. On the other hand,

$$\sum\nolimits_{i=1}^{N} (p_{i,s_{uc}} - y_i) = \alpha \sum\nolimits_{i=1}^{N} (p_{i,t_{uc}} - y_i) \neq 0 \implies \sum\nolimits_{i=1}^{N} p_{i,s_{uc}} \neq \sum\nolimits_{i=1}^{N} y_i,$$

which completes the proof. □

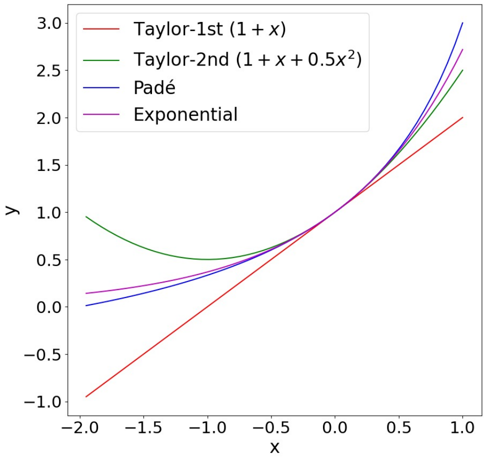

Figure S1: Padé vs Taylor for a simple exponential function. Note that Padé approximants offer superior reliability compared to the extensively used Taylor approximants.

2.3 PADÉ VS TAYLOR APPROXIMANTS

Employing approximants to derive theoretical outcomes in `DNNs` is commonplace due to the intricacies of dealing with highly nonlinear equations. We illustrate the difference between Padé and Taylor's approximation as follows: Padé approximants have a wider range of convergence than the corresponding Taylor series, and can even converge where the Taylor series does not. A simple example of Padé approximant is, $e^x = \frac{e^{0.5x}}{e^{-0.5x}} \approx \frac{(1+0.5x)}{(1-0.5x)} = (1+0.5x)(1-0.5x)^{-1}$, which for $|x| < 2$ can further be expanded to $e^x \approx (1+0.5x)(1-0.5x)^{-1} = (1+0.5x)(1+0.5x+0.25x^2+\dots)$. Thus, despite using first-order approximations for both the numerator and denominator terms, the above Padé approximant very closely follows the original exponential function. This is in contrast to Taylor's expansion, and even a second-order Taylor's expansion does not mimic the exponential function, except for a very small interval around the origin. Please refer to Figure S1 for further details.

This is precisely why we restrict using Padé approximants in our theoretical exploration since they are still potentially non-divergent in regimes even when $z_{i,t}$ and $z_{i,s}$ are not vanishingly small. It must also be remarked that **exact characterization of weights of student network is a theoretically hard problem**, and such practical approximations are useful to obtain important theoretical insights.

3 RATIONALE ON THE SUPERIOR PERFORMANCE OF OUR KD(C) FRAMEWORK

The essence of Theorem 2 lies in its assertion that uncalibrated teachers can only transfer their lack of calibration to their student counterparts, whereas calibrated teachers enable the distillation of calibrated students. This theorem underscores the crucial significance of utilizing calibrated teachers in the knowledge distillation process. In light of this observation, we advocate for a novel approach to achieving accurate and calibrated models: calibrating a model through distillation from another model that is already calibrated. To validate the efficacy of this approach, we conducted an extensive series of experiments, showcasing the capabilities of our framework, KD(C). Our experimental results provide compelling evidence that KD(C) yields student models characterized by two key attributes: dynamic calibration at the sample level and semantic calibration. These findings substantiate the effectiveness of our proposed framework in achieving both sample-level and semantic calibration in student models.

3.1 CLASSIFICATION OF LABEL SMOOTHING

Standard/static label smoothing. Label Smoothing (`LS`) serves as a regularization technique designed to address potential inaccuracies within datasets. It recognizes that maximizing the likelihood directly, denoted as $P(y|\mathbf{x})$, may be detrimental due to the possibility of errors in the training labels.

To mitigate this issue, `LS` introduces controlled noise into the labeling process. In essence, `LS` operates as follows: Given a small constant value ϵ, it considers the training label y to be correct with a probability of $(1-\epsilon)$ and incorrect otherwise. Specifically, in the context of a softmax model with k outputs, it replaces the traditional binary classification targets of 0 and 1 with modified targets. These modified targets consist of $\frac{\epsilon}{(k-1)}$ for incorrect labels and $(1-\epsilon)$ for correct labels Szegedy et al. (2015); Müller et al. (2019). This approach ensures that all output probabilities undergo uniform regularization, thereby helping to combat overfitting and improve model generalization.

Adaptive Label Smoothing. In this method the level of regularization applied to training labels, which are typically one-hot encoded, is dynamically adjusted based on the network's output probabilities for different classes Cheng & Vasconcelos (2022); Hebbalaguppe et al. (2022); Park et al. (2023). This method is found to be more beneficial than conventional static label smoothing (`LS`) proposed in Szegedy et al. (2015).

Conditional Label Smoothing. In this method the training labels go through selective modifications based on specific criteria, such as the application of margin-based penalties Liu et al. (2022). This approach places its emphasis on and applies regularization solely to the probabilities that exhibit miscalibration, thereby demonstrating enhanced calibration capabilities.

3.2 VISUALIZATION OF PENULTIMATE LAYER'S ACTIVATIONS REVEAL KD(C) USING DYNAMIC REGULARIZATION WORKS BETTER THAN STATIC REGULARIZATION

