Critic Loss for Image Classification

Brendan Hogan Rappazzo, Aaron Ferber, Carla Gomes
Department of Computer Science
Cornell University
Ithaca, New York 14850
Email: {bhr54, amf272, cpg5}@cornell.edu

2024 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-7488-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMLA61862.2024.00104

***Abstract*—Modern neural network classifiers achieve remarkable performance across a variety of tasks; however, they frequently exhibit overconfidence in their predictions due to the cross-entropy loss. Inspired by this problem, we propose the Critic Loss for Image Classification (CrtCl, pronounced Critical). CrtCl formulates image classification training in a generator-critic framework, with a base classifier acting as a generator, and a correctness critic imposing a loss on the classifier. The base classifier, acting as the generator, given images, generates the probability distribution over classes and intermediate embeddings. The critic model, given the image, intermediate embeddings, and output predictions of the base model, predicts the probability that the base model has produced the correct classification, which then can be back propagated as a self supervision signal. Notably, the critic does not use the label as input, meaning that the critic can train the base model on both labeled and unlabeled data in semi-supervised learning settings. CrtCl represents a learned loss method for accuracy, alleviating the negative side effects of using cross-entropy loss. Additionally, CrtCl provides a powerful way to select data to be labeled in an active learning setting, by estimating the classification ability of the base model on unlabeled data. We study the effectiveness of CrtCl in low-labeled data regimes, and in the context of active learning. In classification, we find that CrtCl, compared to recent baselines, increases classifier generalization and calibration with various amounts of labeled data. In active learning, we show our method outperforms baselines in accuracy and calibration. We observe consistent results across three image classification datasets.**

I. Introduction

In recent years, significant advances in the architecture of deep learning models have led to the development of powerful automated systems [1], [2], [3]. Central to the efficacy of these models is the rigorous optimization of millions of parameters with respect to a given loss function [4]. While crafting clever network architectures and loss functions has proven important [5], [6], often, the best empirical results come from letting large models learn as end-to-end as possible for the task at hand [7], [8]. That is, where the model is trained to directly map inputs to outputs, and directly optimize for the relevant metric, learning the entire sequence of transformations required without relying on human-crafted intermediate steps or features. End-to-end training enables the model to autonomously discover the most effective representations and relationships for the task at hand.

In order to directly optimize for the most relevant metric for a given task, the metric itself should be represented as a differentiable loss function. For regressions tasks, this presents no problem, the metric of interest (Mean Absolute Error, Mean Squared Error etc.) is directly differentiable. However, for tasks like classification, the true metric of interest, accuracy, is not. To remedy this, proxy loss functions like cross-entropy, which encourages the output probability of the network to match a one-hot encoding of the ground truth label, are used. This works extraordinary well in practice, and large scale image classification networks have reached or surpassed human level performance on a host of tasks [9]. However, this is not purely end-to-end learning, and while cross-entropy is a good proxy for accuracy, there are unintended side effects.

One particularly well-studied side effect, with significant impact on the industrial use of neural networks, is calibration [10], [7]. Calibration is the notion or measure of how well the probability output of a network, matches its accuracy rate. Modern large-scale networks tend to be ill-calibrated in that they are overconfident in their predictions - an artifact of the one-hot encoding of cross-entropy loss [11]. Calibration is extremely important in practice because it allows the users of a model to correctly understand the uncertainty in a prediction. Further, ill-calibration can affect down-stream tasks, in particular it can effect the efficacy of using a model's output in risk analysis, and active learning [12]. In active learning, a well calibrated model can be used to do uncertainty sampling by making predictions on unlabeled data, and it's most uncertain predictions can be used to guide which next samples to collect [13].

Inspired by this problem, we introduce a new learned loss function called Critic Loss For Image Classification (CrtCl, pronounced Critical). CrtCl formulates image classification as directly training a classifier to optimize accuracy by simultaneously training a critic to estimate accuracy. From one perspective, CrtCl treats image classification as a generator critic two-player game, where the base classification model *generates* features and class probabilities, and the *critic* learns to distinguish between correct or incorrect predictions. As the critic network is a differentiable neural network, it can be used as a loss function to teach the generator how to fix incorrect predictions to be correct. As a result, CrtCl allows for end-to-end training of the accuracy metric, which leads to more generalizable and better-calibrated models.

