A. Connection to Example Difficulty

A related line of work is identifying *difficult* examples, or how well a model can generalize to a given unseen example. Jiang *et al*. [20] introduce a per-instance empirical consistency score which estimates the probability of predicting the ground truth label with models trained on data subsamples of different sizes. Unlike our approach, however, this requires training a large number of models. Toneva *et al*. [32] quantifies example difficulty through the lens of a forgetting event, in which the example is misclassified after being correctly classified. However, the metrics that we introduce in § 3, are based on the disagreement of the label at each checkpoint with the final predicted label. Other approaches estimate the example difficulty by: prediction depth of the first layer at which a k-NN classifier correctly classifies an example [2], the impact of pruning on model predictions of the example [17], and estimating the leave-one-out influence of each training example on the accuracy of an algorithm by using influence functions [7]. Closest to our method, the work of Agarwal *et al*. [1] utilizes gradients of intermediate models during training to rank examples by difficulty. In particular, they average pixel-wise variance of gradients for each given input image. Notably, this approach is more costly and less practical than our approach and also does not study the accuracy/coverage tradeoff which is of paramount importance to SC.

B. Alternative Metric Choices

We briefly discuss additional potential metric choices that we investigated but which lead to SC performance worse than s_{avg}. A more elaborate discussion of these results is provided in Appendix C.3.

B.1. Incorporating Estimates of e_t into s_{min} and s_{avg}

Since the results in § 4.3 show that e_t is only nearly 0 almost everywhere, we investigate whether incorporating an estimate of e_t into s_{min} and s_{avg} leads to additional SC improvements. Our results demonstrate that neither an empirical estimate nor a smooth decay function similar to v_t robustly improves over s_{min} and s_{avg} (which assumed $e_t = 0$).

B.2. Jump Score s_{jmp}

We also consider a score which captures the level of disagreement between the predicted label of two successive intermediate models (*i.e*. how much jumping occurred over the course of training). For $j_t = 0$ iff $f_t(\boldsymbol{x}) = f_{t-1}(\boldsymbol{x})$ and $j_t = 1$ otherwise we can compute the jump score as $s_{\text{jmp}} = 1 - \sum v_t j_t$ and threshold it as in § 3.2 and § 3.3.

B.3. Variance Score s_{var} for Continuous Metrics

Finally, we consider monitoring the evolution of continuous metrics that have been shown to be correlated with example difficulty. More specifically, Jiang *et al*. [20] show that example difficulty is correlated with confidence and negative entropy. Moreover, they find that difficult examples are learned later in the training process. This observation motivates designing a score based on these continuous metrics that penalises changes later in the training process more heavily. We consider the maximum softmax class probability known as confidence, the negative entropy and the gap between the most confident classes for each example instead of the model predictions. Assume that any of these metrics is given by a sequence $z = \{z_1, \ldots, z_T\}$ obtained from T intermediate models. Then we can capture the uniformity of z via a (weighted) variance score $s_{\text{var}} = \sum_t w_t (z_t - \mu)^2$ for mean $\mu = \frac{1}{T}\sum_t z_t$ and an increasing weighting sequence $w = \{w_1, \ldots, w_T\}$.

In order to show the effectiveness of the variance score s_{var} for continuous metrics, we provide a simple bound on the variance of confidence $\max_{y \in \mathcal{Y}} f_t(\boldsymbol{x})$ in the final checkpoints of the training. Assuming that the model has converged to a local minima with a low learning rate, we can assume that the distribution of model weights can be approximated by a Gaussian distribution.

We consider a linear regression problem where the inputs are linearly separable.

Lemma 2. *Assume that we have some Gaussian prior on the model parameters in the logistic regression setting across m final checkpoints. More specifically, given T total checkpoints of model parameters $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T\}$ we have $p(W = \boldsymbol{w}_t) = \mathcal{N}(\boldsymbol{w}_0 \mid \boldsymbol{\mu}, s\boldsymbol{I})$ for $t \in \{T - m + 1, \ldots, T\}$ and we assume that final checkpoints of the model are sampled from this distribution. We show that the variance of model confidence $\max_{y \in \{-1,1\}} p(y \mid \boldsymbol{x}_i, \boldsymbol{w}_t)$ for a datapoint $(\boldsymbol{x}_i, y_i)$ can be upper bounded by a factor of probability of correctly classifying this example by the optimal weights.*

Proof. We first compute the variance of model predictions $p(y_i \mid \boldsymbol{x}_i, W)$ for a given datapoint $(\boldsymbol{x}_i, y_i)$. Following previous work [4, 30], the variance of predictions over these checkpoints can be estimated as follows:

Taking two terms in Taylor expansion for model predictions we have $p(y_i \mid \boldsymbol{x}_i, W) \simeq p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}) + g_i(\boldsymbol{w})^\top (W - \boldsymbol{w})$ where W and $\boldsymbol{w}$ are current and the expected estimate of the parameters and $g_i(\boldsymbol{w}) = p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})(1 - p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}))\boldsymbol{x}_i$ is the gradient vector. Now we can write the variance with respect to the model prior as:

$$\mathbb{V}\left(p(y_i \mid \boldsymbol{x}_i, W)\right) \simeq \mathbb{V}\left(g_i(\boldsymbol{w})^\top (W - \boldsymbol{w})\right) = g_i(\boldsymbol{w})^\top F^{-1} g_i(\boldsymbol{w})$$

where F is the variance of posterior distribution $p(W \mid X, Y) \sim \mathcal{N}(W \mid \boldsymbol{w}, F^{-1})$. This suggests that the variance of probability of correctly classifying $\boldsymbol{x}_i$ is proportional to $p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})^2 (1 - p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}))^2$. Now we can bound

the variance of maximum class probability or confidence as below:

$$\begin{aligned}\mathbb{V}\left(\max_{y\in\{-1,1\}} p(y \mid \boldsymbol{x}_i, W)\right) &\leq \mathbb{V}\left(p(y_i \mid \boldsymbol{x}_i, W)\right) \\ &\quad + \mathbb{V}\left(p(-y_i \mid \boldsymbol{x}_i, W)\right) \\ &\approx 2p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w})^2 \\ &\quad \cdot (1 - p(y_i \mid \boldsymbol{x}_i, \boldsymbol{w}))^2 \\ &\quad \cdot \boldsymbol{x}_i^\top F^{-1} \boldsymbol{x}_i\end{aligned}$$

□

We showed that if the probability of correctly classifying an example given the final estimate of model parameters is close to one, the variance of model predictions following a Gaussian Prior gets close to zero, we expect a similar behaviour for the variance of confidence under samples of this distribution.

