BAYESIAN SELF-DISTILLATION FOR IMAGE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Supervised training of deep neural networks for classification typically relies on hard targets, which promote overconfidence and often limit calibration, generalization, and robustness. Self-distillation methods aim to mitigate this by leveraging inter-class and sample-specific information present in the model's own predictions, but often remain dependent on hard targets, limiting their effectiveness. With this in mind, we propose Bayesian Self-Distillation (BSD), a principled method for constructing sample-specific target distributions via Bayesian inference using the model's own predictions. Unlike existing approaches, BSD does not rely on hard targets after initialization. BSD consistently yields higher test accuracy (e.g. +1.4% for ResNet-50 on CIFAR-100) and significantly lower Expected Calibration Error (ECE) (-40% ResNet-50, CIFAR-100) than existing architecture-preserving self-distillation methods for a range of deep architectures and datasets. Additional benefits include improved robustness against data corruptions, perturbations, and label noise. When combined with a contrastive loss, BSD achieves state-of-the-art robustness under label noise for single-stage, single-network methods. Code is available in the supplementary material.

1 INTRODUCTION

Despite the widespread use of deep neural networks for classification tasks, the implications of using hard targets for loss computation have received relatively little attention. Intuitively, classes can exhibit varying degrees of similarity, and individual samples may exhibit different levels of resemblance to both their assigned and other classes. This nuanced, sample-specific information – often referred to as dark knowledge (Hinton, 2014) – is not taken into account when relying solely on hard targets. For instance, misclassifying an image of a dog as a teapot is fundamentally different from misclassifying it as a cat. Nevertheless, deep neural networks are typically trained with hard targets, which fail to capture inter-class relationships or sample-specific ambiguities, contributing to poor calibration (Guo et al., 2017).

Self-distillation, an efficient variant of *knowledge distillation* (Hinton et al., 2015), leverages dark knowledge by using a model's own predictions as soft targets during training. In contrast, conventional knowledge distillation provides soft targets to a model by leveraging another, often larger, network. Existing self-distillation methods, however, exhibit varying limitations. Some modify the network architecture with intermediate classifiers (Zhang et al., 2019; 2021), increasing parameter count and computational cost. Others impose a consistency loss between current predictions and those from the last epoch or mini-batch (Kim et al., 2021; Shen et al., 2022). Yet, epoch-wise construction introduces variance under augmentations, while targets from the last mini-batch may be of limited effectiveness, as they originate from nearly identical model states.

Fundamentally, existing self-distillation frameworks constrain their predictions by incorporating a loss term derived from the hard targets (e.g. Furlanello et al. (2018); Kim et al. (2021); Shen et al. (2022)). This reliance interferes with the goal of learning richer and more calibrated predictions, and leaves the model sensitive to label noise. In the overparameterized regime, sensitivity to label noise can exacerbate double descent (Nakkiran et al., 2021), a phenomenon where the test error initially decreases, then increases, before decreasing again as model capacity or training time increase. We hypothesize that a self-distillation method decoupled from the hard targets during training could pro-

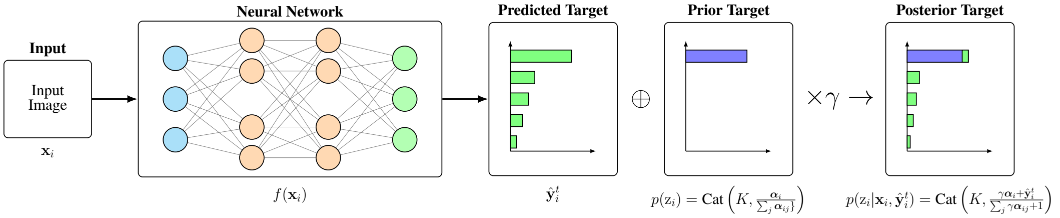

Input
Input Image
$\mathbf{x}_i$
Neural Network
$f(\mathbf{x}_i)$
Predicted Target
$\hat{\mathbf{y}}_i^t$
$\oplus$
Prior Target
$p(\mathrm{z}_i) = \mathrm{Cat}\left(K, \frac{\boldsymbol{\alpha}_i}{\sum_j \boldsymbol{\alpha}_{ij}\}}\right)$
$\times\gamma \rightarrow$
Posterior Target
$p(\mathrm{z}_i|\mathbf{x}_i, \hat{\mathbf{y}}_i^t) = \mathrm{Cat}\left(K, \frac{\gamma\boldsymbol{\alpha}_i + \hat{\mathbf{y}}_i^t}{\sum_j \gamma\boldsymbol{\alpha}_{ij}+1}\right)$

Figure 1: **Bayesian update of a target.** At each epoch, the target distributions are updated using the model's own predictions. Here, $\oplus$ denotes the Bayesian update, and $\times$ indicates discounting of the previous posterior parameters by $\gamma \in [0, 1]$. See Section 3 for details.

vide smoother and more robust probability estimates, thereby mitigating these effects and providing more predictable training dynamics.

To address these limitations, we propose Bayesian Self-Distillation (BSD), an efficient method for leveraging dark knowledge by constructing rich, sample-specific target distributions in a single training run. BSD treats a model's own predictions as evidence for Bayesian inference, without explicitly relying on the hard targets after initialization. A diagram depicting BSD is included in Figure 1.

Our main contributions can be summarized as follows:

- We propose Bayesian Self-Distillation (BSD), a lightweight and principled method for self-distillation that operates independently of hard targets after initialization.
- We provide the first, to our knowledge, theoretical formalization of dark knowledge and use it to show that BSD captures rich sample-specific information.
- We demonstrate through extensive experiments that BSD consistently outperforms both existing architecture-preserving self-distillation methods and conventional knowledge distillation in generalization and calibration for a variety of datasets and network architectures.
- We show that BSD provides robustness against data corruptions, perturbations, and label noise. Notably, it mitigates epoch-wise double descent under label noise and achieves state-of-the-art performance for single-stage, single-network methods when combined with a contrastive loss.

2 Related work

Self-Distillation (SD) originates from Knowledge Distillation (KD) (Hinton et al., 2015), a technique introduced for model compression using a teacher–student framework. Later work demonstrated that identical (Born-Again) networks can be trained sequentially with KD to improve generalization (Furlanello et al., 2018), thereby giving rise to self-distillation.

SD methods differ primarily in how they construct the teacher signal. One family of approaches relies on architectural structure, such as introducing auxiliary classifiers or branches to provide internal supervision (Zhang et al., 2019; 2021; Zhu et al., 2018). Another category leverages temporal knowledge transfer, using model snapshots from earlier epochs (Yang et al., 2019), moving averages of past predictions (Temporal Ensembling, TE) (Laine & Aila, 2016), or moving averages of model parameters (mean teacher) (Tarvainen & Valpola, 2017) over epochs. A third set of methods leverages consistency, encouraging alignment between samples of the same class (Yun et al., 2020) or between consecutive predictions of the same sample (Kim et al., 2021; Shen et al., 2022). This category includes Progressive Self-Knowledge Distillation (PS-KD) (Kim et al., 2021), which uses the model's prediction from the last epoch as a soft target, and Self-Distillation from the Last mini-Batch (DLB) (Shen et al., 2022), which constructs overlapping mini-batches and leverages the prediction from the previous mini-batch. Despite their differences, these strategies remain anchored to the original hard labels, which can be particularly problematic in settings where labels are noisy.

Label noise remains an obstacle in deep learning, contributing to overfitting and the undesirable training dynamics. Proposed solutions range from regularization techniques such as label smoothing (Szegedy et al., 2016) and Mixup (Zhang et al., 2017), to robust loss functions like Symmetric

Cross Entropy (Wang et al., 2019). More sophisticated methods include sample selection via optimal transport Feng et al. (2023); Chang et al. (2023), label correction by interpolation between the prediction and label (Xu et al., 2025), or even complex multi-stage (Liu et al., 2023) or multi-network training pipelines (Zhang et al., 2024). These methods, however, often rely on heuristics rather than *reformulating the learning objective to explicitly model uncertainty*. In contrast, probabilistic modeling provides a principled way to estimate uncertainty and improves interpretability.