We demonstrate the effectiveness of CrtCl for active learning. In many real-world settings, abundant data exist but few are labeled. Furthermore, labeling data can be expensive, especially in medical and cybersecurity domains [14], [15], [16]. It is thus necessary to have models that can learn

1946-0759/24/$31.00 ©2024 IEEE
DOI 10.1109/ICMLA61862.2024.00104

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

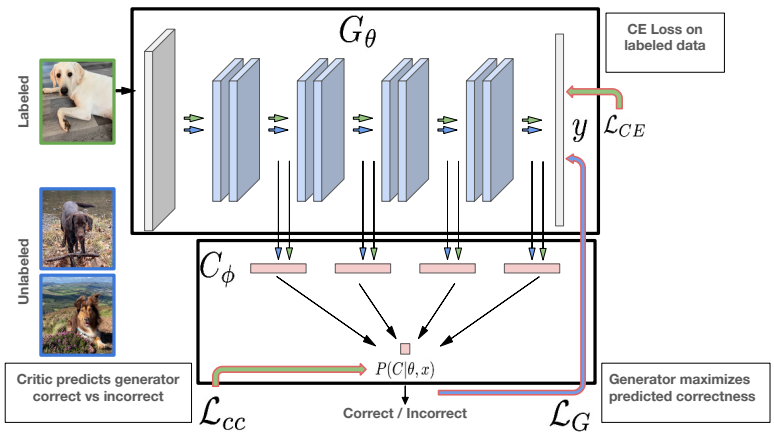

Fig. 1. A schematic of CrtCl, the classifier G_θ takes in images and produces intermediate representations and class probabilities, where the `ARGMAX` of the probabilities is the classification. The critic network, C_ϕ, takes in the representations of G_θ and predicts whether G_θ classifies an example correctly. Once trained, the critic network can be used as a learned loss on both labeled and unlabeled data to train the generator to be more correct, while avoiding miscalibration from cross-entropy loss. Further, the critic model's prediction on unlabeled data points can be used to suggest misclassified points for active learning.

effectively with fewer labels, and also strategically identify which samples to label next. CrtCl, coupled with standard cross-entropy loss, outperforms baseline methods for active learning for image classification on three separate data sets, particularly in realistic low-labeled data regimes. We show that the critic model can be used for semi-supervised learning by applying the critic loss to unlabeled samples and for selecting the most informative points to label to improve accuracy. We also show that the models trained with our method tend to be better calibrated than other methods. Lastly, we run several ablation experiments to understand the effect of CrtCl's use as an auxiliary loss function, as a method to actively sample the next points to be labeled, and how these facets influence each other as a joint method. We show that our method, even just as an auxiliary loss function, outperforms baseline methods in terms of accuracy and network calibration.

Our contributions are 1) we introduce our Critic Loss For Image Classification (CrtCl) method which aims to **train classifiers to optimize accuracy end-to-end**. 2) We show that CrtCl can be deployed in **active learning and semi-supervised learning settings**, outperforming recent baseline techniques, and allowing for both better learning in low-data regimes, and better selection of the data label. 3) We show that CrtCl tends to produce **better calibrated classifications** meaning that end-users have a better grasp on the model's uncertainty.

II. Related Work

1) Calibration: Calibration measures the alignment between a model's uncertainty and observed probabilities, and it is often measured by the expected calibration error (ECE) [17]. It has been shown that modern neural networks, especially larger models with low classification error, tend to be ill-calibrated and are overconfident in their predictions [18]. It has been show that calibration is particularly relevant in online settings [19] and for structure predictions [20].

Method	Auxiliary Networks	Hyper-parameters	Steps Per Epoch	Active Learning	Semi-Supervised	Calib-ration
Label Smoothing [25]	0	2+	1	✗	✗	✓
Temperature Scaling [18]	0	2+	1	✗	✗	✓
Learning Loss [31]	1	2	1	✓	✗	✗
TOD [32]	2	3	1	✓	✓	✗
PT4AL [33]	1	2	1	✓	✗	✗
CrtCl (ours)	1	2	2	✓	✓	✓

TABLE I
Comparison of Active Learning and Calibration Methods

Several loss-based methods have been introduced to improve calibration such as a reguarlization term that also occasionally improves generalization [21], a focal loss [22], a relaxation of ECE [23], a calibration-inducing kernel [24], and label smoothing [25]. Label smoothing, which relaxes the hard one-hot loss of cross-entropy to use "smoother" labels, has been further shown to improve model calibration [26]. Label smoothing and calibration also relate to work in knowledge distillation (KD) [27], in which the authors trained a secondary smaller network to predict the output logits of a larger network. There have been many works on different version of KD, including self-distillation [28]. With the work in [29], the authors study the effect that image augmentation combined with loss functions, such as MixUp [30], can improve calibration.

2) Active Learning: Active learning is a rich and well studied field, with many branches [34]. Methods have been proposed which select examples based on the predicted class probabilities [35], [36], the difference between the top k predicted classes [37], and the entropy of the predicted probabilities [38], [37].

A relatively new class of methods most similar to CrtCl aims to quantify the expected improvement or loss of the current

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

model to use as proxy for data selection [39], [40], [41]. In the state of the art Temporal Output Discrepancy (TOD) [37], the prediction disagreement on the unlabeled data acts as a proxy to estimate uncertainty, and thus select samples. A similar method uses a generative-adversarial model to predict which samples belong to the labeled or unlabeled data sets [42].

Also similar to our work is a method that first trains a network on the pretext task of predicting image rotation, which requires no labels [33], and uses this loss to select samples.

Most similar to our work, Learning Loss for Active Learning (LL) [31], uses an auxiliary network to predict the loss value of the base network for a given data point, and then uses this prediction on unlabeled data to select data likely to have a high loss value. Further, this method has been extended with more mathematical analysis, particularly for regression tasks [43]. Additionally, similar methods have been explored for segmentation methods, particular to detect adversarial attacks [44].

