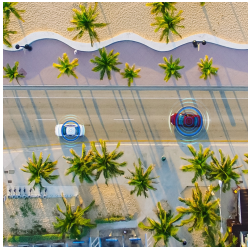
Selective Classification Via Neural Network Training Dynamics

Stephan Rabanser stephan@cs.toronto.edu
Anvith Thudi anvith.thudi@mail.utoronto.ca
Kimia Hamidieh kimia@cs.toronto.edu
Adam Dziedzic ady@vectorinstitute.ai
Nicolas Papernot nicolas.papernot@utoronto.ca

June 7, 2022

Motivation

Machine Learning systems are becoming ubiquitous.




Especially in high-stakes decision-making, it is of vital importance to detect inputs that a machine learning model would predict incorrectly on.

Image credit: `unsplash.com`

Standard Supervised Learning Setup

- Dataset $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ consisting of data points $(\boldsymbol{x}, y)$ with $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$.
  - Covariate space $\mathcal{X} = \mathbb{R}^d$ of dimensionality d.
  - Label space $\mathcal{Y} = \{1, 2, \ldots, C\}$ consisting of C classes.
- $(\boldsymbol{x}, y)$ are sampled iid from the underlying distribution p defined over $\mathcal{X} \times \mathcal{Y}$.
- Goal: learn a prediction function $f : \mathcal{X} \to Y$ which minimizes the classification risk $R(\cdot)$ wrt the underlying data distribution p and an appropriately chosen loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.
- We can derive the optimal parameters $\hat{\boldsymbol{\theta}}$ via empirical risk minimization:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} R(f_{\boldsymbol{\theta}}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)] \approx \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{N} \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i)$$

Selective Classification (SC)

Selective classification introduces a rejection class $\perp$ via a *gating mechanism*. Our goal is to derive a selection function $g : \mathcal{X} \to \mathbb{R}$ which, given an acceptance threshold τ, determines whether a model should predict on a data point $\boldsymbol{x}$.

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

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

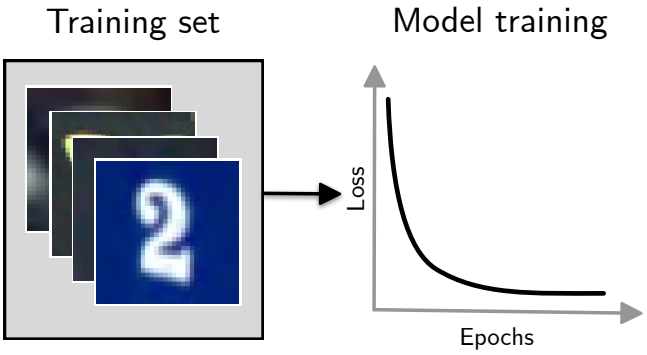
- the *coverage* of (f, g), i.e. what fraction of points we predict on
- the selective *accuracy* of (f, g) on the points it accepts.

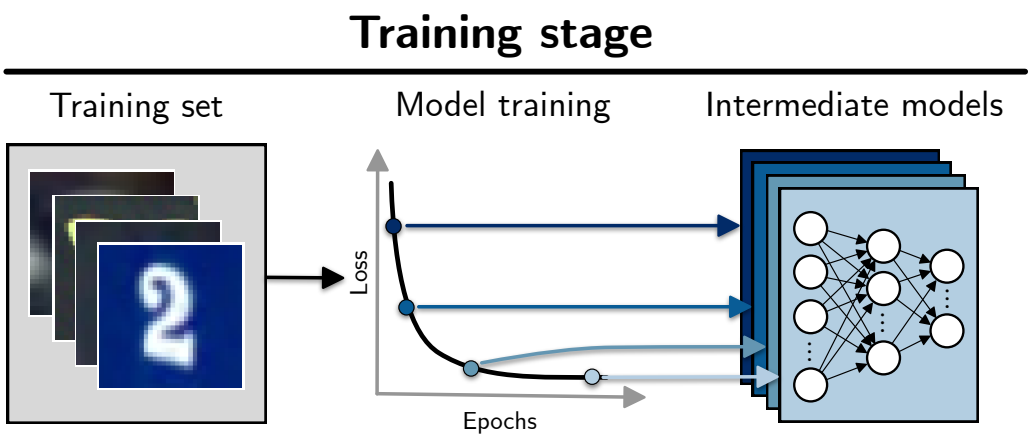
$$\text{cov}(f, g) = \frac{|\{\boldsymbol{x} : g(\boldsymbol{x}) \geq \tau\}|}{|D|} \qquad \text{acc}(f, g) = \frac{|\{\boldsymbol{x} : f(\boldsymbol{x}) = y, g(\boldsymbol{x}) \geq \tau\}|}{|\{\boldsymbol{x} : g(\boldsymbol{x}) \geq \tau\}|}$$