Bayesian methods in deep learning have gained attention for their ability to model predictive uncertainty. Methods like Variational inference (Blundell et al., 2015), Monte Carlo dropout (Gal & Ghahramani, 2016), and deep ensembles (Lakshminarayanan et al., 2017) can be used to quantify uncertainty, improve calibration and increase robustness to out-of-distribution data. Recently, Bayesian optimization of hyperparameters has been paired with born-again networks to improve generalization over generations of networks. Related work has also shown that KD itself can serve as a mechanism to transfer better calibration from a teacher to a student (Hebbalaguppe, 2024).

Our proposed method, Bayesian Self-Distillation (BSD), draws inspiration from these domains. Like SD, it softens target distributions, but views the problem from a Bayesian perspective where model predictions serve as evidence for updating target distributions. Note that BSD is distinct from both Bayesian neural networks and Bayesian optimization, where the former regards distributions over model weights and the latter is a method for optimizing black-box functions, whereas BSD places priors over target distributions and refines them over epochs. This allows targets to evolve independently without the continual influence of hard targets, providing richer and more flexible self-supervision that inherently offers robustness to label noise.

3 Method

As discussed in Section 1, a network trained exclusively on hard targets is hardly encouraged to learn meaningful inter-class relationships or account for sample-specific ambiguities. To address this, we frame the training process as Recursive Bayesian Estimation, with the goal of approximating the true latent target distributions $\mathbf{y}$. Rather than viewing the network as a static generator of fixed predictions, we treat it as a noisy sensor observing the external dataset $\mathcal{D}$. By recursively integrating these noisy signals, we aim to construct stable target distributions that capture both class relationships and sample-specific information, thereby improving generalization and reducing overconfidence. The proposed method, Bayesian Self-Distillation (BSD), is summarized in Algorithm 1.

3.1 Notation

Consider the supervised classification problem with a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i^0)\}_{i=1}^n$, samples $\mathbf{x}_i \in \mathbb{R}^d$ and one-hot targets $\mathbf{y}_i^0 \in \Delta_k$. Let $f : \mathbb{R}^d \to \mathbb{R}^k$ denote a neural network with parameters $\boldsymbol{\theta}$ with a softmax activation as its last layer, $\mathcal{L}$ the loss function, and let $\hat{\mathbf{y}}_i^t$ denote the prediction the model outputs for sample i at epoch t.

3.2 Bayesian Self-Distillation

Training a deep neural network is a stochastic process where randomness enters through stochastic optimization (e.g., SGD), data augmentation, and regularization mechanisms. SGD dynamics approximate a posterior distribution over model parameters (Mandt et al., 2017), while stochastic regularization (e.g. dropout) allows the network's output to be viewed as a draw from a predictive distribution (Gal & Ghahramani, 2016). Consequently, the model's prediction for a sample $\mathbf{x}_i$ at any training step can be seen as a random sample from an implicit predictive distribution.

From this perspective, we treat the network as a noisy sensor measuring the true latent class distribution $\mathbf{y}_i$ of the external input $\mathbf{x}_i$. This interpretation is similar to established frameworks in that it views neural network training as Recursive Bayesian Estimation (Singhal & Wu, 1988). At epoch t, the network provides a prediction $\hat{\mathbf{y}}_i^t$, serving as a noisy measurement of the true state $\mathbf{y}_i$. Although the sensor evolves during training making the measurement noise non-stationary, this dynamic is consistent with adaptive filtering and probabilistic self-training frameworks (e.g., Expectation-Maximization). Importantly, this framing allows for model predictions to be viewed as external and independent evidence. The predictions $\hat{\mathbf{y}}_i^t$ constitute measurements as read by the sensor f, conditioned on the independent input $\mathbf{x}_i$. To formalize this, we introduce a latent class variable $z_i \in \{1, \ldots, k\}$ for each sample $\mathbf{x}_i$ and model its distribution as categorical,

$$\mathrm{z}_i \sim \mathrm{Cat}(\mathbf{y}_i). \tag{1}$$

where $\mathbf{y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,k}]$ represents the probabilities for each of the k classes.

To express prior beliefs about $\mathbf{y}_i$, we use a Dirichlet distribution $\mathbf{y}_i \sim \mathrm{Dir}(\boldsymbol{\alpha}_i)$ due to its conjugacy and representation of accumulated evidence as independent components. Specifically, a Dirichlet distribution can be generated by normalizing a set of independent Gamma variables $v_{i,j} \sim \mathrm{Gamma}(\alpha_{i,j}, 1)$ such that $y_{i,j} = v_{i,j} / \sum_k v_{i,k}$. This allows us to interpret the prior as maintaining independent evidence counters for each class. When the input $\mathbf{x}_i$ triggers a non-zero prediction for a secondary class (e.g., “cat” features in a “dog” image), it acts as an independent sensor update for that specific class. Assuming that the model learns to extract relevant feature representations during training, the predictions will capture semantically meaningful knowledge rather than model hallucinations. We encode prior belief in the class corresponding to the label by letting

$$\alpha^0_{i,j} = \begin{cases} c, & \text{if } j = \operatorname*{argmax}_l \mathrm{y}^0_{i,l}, \\ \epsilon, & \text{otherwise,} \end{cases} \tag{2}$$

where $\epsilon \ll c$. For sufficiently small ϵ, the prior implies $p(\mathrm{z}_i | \boldsymbol{\alpha}^0_i)_j \approx 1$ for the labeled class.

At each epoch t, the model outputs a prediction $\hat{\mathbf{y}}^t_i$ which we treat as a noisy measurement to update our beliefs. We assume the likelihood

$$p\left(\hat{\mathbf{y}}_i | \mathbf{y}_i\right) \propto \prod_{j=1}^{k} y_{i,j}^{\hat{y}_{i,j}}, \tag{3}$$

which generalizes the categorical likelihood to fractional evidence and is conjugate to the prior (Bishop & Nasrabadi, 2006). This likelihood treats the prediction as partial evidence, where higher predicted probabilities correspond to stronger observations.

Formally, after observing a prediction $\hat{\mathbf{y}}^t_i$ at epoch t, the posterior distribution is

$$\mathbf{y}_i | \hat{\mathbf{y}}^t_i, \boldsymbol{\alpha}^{t-1}_i \sim \mathrm{Dir}\left(\boldsymbol{\alpha}^t_i\right), \quad \text{where} \quad \boldsymbol{\alpha}^t_i = \boldsymbol{\alpha}^{t-1}_i + \hat{\mathbf{y}}^t_i, \tag{4}$$

which accumulates the noisy measurements into sample-specific distributions over the class probabilities $\mathbf{y}$. To predict the distribution over the labels, we use the posterior predictive distribution. However, because the quality of measurements is expected to improve during training (as the model learns), the measurement noise is non-stationary. Standard Bayesian updating would weight early, noisy observations equally to later, more accurate ones. To address this, we adopt a discounted Bayesian model (West & Harrison, 2006), effectively forgetting old evidence to adapt to the improving network. For a discounting factor $\gamma \in [0, 1]$, we have $\boldsymbol{\alpha}^t_i = \gamma \boldsymbol{\alpha}^{t-1}_i + \hat{\mathbf{y}}^t_i$, yielding the update rules

$$\mathbf{y}^t_i = \frac{\gamma A^{t-1}_i}{\gamma A^{t-1}_i + 1} \mathbf{y}^{t-1}_i + \left(1 - \frac{\gamma A^{t-1}_i}{\gamma A^{t-1}_i + 1}\right) \hat{\mathbf{y}}^t_i, \quad A^t_i = \gamma A^{t-1}_i + 1, \tag{5}$$

for $A^t_i = \sum_{j=1}^k \alpha^t_{ij}$.

Intuitively, the Dirichlet parameters accumulate the model’s belief about how much each class is supported by the input $\mathbf{x}_i$. Discounting ensures that more recent predictions are weighted more heavily than earlier, likely worse, ones. Early in training, when predictions are more likely to have high variance, the prior dominates. As training progresses, predictions are expected to become more consistent, providing stronger and more reliable evidence.