C. Extension of Empirical Evaluation

C.1. Comparison with One-Sided Prediction

In addition to the main results, we also compare NNTD to one-sided prediction (OSP). For this set of experiments, we evaluate our proposed approach on image dataset benchmarks that are common in the selective classification literature: CIFAR-10, SVHN, and Cats & Dogs. For each dataset, we train a deep neural network following the ResNet-32 architecture [16] and checkpoint each model after processing 50 mini-batches of size 128. All models are trained over 200 epochs. SVHN, Cats & Dogs, and GTSRB are trained using the Adam optimizer with learning rate 10^{-3}. The CIFAR-10 and CIFAR-100 models are trained using momentum-based stochastic gradient descent with an initial learning rate of 10^{-1} on a multi-step learning rate reduction schedule (reduction by 10^{-1} after epochs 100 and 150), momentum 0.9, and weight decay 10^{-4}. We report these results in Tables 4 and 5.

C.2. Full Hyper-Parameters

We document full hyper-parameter settings for our method (NNTD) as well as all baseline approaches in Table 6. Baseline hyper-parameters are consistent with Gangrade *et al.* [10] in the OSP setting and follow the setup from Huang *et al.* [19] (Appendix A.4) for our main set of experiments. Both DE and MC-DO use an ensemble of 10 models.

C.3. Additional Selective Classification Results

C.3.1 Variance Score Experiments on Continuous Metrics

In addition to the evolution of predictions made across intermediate models, we also monitor a variety of easily computable continuous metrics. In our work, we consider the following metrics:

- Confidence (conf): $\max_{c\in\mathcal{Y}} f_t(\boldsymbol{x})$
- Confidence gap between top 2 most confident classes (gap): $\max_{c\in\mathcal{Y}} f_t(\boldsymbol{x}) - \max_{c\neq\hat{y}} f_t(\boldsymbol{x})$
- Entropy (ent): $-\sum_{c=1}^{C} f_t(\boldsymbol{x})_c \log\left(f_t(\boldsymbol{x})_c\right)$

Note that for the purpose of this experiment, we adapt the notation of f to map to a softmax-parameterized output instead of the hard thresholded label, formally $f : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$.

We depict example evolution plots for these metrics in Figures 7, 8, 9, 10, and 11 in the third row.

C.3.2 Individual Metric Evolution Plots

We provide additional individual metric evolution plots for SVHN (Figure 7), GTSRB (Figure 8), CIFAR-10 (Figure 9), CIFAR-100 (Figure 10), and Cats & Dogs (Figure 11).

C.3.3 Incorporating Estimates of e_t into s_{min} and s_{avg}

Our results in § 4.3 show that e_t is only nearly 0 almost everywhere, we investigate whether incorporating an estimate of e_t into s_{min} and s_{avg} leads to additional SC improvements. Recall that in Lemma 1 we gave an upper-bound on the probability that the test point is correctly classified as $\frac{v_t}{|a_t - e_t|^2}$. In the case where e_t is not 0 everywhere, we can adjust

$$s_{\text{min}} = \min_{t \text{ s.t } a_t = 1} \frac{v_t}{|a_t - e_t|^2} \qquad s_{\text{avg}} = \frac{\sum \frac{v_t}{|a_t - e_t|^2} a_t}{\sum a_t} \tag{4}$$

accordingly. In Figure 12, we experimentally test whether incorporating an empirical estimate (first row) or a smooth decay function similar to v_i (second row) robustly improve over s_{min} and s_{avg} with $e_t = 0$. Our results show that neither setting outperforms our default setting with $e_t = 0$.

C.3.4 Performance of Jump Score s_{jmp} and Weighted Variance s_{var}

As discussed in Appendix B, we also investigated whether the jump score s_{jmp} or the weighted variance of continuous metrics s_{var} can be used as an effective score for SC. As we show in Figure 13, none of these metrics robustly outperforms our main method $\texttt{NNTD}(s_{\text{avg}}, 0.05)$.

C.3.5 Concave Weighting for v_t

As we have empirically analyzed in § 4.3, the variances v_t follow a monotonically decreasing and convex trend. This inspires our best performing method $\texttt{NNTD}(s_{\text{avg}}, 0.05)$ to

Table 4. **Performance at low target errors for OSP based setup**. Our NNTD$(s_{\text{avg}}, 0.05)$ method either matches or provides higher test set coverage at a fixed target error than other competing methods. Bold entries are within one standard deviation of each other over 5 random runs.

Dataset	Target	NNTD		OSP		SR		SN		DG	
	Error	Cov.	Error	Cov.	Error	Cov.	Error	Cov.	Error	Cov.	Error
CIFAR-10	2%	**83.3**	1.96	80.6	1.91	75.1	2.09	73.0	2.31	74.2	1.98
	1%	**79.7**	1.05	74.0	1.02	67.2	1.09	64.5	1.02	66.4	1.01
	0.5%	**74.2**	0.49	64.1	0.51	59.3	0.53	57.6	0.48	57.8	0.51
SVHN	2%	**95.7**	1.96	**95.8**	1.99	95.7	2.06	93.5	2.03	94.8	1.99
	1%	**91.2**	0.99	90.1	1.03	88.4	0.99	86.5	1.04	89.5	1.01
	0.5%	**83.9**	0.50	82.4	0.51	77.3	0.51	79.2	0.51	81.6	0.49
Cats & Dogs	2%	**90.5**	2.03	**90.5**	1.98	88.2	2.03	84.3	1.94	87.4	1.94
	1%	**85.6**	1.01	**85.4**	0.98	80.2	0.97	78.0	0.98	81.7	0.98
	0.5%	77.5	0.50	**78.7**	0.49	73.2	0.49	70.5	0.46	74.5	0.48

Table 5. **Performance at high target coverage for OSP based setup**. Similar as demonstrated in Table 4, NNTD$(s_{\text{avg}}, 0.05)$ matches or outperforms competing error rates at fixed coverage levels.

Dataset	Target	NNTD		OSP		SR		SN		DG	
	Coverage	Cov.	Error	Cov.	Error	Cov.	Error	Cov.	Error	Cov.	Error
CIFAR-10	100%	100	**9.57**	100	9.74	99.99	**9.58**	100	11.07	100	10.81
	95%	95.1	**6.13**	95.1	6.98	95.2	8.74	94.7	8.34	95.1	8.21
	90%	90.1	**4.16**	90.0	4.67	90.5	6.52	89.6	6.45	90.1	6.14
SVHN	100%	100	4.11	100	4.27	99.97	**3.86**	100	4.27	100	4.03
	95%	94.8	**1.80**	95.1	**1.83**	95.1	1.86	95.1	2.53	95.0	2.05
	90%	90.1	**0.77**	90.1	1.01	90.0	1.04	90.1	1.31	90.0	1.06
Cats & Dogs	100%	100	**5.18**	100	5.93	100	5.72	100	7.36	100	6.16
	95%	94.9	**2.99**	95.1	**2.97**	95.0	3.46	95.2	5.1	95.1	4.28
	90%	90.0	1.87	90.0	**1.74**	90.0	2.28	90.2	3.3	90.0	2.50

use $k = 0.05$ for $v_i = 1 - i^k$. As a sanity check, we examine whether a concave weighting yielded by $v_i = 1 - i^k$ for $k \geq 1$. As we demonstrate in Figure 14, the best concave weighting is given by $k = 1$. Therefore, we confirm that no concave weighting outperforms the convex weightings analyzed in Figure 4.