Training stage

Training set

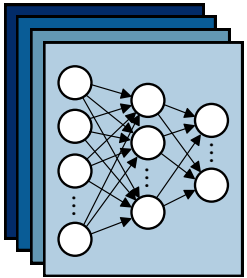
Training stage




Training stage
Training set
Model training
Intermediate models
Loss
Epochs

Testing stage

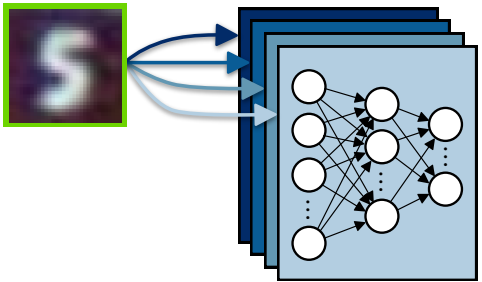
Intermediate models

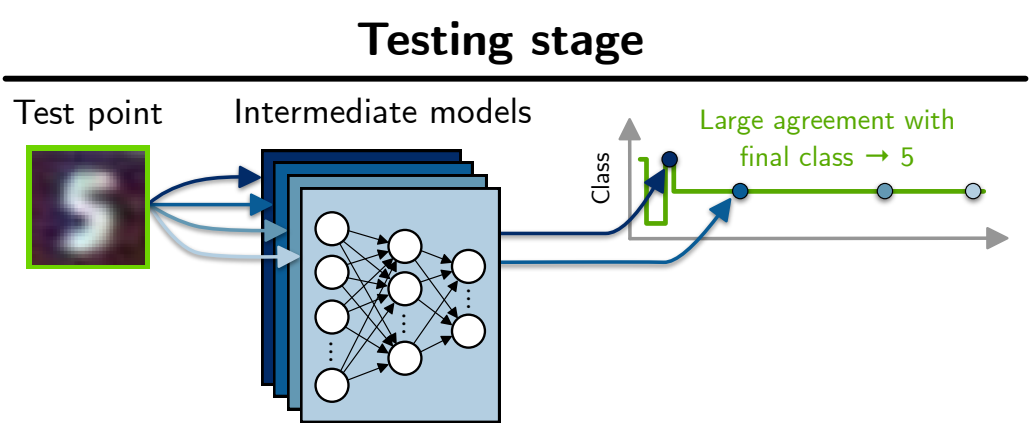


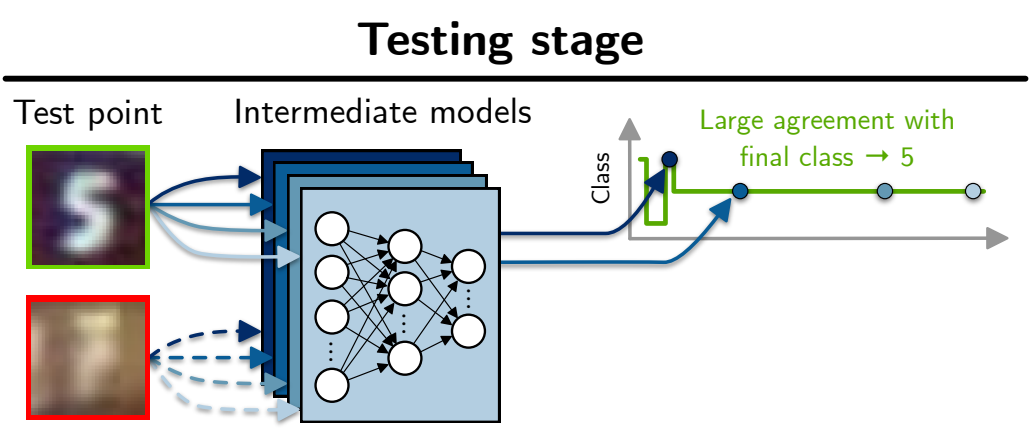
Testing stage

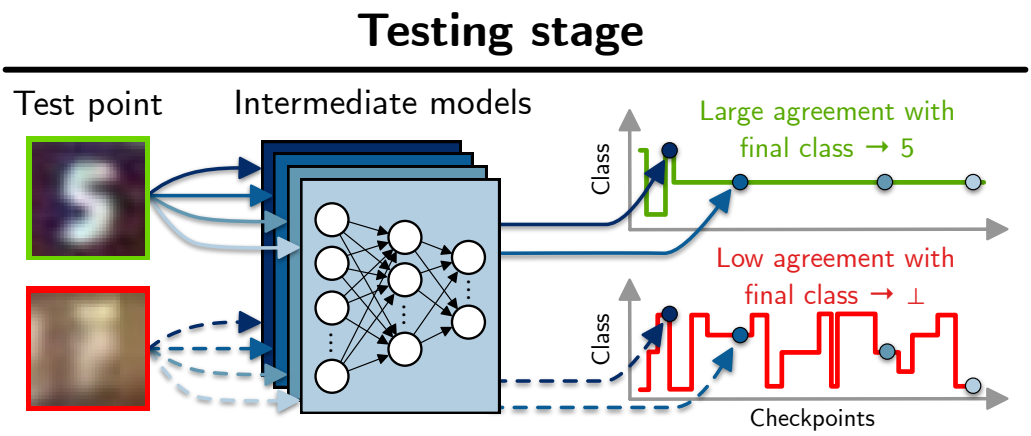
Test point

Intermediate models



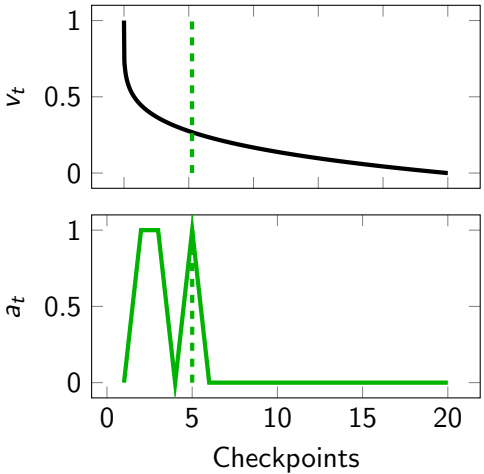

Testing stage
Test point
Intermediate models
Class
Large agreement with
final class → 5


Testing stage
Test point
Intermediate models
Class
Large agreement with
final class → 5


Testing stage
Test point
Intermediate models
Class
Large agreement with final class → 5
Low agreement with final class → ⊥
Class
Checkpoints

