Training Private Models That Know What They Don't Know

Stephan Rabanser
stephan@cs.toronto.edu

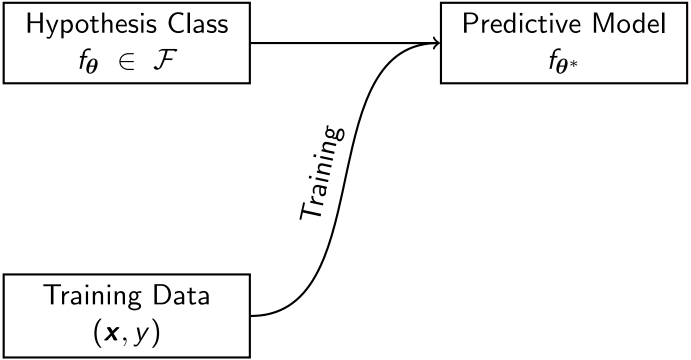
September 21, 2023

Warmup: (Supervised) Machine Learning Pipeline

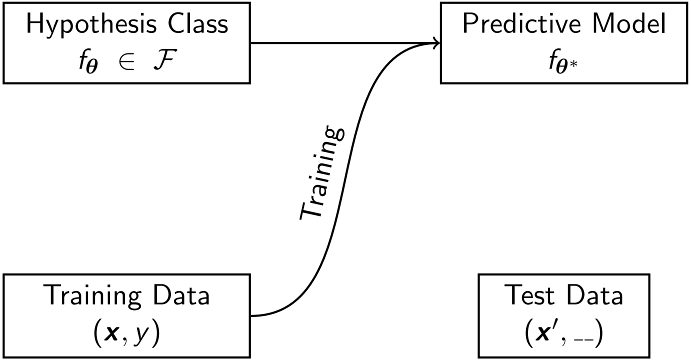
Hypothesis Class
$f_{\boldsymbol{\theta}} \in \mathcal{F}$

Training Data
$(\boldsymbol{x}, y)$

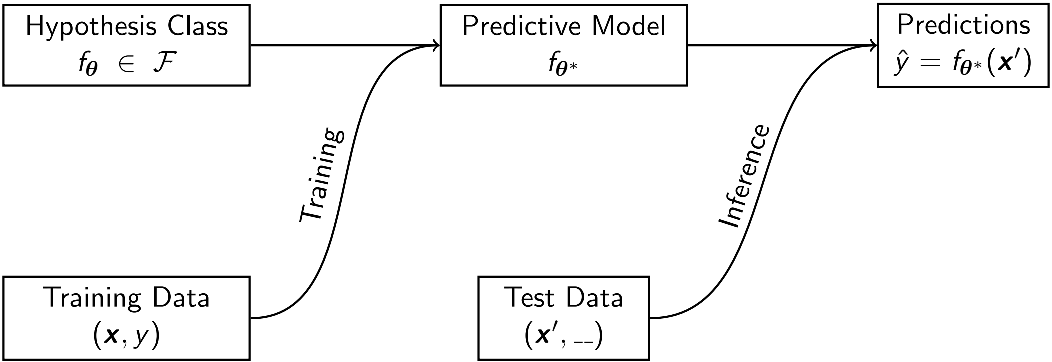
Warmup: (Supervised) Machine Learning Pipeline



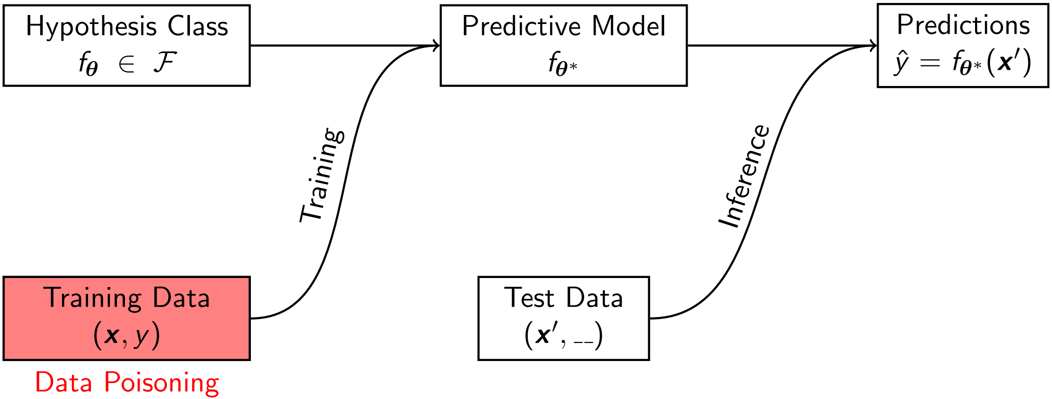
Warmup: (Supervised) Machine Learning Pipeline



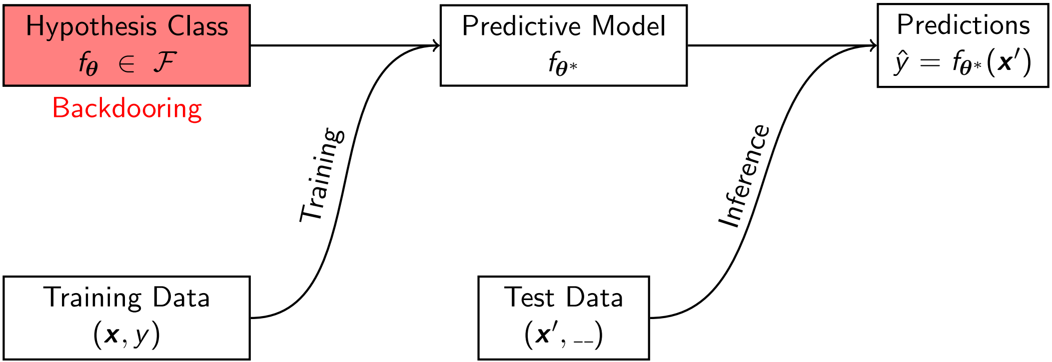
Warmup: (Supervised) Machine Learning Pipeline



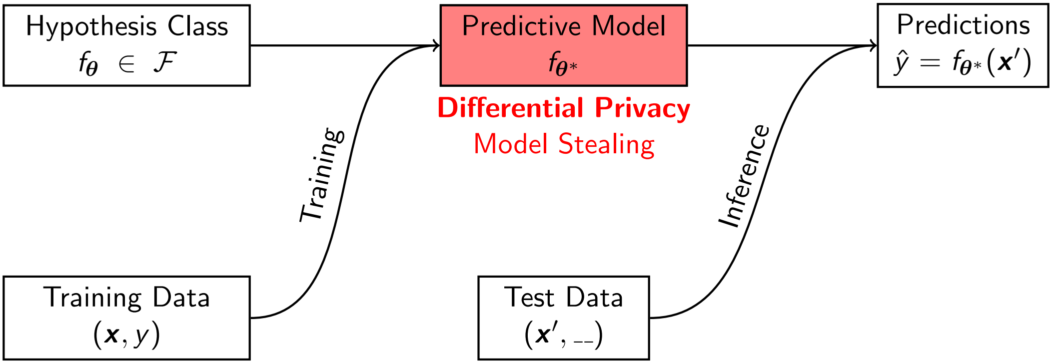
Warmup: (Supervised) Machine Learning Pipeline