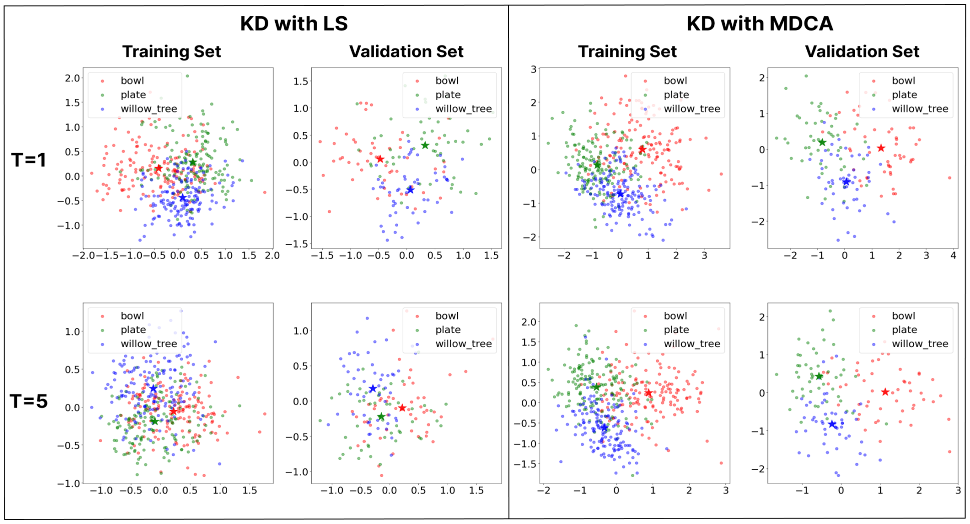

Figure S2: **Visualization of penultimate layer's activations (`Teacher = ResNet56, Student = ResNet8, Dataset = CIFAR100`).** We train `ResNet8` using calibration techniques: `KD with LS (Left column)` and `KD with MDCA(Right Column)`. We follow the same setup and procedure used in papers [Müller et al. (2019); Shen et al. (2021)] We use two semantically similar classes (`bowl`, `plate`) and one semantically dissimilar class (`willow_tree`). A '*' in the plot for each cluster represents its cluster's centroid. A well-calibrated teacher can effectively capture the inter-class relationships and serve as a reliable dynamic label smoothing prior such as `MDCA` Hebbalaguppe et al. (2022). Observe that the classes: `bowl` and `plate` are visually similar and hence the penultimate visualizations of these classes should be closer than the dissimilar class: `willow_tree`. As the temperature T is increased the similar classes diffuse into one in the case of `KD with LS` while `KD with MDCA` offers better separation, retaining the semantic similarity while being well separated from the dissimilar class.

Penultimate Visualization. Müller et al. (2019) introduced this visualization technique wherein they projected the penultimate activations onto the hyperplane defined by the template vectors (weight vectors) corresponding to the selected classes (three classes) for visualization.

Systematic diffusion. The concept of "systematic diffusion," introduced by Chandrasegaran et al. (2022), was developed to address discrepancies observed in prior studies, particularly the contra-

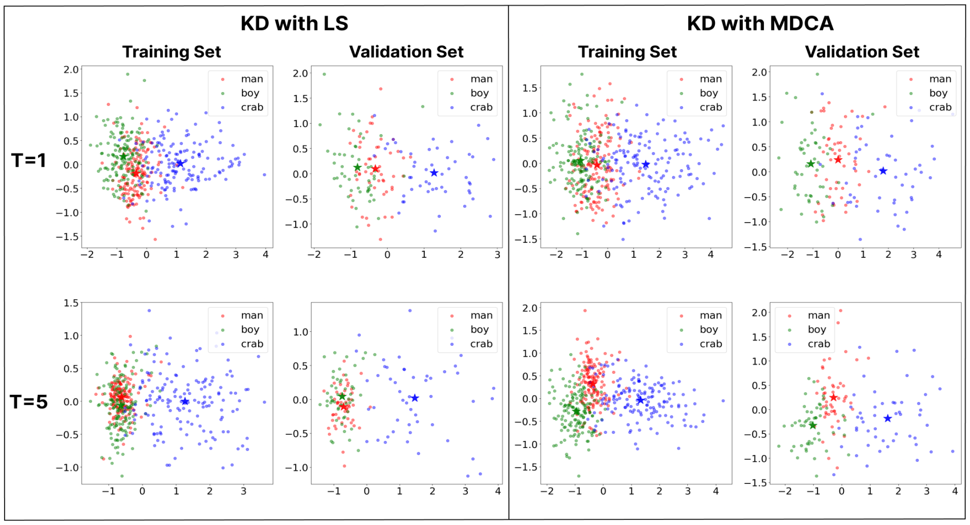

Figure S3: **Visualization of penultimate layer's activations (`Teacher = ResNet56, Student = ResNet8, Dataset = CIFAR100`)**. We train `ResNet8` using calibration techniques: `KD with LS (Left column)` and `KD with MDCA(Right Column)`. We follow the same setup and procedure used in papers [Müller et al. (2019); Shen et al. (2021)] We use two semantically similar classes (`man`, `boy`) and one semantically dissimilar class (`crab`). A '*' for each cluster represents its cluster's centroid. A well-calibrated teacher can effectively capture the inter-class relationships and serve as a reliable dynamic label smoothing prior such as `MDCA` Hebbalaguppe et al. (2022). Observe that the classes: `man` and `boy` are visually similar and hence the penultimate visualizations of these classes should be closer than the dissimilar class: `crab`. As the temperature T is increased the similar classes diffuse into one in the case of `KD with LS` while `KD with MDCA` offers better separation, retaining the semantic similarity while being well separated from the dissimilar class.

dictions between Shen et al. (2021) and the insights presented in `LS` literature Müller et al. (2019). This concept aims to elucidate the compatibility of label smoothing with knowledge distillation. The findings from Chandrasegaran et al. (2022)'s work indicate that when `KD` is conducted at elevated temperatures from a teacher model trained with `LS`, it results in a systematic shift in the relationships between classes. Specifically, for semantically similar classes, the inter-cluster distance decreases, while for the remaining classes, it increases relatively. Importantly, this diffusion of classes is not random; rather, it follows a systematic pattern.

In Fig. S2 and Fig. S3, we provide visual evidence of the limitations associated with `LS`-trained teachers compared to `MDCA` teachers Hebbalaguppe et al. (2022). These Penultimate layer visualizations, inspired by the work of Shen et al. (2021), reveal that semantically similar classes experience systematic diffusion when using `LS`, whereas this phenomenon is not observed with `MDCA` calibration. This observation substantiates our recommendation to opt for dynamic smoothing regularization techniques such as `MDCA`.

Notably, we notice a trend where distilled student models are most calibrated when the distillation temperature (T) is approximately 1. We hypothesize that increasing T leads to the destruction of discriminating features, as outlined by Chandrasegaran et al. (2022), due to systematic diffusion among highly similar classes as seen in the penultimate representations. These discriminating features are crucial for achieving calibration by resolving confusion among similar classes. However, as T increases further, we simultaneously amplify the relationships between somewhat related classes Tang et al. (2020), while diminishing the relationships between very similar classes. This nuanced understanding highlights the intricate interplay between temperature, class relationships, and calibration, shedding light on the optimal conditions for achieving calibration in `KD` scenarios.