C.3.6 Limiting NNTD to a Subset of Last Checkpoints

In order to determine which training stages are important for selective classification, we perform an experiment on CIFAR-10 and SVHN where we limit ourselves to a subset of the last checkpoints. In particular, we examine the coverage/error trade-off for only including the last $\{10\%, 20\%, 50\%, 80\%, 90\%, 100\%\}$ of checkpoints. We report our findings in Table 7. We see that taking into account more than 80% of checkpoints does lead to diminishing returns and that only taking into account 50% or less of the total numbers of checkpoints does not lead to state-of-the-art selective classification performance.

C.3.7 Applying OOD Scores to Selective Classification

Finally, we also compare popular out-of-distribution detection approaches to our NNTD method. In particular, we apply three state-of-the-art approaches, namely energy-based OOD detection (Energy) [26], Mahalanobis-distance-based OOD detection (Mahalanobis) [24], and ODIN (ODIN) [25] to the CIFAR-10 and SVHN test sets and document these results in Table 8. Overall, we find that SOTA out-of-distribution detection techniques are not well equipped to attain SOTA selective classification performance; NNTD outperforms these methods by a large margin.

Table 6. **Hyper-parameters used for all SC algorithms in the `OSP` setup.**

Dataset	SC Algorithm	Hyper-Parameters
CIFAR-10	Softmax Response (`SR`)	$t = 0.0445$
	Selective Net (`SN`)	$\lambda = 32, c = 0.51, t = 0.24$
	Deep Gamblers (`DG`)	$o = 1.179, t = 0.03$
	One-Sided Prediction (`OSP`)	$\mu = 0.49, t = 0.8884$
	Neural Network Training Dynamics (`NNTD`)	$T = 1593, k = 0.05$
SVHN	Softmax Response (`SR`)	$t = 0.0224$
	Selective Net (`SN`)	$\lambda = 32, c = 0.79, t = 0.86$
	Deep Gamblers (`DG`)	$o = 1.13, t = 0.23$
	One-Sided Prediction (`OSP`)	$\mu = 1.67, t = 0.9762$
	Neural Network Training Dynamics (`NNTD`)	$T = 1195, k = 0.05$
Cats & Dogs	Softmax Response (`SR`)	$t = 0.029$
	Selective Net (`SN`)	$\lambda = 32, c = 0.7, t = 0.73$
	Deep Gamblers (`DG`)	$o = 1.34, t = 0.06$
	One-Sided Prediction (`OSP`)	$\mu = 1.67, t = 0.9532$
	Neural Network Training Dynamics (`NNTD`)	$T = 797, k = 0.05$

Table 7. **Coverage/Error trade-off of `NNTD` as a function of the accounted data points (CIFAR-10 left, SVHN right)**. We observe that large parts of the training process contain valuable sigals for selective classification.

Target Coverage	100%	90%	80%	50%	20%	10%
100%	6.07	6.08	6.10	6.15	6.18	6.20
95%	3.24	3.25	3.25	3.31	3.34	3.39
90%	1.83	1.83	1.84	1.88	1.91	1.93
80%	0.64	0.65	0.67	0.72	0.78	0.79
70%	0.34	0.34	0.36	0.38	0.40	0.40

Target Coverage	100%	90%	80%	50%	20%	10%
100%	2.68	2.70	2.73	2.83	2.88	2.92
95%	0.88	0.90	0.93	1.01	1.09	1.11
90%	0.55	0.57	0.59	0.62	0.68	0.73
80%	0.38	0.39	0.40	0.45	0.51	0.53
70%	0.33	0.33	0.35	0.41	0.44	0.45

Table 8. **Performance of `NNTD` in comparison with OOD scores**. We find that `NNTD` significantly outperforms common OOD detection approaches for the purpose of selective classification.

Dataset	Target	`NNTD`		`Energy`		`Mahalanobis`		`ODIN`	
	Coverage	Cov	Err ↓	Cov	Err ↓	Cov	Err ↓	Cov	Err ↓
CIFAR-10	100%	100	**6.07**	100	**6.07**	100	**6.07**	100	**6.07**
	95%	95.0	**3.24**	95.0	3.99	95.0	4.44	95.1	6.04
	90%	90.1	**1.83**	90.0	2.67	90.0	2.93	88.9	6.01
	80%	79.9	**0.64**	80.0	1.10	80.1	1.20	88.9	6.01
	70%	69.8	**0.34**	70.0	0.83	69.9	0.82	68.3	4.41
SVHN	100%	100	**2.68**	100	**2.68**	100	**2.68**	100	**2.68**
	95%	95.0	**0.88**	95.1	1.35	95.1	1.51	95.0	2.67
	90%	90.1	**0.55**	89.9	0.92	90.0	1.04	89.9	2.64
	80%	79.9	**0.38**	79.9	0.70	80.3	0.70	81.9	2.51
	70%	69.8	**0.33**	70.0	0.58	69.8	0.58	74.9	2.18