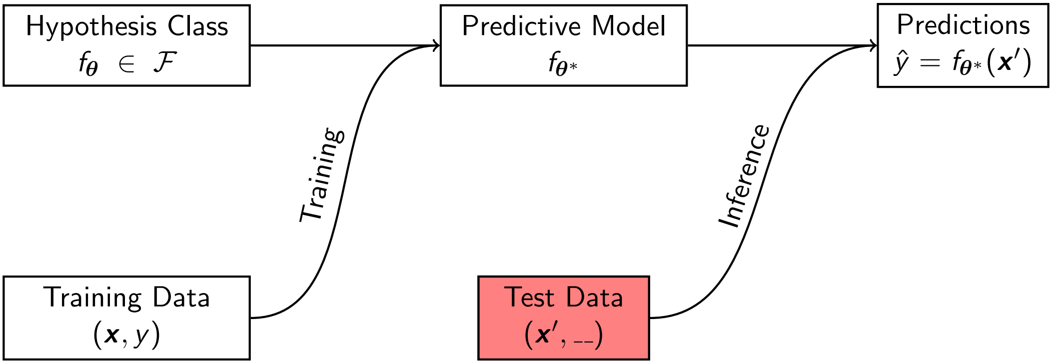
Warmup: (Supervised) Machine Learning Pipeline



Warmup: (Supervised) Machine Learning Pipeline



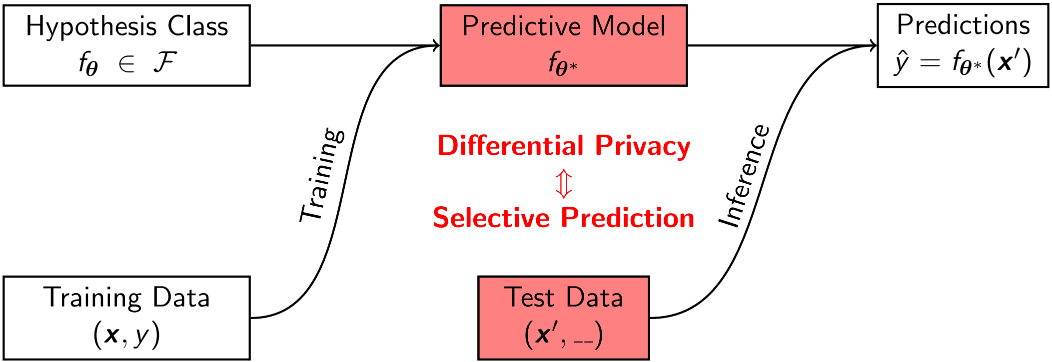
Warmup: (Supervised) Machine Learning Pipeline



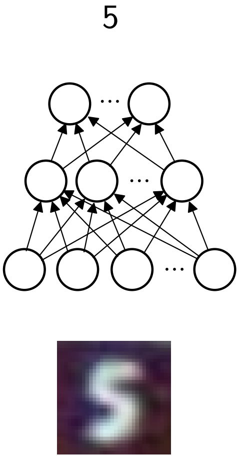
Adversarial Examples

Anomalous Inputs

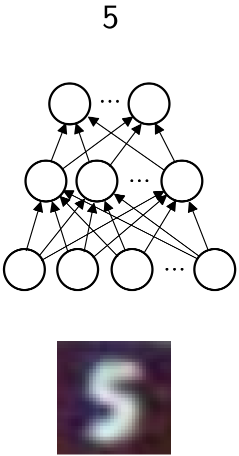
Warmup: (Supervised) Machine Learning Pipeline

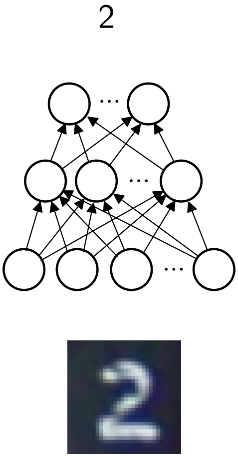


Motivation: Input Sample Rejection

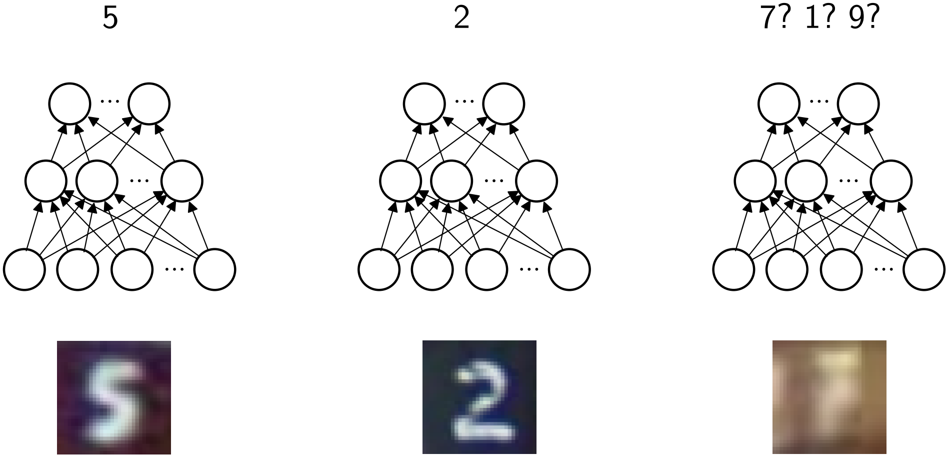


Motivation: Input Sample Rejection





Motivation: Input Sample Rejection



Selective Classification (SC)

Selective classification adds a rejection class $\perp$ via a gating mechanism.

Selective Classification (SC)

Selective classification adds a rejection class $\perp$ via a gating mechanism.

Goal: Derive a selection function $g : \mathcal{X} \to \mathbb{R}$ which, given an acceptance threshold τ, determines whether a model $f : \mathcal{X} \to \mathcal{Y}$ should predict on a data point $\boldsymbol{x}$.

$$(f, g)(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & g(\boldsymbol{x}) \leq \tau \\ \perp & \text{otherwise.} \end{cases}$$

Selective Classification (SC)

Selective classification adds a rejection class $\perp$ via a gating mechanism.

Goal: Derive a selection function $g : \mathcal{X} \rightarrow \mathbb{R}$ which, given an acceptance threshold τ, determines whether a model $f : \mathcal{X} \rightarrow \mathcal{Y}$ should predict on a data point $\boldsymbol{x}$.

$$(f, g)(\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & g(\boldsymbol{x}) \leq \tau \\ \perp & \text{otherwise.} \end{cases}$$

The performance of a selective classifier (f, g) on a dataset D is assessed based on

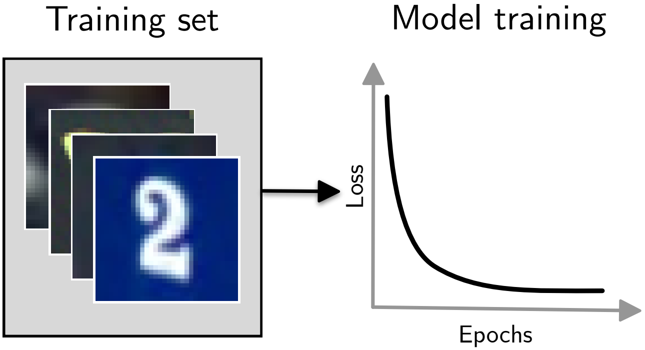
- the *coverage* of (f, g), i.e. what fraction of points we predict on; and
- the selective *accuracy* of (f, g) on the points it accepts.

$$\text{cov}_\tau(f, g) = \frac{|\{\boldsymbol{x} : g(\boldsymbol{x}) \leq \tau\}|}{|D|} \qquad \text{acc}_\tau(f, g) = \frac{|\{\boldsymbol{x} : f(\boldsymbol{x}) = y, g(\boldsymbol{x}) \leq \tau\}|}{|\{\boldsymbol{x} : g(\boldsymbol{x}) \leq \tau\}|}$$