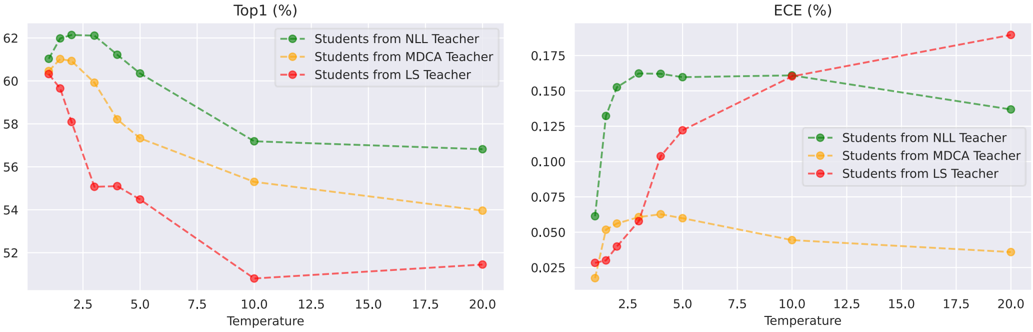

Figure S4: **[Study of ECE variablity in case of KD(C), specifically we consider KD with LS and KD with MDCA and study variation of accuracy and calibration as a function of temperature]**: Comparison of Top 1% accuracy and ECE when train-time calibration method is changed from Label Smoothing Szegedy et al. (2015) and MDCA Hebbalaguppe et al. (2022): We use ResNet56 teacher on CIFAR100 and distill to ResNet8. Note that MDCA-based students have lower accuracy than NLL, however, ECE is largely stable when temperature T is varied.

4 ILLUSTRATION OF THE GENERALITY OF KD(C) FRAMEWORK

Fig. S5 presents a novel framework KD(C) that leverages calibrated teachers through KD to produce DNNs with the least calibration error. This comprehensive framework encompasses the full spectrum, enabling models with varying capacity (smaller/larger) to distill student models with the least calibration error and better accuracy compared to the SOTA post-hoc/train-time calibration methods.

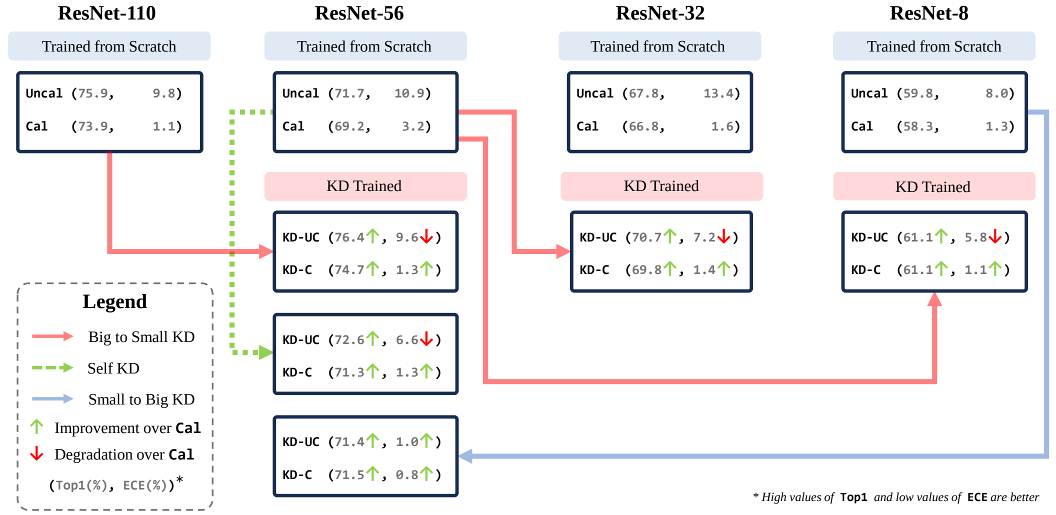

Figure S5: **An illustration of KD(C) framework's generality using calibration method as MDCA.** We can distill a calibrated student from a large teacher and vice-versa yielding SOTA calibration without any trade-offs in accuracy. "Uncal" and "Cal" mean uncalibrated and calibrated teachers trained using `NLL` and a recent `SOTA` calibration technique Hebbalaguppe et al. (2022) respectively. KD(UC) and KD(C) refer to students distilled using "Uncal" and "Cal" teachers respectively. Going from a large calibrated teacher to a smaller student yields `SOTA` calibrated student, with an additional boost in accuracy (E.g., compare `ResNet56` "Trained from scratch" with `ResNet56` "KD-trained" student from ResNet110). Self-distillation and going from a smaller teacher to a bigger student also have a similar effect on calibration, however, the gains in accuracy are comparable to respective models trained from scratch. The above results are on `CIFAR100`.

5 ADDITIONAL RESULTS

5.1 RELIABILITY DIAGRAMS AND CONFIDENCE HISTOGRAMS

Reliability diagrams serve as effective visual aids for assessing the calibration of `DNNs`. They involve partitioning the predicted probabilities generated by `DNNs` into a predetermined number of bins along the x-axis. The y-axis represents the normalized count of events (e.g., class = "dog") within each bin. A well-calibrated model will exhibit points that closely align with the main diagonal, spanning from the bottom left to the top right of the plot. Reliability diagrams corresponding to Fig. S6 are included to show that KD(C) variants obtain near `SOTA` results.

5.2 EFFECT OF HYPER-PARAMETERS LIKE T (TEMPERATURE) AND α

We investigate the influence of hyperparameters T and α on both calibrated and uncalibrated teacher models, as visually depicted in Fig. S7 (big-to-small) and Fig. S8 (small-to-big).

(a) Big teacher, Small student: In this scenario as we increase the value of α, we witness an intuitive rise in calibration. However, this effect is predominantly noticeable for small values of T (depicted in the bottom-right region of Fig. S7). Generally, the calibration errors (`ECE`) incurred by distilling students from a calibrated teacher tend to be markedly lower than those distilled from an uncalibrated teacher, as evident from the bottom row in Fig. S7. **(b) Small teacher, Big student**: Initially, we observe an expected trend: as α increases (signifying a higher dependence on the teacher), accuracy experiences a decrease. This outcome arises from the process of distillation from a weaker teacher. However, when distilling from a calibrated teacher, we discern that elevating α results in enhanced calibration. Nevertheless, this improvement in calibration is accompanied by a trade-off with accuracy.