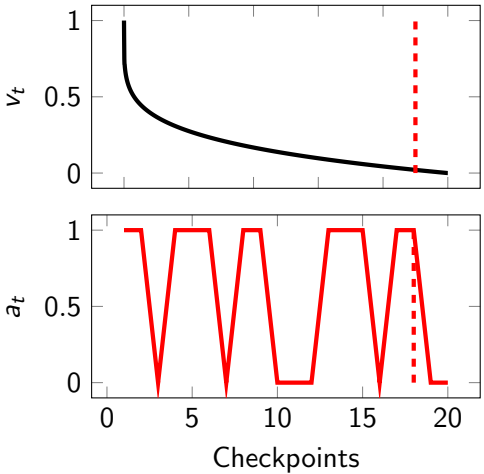
The Minimum Score s_{min}

1. Denote $L = f_T(\boldsymbol{x})$, i.e. the label our final model predicts.
2. If $\exists t \; s.t \; a_t = 1$ then compute $s_{\text{min}} = \min_{t \; s.t \; a_t = 1} v_t$, else accept $\boldsymbol{x}$ with prediction L.
3. If $s_{\text{min}} > \tau$ accept $\boldsymbol{x}$ with prediction L, else reject ($\perp$).



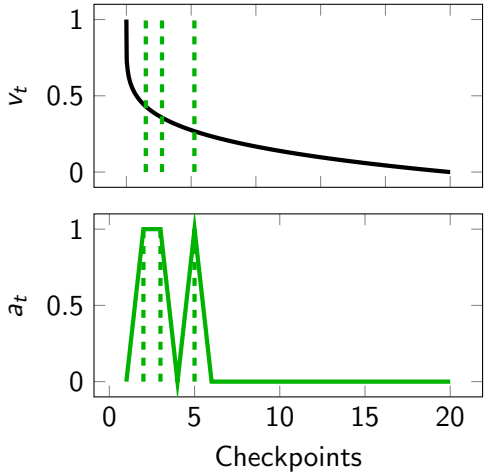
The Minimum Score s_{min}

1. Denote $L = f_T(\boldsymbol{x})$, i.e. the label our final model predicts.
2. If $\exists t \; s.t \; a_t = 1$ then compute $s_{\text{min}} = \min_{t \; s.t \; a_t = 1} v_t$, else accept $\boldsymbol{x}$ with prediction L.
3. If $s_{\text{min}} > \tau$ accept $\boldsymbol{x}$ with prediction L, else reject ($\perp$).



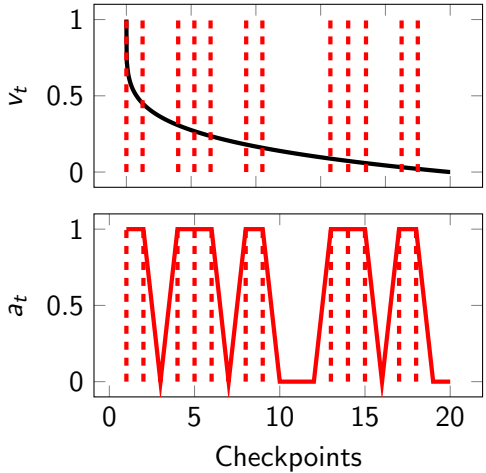
The Average Score s_{avg}

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 $s_{\text{avg}} = \frac{\sum v_t a_t}{\sum a_t}$, else accept $\boldsymbol{x}$ with prediction L.
3. If $s_{\text{avg}} > \tau$ accept $\boldsymbol{x}$ with prediction L, else reject ($\perp$).



The Average Score s_{avg}

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 $s_{\text{avg}} = \frac{\sum v_t a_t}{\sum a_t}$, else accept $\boldsymbol{x}$ with prediction L.
3. If $s_{\text{avg}} > \tau$ accept $\boldsymbol{x}$ with prediction L, else reject ($\perp$).