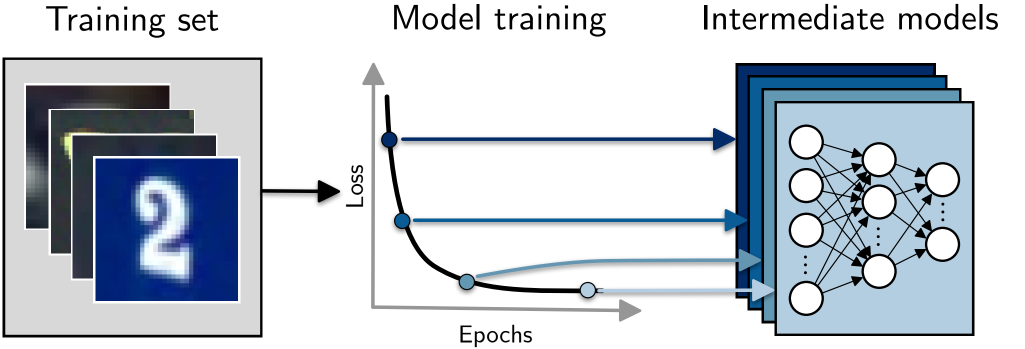
Training stage

Training set

Training stage

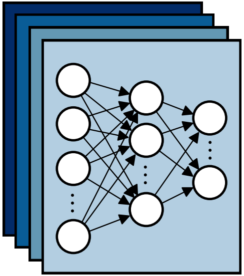


Training stage



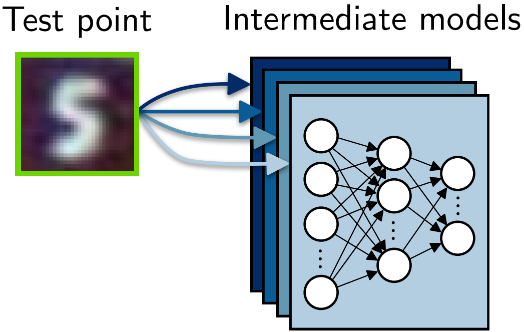
Testing stage

Intermediate models



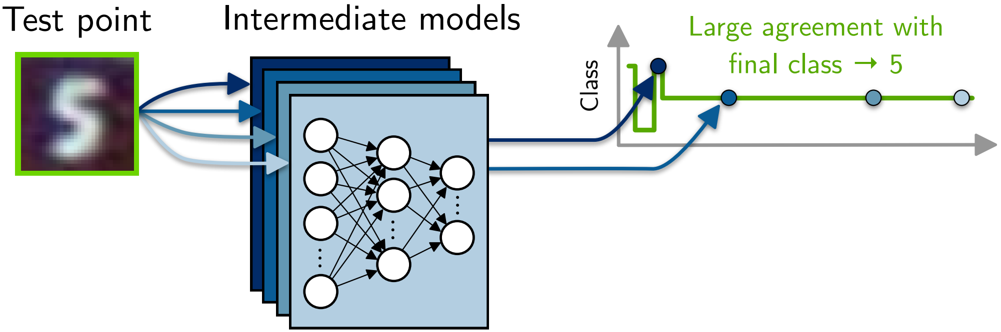
SC via Neural Net Training Dynamics: Method Overview (cont'd)

Testing stage



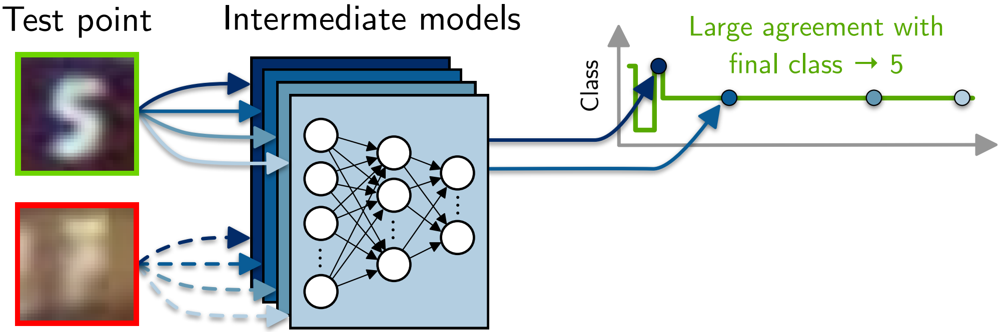
SC via Neural Net Training Dynamics: Method Overview (cont'd)

Testing stage



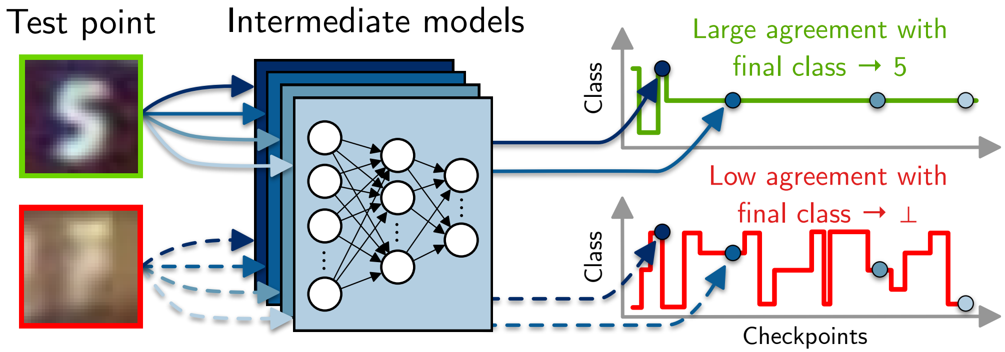
SC via Neural Net Training Dynamics: Method Overview (cont'd)

Testing stage



SC via Neural Net Training Dynamics: Method Overview (cont'd)

Testing stage



Differential Privacy

Definition: Differential Privacy

A randomized algorithm $\mathcal{M}$ satisfies (ε, δ) differential privacy, if for any two datasets $D, D' \subseteq \mathcal{D}$ that differ in any one record and any set of outputs S the following inequality holds:

$$\mathbb{P}\left[\mathcal{M}(D) \in S\right] \leq e^{\varepsilon}\mathbb{P}\left[\mathcal{M}(D') \in S\right] + \delta$$

The above DP bound is governed by two parameters:

- $\varepsilon \in \mathbb{R}_{+}$ which specifies the privacy level; and
- $\delta \in [0, 1]$ which allows for a small violation of the bound.

The most widely used implementation for ensuring DP in deep neural nets is DP-SGD.

Impacts of SC on DP Guarantees

Post-Processing

If a function $\phi(x)$ satisfies (ε, δ)-DP, then for any deterministic or randomized function $\psi(\cdot)$,$\psi \circ \phi(x)$ continues to satisfy (ε, δ)-DP.

Applicable to: Softmax Response (`SR`), Monte-Carlo Dropout (`MCDO`), Deep Gamblers (`DG`), Self-Adaptive Training (`SAT`), Selective Classification Training Dynamics (`SCTD`)

Impacts of SC on DP Guarantees

Post-Processing

If a function $\phi(x)$ satisfies (ε, δ)-DP, then for any deterministic or randomized function $\psi(\cdot)$,$\psi \circ \phi(x)$ continues to satisfy (ε, δ)-DP.

Applicable to: Softmax Response (SR), Monte-Carlo Dropout (MCDO), Deep Gamblers (DG), Self-Adaptive Training (SAT), Selective Classification Training Dynamics (SCTD)

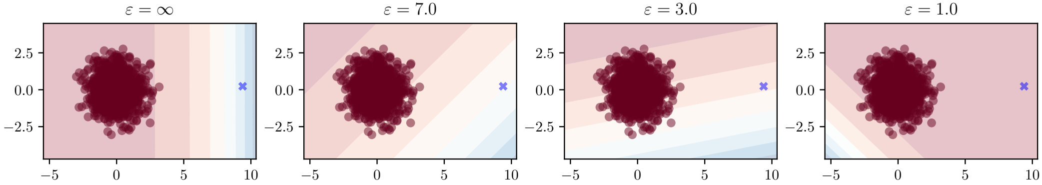
Advanced Sequential Composition

If for a set $\{\phi_1(x), \dots, \phi_M(x)\}$ each $\phi_i(x)$ satisfies (ε, δ)-DP, then releasing $\psi(x) = (\phi_1(x), \dots, \phi_M(x))$ satisfies $\approx (\sqrt{M}\varepsilon, M\delta)$-DP.
If the original (ε, δ)-DP constraint should be maintained, each function needs to satisfy $\approx (\frac{\varepsilon}{\sqrt{M}}, \frac{\delta}{M})$-DP.