The motivation of our work, to find an alternative loss function for image classification is also similar to work in energy models [45], however we argue these models have a similar issue of optimizing a proxy loss function.

3) Semi-supervised Learning: Semi-supervised learning is well studied area of research with the work by [46] provides an excellent survey of modern Semi-supervised learning methods.

4) Overview: We compare several salient approaches in Table I. Notably, CrtCl is the only approach geared towards improving both active learning and semi-supervised learning while also aiming to improve network calibration and uncertainty estimation, something especially important in low-labeled data regimes. These works were picked as baselines because of their similarity to our method, and because of their relatively recent publication and performance on active learning leader boards (paperswithcode.com).

III. Method

In this section, we formalize CrtCl, which aims to improve model generalization and active learning. The method involves the formulation of image classification in a generator-critic framework. Here, the classification network acts as a generator, aiming to generate correct predictions with respect to the critic network. The critic network aims to discriminate between correct and incorrect predictions. We first describe the setup of the two networks and CrtCl's training algorithm.

A. Problem Statement

In semi-supervised learning, we are given sets of labeled examples, $\mathcal{D}_L$, unlabeled examples $\mathcal{D}_U$, and a test set $\mathcal{D}_{Test}$. We aim to train a classification neural network G_θ that maximizes predictive performance on $\mathcal{D}_{Test}$ having only trained on $\mathcal{D}_L$ and $\mathcal{D}_U$. Here predictive performance is measured both in terms of accuracy as well as model calibration. In the active learning setting, we can also iteratively select unlabeled samples to label $x_i \in \mathcal{D}_U$. Ideally, we want a method which has high predictive performance with various amounts of labeled data in both active learning and semi-supervised learning settings.

B. Classification Network

The classification network, represented as a generator is a function $G_\theta(x)$, parameterized by θ, maps an input $x \in \mathcal{X}$ (e.g., images) to a probability distribution over the class labels. Specifically, for a given input x, the classification network produces a vector $z_i \leftarrow G_\theta(x)$ where $z_i \in \mathbb{R}^K$ where K is the number of classes. Each element of this vector represents the predicted probability of the corresponding class. The predicted label $\hat{y}$ is then determined as the class with the highest probability, i.e., $\hat{y} = \texttt{ARGMAX}(z_i)$. The architecture of G_θ can be varied; however, for this work we use convolutional neural networks for image classification with more implementation specifics provided in the experimental section.

C. Critic Network

The Critic is a function $C_\phi(\cdot)$, parameterized by ϕ, which operates on the feature set generated by the classification network. For a given input x, the classification network $G_\theta(x)$ also produces a feature set $F_\theta(x) \in \mathbb{R}^M$, where M represents the dimensionality of the feature space. The Critic then evaluates these features, $C_\phi(F_\theta(x))$, and outputs a scalar value. This value quantifies the Critic's confidence that, the classification network's prediction, equals the ground truth prediction y, that is that $y = \texttt{ARGMAX}(G_\theta(x))$.

There are many options for how and which features F_θ are passed to C_ϕ, which is largely dependant on the architecture of G_θ. Following the learning loss module introduced in [31], our Critic network takes in several intermediate feature layers from G_θ, which are then passed through a global average pooling layer, a fully connected layer, and finally concatenated to produce an embedding of the generator's predictions, from which a fully connected layer outputs a single scalar value, representing the probability $G_\theta(x)$ is correctly predicting for x.

D. Critic Loss Procedure

The training procedure for critic loss uses two loss functions and two additional steps in addition to the standard cross-entropy loss. Let $\mathcal{X}$ be the input space (e.g., images) and $\mathcal{Y}$ be the output space (e.g., class labels). The training dataset is denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ and the unlabeled dataset is denoted as $\mathcal{D}_u = \{(x_i)\}_{i=1}^M$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$,

It works as follows: for a given training input x, we compute the feature set $F_\theta(x)$ and output probabilities z by computing $G_\theta(x)$. The cross-entropy loss $\mathcal{L}_{CE}$ for the classification task is given by:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(z_{i,c}) \tag{1}$$

where $y_{i,c}$ is the ground truth label per sample i and per class c, and $z_{i,c}$ is the per i and c. Subsequently, predictions are categorized as correct or incorrect based on whether $\hat{y}$ matches the true label y. Let C_R denote the set of correct predictions and I denote the set of incorrect predictions. Then, the Critic loss $\mathcal{L}_{cc}$ is calculated using the Wasserstein distance,

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Training Procedure for Image Classification with Critic Network