3.3 Relationship between BSD and other methods

If initialized at its fixed point $A^0_i = \frac{1}{1-\gamma}$, the recurrence $A^t_i = A^0_i = \frac{1}{1-\gamma}$ gives the EMA update

$$\mathbf{y}^t_i = \gamma \mathbf{y}^{t-1}_i + (1 - \gamma) \hat{\mathbf{y}}^t_i. \tag{6}$$

Algorithm 1 Bayesian Self-Distillation (BSD)

```
Input: Training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i^0)\}_{i=1}^n$
Model: Neural network $f$ with parameters $\boldsymbol{\theta}$, optimizer $h$
Parameters: Number of epochs $T$, Dirichlet prior $\boldsymbol{\alpha}_i^0 = [\alpha_{i,1}^0, \ldots, \alpha_{i,k}^0]$, discount factor $\gamma$.
Initialize: $A_i^0 = \sum_{j=1}^k \alpha_{ij}^0$
  for epoch $t \leftarrow 1$ to $T$ do
      for mini-batch $B \subseteq \mathcal{D}$ do
          $\hat{\mathbf{y}}_i^t \leftarrow f(\mathbf{x}_i; \boldsymbol{\theta}), \quad \forall i \in B$    ▷ Model prediction
          $\mathcal{L} \leftarrow \frac{1}{|B|k} \sum_{i \in B} \ell\big(\hat{\mathbf{y}}_i^t, \mathbf{y}_i^{t-1}\big)$    ▷ Compute loss
          Compute $\nabla_{\boldsymbol{\theta}} \mathcal{L}$
          $\theta \leftarrow h\left(\boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}} \mathcal{L}\right)$    ▷ Gradient descent step
          $\mathbf{y}_i^t \leftarrow \frac{\gamma A_i^{t-1}}{\gamma A_i^{t-1}+1} \mathbf{y}_i^{t-1} + \left(1 - \frac{\gamma A_i^{t-1}}{\gamma A_i^{t-1}+1}\right) \hat{\mathbf{y}}_i^t, \quad \forall i \in B$    ▷ Update targets
          $A_i^t = \gamma A_i^{t-1} + 1, \quad \forall i \in B$    ▷ Update prior hyperparameters
      end for
  end for
  Return $\boldsymbol{\theta}$    ▷ Trained parameters
```

Similarly,

$$A_i^t = \gamma A_i^{t-1} + 1 = \gamma^t A_i^0 + \sum_{j=0}^{t-1} \gamma^j = A_i^0 \gamma^t + \frac{1 - \gamma^t}{1 - \gamma} \tag{7}$$

so in the limit we have that

$$\lim_{t \to \infty} A_i^t = \frac{1}{1 - \gamma}, \tag{8}$$

i.e. the weight of new observations converges to $1 - \gamma$ exponentially. In the case of non-zero ϵ, and if we initialize at the fixed point $A_i^0 = \frac{1}{1-\gamma}$ for γ close to 1, BSD will approximate label smoothing (Szegedy et al., 2016).

Related self-distillation methods can be interpreted as special cases of BSD. Conventional training corresponds to taking $c \to \infty$, which fixes the target distribution and prevents Bayesian updating. In this limit, the framework reduces to standard distribution matching, as KL-divergence and cross-entropy are equivalent under fixed one-hot targets. PS-KD (Kim et al., 2021) is recovered by setting $\gamma = 0$, so that only the most recent prediction contributes to the target. DLB (Shen et al., 2022) performs the same Bayesian update as PS-KD but applies it at the mini-batch level rather than epoch-wise. Meanwhile, TE (Laine & Aila, 2016) corresponds to assuming an improper prior (a zero vector) and applying an explicit weighting schedule to the accumulated evidence. Although these methods differ from BSD in that they include a supervised loss term based on one-hot labels, apply temperature scaling or modify the loss function, all of them implicitly construct a target distribution from an accumulation of past predictions. BSD makes this shared structure explicit by interpreting it as Bayesian evidence aggregation under a Dirichlet prior.

4 Experiments

Experimental Setup. We evaluate ResNet (He et al., 2016a), DenseNet (Huang et al., 2017), and ViT (Dosovitskiy et al., 2020) models on CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), Tiny ImageNet (Stanford CS231n, 2017) and ImageNet Russakovsky et al. (2015). We report the average of three runs. For BSD, we set $\epsilon = 0$, $\gamma = 0.95$, and use $c = 1000$ and $c = 50$ for the CNNs and ViTs, respectively, on all datasets except for ImageNet. On Imagenet, we set $\epsilon = 0.05$, $\gamma = 0.99$, and use $c = 400$. The experimental setup is described in more detail in the appendix, including the hyperparameters used for the baselines.

4.1 Emergence of Dark Knowledge

We first validate that dark knowledge emerges naturally during training, and show that BSD promotes further discovery of underlying structures in the data. In an attempt to formalize the notion of dark knowledge, we decompose the output of the network, for an input $\mathbf{x}_i$ with integer label $\mathrm{y}_i = \operatorname*{argmax}_k \mathbf{y}^0_{i,k}$, as

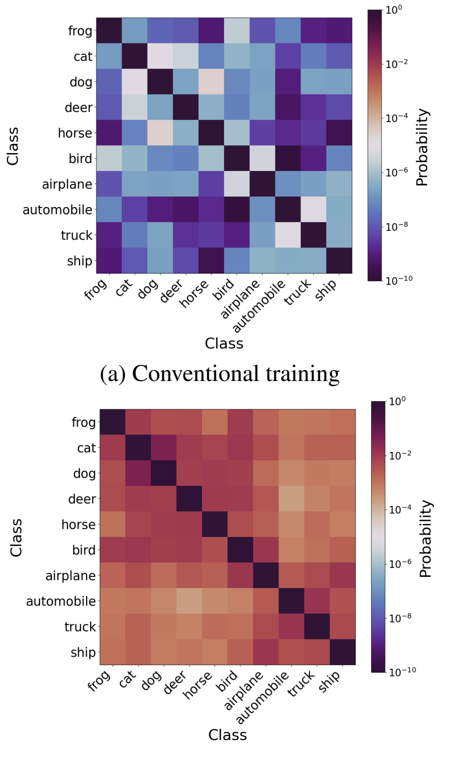
$$f(\mathbf{x}_i) = \boldsymbol{\mu}_{y_i} + \delta(\mathbf{x}_i). \tag{9}$$

Here, $\boldsymbol{\mu}_{y_i} \in \Delta^K$ is a row of the matrix with elements

$$\mu_{ij} = \frac{1}{2}\left(\mathbb{E}[f(X) \mid Y = i]_j + \mathbb{E}[f(X) \mid Y = j]_i\right) \tag{10}$$

and captures the inter-class component of dark knowledge under a symmetry constraint. The function $\delta : \mathbb{R}^d \to \mathbb{R}^K$ captures the sample-specific deviation from $\boldsymbol{\mu}_{y_i}$ and is calculated as $\delta(\mathbf{x}_i) = f(\mathbf{x}_i) - \boldsymbol{\mu}_y$.

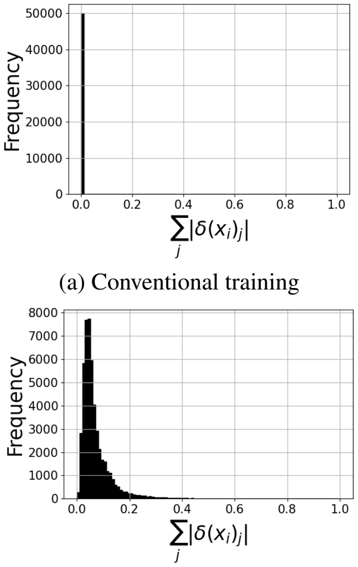
We visualize $\boldsymbol{\mu}$ and δ from a ResNet-18 trained on CIFAR-10 in Figures 3 and 4. The log-scale heatmaps of $\boldsymbol{\mu}$ in Figure 3 reveal that the networks learn meaningful inter-class relationships. For instance, they capture similarities between animals but also between bird and airplane, with these clusters being more prominent and the probabilities larger for BSD. The per-sample absolute deviation from $\boldsymbol{\mu}$ presented in Figure 4 validates the emergence of δ and shows that BSD promotes learning of sample-specific information.



(a) Conventional training

(b) BSD

Figure 3: **Inter-class component $\boldsymbol{\mu}$ of dark knowledge.** Semantical patterns emerge between classes, accentuated by BSD (ResNet-18, CIFAR-10).



(a) Conventional training

(b) BSD

Figure 4: **Sample-specific component of dark knowledge.** BSD promotes learning of sample-specific information (ResNet-18, CIFAR-10).