Applicable to: Deep Ensembles (DE), SelectiveNet (SN)

Impacts of DP on SC Performance

- We expect DP to impact SC beyond a loss in utility.
- Sample points from a majority class and an outlier point $\boldsymbol{x}^*$ from a minority class.
- Train multiple differentially private models with $\varepsilon \in \{\infty, 7, 3, 1\}$.
- Non-private model has best accuracy (and uncertainty) but is influenced by $\boldsymbol{x}^*$.
- All models with $\varepsilon \in \{7, 3, 1\}$ misclassify the outlier and the changing decision boundary increases wrongful overconfidence as ε decreases.



Evaluating SC under DP

- Default approach to quantify SC performance without accuracy bias is to align different SC approaches/models at the same accuracy and evaluate

$$s_{\text{AUC}}(f, g) = \int_0^1 \text{acc}_c(f, g) dc \qquad \text{acc}_c(f, g) = \text{acc}_\tau(f, g) \quad \text{for } \tau \text{ s.t. } \text{cov}_\tau(f, g) = c$$

- Accuracy-aligning can have unintended consequences on SC performance.
- Early-stopping is the de-facto way of ensuring accuracy-alignment.
- **But:** Training for less leads to expending less privacy budget.
- Early-stopping yields a DP model with greater privacy than the targeted ε.

How do we quantify performance across SC methods and ε-levels where accuracy-alignment is not possible?

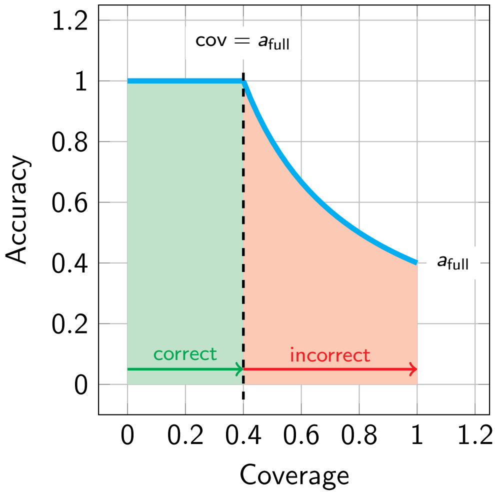
Upper Bound On Selective Classification Performance

Definition: Upper SC Perf. Bound

The upper bound for selective classification performance for a fixed full-coverage accuracy $a_{full} \in [0, 1]$ and a variable coverage level $c \in [0, 1]$ is given by

$$\overline{acc}(a_{full}, c) = \begin{cases} 1 & 0 < c \leq a_{full} \\ \frac{a_{full}}{c} & a_{full} < c < 1 \end{cases}$$

Optimal SC methods accept all correct points first and incorrect points afterwards.



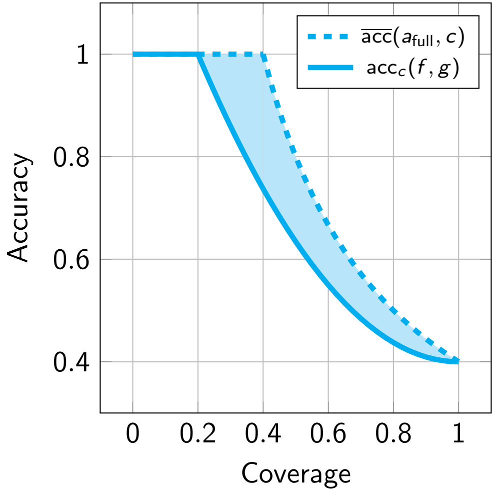
Accuracy-Normalized Score For Selective Classification

Definition: Acc-normalized SC Score

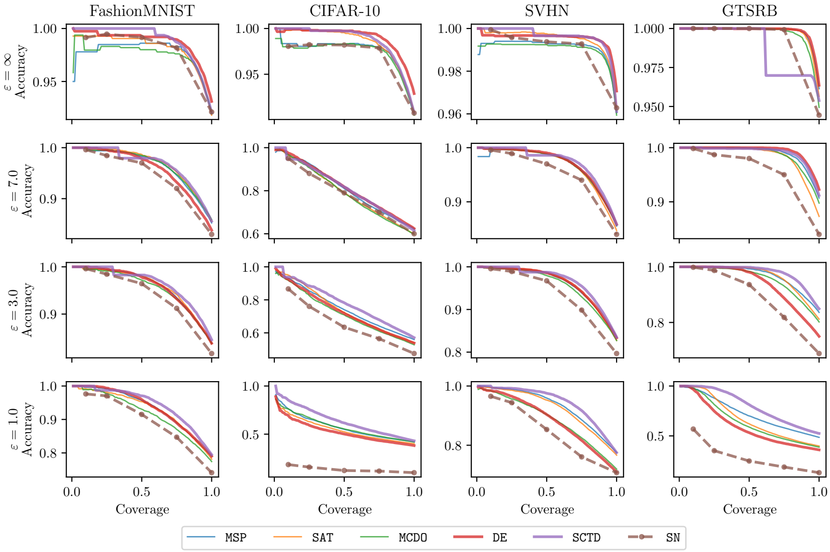
The accuracy-normalized selective classification score $s_{a_{full}}(f,g)$ for a selective classifier (f,g) with full-coverage accuracy a_{full} is given by

$$s_{a_{full}}(f,g) = \int_0^1 (\overline{acc}(a_{full}, c) - acc_c(f,g))dc$$
$$\approx \sum_c (\overline{acc}(a_{full}, c) - acc_c(f,g))$$

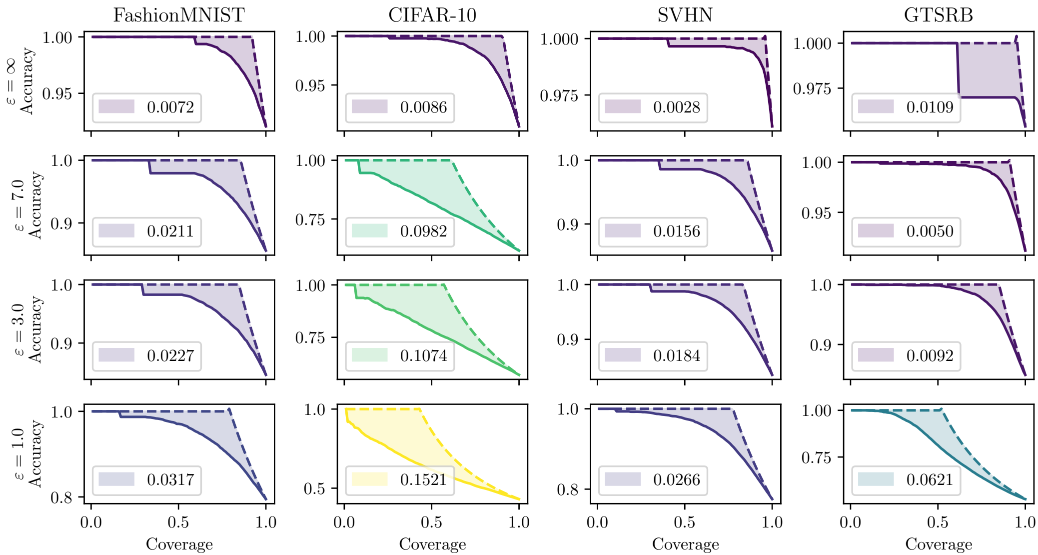
A good selective classifier should achieve a low score ($s_{a_{\text{full}}}(f,g) \approx 0$), indicating closeness to the optimal bound $\overline{\text{acc}}(a_{\text{full}}, c)$.