Input: Labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, Unlabeled dataset $\mathcal{D}_u = \{x_j\}_{j=1}^{M}$
Initialize: Classifier G_θ, Critic C_ϕ
Parameters: Learning rate η, epochs E, loss weight γ, stopping epoch E'
for epoch $= 1, \ldots, E$ **do**
  for each $(x_i, y_i) \in \mathcal{D}$ **do**
    $F_\theta(x_i), z_i \leftarrow G_\theta(x_i)$
    $\mathcal{L}_{CE} \leftarrow -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(z_{i,c})$
    Partition: $C_R, I \leftarrow \{\texttt{ARGMAX}(z_i) = y_i\}, \{\texttt{ARGMAX}(z_i) \neq y_i\}$
    $\mathcal{L}_{cc} \leftarrow \sum_{x\in I}\log C_\phi(F_\theta(x)) - \sum_{x\in C_R} C_\phi(F_\theta(x))$
    $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{CE}$
    $\phi \leftarrow \phi - \eta\nabla_\phi\mathcal{L}_{CC}$
    if epoch $< E'$ **then**
      $\mathcal{L}_G \leftarrow -\gamma\sum_{x\in\mathcal{D}_u}\log C_\phi(G_\theta(x))$
      $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_G$
    end if
  end for
end for
return Optimized G_θ and C_ϕ

also known as the Earth Mover's distance, which is defined for probability distributions P_r, the distribution of incorrect samples, and P_g, the distribution of correct samples as:

$$W(P_r, P_g) = \inf_{\gamma\in\Pi(P_r,P_g)} \mathbb{E}_{(r,g)\sim\gamma}[\|r - g\|], \tag{2}$$

where $\Pi(P_r, P_g)$ denotes the set of all joint distributions $\gamma(r, g)$ whose marginals are P_r and P_g respectively, with r being the prediction of the critic for incorrect samples and g being for correct. Which is formalized by applying the Kantorovich-Rubinstein duality as:

$$\min_G \max_{C\in\mathcal{C}'} \mathbb{E}_{x\sim P_r}[C(G(x)] - \mathbb{E}_{x\sim P_g}[C(G(x))] \tag{3}$$

The set $\mathcal{C}'$ represents the space of 1-Lipschitz functions, ensuring that the discriminator is constrained to be a 1-Lipschitz function.

To enforce the 1-Lipschitz condition on the discriminator, [47] proposed clipping the weights of the discriminator to a compact space $[-c, c]$, where c is a hyperparameter. This can be formally represented as $w \leftarrow \text{clip}(w, -c, c)$, for every weight w in the discriminator. Weight clipping directly constrains the capacity of the discriminator, ensuring that the gradient norms are bounded, which is a necessary condition for the 1-Lipschitz continuity.

Thus using the Earth Mover's distance for our critic loss, we have $\mathcal{L}_{cc}$ formalized as:

$$\mathcal{L}_{cc} = \sum_{x\in I} C_\phi(F_\theta(x)) - \sum_{x\in C_R} C_\phi(F_\theta(x)) \tag{4}$$

Here, $C_\phi(F_\theta(x))$ represents the Critic's assessment of the classification network's feature set, aiming to maximize the features of correct predictions and minimize incorrect predictions, and with C_R and I being the sets of correctly and incorrectly labeled examples respectively. In this step, $\mathcal{L}_{cc}$ is backpropagated to C_ϕ, and $\mathcal{L}_{CE}$ is backpropagated to F_θ.

In the semi supervised setting, we then sample a batch of data $\{x_1...x_b\} \in \mathcal{D}_u$. For each input $x \in \{x_1...x_b\}$, the generator's output $F_\theta(x)$, is passed to the Critic. The Critic then assesses these outputs, and the following loss is computed $\mathcal{L}_G$, is defined as:

$$\mathcal{L}_G = -\sum_{x\in\mathcal{D}_u} C_\phi(F_\theta(x)) \tag{5}$$

Here, $C_\phi(F_\theta(x))$ represents the Critic's assessment of unlabeled outputs from the classification network. By backpropagating this loss through the classification network, the model learns to adjust its predictions, aiming to correctly label these unlabeled data points. Note, in this work we didn't explore other types of GAN loss/training methods, as W-GAN's often lead to the most stable performance, however exploring other loss functions is a promising future work.

E. Active Learning

In an active learning context, our model utilizes the Critic network to efficiently select samples from an unlabeled dataset for labeling. Let $\mathcal{D}_u$ denote the unlabeled dataset and $\mathcal{D}_L$ denote the labeled dataset. The active learning cycle proceeds as follows:

1) For each unlabeled sample $x_u \in \mathcal{D}_u$, compute the feature set $F_\theta(x_u)$ using the classification network.
2) Apply the Critic network to assess the probability the classifier is correct: $p_u = C_\phi(F_\theta(x_u))$.
3) Rank the samples in $\mathcal{D}_u$ based on p_u, identifying those least likely to be correct.
4) Select a subset $\mathcal{S} \subset \mathcal{U}$, comprising n samples with the lowest p_u values.
5) Obtain labels for the samples in $\mathcal{S}$ from a human oracle, resulting in a set of newly labeled pairs $\{(x_s, y_s)\}_{s\in\mathcal{S}}$.
6) Update the labeled dataset: $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup \{(x_s, y_s)\}_{s\in\mathcal{S}}$.
7) Retrain the model with the updated dataset $\mathcal{D}_L$.