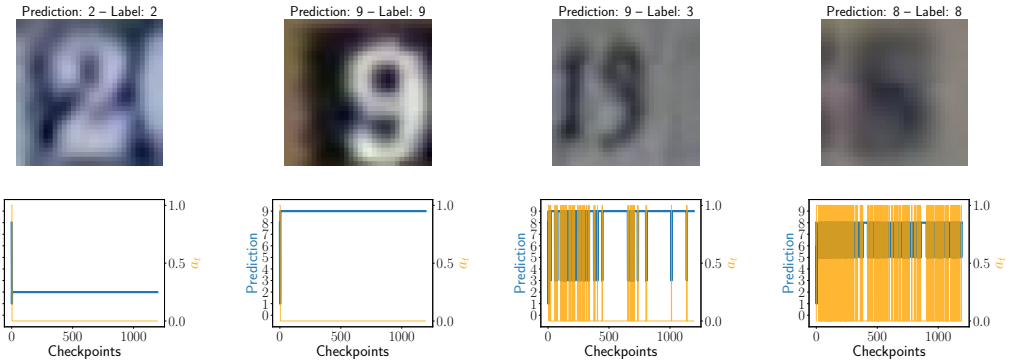
Performance at Low Target Error

Dataset	Target Error	NNTD		OSP		SR		SN		DG	
		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

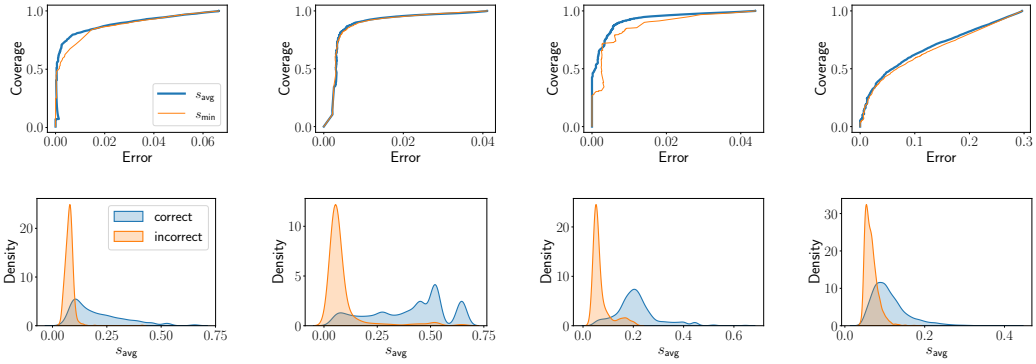
Performance at High Target Coverage

Dataset	Target Coverage	NNTD Cov.	NNTD Error	OSP Cov.	OSP Error	SR Cov.	SR Error	SN Cov.	SN Error	DG Cov.	DG 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

Individual SVHN Example

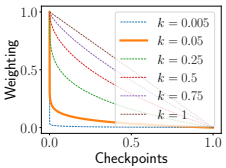


Performance of s_{min} and s_{avg}

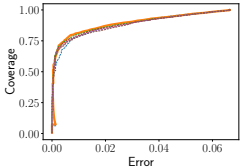


(a) CIFAR-10 (b) SVHN (c) GTSRB (d) CIFAR-100

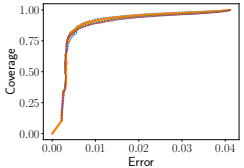
Hyperparamters: Weighting Parameter k Checkpointing Resolution T