Accuracy-Coverage Tradeoff Across Datasets & ε Levels



Upper Bound Closeness for SCTD



Accuracy-Normalized Selective Classification Performance

	FashionMNIST				CIFAR-10			
	$\epsilon = \infty$	$\epsilon = 7$	$\epsilon = 3$	$\epsilon = 1$	$\epsilon = \infty$	$\epsilon = 7$	$\epsilon = 3$	$\epsilon = 1$
MSP	0.019 (±0.000)	0.023 (±0.000)	0.027 (±0.002)	0.041 (±0.001)	0.019 (±0.000)	0.105 (±0.002)	0.133 (±0.002)	0.205 (±0.001)
SAT	0.014 (±0.000)	**0.020 (±0.001)**	0.026 (±0.002)	0.043 (±0.002)	0.010 (±0.000)	0.107 (±0.000)	0.128 (±0.000)	0.214 (±0.002)
MCDO	0.020 (±0.002)	0.023 (±0.001)	0.030 (±0.003)	0.053 (±0.001)	0.021 (±0.001)	0.110 (±0.000)	0.142 (±0.000)	0.201 (±0.000)
DE	**0.010 (±0.003)**	0.027 (±0.002)	0.027 (±0.002)	0.039 (±0.000)	**0.007 (±0.001)**	**0.099 (±0.002)**	0.138 (±0.000)	0.222 (±0.000)
SN	**0.008 (±0.002)**	0.058 (±0.001)	0.056 (±0.001)	0.064 (±0.002)	0.015 (±0.000)	0.155 (±0.003)	0.154 (±0.002)	0.173 (±0.001)
SCTD	**0.007 (±0.001)**	**0.021 (±0.001)**	**0.023 (±0.003)**	**0.032 (±0.002)**	**0.009 (±0.002)**	**0.098 (±0.001)**	**0.107 (±0.001)**	**0.152 (±0.001)**
	SVHN				GTSRB			
MSP	0.008 (±0.001)	0.020 (±0.001)	0.024 (±0.001)	0.040 (±0.001)	**0.001 (±0.001)**	0.006 (±0.002)	0.017 (±0.000)	0.109 (±0.002)
SAT	**0.004 (±0.000)**	0.019 (±0.000)	**0.021 (±0.002)**	0.044 (±0.002)	**0.001 (±0.001)**	0.008 (±0.001)	0.014 (±0.000)	0.089 (±0.000)
MCDO	0.009 (±0.000)	0.019 (±0.001)	0.027 (±0.002)	0.069 (±0.001)	0.002 (±0.001)	0.007 (±0.001)	0.023 (±0.001)	0.110 (±0.001)
DE	**0.004 (±0.001)**	0.018 (±0.001)	**0.022 (±0.002)**	0.067 (±0.003)	**0.001 (±0.000)**	**0.003 (±0.002)**	0.027 (±0.004)	0.127 (±0.002)
SN	**0.004 (±0.000)**	0.055 (±0.001)	0.052 (±0.000)	0.096 (±0.000)	**0.001 (±0.001)**	0.050 (±0.004)	0.044 (±0.001)	0.091 (±0.004)
SCTD	**0.003 (±0.001)**	**0.016 (±0.001)**	**0.018 (±0.002)**	**0.027 (±0.001)**	0.011 (±0.002)	**0.005 (±0.000)**	**0.009 (±0.000)**	**0.062 (±0.001)**

SCTD **has the strongest performance, i.e. the lowest bound distance.**

Coverage Required For Non-Private Full-Coverage Accuracy

	FashionMNIST			CIFAR-10		
	$\varepsilon = 7$	$\varepsilon = 3$	$\varepsilon = 1$	$\varepsilon = 7$	$\varepsilon = 3$	$\varepsilon = 1$
MSP	0.83 (±0.01)	0.80 (±0.01)	0.65 (±0.03)	**0.29 (±0.02)**	0.14 (±0.04)	0.00 (±0.00)
SAT	**0.86 (±0.00)**	0.81 (±0.01)	0.67 (±0.02)	0.25 (±0.01)	**0.19 (±0.02)**	0.00 (±0.00)
MCDO	**0.84 (±0.02)**	0.79 (±0.00)	0.56 (±0.02)	0.25 (±0.01)	0.12 (±0.02)	0.00 (±0.00)
DE	0.75 (±0.00)	0.75 (±0.01)	0.61 (±0.01)	0.22 (±0.01)	0.09 (±0.00)	0.00 (±0.00)
SCTD	**0.86 (±0.01)**	**0.84 (±0.02)**	**0.73 (±0.01)**	**0.26 (±0.03)**	**0.20 (±0.03)**	**0.04 (±0.04)**
	SVHN			GTSRB		
MSP	0.74 (±0.00)	0.67 (±0.01)	0.49 (±0.02)	0.90 (±0.01)	0.71 (±0.03)	0.13 (±0.00)
SAT	0.72 (±0.00)	0.67 (±0.01)	0.45 (±0.02)	0.86 (±0.00)	0.74 (±0.00)	0.20 (±0.03)
MCDO	0.74 (±0.00)	0.64 (±0.00)	0.23 (±0.03)	0.90 (±0.01)	0.69 (±0.01)	0.14 (±0.01)
DE	0.69 (±0.01)	0.62 (±0.01)	0.22 (±0.00)	**0.93 (±0.00)**	0.57 (±0.08)	0.10 (±0.04)
SCTD	**0.78 (±0.01)**	**0.72 (±0.00)**	**0.59 (±0.02)**	**0.93 (±0.01)**	**0.83 (±0.03)**	**0.30 (±0.02)**

SCTD **retains the largest amount of coverage.**

Conclusion

- Analyzed how SC impacts DP guarantees and how DP impacts SC performance.
- Introduced a novel score to disentangle SC performance from baseline utility.
- SC performance degrades with stronger privacy (i.e. as $\varepsilon \to 0$).
- `SCTD` works best to quantify uncertainty under DP.