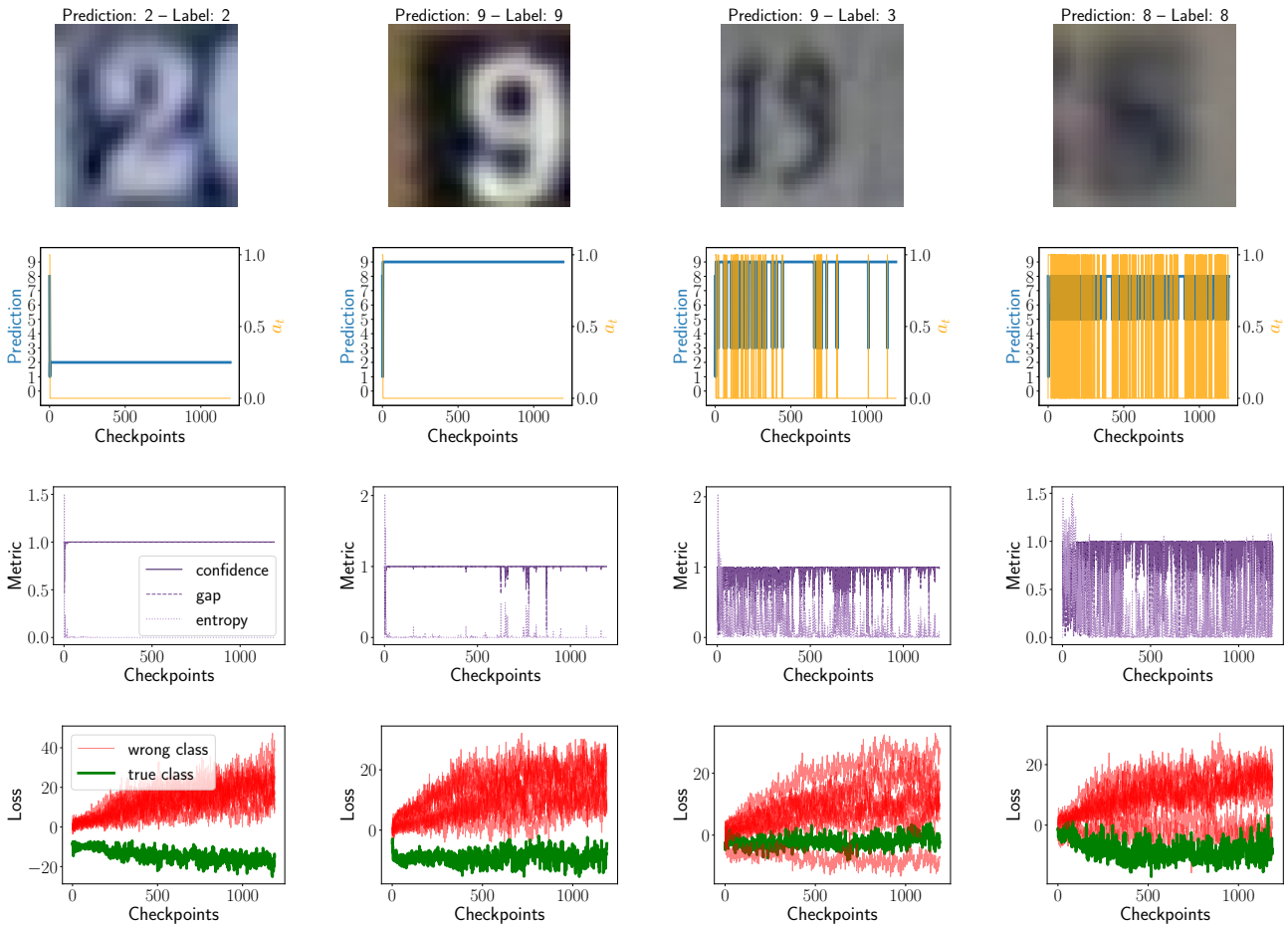


Figure 7. **Individual SVHN examples**. We extend Figure 2 by also including the evolution of the confidence, the gap, and the entropy over the course of training in the third row. In the final row, we further plot the loss evolution for all candidate classes. We note that the first two examples are easy to classify as indicated by mostly stationary metrics and a clear loss separation, while the last two examples are harder to classify since they exhibit highly erratic metrics and ambiguous loss curves.

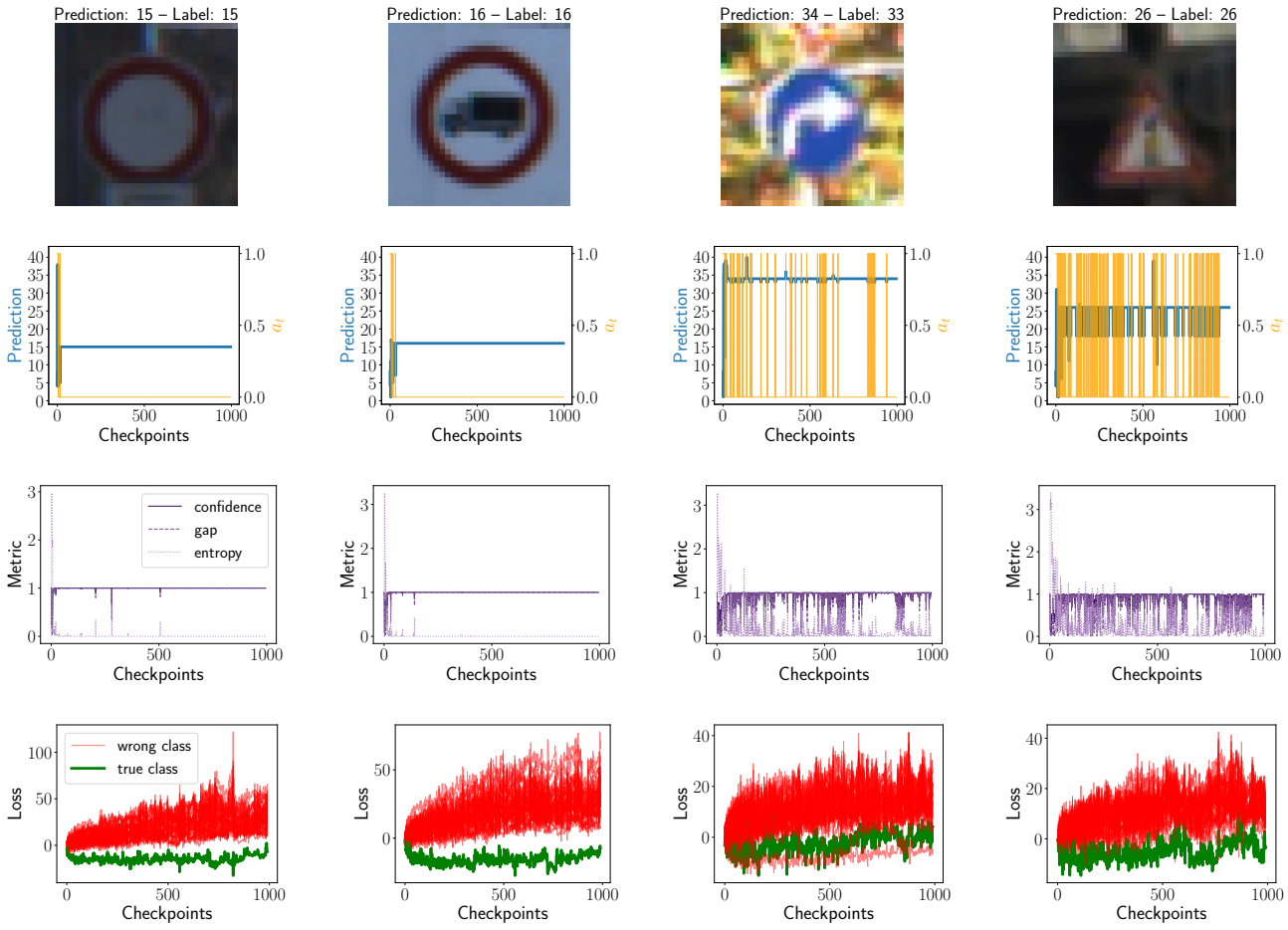


Figure 8. **Individual GTSRB examples**. Similar as Figure 7.

