Keep Learning: Self-supervised Meta-learning for Learning from Inference

Akhil Kedia
Samsung Research, Seoul, South Korea
akhil.kedia@samsung.com

Sai Chetan Chinthakindi
Samsung Research, Seoul, South Korea
sai.chetan@samsung.com

Abstract

A common approach in many machine learning algorithms involves self-supervised learning on large unlabeled data before fine-tuning on downstream tasks to further improve performance. A new approach for language modelling, called dynamic evaluation, further fine-tunes a trained model during inference using trivially-present ground-truth labels, giving a large improvement in performance. However, this approach does not easily extend to classification tasks, where ground-truth labels are absent during inference. We propose to solve this issue by utilizing self-training and back-propagating the loss from the model's own class-balanced predictions (pseudo-labels), adapting the Reptile algorithm from meta-learning, combined with an inductive bias towards pre-trained weights to improve generalization. Our method improves the performance of standard backbones such as BERT, Electra, and ResNet-50 on a wide variety of tasks, such as question answering on SQuAD and NewsQA, benchmark task SuperGLUE, conversation response selection on Ubuntu Dialog corpus v2.0, as well as image classification on MNIST and ImageNet without any changes to the underlying models. Our proposed method outperforms previous approaches, enables self-supervised fine-tuning during inference of any classifier model to better adapt to target domains, can be easily adapted to any model, and is also effective in online and transfer-learning settings.

1 Introduction

It is a common consensus that the performance of Machine Learning algorithms improves with increasing data. However, due to the difficulty of obtaining large quantities of labelled data, many models (particularly in Natural Language Processing domain) such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018) and UniLM (Dong et al., 2019) rely on unsupervised pre-training on unlabelled data to learn useful features which are then fine-tuned for other downstream tasks. While this approach leads to large gains in performance, it leads to a mismatch between a network's pre-training and final fine-tuning. Some approaches such as pseudo-labelling (Lee, 2013) have proposed utilizing data-augmentation of unlabelled data with the model's own predictions to better pre-train a model.

While these methods are limited to the training phase, Krause et al. (2018) proposed to continue training a language modeling model (which is the task of predicting the next token in a sequence of tokens) during the evaluation stage, achieving significant improvements as the model learns to better adapt to the inference data, without any modifications to the model architecture or any access to training data. For language modeling, the ground truth labels are the next input token, which are trivially accessible to the model to facilitate this learning. However, this method does not easily generalize to standard classification tasks due to the unavailability of labels during inference. This is the setting which we further explore in this paper, in which we are provided with a classification model already trained on training data, but with no access to the training data, and the aim is to further improve the performance of the model by utilizing self-training on the inference data.

To solve the above problem, we propose a method to train any classifier model during inference, utilizing methods used in domain adaptation, noisy-label learning, and multi-task meta-learning. With ground truth labels being absent, we utilize the model's own predictions as the pseudo-labels for those samples and utilize Class Balanced Self Training (CBST) (Zou et al., 2018) to filter samples based on the model's confidence while retaining class balance. However, naive online learning or

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 63–77
April 19 - 23, 2021.

re-training on the inference data is not optimal due to the noise in the labels biasing the network, as well as the small size of the inference set. We solve this issue by leveraging the Reptile Meta Learning Algorithm (Nichol et al., 2018) to improve generalization, supplemented with an explicit inductive bias towards the model's pre-trained weights.

Our experimental results and ablation studies show that our method improves the performance of standard backbones such as BERT, Electra (Clark et al., 2020) and ResNet (He et al., 2016) on a wide variety of tasks, such as question answering on SQuAD (Rajpurkar et al., 2018) and NewsQA (Trischler et al., 2017), benchmark task SuperGLUE (Wang et al., 2019), and conversation re-ranking on Ubuntu Dialog corpus v2.0 (Lowe et al., 2017) for NLP, as well as object classification on MNIST (Deng, 2012) and ImageNet (Deng et al., 2009) without any changes to the underlying models, while outperforming previous approaches. Our method can also be utilized for continual self-supervised fine-tuning of classifiers on target domains, as well as in transfer-learning settings, without any model-level modifications.