This approach allows for a focused expansion of the labeled dataset, prioritizing samples that are likely to provide the most informative feedback for model retraining. By iteratively applying this process, the model can effectively improve its performance with fewer labeled examples.

IV. Experiments

In this work we focus on the efficacy of our model in the domain of active learning for image classification. We test our method in an active learning setting, where the model starts with few labeled data, but over the course of several rounds picks new data to be labeled. We gauge the efficacy of our method in both low-labeled data regimes, and in its ability to pick the most informative samples for labeling. Lastly, we provide two ablation studies to tease apart how much of

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

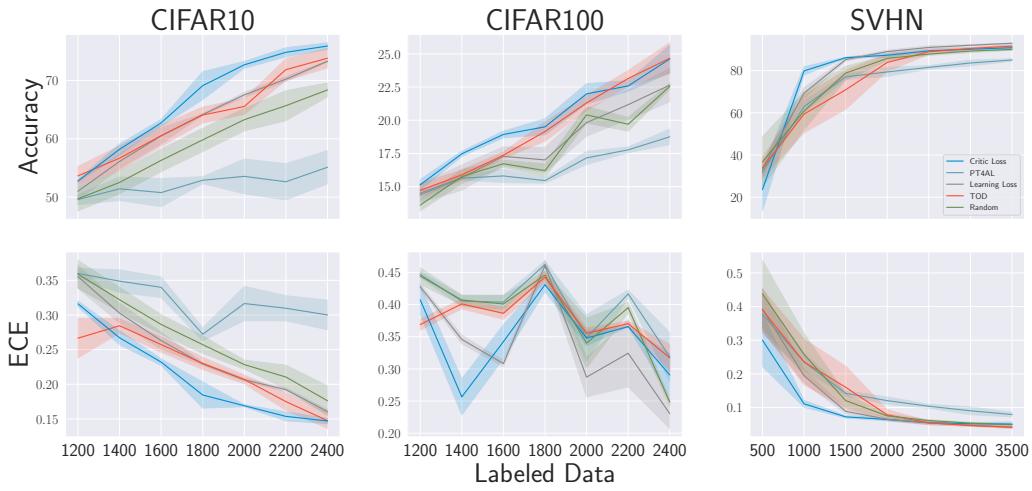

Fig. 2. The accuracy and expected calibration error (ECE) results of our method, compared to Learning Loss [43], TOD [32], PT4AL [33], and a standard training baseline for all three data sets. For all datasets, in the majority of active learning cycles, our method produced the more generalizable models (higher test accuracy), and better calibrated models (lower ECE).

the performance increase is coming from the samples being selected, versus better features being learned from the loss function.

A. Datasets

For all experiments we use the SVHN [48], CIFAR10 and CIFAR100 [49], data sets. CIFAR10 and CIFAR100 have an available labeled data set of 50,000 images and a test set of 10,000 images, with 10 or 100 classes respectively that represent common objects. SVHN has 60,000 training images, and 10,000 test images, for 10 classes of pictures of house number digits. For all data set, the images were normalized to have zero-mean and unit variance. All images were cropped to 32x32 pixels. For the training sets random horizontal flipping was used as an augmentation.

V. Baselines

We compare our method to recent state of the art methods, Learning Loss (LL) [31], Temporal Output Discrepancy (TOD) [50] and Using Self-Supervised Pretext Tasks for Active Learning (PT4AL) [33].

LL uses an auxiliary network, which takes the features produced by the base network, to predict the loss value the base network will have on a given data point. In this way it can be used as an auxiliary loss, by backpropagating it's error through the features of the base network. Additionally, it uses its estimates of loss for unlabeled data to suggest which data will be best to have labeled by considering the data with the highest estimates of loss.

The TOD method uses the output of the base model over different optimization steps to estimate the loss value of the model on specific data points. This difference between outputs at different optimization steps can be used to estimate loss for unlabeled data, and thus can be used as a selection criteria. Additionally, this difference of estimates on unlabeled data, and is used to train the base model in an unsupervised/semi-supervised manner.

PT4AL is a recent work that has show state-of-the-art results for active learning and works by pre-training an auxiliary model on the pretext task of predicting image rotation. Then, for the unlabeled data pool, the rotation loss is used as a proxy to sample which data point to be next labeled. The method gives impressive results for active learning, however it does not have a way to train in a semi-supervised manner.

Lastly, for all data sets and settings we compare to a standard Resnet18, trained with only cross-entropy loss, and with a random selection of the next points to include at each active learning cycle.

A. Implementation Details

Following [32] for all experimental settings, we use a Resnet18 architecture. We use the last four feature layers of the network, as well as the input image and output probability distribution as the extracted features to train the critic network. The critic network takes these features as input, and use Global Average Pooling to pool them all. Then a linear layer is applied to each pooled feature layer, to project to a 128 dimensional space. Then each 128 dimensional space is concatenated, and a single linear layer projects the output to a single scalar value. The ReLU activation is used for all layers except the final layer.