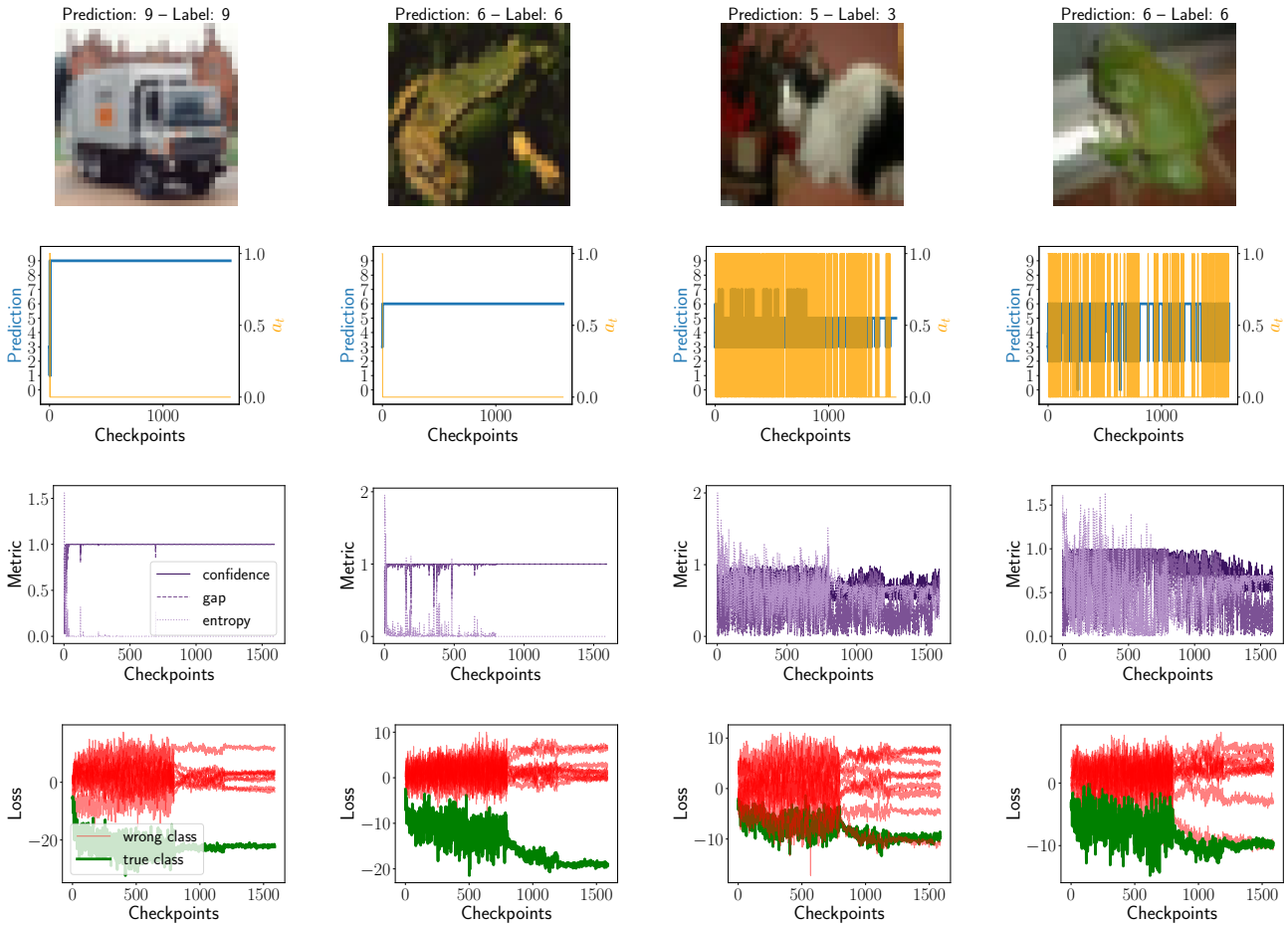


Figure 9. **Individual CIFAR-10 examples**. Similar as Figure 7.