2 Proposed Method

Our proposed technique is the self-supervised training of a classifier model during inference, consisting of three parts – using confident predictions as pseudo-labels, utilizing the Reptile algorithm to improve generalization, and an explicit inductive bias to minimize the effect of noisy labels.

2.1 Class Balanced Pseudo-labels

We utilize our classifier's most likely predicted class during inference as hard ground truth labels (pseudo-labels). Hendrycks and Gimpel (2017) show that using a model's own softmaxed probability values, $max_k\{p(y=k|x)\}$, where k are the classes, x is the input, and y is the predicted class is a reasonable proxy for its expected accuracy. To filter out samples with low maximum probabilities, one can simply threshold the output with some fixed value p_t. As proposed by Zou et al. (2018), a separate threshold $p_t(k_{max})$ for each class, where k_{max} is the class with the maximum predicted probability, works better by reducing skewing in favour of easier classes.

In CBST $p_t(k)$ are automatically selected for each k such that a fixed fraction f of examples of each predicted-class are filtered out from the inference set, i.e.,

$$P^k = \{p(y=k|x) | \text{Argmax}_k p(y=k|x) = k\},$$
$$p_t(k) = \max(\{i \mid \frac{|p > i, p \in P^k|}{|P^k|} <= f\}),$$
$$X_t = \{x \mid max_k\{p(y=k|x)\} > p_t(k)\},$$
$$Y_t = \{k \mid max_k\{p(y=k|x)\} > p_t(k)\}$$

These thresholds $p_t(k)$ can be kept fixed based on the validation set, or can be a running estimate in an online setting. Unlike the original CBST, we do not further normalize the class probabilities with these thresholds, as that led to a drastic reduction in the accuracy of pseudo-label classification. X_t inputs with hard pseudo-labels Y_t are used as a training set to further fine-tune the model, using the Reptile Algorithm below. This approach is also unaffected by a lack of model calibration, as long as the model's accuracy on X_t is acceptably high.

2.2 Reptile Algorithm, but for Single Task

Naively using the confident inferred labels for fine-tuning the model is not optimal due to small size of the test set compared to the train set as well as label noise, lowering generalization, and reducing the gains that can be achieved using the pseudo-labels. Since aligned gradients between samples improve a model's generalization, as shown in Chatterjee (2020) and Fort et al. (2019), we leverage the Reptile Meta-Learning Algorithm to this end. The meta-gradient for the Reptile algorithm contains as a component the gradient for maximizing the inner product between different mini-batches from the same task, as we prove in Section 3.

Algorithm 1: REPTILE + l^2sp

Input: Batches $B = \{b_0, b_1, \ldots, b_n\}$
$W = \theta_{0,0} \leftarrow$ Initial network params
Output: Final fine-tuned θ
for $i \leftarrow 0$ ***to*** $\lfloor n/k \rfloor$ **do**
  for $j \leftarrow 0$ ***to*** $k-1$ **do**
    $\nabla_{\text{inner}} \leftarrow$ grad from $\theta_{i,j}(b_{i*k+j})$
    $\nabla_{\text{LR}} \leftarrow LR_{\text{inner}} * \nabla_{\text{inner}}$
    $l^2sp \leftarrow decay * (\theta_{i,j} - W)$
    $\theta_{i,j+1} \leftarrow \theta_{i,j} - \nabla_{LR} - l^2sp$
  $\nabla_{\text{outer}} \leftarrow (\theta_{i,0} - \theta_{i,k})$
  $\theta_{i+1,0} \leftarrow \theta_{i,0} - LR_{\text{outer}} * \nabla_{\text{outer}}$
return $\theta_{\lfloor n/k \rfloor + 1,0}$