Stephan

Anvith

Abhradeep

Dj

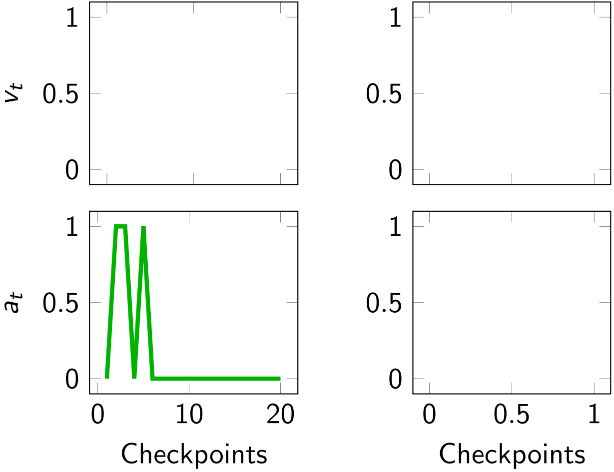
Nicolas

Training Private Models That Know What They Don't Know

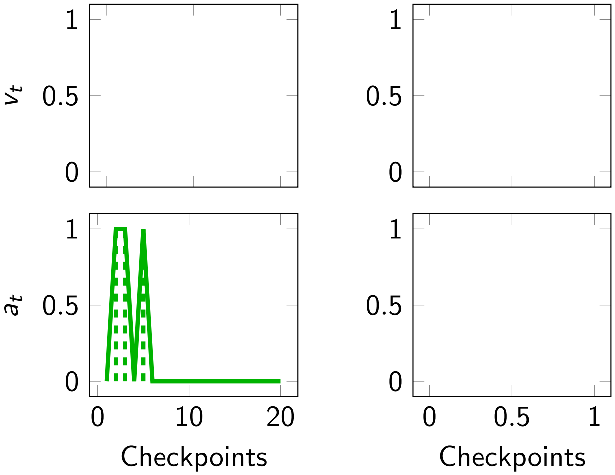
`https://arxiv.org/abs/2305.18393`

Backup

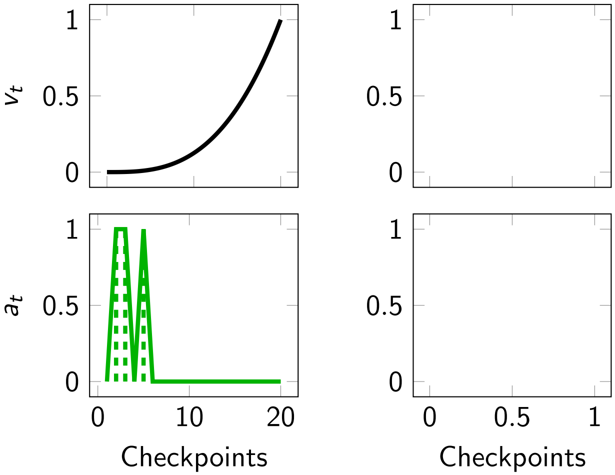
The Sum Score s_{sum}



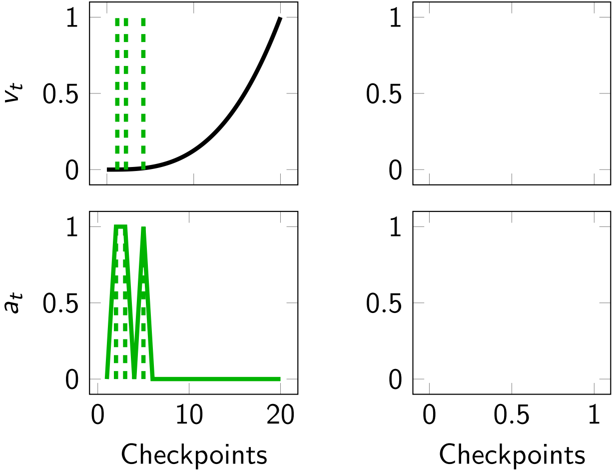
The Sum Score s_{sum}



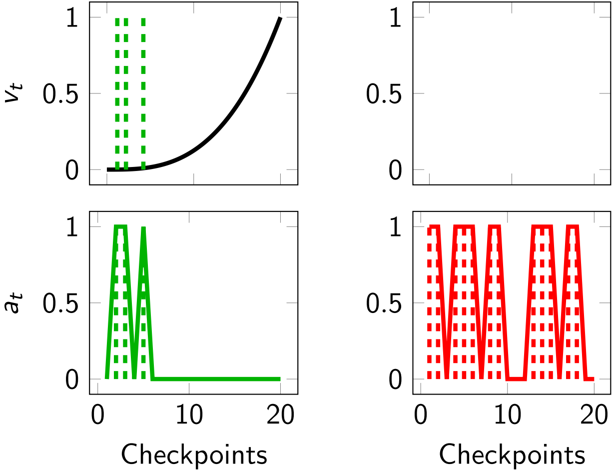
The Sum Score s_{sum}



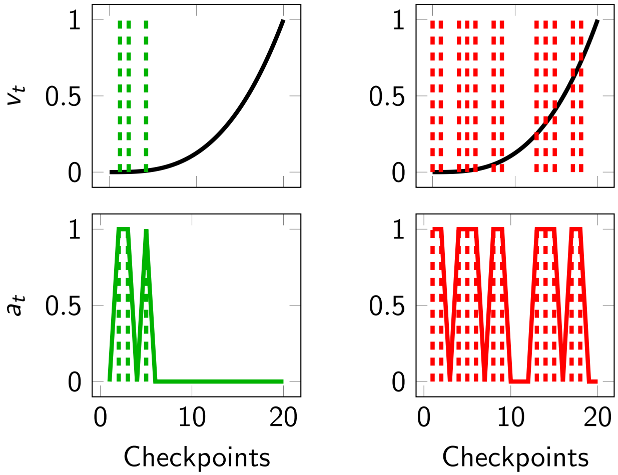
The Sum Score s_{sum}



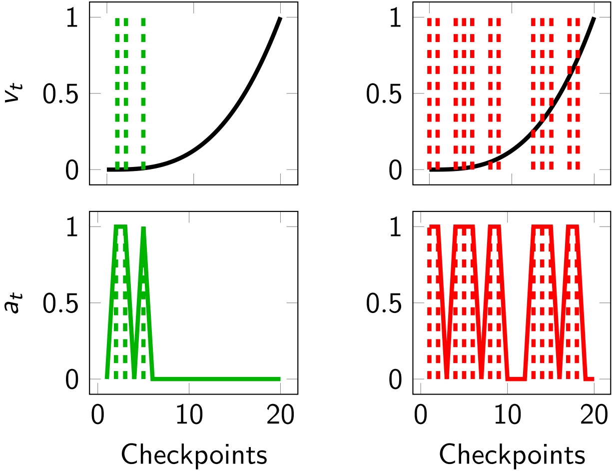
The Sum Score s_{sum}



The Sum Score s_{sum}



The Sum Score s_{sum}



1. Denote $L = f_T(\boldsymbol{x})$, i.e. the label our final model predicts.
2. If $\exists t\ s.t\ a_t = 1$, compute

$$s_{\text{sum}} = \sum v_t a_t$$

   else accept $\boldsymbol{x}$ with prediction L.
3. If $s_{\text{sum}} < \tau$ accept $\boldsymbol{x}$ with prediction L, else reject ($\perp$).

The SCTD Algorithm

Algorithm 0: SCTD

Require: Checkpointed model sequence $\{f_1, \ldots, f_T\}$, query point $\boldsymbol{x}$, weighting parameter $k \in [0, \infty)$.

Compute prediction of last model: $L \leftarrow f_T(\boldsymbol{x})$

Compute disagreement and weighting of intermediate predictions:

for $t \in [T]$ **do**

  if $f_t(\boldsymbol{x}) = L$ **then** $a_t \leftarrow 0$ **else** $a_t \leftarrow 1$

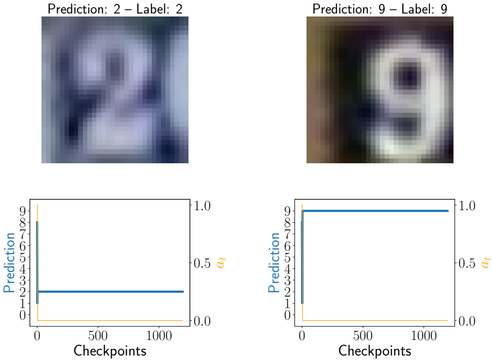
  $v_t \leftarrow 1 - (\frac{t}{T})^k$

end for

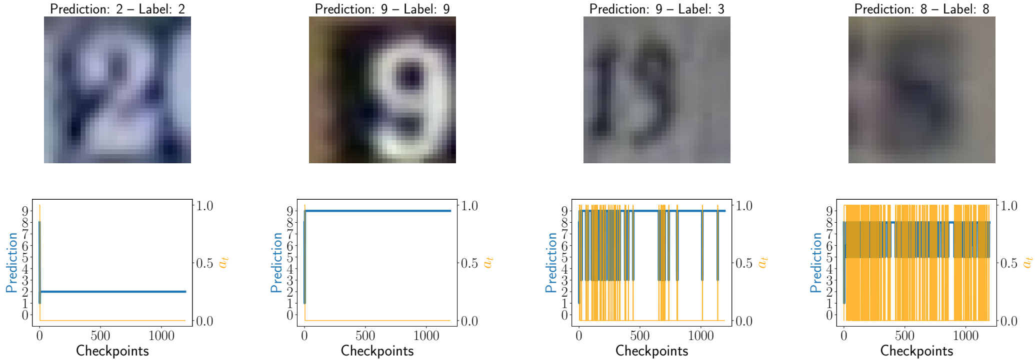
Compute sum score: $s_{\text{sum}} \leftarrow \sum_t a_t v_t$

if $s_{\text{sum}} \leq \tau$ **then** accept $f(\boldsymbol{x}) = L$ **else** reject with $f(\boldsymbol{x}) = \perp$