For $\mathcal{L}_G$, we find our method performs optimally, and only backpropagate to the last layer of the generator (which then is backpropagated to all other layers) instead of directly propagating to all feature layers. For the base network, in all cases we use Stochastic Gradient Descent as the optimizer with a learning rate of 0.1, and a momentum of 0.9. For the critic network we use the AdamW optimizer with a learning rate of 0.001. We train for 200 epochs, and decrease the learning rate by a multiplicative factor of 0.1 at epoch 160. For CIFAR10

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

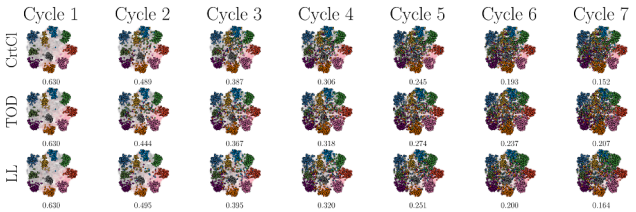

Fig. 3. For this setting we took a standard network trained with cross-entropy on CIFAR10, and produced t-SNE embeddings for the entire data set. We then show, for the highest performing methods, at each cycle which data points are selected by each method to be labeled. Fully opaque points represent those chosen to be labeled, and the unlabeled points are shown as highly transparent. Additionally we compute the mean silhouette score for each cluster. Intuitively, higher scores indicate the model is selecting points more similar to the data already in the labeled set. Whereas, the lower the score, the more the model is selecting more diverse samples that are spread out further in the feature space. We observe that early on in training, our model selects more similar samples, and later on in training it achieves the lowest clustering score, indicating it selects the most diverse samples per class.

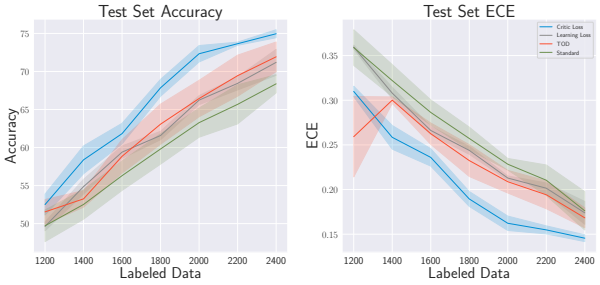

Fig. 4. Our first ablation study which, for all methods, samples the next data points to be labeled randomly, but still uses the auxiliary loss function of all methods for the CIFAR10 data set. It is shown that our loss functions leads to better generalization (higher accuracy), and better calibration (lower ECE).

and CIFAR100 we use a γ value, which is the weighting of the $\mathcal{L}_G$, of 0.04, for SVHN we use 0.02. Additionally similarly to [31], we set E', the epoch we stop using $\mathcal{L}_G$, to 130 for all experiments.

For the active learning setting we start with a randomly initialized set of size k, where k is 1200, 1200, and 500 for CIFAR10, CIFAR100, and SVHN respectively. We then train for 7 cycles, where for each cycle we add t more labeled samples to the labeled training data set according the the active learning selection criteria, where t is 200, 200, and 500 for CIFAR10, CIFAR100 and SVHN respectively. In this way, we aim to study the efficacy of our model in extremely low-labeled data regimes, and where each iteration very little labeled data is added to better mimic real-world settings. At the end of each cycle, the test accuracy and expected calibration error were recorded. All experiments are run for a total of 3 randomized trials, and the mean results with the +/- standard deviation are reported.

For all other hyperparameters and baselines, we follow the methods described in [31], [32] and [33].

All experiments were run on 2xA100 NVIDIA GPUs, and while the time varied from experiment to experiment a single trial took approximately 1 hour of compute time.

B. Evaluation

The primary metric of interest is accuracy on the held out test set for each dataset to gauge model generalization. Further, we are also interested in model calibration which we measure using the expected calibration error (ECE). The expected calibration error (ECE) is a common metric for assessing the calibration of probabilistic models in classification tasks, where the lower ECE, the better the model is calibrated. Given a model with c classes, we divide predictions into M bins based on their predicted confidence. Let B_m represent the set of indices of samples that fall into bin m, for $m = 1, \ldots, M$. The ECE is calculated as follows:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{6}$$

C. Active Learning Setting

In Figure 2 we compare the accuracy of CrtCl, the learning loss [31], TOD [32], PT4AL [33], and a network trained with cross-entropy on randomly-selected active learning samples. As can be seen, for almost all cycles in all data sets our method outperforms the baselines in terms of accuracy, especially in the very low-labeled data regimes. We also show the resulting ECE for each method, which again shows in almost all cases that are model has the lowest calibration error. Showing that the learned classifier is better at estimating it's own uncertainty.

D. Clustering Analysis

In our second experimental setting we aim to show both qualitatively and quantitatively how our method selects data points from the overall data set. To this end, we study the CIFAR10 data set specifically. We take a Resnet18 network, trained in a standard setting with just cross-entropy loss until convergence on CIFAR10. We then use this network to extract the output features for the entire training set of CIFAR10, and compute a t-SNE projection. We then, for each method, for each cycle for a single trial, plot the data selection as seen in Figure Figure 3. Further, for each method and cycle we compute the silhouette score, for each label.