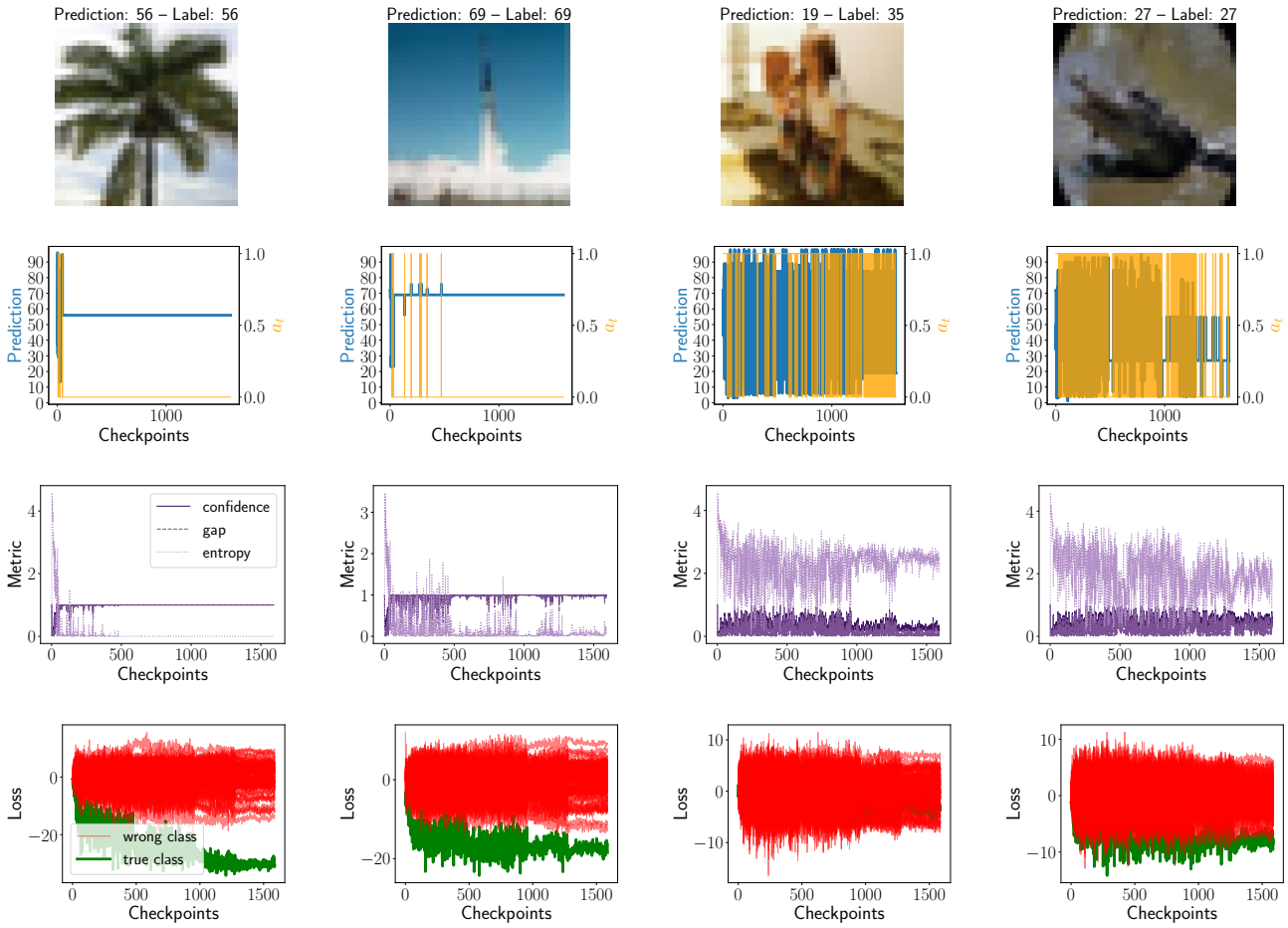


Figure 10. **Individual CIFAR-100 examples**. Similar as Figure 7.

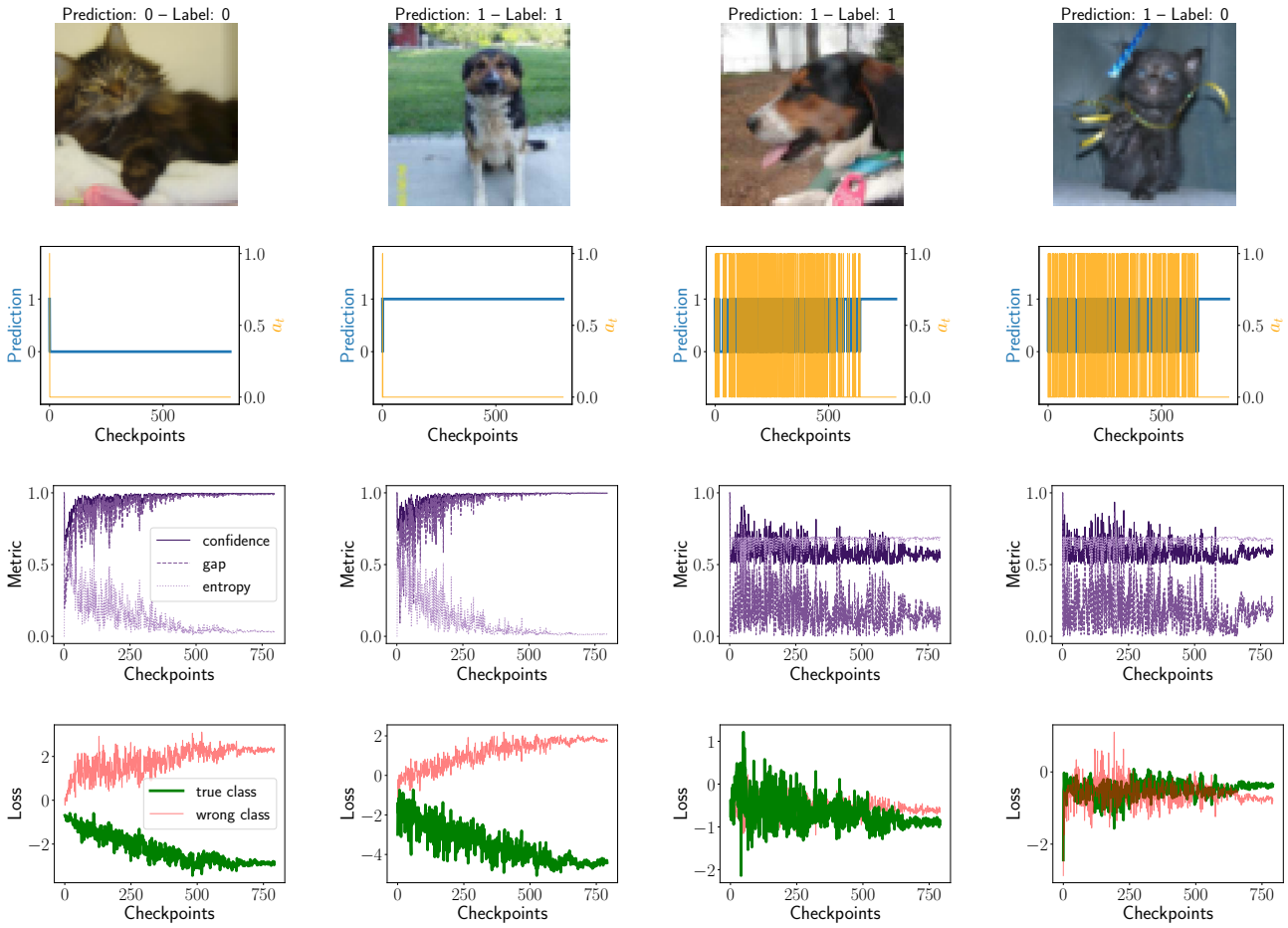
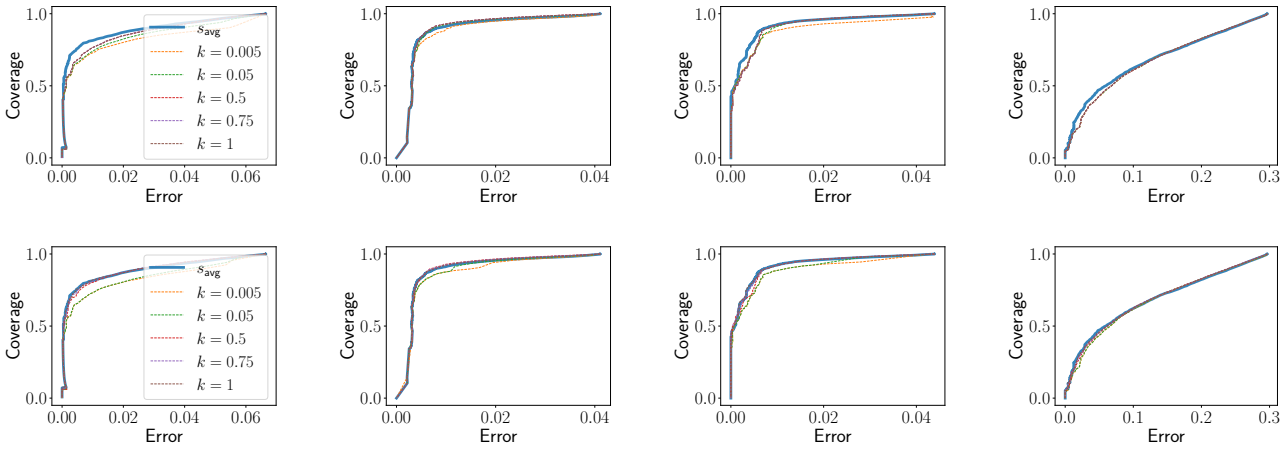
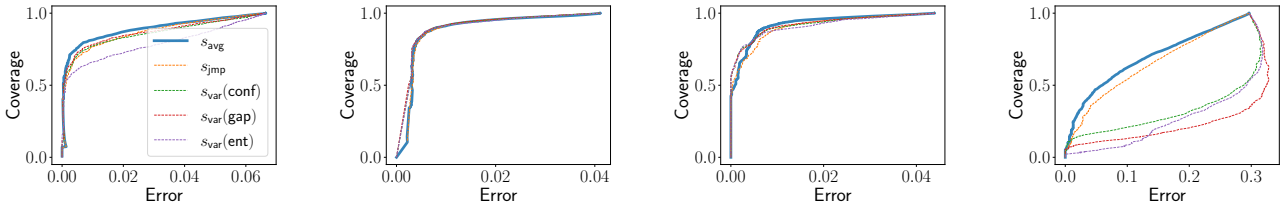