4.2 Classification Results

Methods compared. We benchmark BSD against conventional training (baseline), Temporal Ensembling (TE) (Laine & Aila, 2016), self-Distillation from Last mini-Batch (DLB) (Shen et al., 2022) and Progressive Self-Knowledge Distillation (PS-KD) (Kim et al., 2021). We do not compare with methods such as (Zhang et al., 2019; 2021), as our focus is on modifying label distributions rather than model architectures. An ensemble of three models is included for reference, whose knowledge is also distilled into a single model of the same architecture, for a comparison with conventional knowledge distillation.

4.2.1 Generalization

The main results are reported in Table 1. In all experiments, BSD improves test accuracy relative to both the baseline and related methods, surpassing conventional knowledge distillation and approaching ensemble performance.

The performance gains are most pronounced on CIFAR-100 and TinyImageNet. For ResNet and DenseNet, BSD boosts accuracy by about 3 percentage points (pp) over the baseline and by more than 1 pp over the strongest related methods on both datasets. In contrast, on CIFAR-10, where baseline performance

Table 1: **Test set accuracy on CIFAR-10, CIFAR-100, and TinyImageNet.** The models are trained using standard training (baseline), related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021), and the proposed method (BSD). The best results are highlighted in bold.

Dataset	Model	Baseline (%)	TE (%)	DLB (%)	PS-KD (%)	BSD (%)	KD (%)	Ens. (%)
CIFAR-10	ResNet-18	94.39±0.18	94.52±0.08	94.52±0.11	94.56±0.06	**94.88** ±0.06	94.73±0.18	95.33
	DenseNet-121	94.86±0.13	95.20±0.13	95.15±0.15	95.10±0.04	**95.38** ±0.13	95.28±0.09	95.67
	ViT-B/16	98.28±0.02	98.34±0.05	98.39±0.03	98.37±0.08	**98.44** ±0.01	98.32±0.02	98.56
CIFAR-100	ResNet-50	75.82±0.41	76.07±0.40	77.22±0.23	77.71±0.20	**79.09** ±0.11	77.69±0.08	79.29
	DenseNet-169	76.63±0.28	76.33±0.11	78.43±0.09	77.88±0.04	**79.47** ±0.12	78.72±0.18	79.83
	ViT-B/16	89.16±0.05	89.11±0.23	88.72±0.04	89.36±0.11	**89.54** ±0.13	89.35±0.11	90.27
TinyImageNet	ResNet-101	64.22±0.21	64.45±0.12	65.83±0.28	65.65±0.12	**67.41** ±0.31	66.36±0.16	69.78
	DenseNet-201	64.53±0.38	64.60±0.19	66.66±0.20	66.11±0.08	**67.74** ±0.04	66.94±0.09	69.44
	ViT-B/16	88.99±0.20	89.02±0.23	89.29±0.13	89.32±0.13	**89.65** ±0.17	89.16±0.10	90.23
ImageNet	ResNet-152	78.55±0.05	78.45±0.07	77.46±0.14	78.99±0.10	**79.47** ±0.07	79.37±0.04	80.24

is already high, the improvements are more modest. Similarly, for ViT-B/16, we observe small but consistent gains across all datasets. The trend continues on ImageNet, where BSD improves accuracy by roughly 1 pp over the baseline and 0.5 pp over the best-performing related method, indicating that it also scales effectively to more complex datasets.

4.2.2 Calibration

Given the importance of calibrated probability estimates in many tasks, we evaluate the calibration of models trained on CIFAR-100 using Expected Calibration Error (ECE) (Naeini et al., 2015) and Negative Log Likelihood (NLL). The results are included in Table 2, where BSD demonstrates superior calibration when compared to related methods (DLB). Figure 5 includes reliability diagrams where BSD's curves lie closest to the diagonal line, indicating better calibration. Furthermore, we compare BSD with standard and distillation-based calibration methods in Table 7 (appendix), where it achieves state-of-the-art results across all metrics.

Table 2: **ECE and NLL on CIFAR-100.** The models are trained using standard training (baseline), related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021), and the proposed method (BSD). The best results are highlighted in bold.

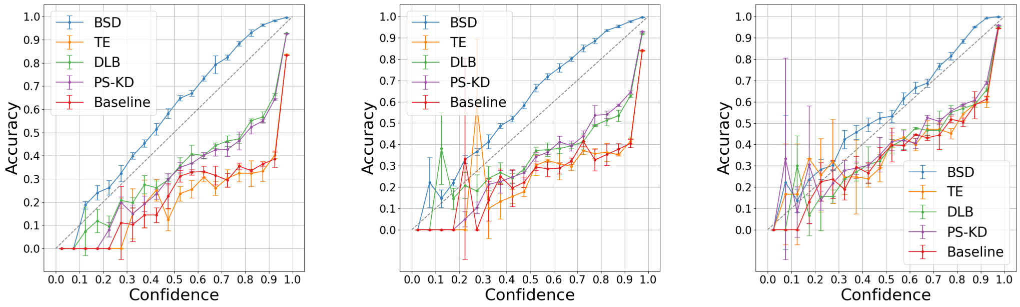
Architecture	Metric	Baseline	TE	DLB	PS-KD	BSD
ResNet-50	ECE (%)	20.41±0.45	20.21±0.34	11.82±0.28	12.41±0.17	**7.17** ±0.40
	NLL	2.94±0.04	2.91±0.04	1.09±0.01	1.09±0.02	**0.77** ±0.00
DenseNet-169	ECE (%)	19.33±0.24	19.63±0.16	12.42±0.09	12.35±0.10	**7.67** ±0.10
	NLL	2.59±0.03	2.63±0.01	1.06±0.01	1.08±0.01	**0.75** ±0.00
ViT-B/16	ECE (%)	7.53±0.15	7.38±0.26	6.89±0.08	5.93±0.26	**5.89** ±0.09
	NLL	0.56±0.02	0.55±0.01	0.52±0.00	0.44±0.01	**0.42** ±0.00

The improvements are significant compared to the baseline, with BSD reducing ECE by more than 60% and NLL by over 70% for the convolutional networks. Relative to the best-performing related method, BSD lowers ECE by nearly 40% and NLL by about 29%. For ViT-B/16, the gains are more modest, with a 22% reduction in ECE and a 25% reduction in NLL over the baseline, alongside small but consistent improvements over related methods. As discussed in Section 4.2.5, careful hyperparameter selection can further improve BSD's calibration, e.g. an ECE of 1.33% for ResNet-50 on CIFAR-100.

4.2.3 Robustness

To assess our method's performance under less ideal conditions, we evaluate the models trained on CIFAR-10 on corrupted and perturbed images, and introduce symmetric and asymmetric label noise.

Corruptions and perturbations. We evaluate the robustness to corruptions and perturbations of BSD and related methods by evaluating the models trained on CIFAR-10 on CIFAR-10-C and CIFAR-10-P (Hendrycks & Dietterich, 2019). The results are included in Table 3, where we report test set accuracy and mean Flip Probability (mFP) for CIFAR-10-C and CIFAR-10-P, respectively. BSD yields the highest accuracy and the lowest mFP for all models.



(a) ResNet-50 (b) DenseNet-169 (c) ViT-B

Figure 5: **Reliability Diagrams for CIFAR-100 Models.** A curve closer to the diagonal indicates better calibration. The models are trained using standard training (baseline), related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021), and the proposed method (BSD).

Table 3: **Performance on CIFAR-10-C and CIFAR-10-P.** Accuracy and Mean Flip Probability of BSD, baseline, and related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021) under corruptions and perturbations. The best results for each metric are highlighted in bold.

Model	CIFAR-10-C (Acc., %)					CIFAR-10-P (mFP, %)				
	Baseline	TE	DLB	PS-KD	BSD	Baseline	TE	DLB	PS-KD	BSD
ResNet-18	73.31±0.31	73.22±0.27	73.94±0.31	74.28±0.50	**74.88** ±0.28	7.20±0.21	7.23±0.04	7.08±0.06	7.10±0.28	**6.09** ±0.08
DenseNet-121	73.36±0.32	73.09±0.86	74.81±0.23	74.30±0.07	**74.85** ±0.32	7.49±0.20	7.46±0.18	7.06±0.22	7.21±0.16	**6.39** ±0.11
ViT-B	91.17±0.07	91.31±0.22	91.20±0.26	91.38±0.24	**91.57** ±0.12	2.53±0.04	2.40±0.03	2.45±0.05	2.35±0.06	**2.11** ±0.07

On CIFAR-10-C, BSD improves robustness across all architectures, raising accuracy by about 1.5 pp for both ResNet-18 and DenseNet-121 and by 0.4 pp for ViT-B over the baseline, slightly surpassing the best related methods. On CIFAR-10-P, BSD achieves the lowest mean flip probability for all models, reducing mFP by around 10% compared to the best-performing related methods.