In this case, higher scores mean the method is selecting active learning points that are most similar to those already in the labeled data set. Whereas lower scores means the clusters are more spread out over the embedding space, indicating the model is selecting data points for each class that are more diverse.

Although largely qualitative, this analysis provides insight into the mechanism behind our method, and the comparable baselines.

We observe that for the first three cycles our method's silhouette score is in between the other two, suggesting our method, in early stages prefers data points that reinforce current knowledge, whereas after cycle four it has the lowest

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

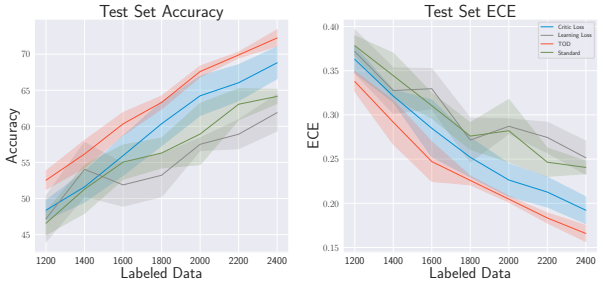

Fig. 5. Our second ablation study which, for all methods, samples the next data points using the described methods, but does not backpropagate the auxiliary loss, for the CIFAR10 dataset. While our method still outperforms Learning Loss, TOD performs the best. Indicating that the benefits of our model are more entangled with the critic loss, which intuitively is expected given the nature of the generator-critic learning process.

score, indicating it is successfully exploring more of the embedding space for each class compared to other models.

E. Ablation Studies

All evaluated methods, besides PT4AL, have both an active learning selection mechanism, as well as a method to compute and backpropagate an auxiliary loss, whether on just the labeled set, or as a semi-supervised loss on the unlabeled data set. The natural question arises - how much of the improvement is from the auxiliary loss versus the data labeling selection mechanism. To this end, we designed and ran two ablation experiments where we test the efficacy of each model (besides PT4AL) with and without the auxiliary loss, and with and without the selection mechanism.

1) Auxiliary Loss Only : In the first ablation study, we backpropagate the different auxiliary loss functions to the classification network, but when selecting points to label we randomly sample. Here, we aim to study how much of the benefit is purely coming from the CrtCl loss term. We plot the results for CIFAR10 in Figure 4 which shows improved test accuracy and calibration for all cycles.

Further, the improvement is more exaggerated than when comparing to the full implementation. This suggests that our method, all else being equal, in just semi-supervised learning settings, may provide the best way to optimize a base network for highest accuracy, while preserving network calibration. That is, our generator-critic network loss function provides improvement over SOTA baselines, as well as standard cross-entropy loss in a standard image classification training setting.

2) Active Learning Sampling Only: In the second ablation study, we train the auxiliary networks as described for each method, but do not backpropagate the losses to the main network, and instead only use the selection mechanism for labeling data. In Figure 5, we show the results for CIFAR10, which show that in this setting the TOD method outperforms all others, including our own. This result and the previous ablation study suggest that CrtCl performs well as an auxiliary loss and a joint loss and data selection method, whereas TOD performs better for data selection.

F. Trade-Off

It should be noted that while our method yields superior results, it is more computationally expensive, requiring roughly double the number of gradient descent steps of standard deep learning training, as well as roughly twice the memory. While generally this range is acceptable, it is a significant trade-off and worth noting.

VI. Conclusion

In this paper we introduce CrtCl, a novel learned loss method, which formulates training a classification network as a two player, generator-critic framework, where the base network generates features and probability distributions over classes, and the critic network produces a estimate that the generator network is correct. This critic network can be used to provide semi-supervision over unlabeled data, as well as to select data to be labeled in an active learning setting. Our method outperforms SOTA methods and standard baselines, in terms of accuracy, calibration and diverse sampling for three data sets.

VII. Acknowledgements

This project is partially supported by the National Science Foundation (NSF); the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a program of Schmidt Sciences, LLC; the National Institute of Food and Agriculture (US-DA/NIFA); the Air Force Office of Scientific Research) (AFOSR), and Toyota Research Institute (TRI).

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," vol. 550, no. 7676, pp. 354–359. [Online]. Available: https://doi.org/10.1038/nature24270

[3] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[4] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[6] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[7] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.

[8] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," 2023.

[9] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," 2023.

[10] C. Wang, "Calibration in deep learning: A survey of the state-of-the-art," 2024.

[11] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *CoRR*, vol. abs/1706.04599, 2017. [Online]. Available: http://arxiv.org/abs/1706.04599

[12] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1721–1730. [Online]. Available: https://doi.org/10.1145/2783258.2788613

[13] B. Settles, "Active learning, volume 6 of synthesis lectures on artificial intelligence and machine learning," *Morgan & Claypool*, 2012.