Figure 11. **Individual Cats & Dogs examples**. Similar as Figure 7.



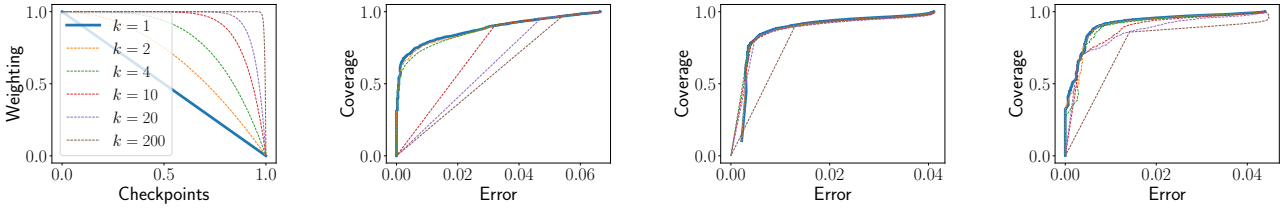
(a) CIFAR-10 (b) SVHN (c) GTSRB (d) CIFAR-100

Figure 12. **Coverage/error trade-off when incorporating e_t into s_{min} and s_{avg}.** In the first row, the solid blue line corresponds to $\texttt{NNTD}(s_{\text{avg}}, 0.05)$ while the dashed lines show the performance when we incorporate an empirical estimate of e_t for various weightings v_i. In the second row, maintaining the solid blue line as $\texttt{NNTD}(s_{\text{avg}}, 0.05)$, we fix v_t at $k = 0.05$ and test various continuous approximations of e_t as $e_t = 1 - t^k$ for $k \in (0, 1]$. Overall, we find that introducing an adaptable e_t does not further improve the performance of s_{min} and s_{avg} with $e_t = 0$.



(a) CIFAR-10 (b) SVHN (c) GTSRB (d) CIFAR-100

Figure 13. **Coverage/error trade-off of `NNTD` for alternate scores**. We find that neither the jump score, nor any of the weighted variance metrics at their optimal k for v_i outperform $\texttt{NNTD}(s_{\text{avg}}, 0.05)$.



(a) Concave weighting (b) CIFAR-10 (c) SVHN (d) GTSRB

Figure 14. **Concave weighting used in $v_t = 1 - t^k$.** Overall, since the best concave weighting is given by $k = 1$, we conclude that no concave weighting outperforms any convex weighting.

CVPR 2023 Submission #12608. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Table 9. **Performance at low target errors**. Same results as in Table 1 but includes standard deviations.

Dataset	Target	NNTD		SAT		DG		SN		SR		MC-DO	
	Error	Cov ↑	Err	Cov ↑	Err	Cov ↑	Err	Cov ↑	Err	Cov ↑	Err	Cov ↑	Err
CIFAR-10	2%	**91.2** (± 0.05)	1.99 (± 0.01)	90.3 (± 0.03)	1.97 (± 0.02)	89.1 (± 0.04)	2.02 (± 0.01)	88.3 (± 0.03)	2.03 (± 0.02)	85.8 (± 0.08)	1.98 (± 0.02)	86.1 (± 0.07)	2.01 (± 0.01)
	1%	**86.4** (± 0.03)	1.00 (± 0.01)	**86.1** (± 0.02)	1.02 (± 0.02)	85.5 (± 0.07)	1.03 (± 0.02)	84.4 (± 0.06)	0.98 (± 0.02)	79.1 (± 0.08)	1.01 (± 0.03)	79.9 (± 0.08)	1.01 (± 0.03)
	0.5%	**75.9** (± 0.04)	0.49 (± 0.02)	**76.0** (± 0.05)	0.51 (± 0.02)	75.2 (± 0.08)	0.5 (± 0.03)	74.7 (± 0.09)	0.49 (± 0.01)	71.2 (± 0.05)	0.51 (± 0.02)	72.0 (± 0.03)	0.50 (± 0.03)
SVHN	2%	**98.5** (± 0.05)	1.98 (± 0.03)	**98.2** (± 0.04)	1.99 (± 0.02)	97.8 (± 0.03)	2.06 (± 0.05)	97.7 (± 0.06)	2.03 (± 0.02)	97.6 (± 0.05)	1.99 (± 0.03)	97.9 (± 0.03)	2.00 (± 0.04)
	1%	**96.3** (± 0.03)	0.99 (± 0.02)	95.7 (± 0.02)	1.03 (± 0.03)	94.8 (± 0.04)	0.99 (± 0.01)	94.5 (± 0.03)	1.04 (± 0.02)	93.5 (± 0.05)	1.01 (± 0.03)	94.1 (± 0.06)	0.97 (± 0.02)
	0.5%	**88.1** (± 0.02)	0.50 (± 0.02)	**87.9** (± 0.05)	0.51 (± 0.01)	86.4 (± 0.04)	0.51 (± 0.01)	86.0 (± 0.04)	0.51 (± 0.01)	70.0 (± 0.09)	0.50 (± 0.04)	70.1 (± 0.04)	0.49 (± 0.03)
Cats & Dogs	2%	97.7 (± 0.09)	2.01 (± 0.03)	**98.2** (± 0.02)	1.98 (± 0.03)	**98.0** (± 0.05)	2.03 (± 0.02)	97.4 (± 0.05)	1.98 (± 0.03)	95.1 (± 0.12)	1.99 (± 0.04)	95.7 (± 0.10)	1.99 (± 0.03)
	1%	93.1 (± 0.03)	1.01 (± 0.02)	**93.6** (± 0.02)	0.98 (± 0.03)	92.6 (± 0.08)	0.97 (± 0.04)	92.2 (± 0.05)	0.98 (± 0.02)	86.9 (± 0.13)	0.98 (± 0.04)	88.6 (± 0.09)	1.01 (± 0.02)
	0.5%	**85.7** (± 0.06)	0.51 (± 0.02)	**86.0** (± 0.04)	0.49 (± 0.01)	85.3 (± 0.02)	0.49 (± 0.02)	84.8 (± 0.03)	0.46 (± 0.05)	68.4 (± 0.10)	0.48 (± 0.02)	70.1 (± 0.08)	0.51 (± 0.03)

Table 10. **Performance at high target coverage**. Same results as in Table 2 but includes standard deviations.