The Reptile Algorithm is a batched First-Order MAML (FO-MAML) Algorithm, originally in-

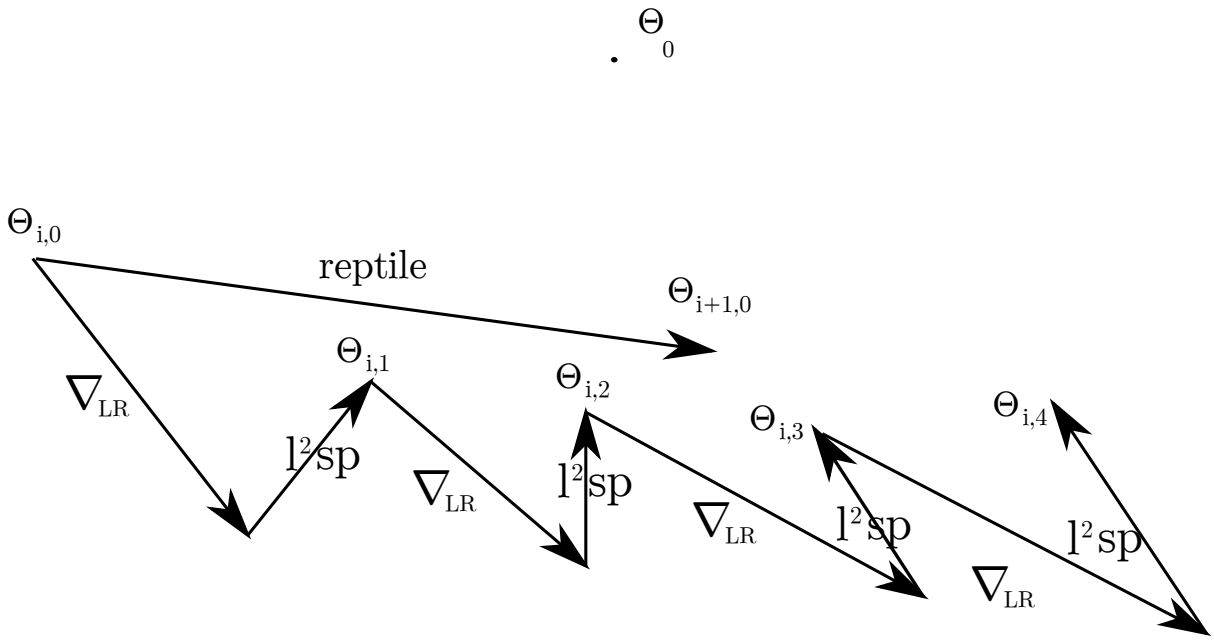

Figure 1: Overview of Reptile with l^2sp update for 4 inner steps.

tended for multi-task meta-learning. We use this algorithm in a single-task setting, as shown in Algorithm 1. The Reptile algorithm consists of $k > 1$ inner steps of standard SGD updates with learning rate LR_{inner}. The difference between original network weights $\theta_{i,0}$ and the final network weights $\theta_{i,k}$ is used as a meta-gradient for SGD for updating the network parameters with a learning rate LR_{outer}, where i is the outer step. The SGD optimizer can be replaced with any other, such as Adam.

The Reptile algorithm for this single task setting is First Order, requiring little extra compute compared to standard optimization, and can be plugged in to any model with ease. Some other multi-task algorithms with Experience-Replay, such as Riemer et al. (2018), may exhibit better learning but are computationally orders of magnitude more expensive and are hence infeasible for large datasets and models.

2.3 Explicit Inductive Bias

While all the models we use employ a weight decay towards 0 in their training phase, given the usually smaller size of the inference set, we regularize the model by biasing the network towards its pre-trained weights instead. For this, we use the l^2sp decay (Li et al., 2018), slowly decaying the model weights between updates towards the initial trained model weights. An example of the update steps involved for $k = 4$ is shown in Fig.1. We conjecture that this will also make the learning more stable to the noisy pseudo-labels.