[14] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 417–424. [Online]. Available: https://doi.org/10.1145/1143844.1143897

[15] X. Li, M. Xia, J. Jiao, S. Zhou, C. Chang, Y. Wang, and Y. Guo, "HAL-IA: A hybrid active learning framework using interactive annotation for medical image segmentation," vol. 88, p. 102862. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841523001226

[16] N. Nissim, A. Cohen, R. Moskovitch, A. Shabtai, M. Edry, O. Bar-Ad, and Y. Elovici, "Alpd: Active learning framework for enhancing the detection of malicious pdf files," in *2014 IEEE Joint Intelligence and Security Informatics Conference*, 2014, pp. 91–98.

[17] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," vol. 29, no. 1. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9602

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *CoRR*, vol. abs/1706.04599, 2017. [Online]. Available: http://arxiv.org/abs/1706.04599

[19] V. Kuleshov and S. Ermon, "Reliable confidence estimation via online learning," 07 2016.

[20] V. Kuleshov and P. S. Liang, "Calibrated structured prediction," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/52d2752b150f9c35ccb6869cbf074e48-Paper.pdf

[21] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," *CoRR*, vol. abs/1701.06548, 2017. [Online]. Available: http://arxiv.org/abs/1701.06548

[22] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. S. Torr, and P. K. Dokania, "Calibrating deep neural networks using focal loss," *CoRR*, vol. abs/2002.09437, 2020. [Online]. Available: https://arxiv.org/abs/2002.09437

[23] A. Karandikar, N. Cain, D. Tran, B. Lakshminarayanan, J. Shlens, M. C. Mozer, and B. Roelofs, "Soft calibration objectives for neural networks," 2021.

[24] A. Kumar, S. Sarawagi, and U. Jain, "Trainable calibration measures for neural networks from kernel mean embeddings," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 2805–2814. [Online]. Available: https://proceedings.mlr.press/v80/kumar18a.html

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[26] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *CoRR*, vol. abs/1906.02629, 2019. [Online]. Available: http://arxiv.org/abs/1906.02629

[27] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[28] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," *CoRR*, vol. abs/1905.08094, 2019. [Online]. Available: http://arxiv.org/abs/1905.08094

[29] L. Zhang, Z. Deng, K. Kawaguchi, and J. Zou, "When and how mixup improves calibration," 2022.

[30] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2018.

[31] D. Yoo and I. S. Kweon, "Learning loss for active learning," *CoRR*, vol. abs/1905.03677, 2019. [Online]. Available: http://arxiv.org/abs/1905.03677

[32] S. Huang, T. Wang, H. Xiong, B. Wen, J. Huan, and D. Dou, "Temporal output discrepancy for loss estimation-based active learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[33] J. S. K. Yi, M. Seo, J. Park, and D. Choi, "Using self-supervised pretext tasks for active learning," *CoRR*, vol. abs/2201.07459, 2022. [Online]. Available: https://arxiv.org/abs/2201.07459

[34] W. H. Beluch, T. Genewein, A. Nurnberger, and J. M. Kohler, "The power of ensembles for active learning in image classification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9368–9377.

[35] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *International Conference on Machine Learning*, 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:5319590

[36] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," 1994.

[37] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2372–2379.

[38] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, M. Lapata and H. T. Ng, Eds. Honolulu, Hawaii: Association for Computational Linguistics, Oct. 2008, pp. 1070–1079. [Online]. Available: https://aclanthology.org/D08-1112

[39] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," in *International Conference on Machine Learning*, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:14293159

[40] A. Freytag, E. Rodner, and J. Denzler, "Selecting influential examples: Active learning with expected model output changes," in *European Conference on Computer Vision*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:2013441

[41] C. Käding, E. Rodner, A. Freytag, and J. Denzler, "Active and continuous exploration with deep neural networks and expected model output changes," *CoRR*, vol. abs/1612.06129, 2016. [Online]. Available: http://arxiv.org/abs/1612.06129

[42] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," *CoRR*, vol. abs/1904.00370, 2019. [Online]. Available: http://arxiv.org/abs/1904.00370

[43] M. Shukla and S. Ahmed, "A mathematical analysis of learning loss for active learning in regression," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3315–3323. [Online]. Available: http://arxiv.org/abs/2104.09315

[44] V. Besnier, A. Bursuc, D. Picard, and A. Briot, "Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation," 2021. [Online]. Available: https://arxiv.org/abs/2108.01634

[45] X. Yang and S. Ji, "JEM++: improved techniques for training JEM," *CoRR*, vol. abs/2109.09032, 2021. [Online]. Available: https://arxiv.org/abs/2109.09032

[46] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, pp. 373 – 440, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:254738406

[47] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017. [Online]. Available: https://arxiv.org/abs/1701.07875

[48] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng, "Reading digits in natural images with unsupervised feature learning," 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:16852518

[49] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

[50] S. Huang, T. Wang, H. Xiong, J. Huan, and D. Dou, "Semi-supervised active learning with temporal output discrepancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Authorized licensed use limited to: Cornell University Library. Downloaded on March 06,2025 at 20:24:00 UTC from IEEE Xplore. Restrictions apply.