Dataset	Target	NNTD		SAT		DG		SN		SR		MC-DO	
	Coverage	Cov	Err ↓	Cov	Err ↓	Cov	Err ↓	Cov	Err ↓	Cov	Err ↓	Cov	Err ↓
CIFAR-10	100%	100 (± 0.00)	**6.07** (± 0.05)	100 (± 0.00)	**6.06** (± 0.03)	100 (± 0.00)	6.11 (± 0.05)	100 (± 0.00)	6.13 (± 0.03)	100 (± 0.00)	**6.07** (± 0.05)	100 (± 0.00)	**6.07** (± 0.05)
	95%	95.0 (± 0.01)	**3.24** (± 0.03)	95.1 (± 0.01)	3.32 (± 0.05)	95.1 (± 0.02)	3.47 (± 0.06)	95.0 (± 0.01)	4.08 (± 0.08)	94.9 (± 0.03)	4.48 (± 0.07)	95.1 (± 0.04)	4.48 (± 0.09)
	90%	90.1 (± 0.02)	**1.83** (± 0.04)	89.9 (± 0.02)	1.90 (± 0.02)	90.0 (± 0.01)	2.19 (± 0.05)	90.1 (± 0.02)	2.29 (± 0.03)	90.1 (± 0.02)	2.78 (± 0.06)	90.0 (± 0.02)	2.87 (± 0.07)
	80%	79.9 (± 0.02)	**0.64** (± 0.03)	80.0 (± 0.00)	**0.65** (± 0.04)	80.1 (± 0.03)	**0.66** (± 0.04)	80.1 (± 0.01)	0.81 (± 0.07)	79.8 (± 0.02)	1.05 (± 0.08)	79.9 (± 0.01)	1.01 (± 0.03)
	70%	69.8 (± 0.03)	**0.34** (± 0.04)	69.9 (± 0.03)	**0.32** (± 0.05)	69.8 (± 0.04)	0.41 (± 0.04)	70.2 (± 0.03)	0.30 (± 0.02)	70.0 (± 0.01)	0.47 (± 0.07)	70.1 (± 0.02)	0.42 (± 0.05)
SVHN	100%	100 (± 0.00)	**2.68** (± 0.02)	100 (± 0.00)	**2.71** (± 0.03)	100 (± 0.00)	**2.72** (± 0.05)	100 (± 0.00)	2.77 (± 0.06)	100 (± 0.00)	**2.68** (± 0.02)	100 (± 0.00)	**2.68** (± 0.02)
	95%	95.0 (± 0.01)	**0.88** (± 0.02)	95.1 (± 0.02)	0.95 (± 0.03)	95.1 (± 0.01)	1.01 (± 0.06)	95.0 (± 0.02)	1.07 (± 0.05)	94.9 (± 0.03)	1.15 (± 0.11)	95.1 (± 0.02)	1.12 (± 0.07)
	90%	90.1 (± 0.02)	**0.55** (± 0.4)	89.9 (± 0.01)	**0.58** (± 0.01)	90.0 (± 0.00)	0.63 (± 0.06)	90.1 (± 0.01)	0.71 (± 0.04)	90.1 (± 0.02)	0.82 (± 0.06)	90.0 (± 0.02)	0.76 (± 0.03)
	80%	79.9 (± 0.01)	**0.38** (± 0.02)	80.0 (± 0.01)	**0.37** (± 0.02)	80.1 (± 0.01)	0.43 (± 0.01)	80.1 (± 0.00)	0.48 (± 0.02)	79.8 (± 0.03)	0.55 (± 0.09)	79.9 (± 0.02)	0.53 (± 0.08)
	70%	69.8 (± 0.03)	**0.33** (± 0.03)	69.9 (± 0.02)	**0.33** (± 0.01)	69.8 (± 0.04)	**0.35** (± 0.02)	70.2 (± 0.02)	0.45 (± 0.06)	70.0 (± 0.01)	0.50 (± 0.05)	70.1 (± 0.02)	0.49 (± 0.06)
Cats & Dogs	100%	100 (± 0.00)	3.48 (± 0.01)	100 (± 0.00)	**3.45** (± 0.01)	100 (± 0.00)	**3.41** (± 0.03)	100 (± 0.00)	3.56 (± 0.04)	100 (± 0.00)	3.48 (± 0.01)	100 (± 0.00)	3.48 (± 0.01)
	95%	95.1 (± 0.02)	1.51 (± 0.03)	95.1 (± 0.03)	**1.45** (± 0.02)	95.0 (± 0.01)	**1.43** (± 0.02)	94.9 (± 0.02)	1.61 (± 0.05)	94.8 (± 0.03)	1.92 (± 0.12)	95.1 (± 0.02)	1.95 (± 0.08)
	90%	90.1 (± 0.03)	**0.60** (± 0.03)	89.9 (± 0.02)	**0.57** (± 0.03)	90.0 (± 0.01)	0.69 (± 0.04)	90.1 (± 0.02)	0.95 (± 0.11)	90.1 (± 0.03)	1.13 (± 0.07)	90.0 (± 0.01)	1.09 (± 0.06)
	80%	79.9 (± 0.01)	**0.42** (± 0.04)	80.0 (± 0.01)	**0.41** (± 0.03)	80.1 (± 0.02)	0.56 (± 0.02)	80.1 (± 0.03)	0.39 (± 0.05)	79.8 (± 0.02)	0.69 (± 0.07)	79.9 (± 0.02)	0.58 (± 0.05)
	70%	69.8 (± 0.02)	**0.36** (± 0.05)	69.9 (± 0.03)	**0.33** (± 0.03)	69.8 (± 0.03)	0.45 (± 0.05)	70.2 (± 0.03)	**0.33** (± 0.03)	70.0 (± 0.01)	0.62 (± 0.06)	70.1 (± 0.02)	0.51 (± 0.04)

CVPR #12608