Notably, we find that optimal calibration is generally achieved when $T \approx 1$, regardless of the size of the teacher model employed. This observation aligns with the findings presented in Stanton et al.

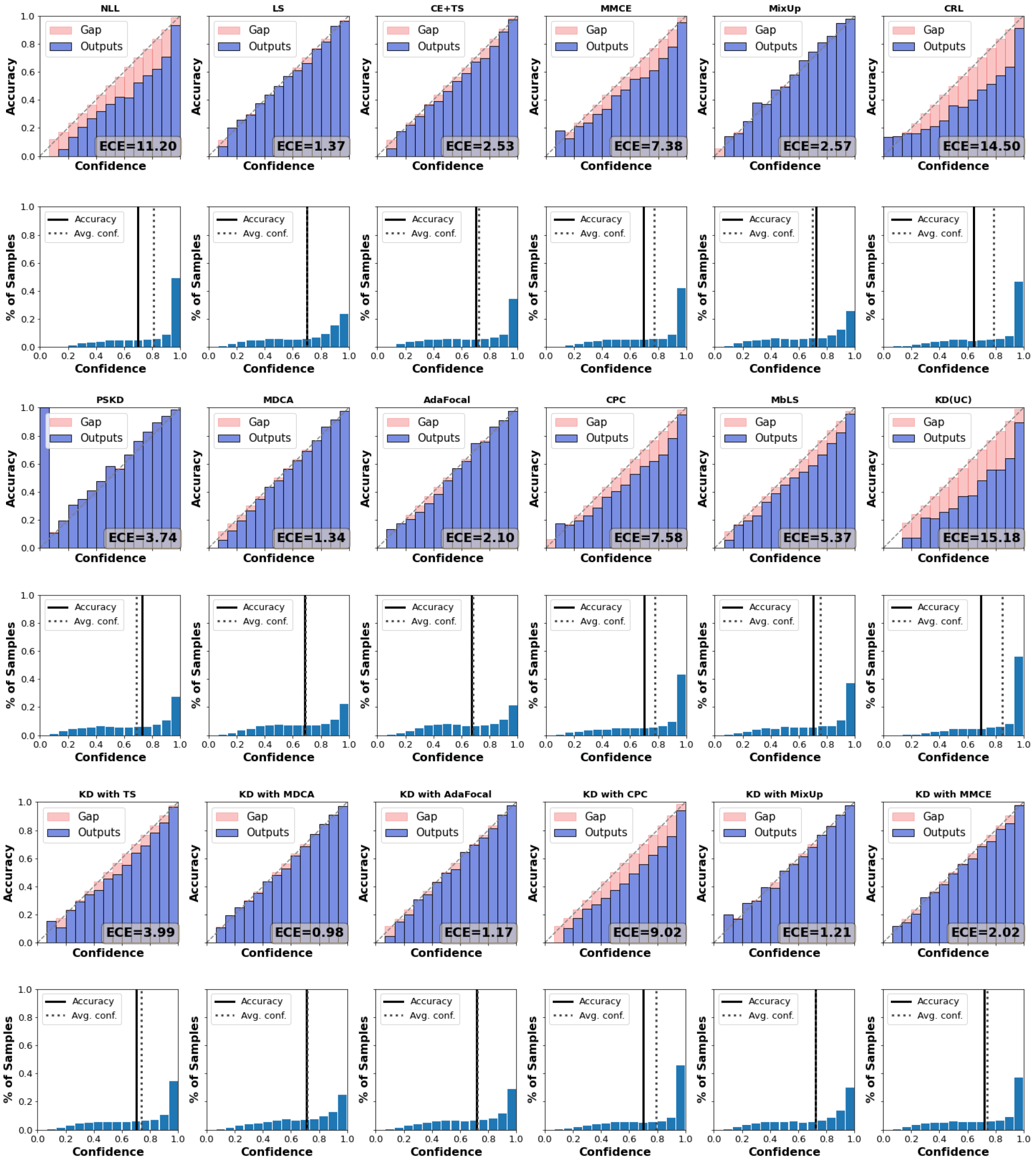

Figure S6: Reliability Plots for `top-5` KD with **(Ours)** techniques on WideResNet-40-1 on `CIFAR100`. Teacher used: WideResNet-40-2. **KD(C)** framework achieves competitive calibration results for `KD with MDCA`, `KD with AdaFocal` and `KD with MixUp`.

(2021), which suggest that maximizing fidelity with the teacher model yields the best transfer of properties.

5.3 CALIBRATION PERFORMANCE UNDER DATASET DRIFT

`DNNs` are found to be over-confident and highly uncalibrated under dataset/domain shift Tomani et al. (2020). We investigate the robustness of our method KD(C) by examining the degradation in calibration under natural/non-semantic shift (images with the same label but different distribution). We carry out this study for `ResNet56` pre-trained on the `CIFAR100` dataset along with various calibration techniques and report the evaluation results on `CIFAR100-C` Hendrycks & Gimpel (2016) in Fig. S9. We used `ResNet56` models that were trained with `ResNet110` as teacher

Calibration Method	**Top1 (%)** ↑	**ECE (%)** ↓	**SCE (%)** ↓	**AECE (%)** ↓
NLL	50.43	13.72	0.24	13.72
LS Szegedy et al. (2015)	51.20	3.84	**0.19**	3.88
CE with TS Guo et al. (2017)	50.43	13.72	0.24	13.72
MMCE Kumar et al. (2018)	50.30	11.32	0.21	11.32
MixUp Thulasidasan et al. (2019)	52.02	4.74	**0.19**	4.73
PSKD Kim et al. (2021)	**53.66**	13.27	0.21	13.27
MDCA Hebbalaguppe et al. (2022)	46.81	1.52	**0.19**	**1.11**
CPC Cheng & Vasconcelos (2022)	51.27	12.01	0.21	12.01
MbLS Liu et al. (2022)	50.11	8.87	0.20	8.87
KD(UC)	49.31	4.20	0.20	4.20
Ours (KD with MDCA)	45.79	**0.85**	0.21	1.17
Ours (KD with LS)	49.69	2.69	**0.19**	2.68
Ours (KD with MbLS)	49.33	2.86	0.20	2.89

Table T1: **[Self-distillation]** using MobileNetV2 feature extractor on `Tiny-ImageNet` dataset. Note that the main paper reported self-distillation results using the MobileNetV2 feature extractor on the `CIFAR10` dataset. `Top3` best KD(C) variants are reported. `KD with MDCA` variant of KD(C) achieve competitive calibration results with `SOTA`.