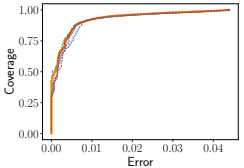
(a) Convex weighting



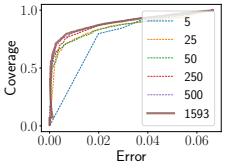
(b) CIFAR-10



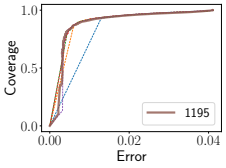
(c) SVHN



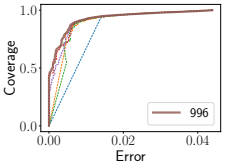
(d) GTSRB



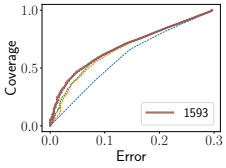
(a) CIFAR-10



(b) SVHN



(c) GTSRB

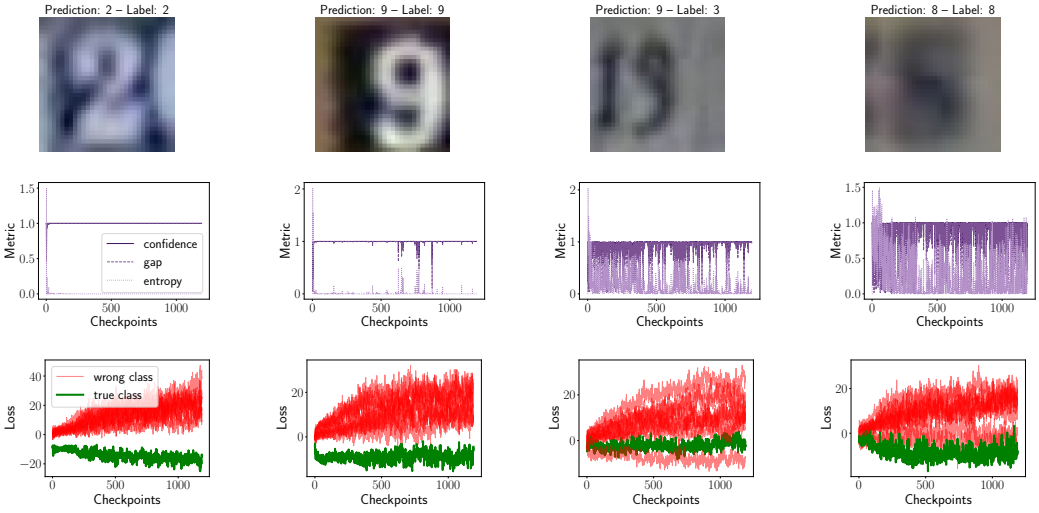


(d) CIFAR-100

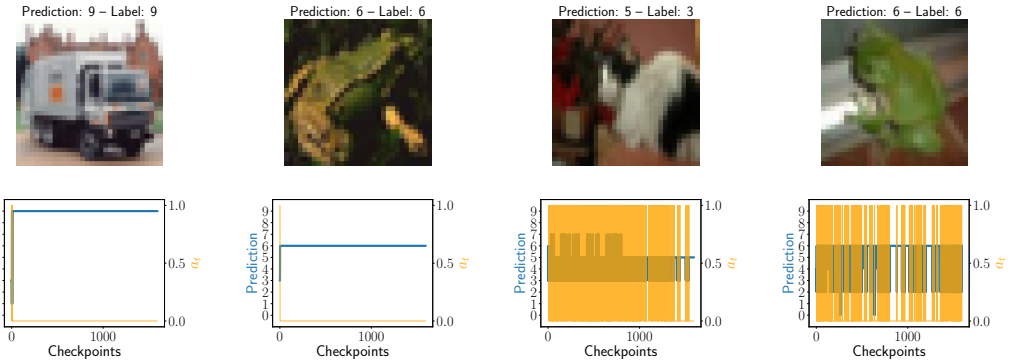
Thanks! :)