Label noise. We generate symmetric label noise by randomly reassigning a percentage of all labels, and asymmetric noise using the class-dependent definition from prior work (Tanaka et al., 2018). Figure 6 shows test error over epochs for the previously mentioned self-distillation methods under 20% label noise for ResNet-18 trained on CIFAR-10. BSD and TE are obtain the highest accuracy under label noise, while the methods that rely more on the hard targets perform worse. Notably, BSD flattens the characteristic double descent curve observed for other methods, indicating a more robust learning process.

We compare BSD with lightweight regularization methods under noisy labels, including Label Smoothing (LS) (Szegedy et al., 2016), Symmetric Cross Entropy Learning (SL) (Wang et al., 2019), MixUp (Zhang et al., 2017) and Temporal Ensembling (TE) Laine & Aila (2016). We report the average best obtained accuracy, which for BSD is typically close to the final accuracy, but diverges in varying amounts for the remaining methods. The results are included in upper section of Table 4, where BSD yields the highest accuracy for all noise levels except under 10% asymmetric noise. We hypothesize that the strong performance of BSD is due to the complete detachment from the original targets during training, which would explain the robustness under high noise levels.

To compare BSD with state-of-the-art methods for learning with noisy labels, we combine it with techniques from self-supervised learning (for more information refer to the Appendix). We restrict our analysis to single-stage, single-network approaches, as multi-stage or ensemble methods may be used to further enhance BSD's performance. We include OT-Filter (Feng et al., 2023), Curriculum and Structure-aware Optimal Transport (CSOT) (Chang et al., 2023) and Dynamic and Uniform Label Correction (DULC) (Xu et al., 2025). Like other works, we consider a PreAct ResNet-18 (He et al., 2016b), trained with SGD for 300 epochs with a batch size of 128 and weight decay of $5e^{-4}$, learning rate of 0.02 and report the average best obtained accuracy. Results are summarized in the lower section of Table 4, where BSD consistently matches or outperforms existing state-of-the-art methods.

Table 4: **Test set accuracy under symmetric and asymmetric label noise.** Performance of lightweight methods (top, ResNet-18) (Szegedy et al., 2016; Wang et al., 2019; Zhang et al., 2017; Laine & Aila, 2016) and state-of-the-art methods (bottom, PreAct ResNet-18) (Feng et al., 2023; Chang et al., 2023; Xu et al., 2025) on CIFAR-10. The best results are highlighted in bold.

Method	Symmetric			Asymmetric		
	20%	50%	80%	10%	30%	40%
Baseline	86.76±0.37	81.49±0.26	63.63±0.38	90.07±0.26	85.23±0.29	80.30±0.60
LS	87.85±0.10	81.49±0.51	64.70±1.17	90.35±0.24	86.10±0.59	81.96±0.52
SL	91.86±0.12	86.39±0.32	72.89±0.45	91.95±0.14	86.39±0.21	80.57±0.07
MixUp	89.87±0.18	83.38±0.20	69.23±0.56	91.78±0.18	88.26±0.58	84.49±0.54
TE	93.07±0.07	90.70±0.14	72.61±0.74	94.37±0.15	92.86±0.07	90.81±0.07
BSD	93.39±0.05	90.71±0.26	77.50±0.49	94.18±0.07	93.20±0.14	91.65±0.06
OT-Filter	96.0	95.3	94.0	-	-	95.1
CSOT	96.6 ±0.10	**96.2** ±0.11	94.4 ±0.16	-	-	95.5 ±0.06
DULC	96.6	96.0	95.0	96.7	95.5	95.2
BSD+	**96.9** ±0.04	**96.2** ±0.06	**95.1** ±0.06	**97.0** ±0.12	**96.3** ±0.13	**95.6** ±0.25

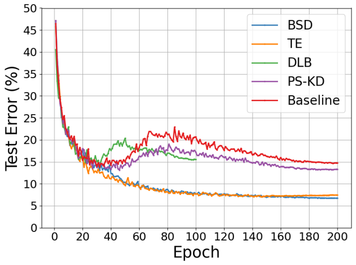


Figure 6: **Test Error Over Epochs Under 20% Label Noise on for ResNet-18 on CIFAR-10.** The models are trained using standard training (baseline), related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021), and the proposed method (BSD).

4.2.4 Augmentation

As data augmentation is a standard component in image classification, we examine the interaction of BSD with two widely used augmentation methods, CutOut (DeVries & Taylor, 2017) and CutMix (Yun et al., 2019). CutOut masks out a patch of an image with zeros, while CutMix replaces the patch with a segment from another image and interpolates the labels. For BSD, the augmentations are applied to half of the images in each mini-batch, while the remaining half is used for distillation. This reduces the frequency of label updates but also changes CutMix to combine soft targets rather than one-hot labels. To compensate, we set $\gamma = 0.9$ and $c = 500$. We include DLB (Shen et al., 2022) and conventional training for comparison. Results are reported in Table 5.

CutMix. BSD consistently improves upon both the baseline and DLB across datasets and architectures. On CIFAR-100, BSD boosts convolutional networks by over 3 pp relative to the baseline, and by about +2.3 pp (DenseNet-169) and +1.3 pp (ResNet-50) compared to DLB. On TinyImageNet, BSD surpasses the baseline by roughly +2 pp for ResNet-101 and over +4 pp for DenseNet-201, both exceeding DLB by more than 1 pp. For ViT-B/16, BSD achieves gains of +0.5 pp over the baseline and +0.3 pp over DLB on CIFAR-100, and +1.15 pp and +0.6 pp on TinyImageNet. On CIFAR-10, improvements are smaller but consistent across all models. Compared with Table 1, CutMix enhances BSD's performance, offering consistent gains over both standard training and DLB.

CutOut. The effect of CutOut is more nuanced. On CIFAR-10 and CIFAR-100, BSD consistently performs best, improving convolutional networks by more than +3 pp over the baseline and over +2 pp over DLB on CIFAR-100. For ViT-B/16, BSD remains competitive with the baseline and performs better than DLB, though the gains are modest given the already high baseline accuracy. On TinyImageNet, however, CutOut does not provide additional benefits: BSD still surpasses the baseline, but the accuracy is lower than without CutOut (cf. Table 1), and ViT-B/16 shows little change relative to baseline or DLB. We speculate that the poor results for BSD on TinyImageNet and DLB overall is due to over-regularization. Overall, BSD remains stronger than DLB under CutOut, but the augmentation itself appears less complementary to BSD than CutMix.

4.2.5 Ablation

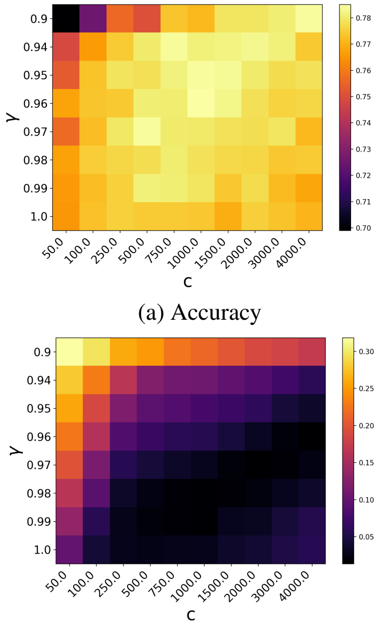
We perform ablations on the hyperparameters γ and c, analyzing their effect on accuracy and ECE for ResNet-50 on CIFAR-100. The results are plotted in Figure 8. We observe high accuracy and low ECE for a range of values. Interestingly, the set of hyperparameters that minimizes ECE does not coincide exactly with the values that maximizes accuracy.

Table 5: **Accuracy on CIFAR-10, CIFAR-100, and TinyImageNet with CutMix and CutOut.** The models are trained using standard training (baseline), DLB (Shen et al., 2022) and the proposed method (BSD). The best results are highlighted in bold.