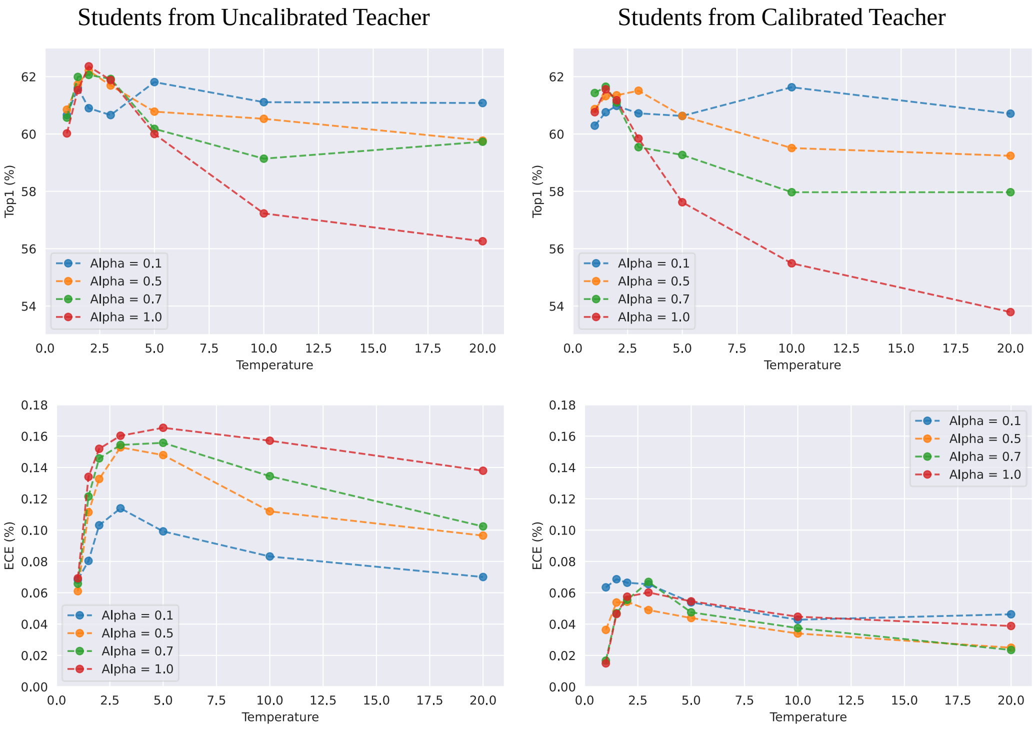

Figure S7: We study the effect of varying temperature, T and distillation weight α, on `ECE` and top 1% accurracy when `ResNet56` teacher model is used and `ResNet8` as student on `CIFAR100` dataset. Observe the optimal values of `ECE` and top 1% accuracy when T is set around 1. For calibration `KD with MDCA` was used.

for KD(UC) and KD(C). We observe KD(UC) and KD(C) achieve the highest accuracy across all severities, with the latter achieving close to the best `ECE` (`LS` achieves best `ECE`), however, KD(C) achieves the best `AUROC` score in comparison to any other calibration technique. This indicates, KD(C) is better across all metrics measuring reliability (be it calibration or refinement, while also giving an additional boost in accuracy).

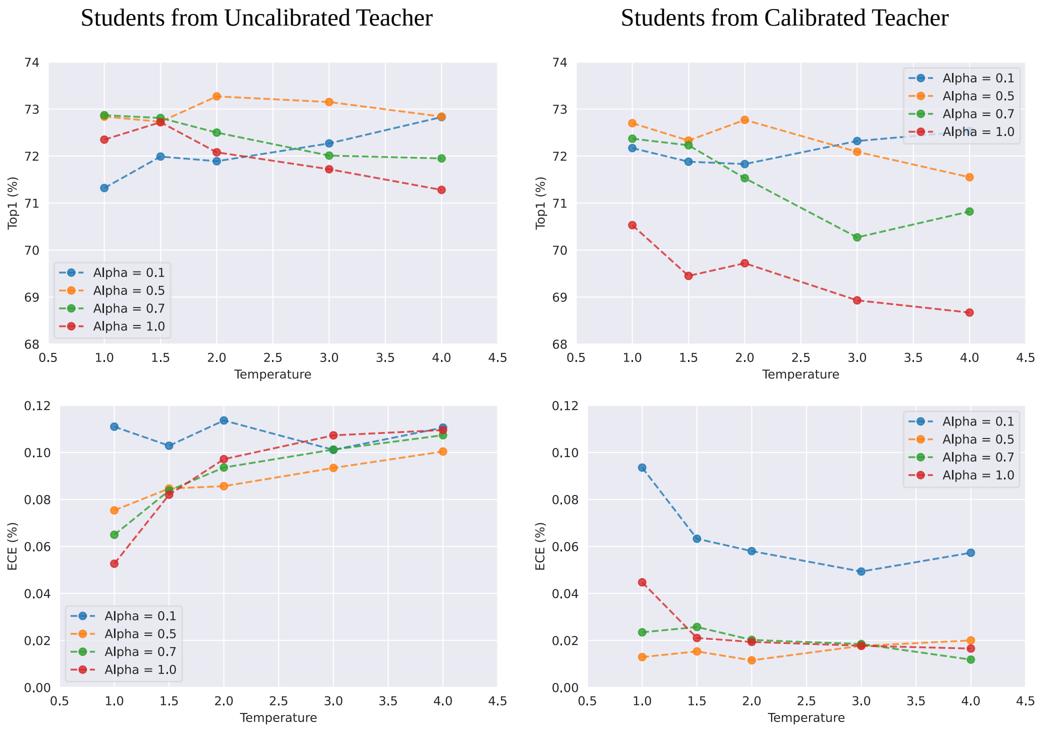

Figure S8: We study the effect of varying temperature, T and distillation weight α, on `ECE` and top 1% accuracy when `ResNet32` teacher model is used and `ResNet56` as student on `CIFAR100` dataset. `KD with MDCA` was used for calibration.