Backup

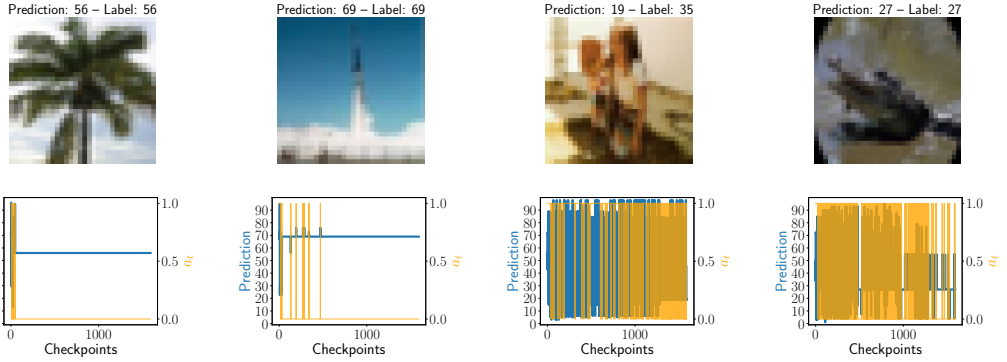
Extended SVHN Example



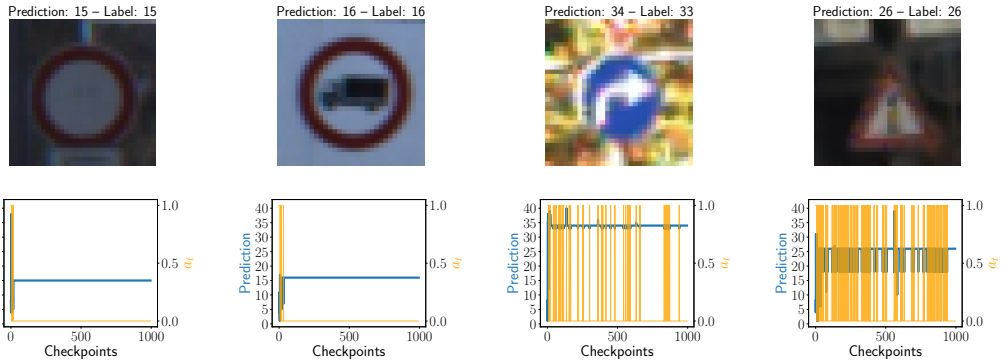
CIFAR-10 Example



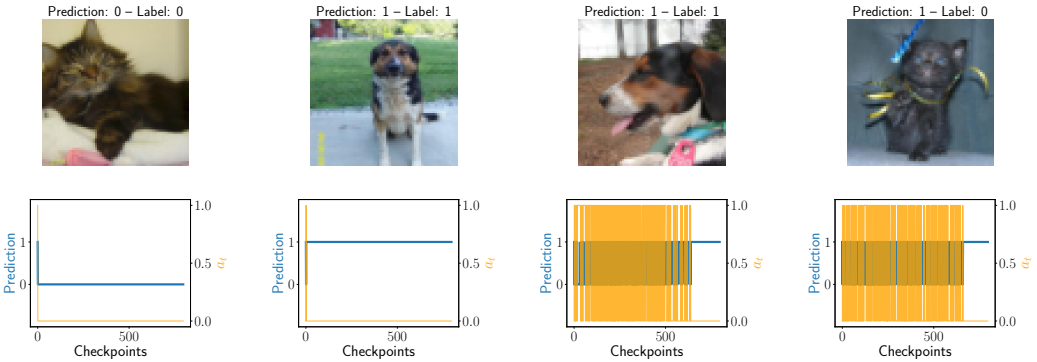
CIFAR-100 Example



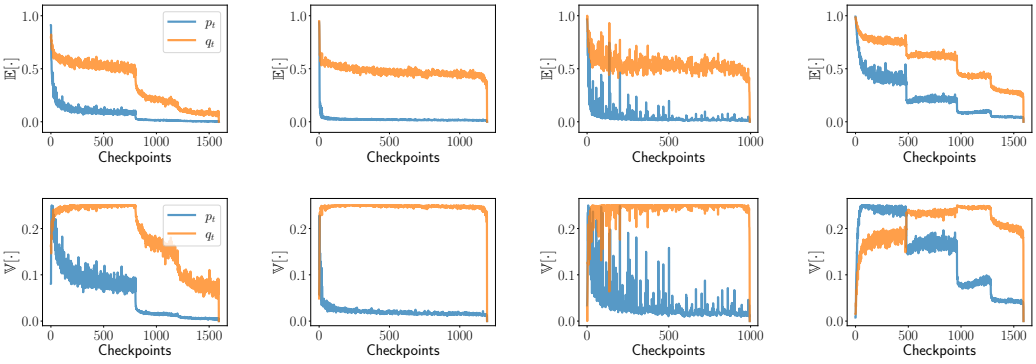
GTSRB Example



Cats & Dogs Example

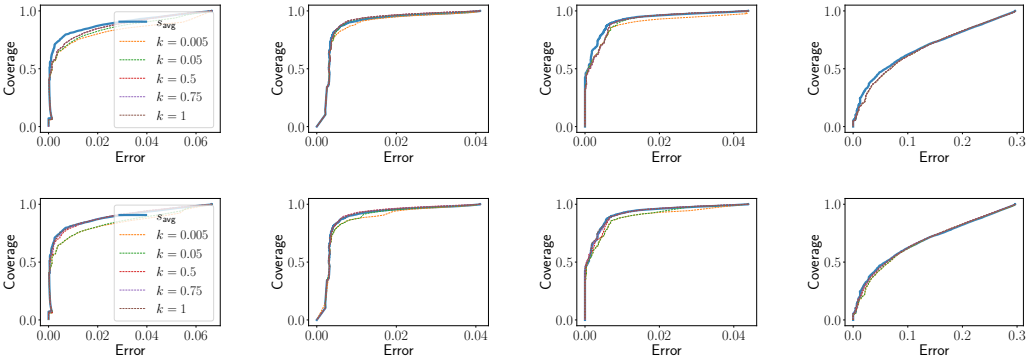


Examining the Convergence Behavior of Correct & Incorrect Points



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

Incorporating Empirical Estimates of e_t and v_t



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

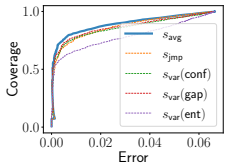
Alternate Metrics

Jump Score

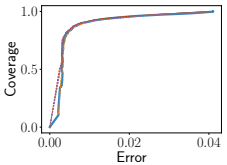
Measure level of disagreement between the predicted label of two successive intermediate models: for $j_t = 0$ iff $f_t(\boldsymbol{x}) = f_{t-1}(\boldsymbol{x})$ and $j_t = 1$ otherwise we can compute the jump score as $s_{\text{jmp}} = 1 - \sum v_t j_t$.