Dataset	Model	+ CutOut (%)			+ CutMix (%)		
		Baseline	DLB	BSD	Baseline	DLB	BSD
CIFAR-10	ResNet-18	95.71±0.09	95.47±0.07	**95.94** ±0.07	95.75±0.02	95.70±0.04	**96.23** ±0.06
	DenseNet-121	96.09±0.08	95.87±0.12	**96.21** ±0.12	96.17±0.20	96.34±0.10	**96.84** ±0.08
	ViT-B/16	98.65±0.04	98.58±0.06	**98.68** ±0.08	98.82±0.02	98.69±0.04	**98.86** ±0.04
CIFAR-100	ResNet-50	76.61±0.43	76.38±0.33	**79.70** ±0.16	79.73±0.13	80.15±0.30	**81.42** ±0.08
	DenseNet-169	77.10±0.17	77.87±0.16	**80.14** ±0.35	79.31±0.27	79.93±0.11	**82.18** ±0.14
	ViT-B/16	90.30±0.06	89.91±0.10	**90.31** ±0.13	90.31±0.05	90.50±0.11	**90.87** ±0.04
TinyImageNet	ResNet-101	64.08±0.35	64.26±0.44	**66.45** ±0.18	68.32±0.16	68.74±0.14	**70.29** ±0.14
	DenseNet-201	65.58±0.17	64.84±0.34	**67.62** ±0.21	66.54±0.60	69.22±0.48	**70.63** ±0.30
	ViT-B/16	89.63±0.04	89.72±0.03	**89.96** ±0.20	89.48±0.03	90.03±0.05	**90.63** ±0.14

The lowest ECE is achieved for $\gamma = 0.97$ and $c = 2000$, while the highest accuracy is observed at $\gamma = 0.96$ and $c = 1000$. Notably, the lowest ECE obtained is 1.33% on the validation set, which is substantially lower than for the other methods in Table 2. While this comes at the cost of a slight decrease in accuracy, calibration may be prioritized for certain tasks or applications. Furthermore, by comparing $\gamma = 1.0$ with the remainder of the values in Figure 8, we note that the discounting is a valuable addition that appears to improve both accuracy and ECE.

While BSD appears offer low ECE and high accuracy for wide range of values of γ and c, we note that if hyperparameters are chosen poorly (e.g. $\gamma = 0.9$, $c = 50.0$), performance deteriorates. This is likely due to underfitting caused by relying too much on predictions too early.



(a) Accuracy

(b) ECE

Figure 8: **Impact of γ and c.** Validation accuracy and ECE for different values of γ and c (ResNet-50, CIFAR-100).

5 Discussion

Overall, our results demonstrate that BSD can improve test set accuracy, ECE and NLL across a variety of datasets, architectures, and augmentation strategies compared to conventional training and related self-distillation methods. BSD does not seem to overfit to noise in the same way as conventional training and contrastive self-distillation methods, and achieves higher test set accuracy under label noise than other architecture-preserving self-distillation methods. Additionally, BSD+ yields state-of-the-art accuracy under label noise on CIFAR-10.

Computational cost and memory requirements. The Bayesian update in Equation 5 requires first computing a per-example scalar weight from A and then applying that weight to the two target tensors. For a mini-batch of size B and K classes, this is $\mathcal{O}(BK)$ and negligible compared with a forward-backward pass. Memory-wise, we store the target distributions $\mathbf{y}$ and per-example counts A. With float16 this requires $2NK$ bytes for $\mathbf{y}$ and $2N$ bytes for A. TE and PS-KD require the same amount of memory for storing the targets, whereas DLB requires a batch-wise buffer of $2BK$ bytes.

REFERENCES

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006. See Section 2.2.1 on page 76.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Wanxing Chang, Ye Shi, and Jingya Wang. Csot: Curriculum and structure-aware optimal transport for learning with noisy labels. *Advances in Neural Information Processing Systems*, 36:8528–8541, 2023.

Jiacheng Cheng and Nuno Vasconcelos. Calibrating deep neural networks by pairwise constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13709–13718, 2022.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Chuanwen Feng, Yilong Ren, and Xike Xie. Ot-filter: An optimal transport filter for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16164–16174, 2023.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pp. 1607–1616. PMLR, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Arindam Ghosh, Thomas Schaaf, and Matthew Gormley. Adafocal: Calibration-aware adaptive focal loss. *Advances in Neural Information Processing Systems*, 35:1583–1595, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

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, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645. Springer, 2016b.

R. et al. Hebbalaguppe. Calibration transfer via knowledge distillation. In *ACCV*, 2024.

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*, pp. 16081–16090, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Geoffrey Hinton. Dark knowledge. Distinguished Lecture Series, Toyota Technological Institute at Chicago, October 2014. https://www.ttic.edu/dls-2014-2015/.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

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*, pp. 6567–6576, 2021.

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

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

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.

Zixuan Liu, Xin Zhang, Junjun He, Dan Fu, Dimitris Samaras, Robby Tan, Xiao Wang, and Sheng Wang. Chimera: Learning with noisy labels by contrasting mixed-up augmentations. *arXiv preprint arXiv:2310.05183*, 2023.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

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.

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

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, 2011.

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 International Conference on Computer Vision*, pp. 3936–3945, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL `https://arxiv.org/abs/1912.01703`.

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.

Yiqing Shen, Liwu Xu, Yuzhe Yang, Yaqian Li, and Yandong Guo. Self-distillation from the last mini-batch for consistency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11943–11952, 2022.

Sharad Singhal and Lance Wu. Training multilayer perceptrons with the extended kalman algorithm. *Advances in neural information processing systems*, 1, 1988.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pp. 369–386. SPIE, 2019.

Stanford CS231n. Tiny imagenet challenge. `http://cs231n.stanford.edu/tiny-imagenet-200.zip`, 2017. Accessed: 2025-02-20.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5552–5560, 2018.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 322–330, 2019.

Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*, chapter 6, pp. 193–202. Springer, 2006.

Yuanzhuo Xu, Xiaoguang Niu, Jie Yang, Ruiyi Su, Jian Zhang, Shubo Liu, and Steve Drew. Revisiting interpolation for noisy label correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21833–21841, 2025.

Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2859–2868, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13876–13885, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3713–3722, 2019.

Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4388–4403, 2021.

Qian Zhang, Ge Jin, Yi Zhu, Hongjian Wei, and Qiu Chen. Bpt-plr: A balanced partitioning and training framework with pseudo-label relaxed contrastive loss for noisy label learning. *Entropy*, 26(7):589, 2024.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31, 2018.

A Appendix

A.1 Experimental Details

Unless otherwise stated, all models trained on CIFAR and TinyImageNet datasets use the Adam optimizer (Kingma, 2014) with a batch size of 256, a maximum learning rate of 0.01 scheduled via the 1cycle policy (Smith & Topin, 2019), and basic augmentations (random cropping and horizontal flipping) for 200 epochs (40 for ViTs, with a maximum learning rate of 5e-5). For ImageNet, we train with SGD for 90 epochs using a per-GPU batch size of 128 on 8 GPUs, a learning rate of 0.5 (scheduled via cosine annealing with 5 epochs of warm-up), a weight decay of $1e{-}4$, and random resized crops of 224 pixels combined with horizontal flipping.

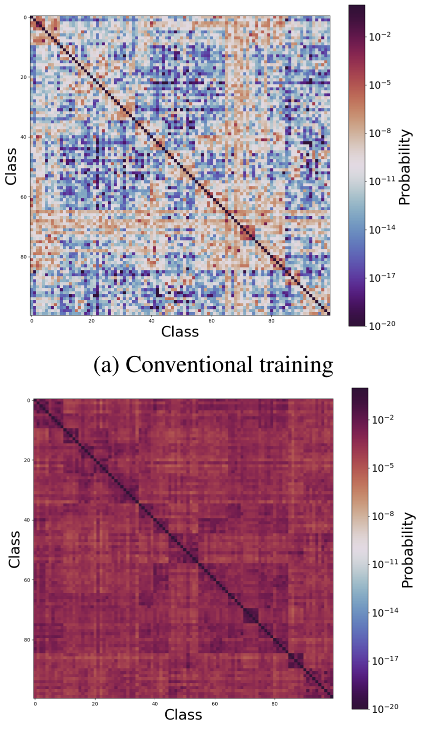
Architectures. Because of the small size of the images, the ResNet and DenseNet networks have been modified to include a 3×3 convolution instead of the usual 7×7 convolution for all datasets except ImageNet. The ViT-B (Dosovitskiy et al., 2020) model has been pretrained on ImageNet for all experiments and is available in Pytorch (Paszke et al., 2019).