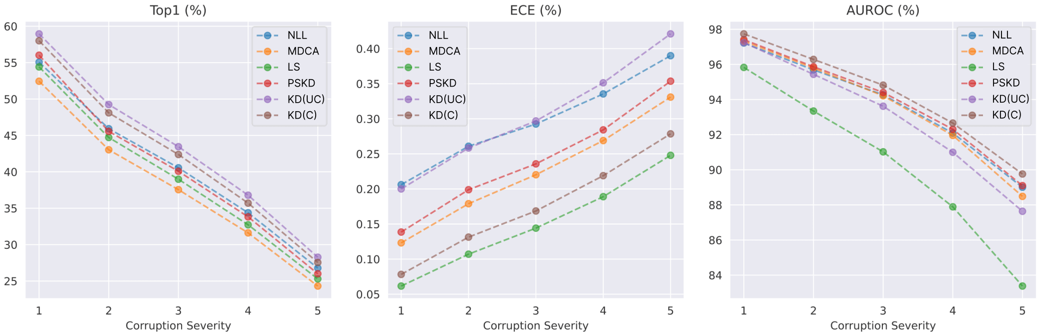

Figure S9: Robustness to corruption, tested on `CIFAR100-C` dataset Hendrycks & Gimpel (2016) using `ResNet-56`. KD(UC) and KD(C) were trained using `ResNet-110` as a Teacher. Note that KD(C) provides a good trade-off between accuracy and calibration, at the same time achieving the highest AUROC (even though LS outperforms KD(C) by a tiny margin in terms of calibration, KD(C) has significantly better AUROC and accuracy. AUROC indicates better inter-class separability in classifiers thereby enhancing trustworthiness in addition to calibration benefits. KD(C) uses `KD with MDCA` variant.

6 TRAINING DETAILS

In this section, we provide a detailed summary of the hyperparameters and training techniques used, in order to ensure reproducibility. All models have been trained on 40GB Nvidia A100 GPUs.

Dataset	Teacher Calibration	Teacher	Top1 (%)	ECE (%)	SCE (%)	ACE (%)	Student	Top1 (%)	ECE (%)	SCE (%)	ACE (%)
CIFAR100	NLL	WRN-40-2	74.10	13.42	0.32	13.42	WRN-40-1	69.60	15.18	0.37	15.18
	LS		74.67	2.44	0.21	2.21		70.54	1.22	0.21	1.20
	CE+TS		74.10	2.18	0.20	2.15		70.07	4.00	0.21	4.00
	MMCE		73.04	5.21	0.21	5.14		72.08	2.02	0.19	1.95
	MixUp		77.46	1.50	0.19	1.43		72.48	1.21	0.20	1.17
	CRL		70.71	12.20	0.32	12.11		69.76	7.21	0.23	7.04
	MDCA		73.74	1.36	0.19	1.26		71.07	0.98	0.20	1.10
	AdaFocal		73.24	2.43	0.20	2.31		71.70	1.19	0.19	1.34
	CPC		75.09	11.06	0.28	10.99		70.00	9.02	0.26	9.01
	MbLS		73.54	5.53	0.22	5.50		71.44	3.70	0.22	3.41
	NLL	RNXT-18x4	63.69	17.20	0.41	17.20	MNV2	66.82	5.40	0.22	5.36
	LS		63.89	5.00	0.25	5.37		66.63	2.48	0.24	2.50
	CE+TS		63.69	2.73	0.23	2.74		67.22	1.63	0.19	1.58
	MMCE		62.40	6.53	0.24	6.56		66.24	1.47	0.19	1.64
	MixUp		65.57	3.15	0.24	3.17		69.92	2.17	0.24	2.10
	CRL		52.98	20.21	0.51	20.21		64.17	2.89	0.23	2.79
	MDCA		63.70	2.14	0.22	2.23		67.17	1.10	0.20	1.17
	AdaFocal		64.55	6.19	0.24	6.15		66.64	1.55	0.20	1.43
	CPC		63.20	8.98	0.28	8.97		67.83	0.88	0.19	0.95
	MbLS		64.42	12.89	0.32	12.88		67.50	3.21	0.20	3.22
CIFAR10	NLL	MNV2	89.87	3.30	0.75	3.28	DN	90.2	2.17	0.60	2.13
	LS		89.60	7.10	1.78	6.75		92.65	4.27	1.23	4.13
	CE+TS		89.90	0.98	0.40	0.77		93.25	0.62	0.43	0.54
	MMCE		89.38	1.20	0.51	0.94		92.13	0.81	0.45	0.69
	MixUp		89.57	9.42	2.07	9.41		91.19	5.20	1.33	5.01
	CRL		90.31	2.92	0.72	2.81		92.24	2.58	0.68	2.54
	MDCA		88.74	0.99	0.46	0.80		90.90	0.53	0.45	0.51
	AdaFocal		88.98	0.79	0.44	0.86		91.68	0.54	0.34	0.61
	CPC		89.26	3.47	0.79	3.44		93.14	0.74	0.43	0.85
	MbLS		89.86	2.83	0.69	2.78		93.1	0.61	0.38	0.40

Table T3: Comparison of evaluation metrics of Teacher-Student pairs. Observe that calibration transfer takes place from a calibrated teacher more or less to a student. The minor differences of calibration values can be attributed to the capacity gap between the teacher and student. WRN: WideResNet, RNXT: ResNeXt, DN: DenseNet. Number of parameters: WRN-40-1: 0.56M ; WRN-40-2: 2.24M ; MNV2: 2.25M ; RNXT-18x4: 25.46M ; DN: 6.95M

The code was written using the PyTorch framework. We make use of automatic mixed precision training in order to reduce training time. We borrow some code from the official implementation of Hebbalaguppe et al. (2022); Mukhoti et al. (2020); Yuan et al. (2021).

For `CIFAR10/100` datasets, we train all `ResNets / WideResNets / ResNeXt-18x4` models using a learning rate of 0.1 for 160 epochs. The learning rate is decayed by a factor of 10 at epoch 80 and 120. We use SGD optimizer with momentum 0.9 and weight decay of $5e-4$. We use a batch size of 128. For the larger models like `ResNet-110`, we train them using a learning rate of 0.05 for 240 epochs. The learning rate is decayed by a factor of 10 at epoch 150, 180 and 210. We use SGD optimizer with momentum 0.9 and weight decay of $5e-4$. We use a batch size of 64. ConvNet2 model has been trained just like all other models, except for the learning rate which is set to 0.01, without a learning rate decay scheduler.