Continuous Metrics (e.g. MSP, Entropy)

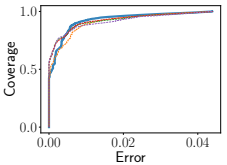
Assume that any of these metrics is given by a sequence $z = \{z_1, \ldots, z_T\}$. Then we can capture the uniformity of z via a variance score $s_{\text{var}} = \sum_t w_t (z_t - \mu)^2$ for mean $\mu = \frac{1}{T} \sum_t z_t$ and an increasing weighting sequence $w = \{w_1, \ldots, w_T\}$.



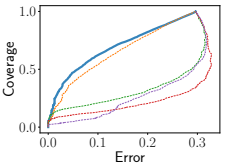
(a) CIFAR-10



(b) SVHN

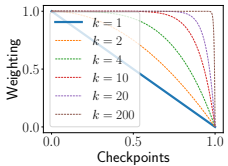


(c) GTSRB

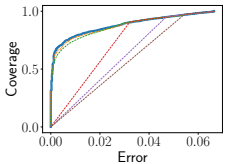


(d) CIFAR-100

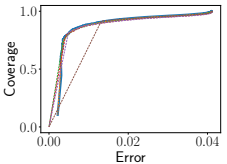
Concave Weighting



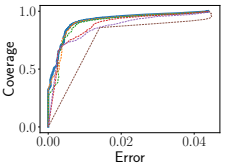
(a) Concave weighting



(b) CIFAR-10



(c) SVHN



(d) GTSRB