Method-specific hyperparameters. For the temporal ensemble, we set the momentum parameter $\alpha = 0.6$, gradually ramping up the distillation loss over the first 100 epochs (45 for ImageNet) and anneal Adam's β_1 to zero during the final 50 epochs. For DLB, we follow Shen et al. (2022) and use a temperature of 3 and set $\alpha = 1.0$, but train for only 100 epochs to offset the doubled mini-batch size. For PS-KD, we let $\alpha_T = 0.8$ ($\alpha_T = 0.3$ for ImageNet). For the knowledge distillation of the ensembles, we use a temperature of 3.

Label Noise. Inspired by methods in semi-supervised learning, we construct BSD+ by combining BSD with a contrastive loss term. We do this by utilizing a strong and a weak set of augmentations, where the weak set is used for BSD, and the strong is used for the contrastive term. We define the contrastive loss as

$$\mathcal{L}_c = \frac{\lambda_a}{m} \sum_{j=1}^{m} KL(f(T_j(\mathbf{x}_i)), \hat{\mathbf{y}}^t), \tag{11}$$

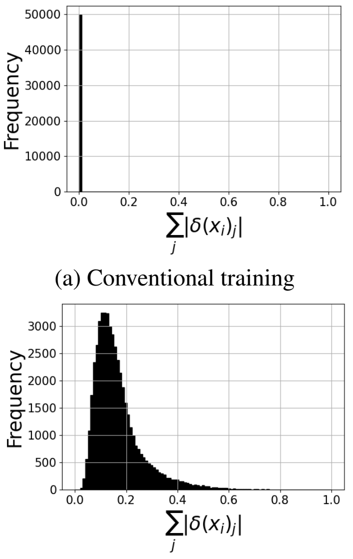
where m is the number of strongly augmented views, and T_j is the corresponding transform. For the strong set of augmentations, we utilize AutoAugment (Cubuk et al., 2018), in combination with CutMix (Yun et al., 2019) and Random Erasing (Zhong et al., 2020). For BSD+, we set $m = 2$ for all noise levels and schedule the learning rate using cosine annealing (Loshchilov & Hutter, 2016).



(a) Conventional training

(b) BSD

Figure 10: **Inter-class component $\boldsymbol{\mu}$ of dark knowledge.** Semantical patterns emerge between classes, accentuated by BSD (ResNet-50, CIFAR-100).



(a) Conventional training

(b) BSD

Figure 11: **Total sample-wise deviation $\delta(\mathbf{x}_i)$ of dark knowledge.** BSD promotes learning of sample-specific information (ResNet-50, CIFAR-100).

Because BSD may excessively smooth the label distributions when labels are noisy, we introduce a sharpening parameter τ into the loss

$$\mathcal{L} \leftarrow \frac{1}{|B|k}\sum_{i\in B}\ell\left(\hat{\mathbf{y}}_i^t, \frac{\left(\mathbf{y}_i^{t-1}\right)^{1/\tau}}{\|\left(\mathbf{y}_i^{t-1}\right)^{1/\tau}\|_1}\right), \tag{12}$$

for both BSD and BSD+, where $|B|$ is the batch size. In the absence of noise, we use $\tau = 1$ to avoid promoting overconfident predictions, while we set $\tau = 0.8$ for all experiments with label noise. For BSD, we set $\gamma = 0.85$ and $c = 2000$ when injecting symmetric noise, and $\gamma = 0.9$ and $c = 1000$ for asymmetric noise. For BSD+, we set $\gamma = 0.95$ and $c = 1000$ for all experiments, and set $\lambda_c = 2$ under asymmetric noise. For BSD+ under symmetric noise we set $\lambda_c = 4$, $\lambda_c = 7$ and $\lambda_c = 14$ for noise levels 20%, 50% and 80%, respectively.

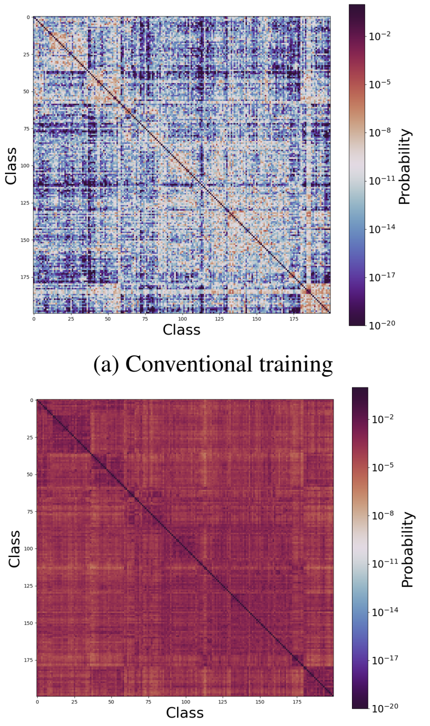
A.2 Additional Experiments

Emergence of dark knowledge. We include visualizations of the inter-class distributions for CIFAR-100 and Tiny ImageNet in Figures 10 and 13, with their corresponding sample-wise deviations plotted in Figures 11 and 14, respectively. We observe similar patterns as for ResNet-18 in Figure 3, but on a larger scale, and larger sample-wise deviations for the larger datasets.

To study the emergence of dark knowledge during training, we compute the average KL divergence between the output distributions of the model over epochs and those of the final model, while adjusting for temperature scaling (Guo et al., 2017). The results for are plotted in Figure 15, where the decrease of KL divergence over epochs suggest that dark knowledge is a property that emerges gradually.

Out-of-distribution calibration and detection. To evaluate model calibration under distributional shifts, we tested models trained on CIFAR-10 against the CIFAR-10-C benchmark, which applies 19 different corruptions (e.g., brightness, blur, noise) across five severity levels. As shown in Figure 16, BSD consistently achieves the lowest Expected Calibration Error (ECE) across all severity levels compared to the baseline and related methods. Importantly, the performance gap widens as data quality degrades, and we observe that the related methods' ECE increases more rapidly at high corruption severities, while BSD maintains a flatter ECE curve. This indicates that BSD reduces overconfidence under increasing covariate shift.

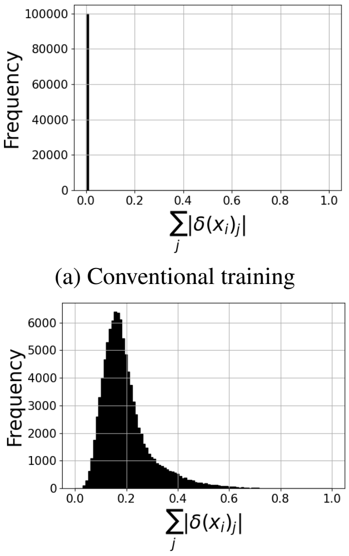
Furthermore, we study the performance of BSD and related methods under domain shifts by measuring the area under the receiver operating characteristic curve (AUROC) for models trained on CIFAR-10 and CIFAR-100, evaluated against the Street View House Numbers (SVHN) dataset Netzer et al. (2011). While the best performing method varies with dataset and



(a) Conventional training

(b) BSD

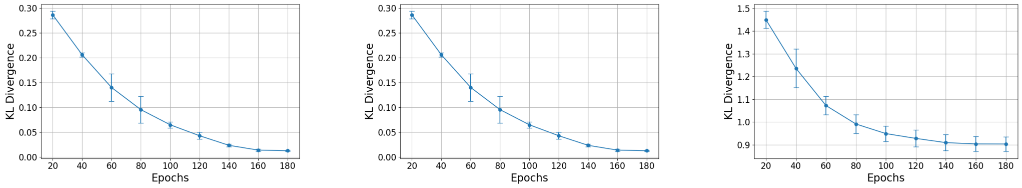
Figure 13: **Inter-class component μ of dark knowledge.** Semantical patterns emerge between classes, accentuated by BSD (ResNet-101, Tiny ImageNet).