Individual SVHN Example



Individual SVHN Example



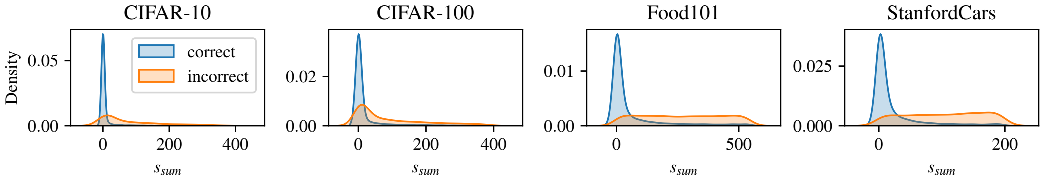
UNIVERSITY OF TORONTO VECTOR INSTITUTE UNIVERSITY OF CAMBRIDGE

SC Performance Over Accuracy-Coverage Curve

Dataset	SR	SAT	DE	SCTD	DE+SCTD
CIFAR-10	0.971	0.978	0.980	0.980	**0.981**
CIFAR-100	0.895	0.900	0.909	0.909	**0.912**
Food101	0.935	0.939	0.945	0.946	**0.947**
StanfordCars	0.920	0.927	0.930	0.931	**0.934**

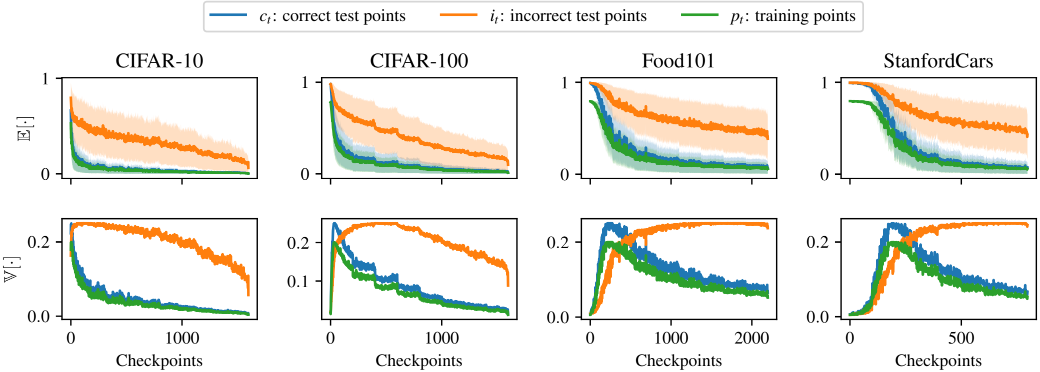
- SCTD offers comparable performance to DE.
- Combining DE with SCTD (DE+SCTD) delivers new SOTA performance.

Score Distributions Of Correct And Incorrect Points



- Correct predictions concentrate at 0 (prediction stability).
- Incorrect predictions spread over a wide score range (prediction instability).

Monitoring $\mathbb{E}[\cdot]$ and $\mathbb{V}[\cdot]$



- Patterns for optimized points overlap with correctly classified test points.
- Correctly classified points have both $\mathbb{E}[c_t]$ and $\mathbb{V}[c_t]$ quickly decreasing to 0.
- Incorrectly classified points exhibit large expectations and variances.

Performance of s_{max} vs s_{sum}

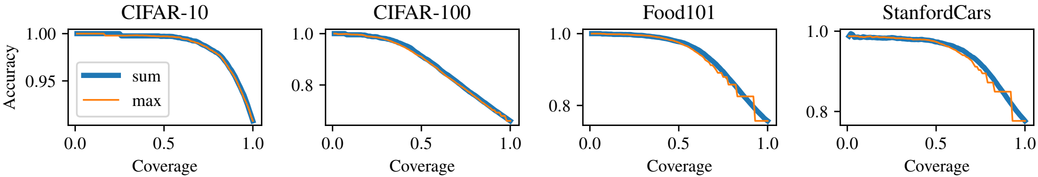


Figure: **Comparing s_{max} and s_{sum} performance**. It is evident that s_{sum} effectively denoises s_{max}.

Ablation On Number of Checkpoints

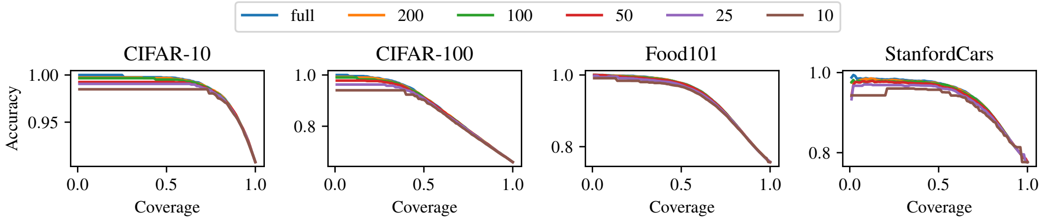


Figure: **Coverage/error trade-off of** $\texttt{SCTD}(s_{\textbf{avg}})$ **for varying total number of checkpoints**. As the checkpointing resolution decreases, accuracy at low coverage increasingly degrades, thereby showing that a detailed characterization of the training dynamics is helpful to attain high target accuracy.

Ablation Over k

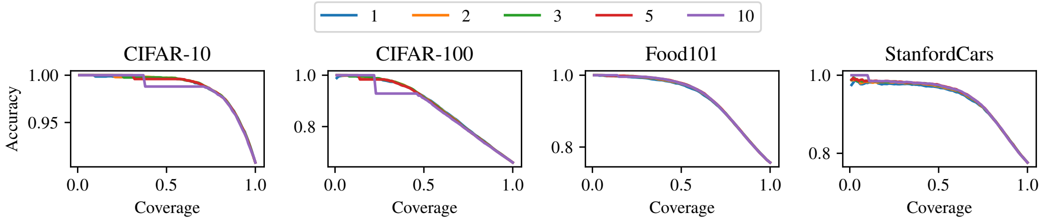


Figure: **Coverage/error trade-off of $\texttt{SCTD}(s_{\textbf{avg}}, k)$ for varying checkpoint weighting k as used in v_t.** We observe strong performance for $k \in [2, 5]$ across datasets.

Incorporating e_t And v_t Into s_{sum}

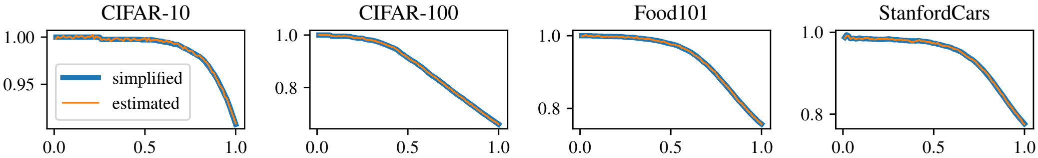


Figure: **Coverage/error trade-off when incorporating e_t and v_t into s_{max} and s_{sum}.** We see that our simplifying assumptions match the performance attained from empirical estimation of e_t and v_t.