For `Tiny-ImageNet` dataset, all models are trained using a maximum learning rate of 0.1 with a cosine annealing learning rate with a warmup of 1000 steps with minimum learning rate $1e-5$. The weight decay and momentum are $5e-4$ and 0.9 respectively. We train the models for 100 epochs with a batch size of 128.

For training students using KD, we use the same hyper-parameters for the respective datasets. For big-to-small KD (e.g. `WideResNet-40-2`$\rightarrow$ `WideResNet-40-1`), we grid search T (temperature) and α (distillation weight) in the ranges $\{1, 1.5, 2, 3, 4, 5, 10, 20\}$ and $\{0.9, 1.0\}$ respectively. For small-to-big KD and self-distillation (e.g. `MobileNetV2` $\downarrow$ `DenseNet-121`, `MobileNetV2` $\rightarrow$ `MobileNetV2`), we grid search T and α in the ranges $\{1, 1.5, 2, 3, 4\}$ and $\{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ respectively.

For baselines, we use the recommended hyperparameters as suggested by the respective authors Hebbalaguppe et al. (2022); Szegedy et al. (2015); Kim et al. (2021); Kumar et al. (2019); Cheng & Vasconcelos (2022); Thulasidasan et al. (2019); Liu et al. (2022); Moon et al. (2020); Ghosh et al.

Dataset	Teacher Model	Student Model	Temperature	Top1 (%)	ECE (%)	SCE (%)	ACE (%)
CIFAR100	WRN-40-2	WRN-40-1	0.10	71.06	26.40	0.55	26.39
			0.20	71.06	23.79	0.52	23.79
			0.50	71.06	15.74	0.38	15.74
			0.75	71.07	8.44	0.27	8.44
			1.00	**71.06**	**0.98**	**0.20**	**1.10**
			1.25	71.06	8.60	0.27	8.60
			1.50	71.06	18.00	0.42	18.00
			1.75	71.06	27.11	0.58	27.11
			2.00	71.06	35.22	0.74	35.22
			2.25	71.06	41.99	0.88	41.99
			2.50	71.06	47.39	0.99	47.39
			2.75	71.06	51.60	1.07	51.60
			3.00	71.06	54.86	1.12	54.86
			3.25	71.06	57.37	1.13	57.37
			3.50	71.06	59.32	1.11	59.32
			3.75	71.06	60.86	1.07	60.86
			4.00	71.06	62.08	1.00	62.08
			4.25	71.06	63.06	0.92	63.06
			4.50	71.06	63.86	0.81	63.86
			4.75	71.06	64.52	0.69	64.52
			5.00	71.06	65.07	0.56	65.07
	RNXT-18x4	MNV2	0.10	67.17	29.85	0.63	29.85
			0.20	67.17	26.81	0.58	26.80
			0.50	67.17	17.33	0.41	17.33
			0.75	67.17	8.96	0.28	8.96
			1.00	**67.17**	**1.10**	**0.20**	**1.17**
			1.25	67.18	9.60	0.28	9.60
			1.50	67.18	19.24	0.44	19.24
			1.75	67.17	28.23	0.62	28.23
			2.00	67.17	35.97	0.78	35.97
			2.25	67.17	42.21	0.91	42.21
			2.50	67.17	47.04	1.00	47.04
			2.75	67.17	50.70	1.05	50.70
			3.00	67.18	53.48	1.06	53.48
			3.25	67.18	55.59	1.05	55.59
			3.50	67.17	57.20	1.01	57.20
			3.75	67.17	58.47	0.94	58.47
			4.00	67.17	59.47	0.85	59.47
			4.25	67.17	60.27	0.75	60.27
			4.50	67.17	60.92	0.64	60.92
			4.75	67.17	61.46	0.52	61.47
			5.00	67.17	61.92	0.40	61.92

Table T4: **[Effect of `TS` on KD(C)]:**The student is calibrated by distilling from an `MDCA` calibrated teacher `KD with MDCA` (a variant of KD(C)). The table shows that further temperature scaling (`TS`) does not impact the models trained with `KD with MDCA` as they are calibrated to start with. Parameters: WRN-40-1: 0.56M ; WRN-40-2: 2.24M ; MNV2: 2.25M ; RNXT-18x4: 25.46M

(2022), i.e. for LS Szegedy et al. (2015), we use smoothing of 0.1; for PSKD Kim et al. (2021) we use $\alpha = 0.8$; for MixUp Zhang et al. (2018) the mixup hyperaparameter was taken as 0.4 as it was reported to be the best by the authors, for MDCA Hebbalaguppe et al. (2022), we grid search for the best performing $\beta \in \{1, 5, 10\}$ and $\gamma \in \{1, 2, 3, 4, 5\}$; For MMCE, we grid search for the best performing $\beta \in \{1, 2, 3, 4, 5\}$.

We provide the metrics for various teachers trained from scratch and used throughout the paper in Tab. T3. These teachers have been used to train various student models for KD(UC) and KD(C) variants mentioned in the paper.

7 REPRODUCIBILITY

In the spirit of reproducible research, we intend to make the source code available post-acceptance. To aid reviewers, the source code for our approach is attached along with the supplemental material. Details of our setup and implementation of the baselines can be found at: `Code/README.md` folder.