(a) Conventional training

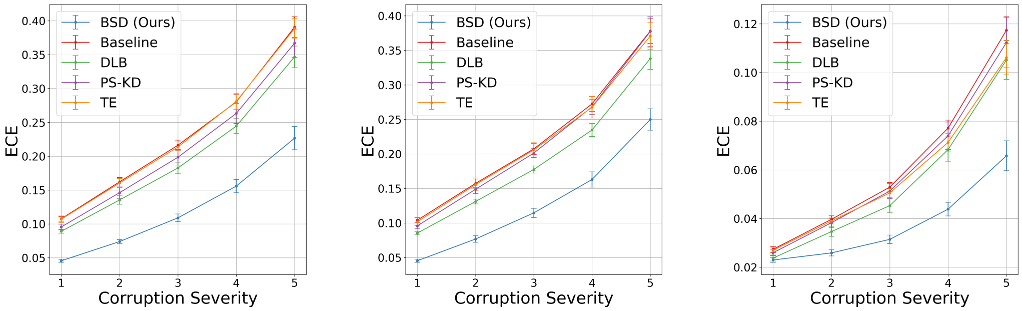
(b) BSD

Figure 14: **Total sample-wise deviation $\delta(\mathbf{x}_i)$ of dark knowledge.** BSD promotes learning of sample-specific information (ResNet-101, Tiny ImageNet).



(a) ResNet-18 on CIFAR-10 (b) ResNet-50 on CIFAR-100 (c) ResNet-101 on Tiny ImageNet

Figure 15: **Evolution of the average Temperature-Adjusted KL Divergence of predictions between the current and final model.** Dark knowledge emerges gradually during training.



(a) ResNet-50 (b) DenseNet-169 (c) ViT-B

Figure 16: **ECE under increasing corruptions.** The models are trained using standard training (baseline), related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021), and the proposed method (BSD), on CIFAR-10 and evaluated on CIFAR-10-C.

Table 6: **Out-of-Distribution Detection Performance (AUROC).** Models trained on in-distribution datasets (CIFAR-10, CIFAR-100) are evaluated against SVHN as the out-of-distribution dataset. The models are trained using standard training (baseline), related methods (Laine & Aila, 2016; Shen et al., 2022; Kim et al., 2021), and the proposed method (BSD). The best results are highlighted in bold.

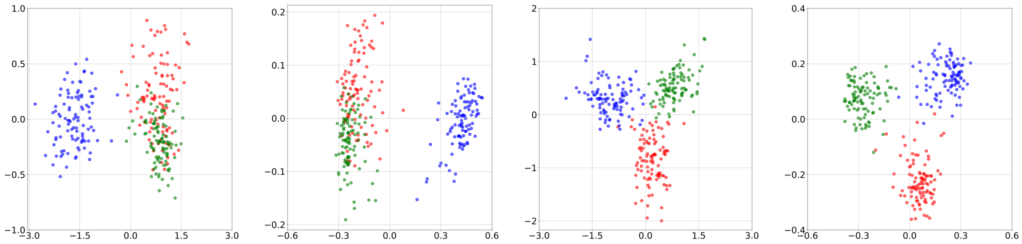
ID Dataset	Model	Baseline (%)	TE (%)	DLB (%)	PS-KD (%)	BSD (%)
CIFAR-10	ResNet-18	90.67±3.42	92.90±0.66	91.25±0.57	92.36±0.48	**94.84** ±0.83
	DenseNet-121	90.96±4.22	**94.05** ±0.98	91.19±1.20	91.11±0.64	93.65±0.16
	ViT-B/16	98.21±0.29	**98.35** ±0.56	98.15±0.22	98.28±0.50	98.03±0.21
CIFAR-100	ResNet-50	74.35±2.23	74.68±1.53	**82.51** ±0.61	76.66±2.40	78.19±1.19
	DenseNet-169	77.92±0.57	68.97±6.85	80.55±3.56	73.95±2.06	**82.41** ±1.79
	ViT-B/16	88.42±1.19	89.12±0.64	89.52±1.16	88.82±0.39	**90.71** ±0.50

model architecture, we observe that BSD yields the most consistent improvement across datasets and architectures. Notably, while TE exhibits instability under this shift on CIFAR-100 (e.g., degrading performance on DenseNet-169 trained on CIFAR-100 with respect to baseline), BSD maintains robust performance. For ViT-B, we observe performance saturation on CIFAR-10 (with all methods $> 98\%$), while for CIFAR-100, BSD yields a notable improvement ($+2.29\%$) over the baseline.

Calibration. We train a WideResNet-40-1 on CIFAR-100 to benchmark BSD against different calibration methods including distillation-based method. We compare against Label Smoothing (LS) (Szegedy et al., 2016), Temperature Scaling (TS) (Guo et al., 2017), MixUp (Zhang et al., 2017), Correctness Ranking Loss (CRL) (Moon et al., 2020), PS-KD (Kim et al., 2021), Multi-class Difference in Confidence and Accuracy (MDCA) (Hebbalaguppe et al., 2022), AdaFocal (Ghosh et al., 2022), Calibration by Pairwise Constraints (CPC) (Cheng & Vasconcelos, 2022), Margin-based Label Smoothing (MbLS) (Liu et al., 2022), Adaptive and Conditional Label Smoothing (ACLS) (Park et al., 2023) and combinations of the aforementioned method with knowledge distillation (Hebbalaguppe, 2024). The results in included in Table 7, where we observe that BSD yields the highest

Table 7: **Calibration Performance for WideResNet-40-1 on CIFAR-100.** Result for all methods (excl. BSD and baseline) are from (Hebbalaguppe, 2024). The best results are highlighted in bold.

Method	Accuracy (%)	ECE (%)	SCE (%)	ACE (%)
Baseline (NLL)	70.04	11.16	0.30	11.19
LS	70.07	1.30	0.21	1.49
TS	70.04	2.57	0.19	2.50
MMCE	69.69	7.34	0.25	7.37
MixUp	72.04	2.57	0.21	2.52
CRL	65.80	13.91	0.37	13.91
PS-KD	72.56	3.73	0.20	3.72
MDCA	68.51	1.35	0.21	1.34
AdaFocal	67.36	2.10	0.21	1.97
CPC	69.99	7.61	0.23	7.55
MBLS	69.97	5.37	0.22	5.37
ACLS	69.92	7.00	0.23	6.99
KD	69.60	15.18	0.37	15.18
KD + MixUp	72.48	1.21	0.20	1.17
KD + AdaFocal	71.70	1.19	0.19	1.34
KD + CPC	70.00	9.02	0.26	9.01
KD + MDCA	71.07	0.98	0.20	1.10
KD + MMCE	72.08	2.02	0.19	1.95
BSD ($\gamma = 0.97, c = 2000$)	72.34±0.26	**0.85** ±0.08	**0.18** ±0.00	**0.87** ±0.19
BSD ($\gamma = 0.96, c = 3000$)	72.62 ±0.20	3.28±0.04	0.19±0.00	3.19±0.12
BSD ($\gamma = 0.95, c = 4000$)	**72.79** ±0.45	7.08±0.40	0.23±0.00	7.08±0.40



(a) Baseline (b) BSD (c) Baseline (d) BSD

Figure 17: **Penultimate layer representations of ResNet-50 on CIFAR-100.** a-b: two semantically similar classes with one dissimilar class. c-d: Three semantically dissimilar classes.

accuracy and the lowest ECE, Static Calibration Error (SCE) and Adaptive Calibration Error (ACE) of all methods.

A.3 Penultimate Layer Representations

Inspired by Müller et al. (2019), we visualize the penultimate layer representations in Figure 17. BSD yields tighter, less overlapping clusters than conventional training, which is somewhat surprising since BSD promotes learning of sample-specific features. It seems that learning similarities between classes can help differentiate among them.

A.4 Limitations

While we experiment with various forms of data augmentation, the interaction with different augmentation schemes as well as regularization techniques warrants further study. Intuitively, augmentations that increase prediction variance may benefit from higher values of the discount factor γ. Finally, BSD requires selecting the discount factor γ and the prior strength c, which, despite the observed performance across a large range of settings, could be viewed as a methodological limitation.

A.5 The Use of Large Language Models

We utilized Large Language Models (LLMs) in a limited capacity to improve the quality of the paper. Specifically, LLMs assisted in writing to improve clarity and grammar, suggested related works for us to consider, and were used as a tool for general feedback. The research ideas, experiments, and methodological design were conceived and carried out by the authors. LLMs did not contribute new scientific insights or results.