Detectability of OOD and Adversarial Examples

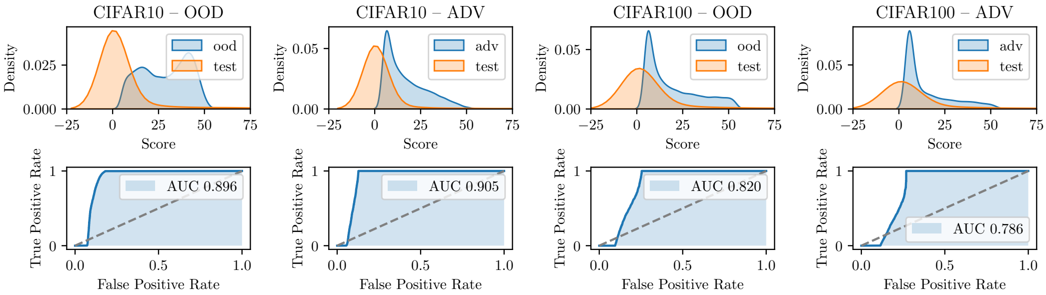


Figure: **Performance of** $\texttt{SCTD}(s_{\textbf{sum}})$ **on out-of-distribution (OOD) and adversarial sample detection on CIFAR-10 and CIFAR-100.** The first row shows the score distribution of the in-distribution test set vs the `SVHN` OOD test set or a set consisting of adversarial samples generated via a PGD attack in the final model. The second row shows the effectiveness of a thresholding mechanism by computing the area under the ROC curve.

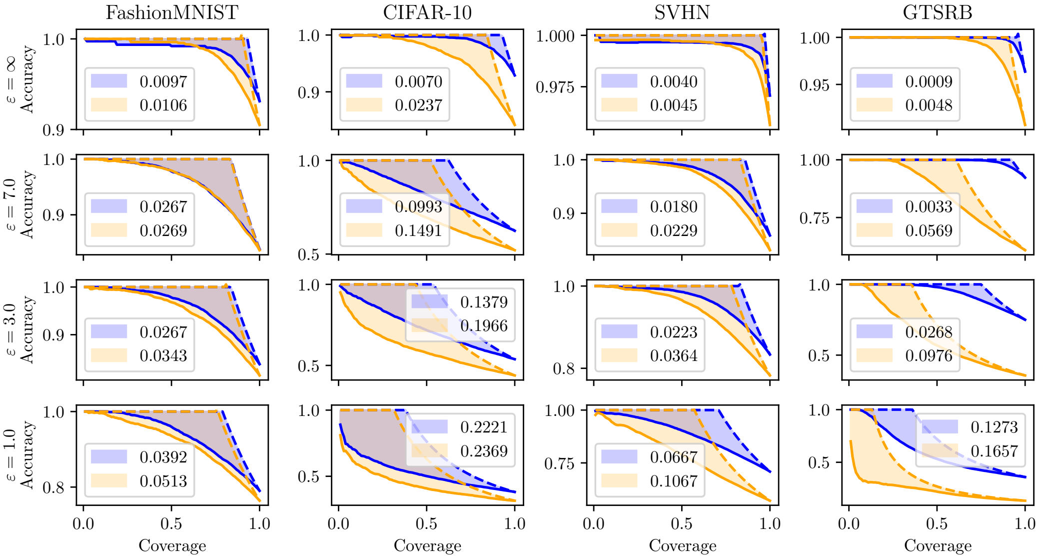
DP-SGD

Algorithm 1: DP-SGD

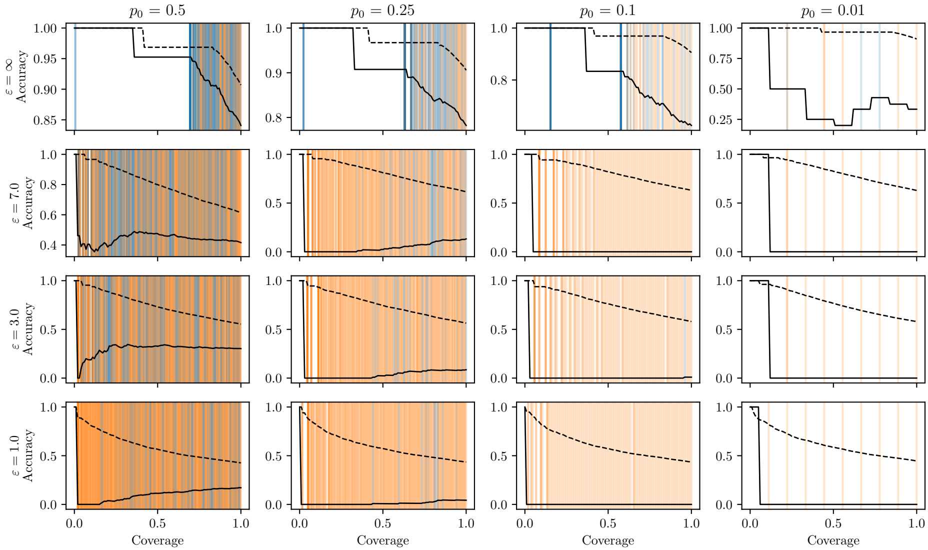
Require: Training dataset D, loss function ℓ, learning rate η, noise multiplier σ, sampling rate q, clipping norm c, iterations T.

Initialize θ_0
for $t \in [T]$ **do**
  1. Per-Sample Gradient Computation
  Sample B_t with per-point prob. q from D
  for $i \in B_t$ **do**
    $g_t(\boldsymbol{x}_i) \leftarrow \nabla_{\theta_t} \ell(\theta_t, \boldsymbol{x}_i)$
  end for
  2. Gradient Clipping
  $\bar{g}_t(\boldsymbol{x}_i) \leftarrow g_t(\boldsymbol{x}_i) / \max\left(1, \frac{\|g_t(\boldsymbol{x}_i)\|_2}{c}\right)$
  3. Noise Addition
  $\tilde{g}_t \leftarrow \frac{1}{|B_t|}\left(\sum_i \bar{g}_t(\boldsymbol{x}_i) + \mathcal{N}(0, (\sigma c)^2 \mathbf{I})\right)$
  $\theta_{t+1} \leftarrow \theta_t - \eta \tilde{g}_t$
end for
Output θ_T, privacy cost (ε, δ) computed via a privacy accounting procedure

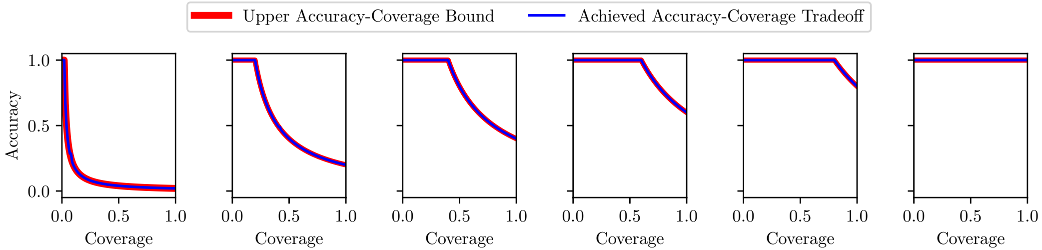
Composition-Based Deep Ensembles VS Partitioned Deep Ensembles



Class Imbalance Results on CIFAR-10



Upper Bound Reachability



- Assume a binary classif. setting with label vector $\boldsymbol{y} \in \{0,1\}^{n_0+n_1}$ and $n_0 = n_1$.
- Generate a prediction vector $\boldsymbol{p}$ which overlaps with $\boldsymbol{y}$ for a fraction of a_{full}.
- Sample a scoring vector $\boldsymbol{s}$ where each correct prediction is assigned a score $s_i \sim \mathcal{U}_{0,0.5}$ and each incorrect prediction is assigned a score $s_i \sim \mathcal{U}_{0.5,1}$.
- This score is optimal since all $s_i < 0.5$ correspond to a correct prediction, while all $s_i \geq 0.5$ correspond to an incorrect prediction.