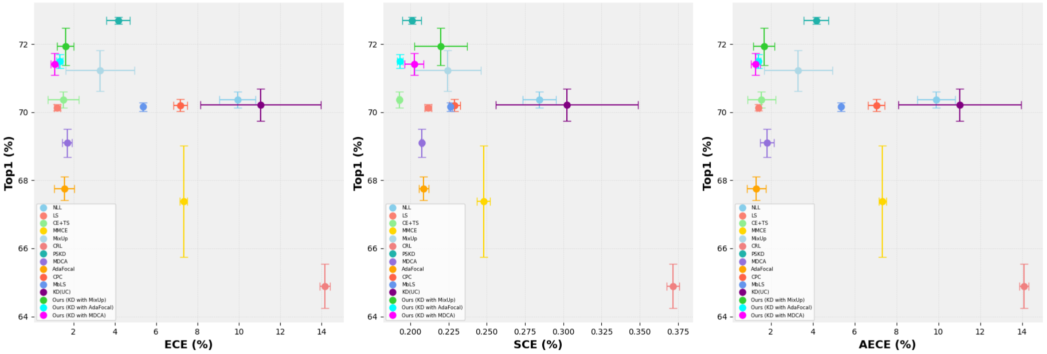

Figure S7: **Comparative study of accuracy vs. calibration trade-offs associated with existing calibration techniques and ours (Top-left is most preferred)**: The mean and one standard scatter error bars for `Top1`, `ECE` and `SCE` of `WideResNet-40-1` trained on `CIFAR100` using `SOTA` calibration techniques. `WideResNet-40-2` was used as Teacher for KD(UC) and the proposed, KD(C) variants. Note: KD(C) variants (magenta, cyan, and green) achieve the best results in terms of `ECE`, `ACE` and `SCE`, along with slight boosts in `Top1` (an inherent KD-property). Further, the lower variances emphasize the reliability of KD(C) variants. All plots were generated by training `WideResNet-40-1` models through every calibration technique on 3 runs.

8 LIMITATIONS

While our work paves way to create optimal lightweight models that are both accurate and calibrated, it is important to acknowledge three potential limitations that we plan to address in future research - (a) principled approach to select hyperparameters, such as the temperature T, distillation weight α, calibration regularization coefficient β, and characterization of optimal student-teacher capacity difference for best calibration, (b) extending theoretical insights to general nonlinear networks, (c) benchmarking KD(C) on natural language processing (NLP) tasks, particularly when the teacher networks belong to the family of large language models (LLMs). This is particularly challenging due to the unavailability of adequate computational resources.

9 BROADER IMPACT

Bigger DNN models aren't necessarily better models. From a deployment standpoint, the size of the weights affects the inference time and storage constraints on edge devices which is crucial in applications such as augmented reality and robotics. Our proposed algorithm has the potential to be employed in trustworthy lightweight models on the edge. In our endeavor to deploy lightweight models that are also reliable, we delve into the realm of knowledge distillation, extending its traditional function of transferring accuracy from teacher networks to student networks. Through this exploration, we have discovered a novel approach to calibrating models effectively. We present, arguably for the first time, compelling evidence that model calibration can be achieved without sacrificing accuracy through knowledge distillation. Notably, our implementation of knowledge distillation not only guarantees enhanced model calibration but also outperforms the accuracy obtained through conventional training from scratch in specific cases. This innovative approach enables us to simultaneously accomplish the dual objectives of optimal calibration and improved accuracy.

Towards this end, we provide extensive theoretical findings that extend beyond the realms of accuracy transfer and calibration alone. We show, through optics of linear teacher and student networks, that the optimization of student network weights through knowledge distillation enables them to exhibit similar behavior and performance as their respective teachers (see Theorem 1 in the main text). Subsequently, the scope of producing trustworthy models can also be extended to incorporate characteristics, such as fairness and refinement. On a more specific note, Theorem 2 in our work shows that there is a definite advantage of working with calibrated teachers over uncalibrated teachers, i.e., calibrated teachers tend to produce calibrated students without compromising on accuracy. Hence our approach, KD(C), centers around train-time calibration of teacher models, enabling them to generate accurate and optimally calibrated students through knowledge distillation. Significantly, based on our empirical evaluations, it is evident that the transfer of calibration operates bidirectionally. This means that larger calibrated models can be utilized to create smaller calibrated models, and conversely, smaller calibrated models can also serve as a foundation for generating larger calibrated models.

Overall, the research contributes to the advancement of model calibration, accuracy, trustworthiness, and scalability, which can have significant implications in various fields relying on the deployment of reliable and lightweight models.

REFERENCES

Keshigeyan Chandrasegaran, Ngoc-Trung Tran, Yunqing Zhao, and Ngai-Man Cheung. Revisiting label smoothing and knowledge distillation compatibility: What was missing? In *International Conference on Machine Learning*, pp. 2890–2916. PMLR, 2022. 6, 7

Jiacheng Cheng and Nuno Vasconcelos. Calibrating deep neural networks by pairwise constraints. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13699–13708, 2022. doi: 10.1109/CVPR52688.2022.01334. 6, 11, 13

Arindam Ghosh, Thomas Schaaf, and Matthew Gormley. Adafocal: Calibration-aware adaptive focal loss. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1583–1595, 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/0a692a24dbc744fca340b9ba33bc6522-Paper-Conference.pdf`. 13

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017. URL `http://arxiv.org/abs/1706.04599`. 11

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16081–16090, June 2022. 2, 6, 7, 8, 9, 11, 13, 14

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016. URL `http://arxiv.org/abs/1610.02136`. 10, 12

Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6567–6576, October 2021. 11, 13, 14

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. *arXiv preprint arXiv:1909.10155*, 2019. 13

Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2805–2814. PMLR, 10–15 Jul 2018. URL `http://proceedings.mlr.press/v80/kumar18a.html`. 11

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. The devil is in the margin: Margin-based label smoothing for network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 80–88, 2022. 6, 11, 13

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *international conference on machine learning*, pp. 7034–7044. PMLR, 2020. 13

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet K. Dokania. Calibrating deep neural networks using focal loss, 2020. 13

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 6, 7

Hyekang Park, Jongyoun Noh, Youngmin Oh, Donghyeon Baek, and Bumsub Ham. Acls: Adaptive and conditional label smoothing for network calibration. In *Proceedings of the IEEE/CVF ICCV*, 2023. 6

Zhiqiang Shen, Zechun Liu, Dejia Xu, Zitian Chen, Kwang-Ting Cheng, and Marios Savvides. Is label smoothing truly incompatible with knowledge distillation: An empirical study. *arXiv preprint arXiv:2104.00676*, 2021. 6, 7

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919, 2021. 9

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6, 8, 11, 13, 14

Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020. 7

Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in neural information processing systems*, 2019. 11, 13

Christian Tomani, Sebastian Gruber, Muhammed Ebrar Erdem, Daniel Cremers, and Florian Buettner. Post-hoc uncertainty calibration for domain drift scenarios. *CoRR*, abs/2012.10988, 2020. URL `https://arxiv.org/abs/2012.10988`. 10

Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2021. 13

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 14