Some recent works such as Goldblum et al. (2020) also show that standard l^2 weight decay towards 0 may not be ideal and recommend biasing weights towards some model-dependent non-zero norm value instead. l^2sp can be seen as a generalization of the same, while simultaneously taking advantage of the pre-training.

3 Theoretical Analysis

In this section, we provide a theoretical analysis of the meta update of $Reptile + l^2sp$. We generalize the Taylor expansion approach for Reptile as used in (Nichol et al., 2018) to accommodate l^2sp, and show how our approach maximizes the inner product of gradients between different mini-batches.

We consider one set of k inner updates. For $i \in [0, k]$, we define -

$$
\begin{aligned}
\theta_i &= \text{network weights before } i^{th} \text{ step}, \\
b_i &= \text{input batch for } i^{th} \text{ step}, \\
L_i &= \text{loss function corresponding to } b_i, \\
W &= \text{pre-trained network weights for } l^2sp, \\
\beta &= l^2sp \text{ weight decay rate}, \\
\alpha &= LR_{inner}, \\
g_i &= L_i'(\theta_i), && \text{(gradient of } i^{th} batch) \\
\overline{g_i} &= L_i'(\theta_0), && \text{(gradient at initial point)} \\
H_i &= L_i''(\theta_i), && \text{(Hessian of } i^{th} batch)
\end{aligned}
$$

$$\overline{H_i} = L_i''(\theta_0), \qquad \text{(Hessian at initial point)}$$

Then, our update rule is -

$$\theta_i = (1-\beta)\theta_{i-1} - \alpha g_{i-1} + \beta W \tag{1}$$

In the following analysis, to keeps the analysis tractable, we assume both α and β are small and comparable, and ignore terms involving $O(\alpha^2)$, $O(\beta^2)$ and $O(\alpha\beta)$. Using the first order Taylor expansion of g_i, we get -

$$g_i = \overline{g_i} + \overline{H_i}(\theta_i - \theta_0) \tag{2}$$

The following equations can be proved using simple induction on Eq (1) and (2) -

$$\theta_i = \theta_0 + i\beta(W - \theta_0) - \alpha \sum_{j=0}^{i-1} \overline{g_j}, \tag{3}$$

$$g_i = \overline{g_i} + i\beta\overline{H_i}(W - \theta_0) - \alpha\overline{H_i} \sum_{j=0}^{i-1} \overline{g_j}, \tag{4}$$

By summing up the displacements from all variable updates, the expectation of the meta-gradient from Reptile + l^2sp under mini-batch sampling is -

$$\mathbb{E}[-(\theta_k - \theta_0)] = \mathbb{E}[\alpha \sum_{i=0}^{k-1} g_i - \sum_{i=0}^{k-1} \beta(W - \theta_i)]$$

When expanding the terms above with Eq (3) and (4) and simplifying, we get -

$$\begin{aligned}\mathbb{E}[-(\theta_k - \theta_0)] &= c_1\mathbb{E}[\overline{g_i}] + c_2(\theta_0 - W) \\ &-c_3\mathbb{E}[\overline{H_j}\overline{g_i}] - c_4\mathbb{E}[\overline{H_j}(\theta_0 - W)],\end{aligned} \tag{5}$$

where each c_i is a positive constant, dependent on k, α and β.

The first term in R.H.S. of Eq (5) is the gradient which takes us to the minimum of the training problem. For the third term, note that -

$$\begin{aligned}\mathbb{E}[\overline{H_j}\overline{g_i}] = \mathbb{E}[\overline{H_i}\overline{g_j}] &= \frac{1}{2}\mathbb{E}[\overline{H_j}\overline{g_i} + \overline{H_i}\overline{g_j}] \\ &= \frac{1}{2}\mathbb{E}[\frac{\partial}{\partial\theta_0}(\overline{g_i} \cdot \overline{g_j})]\end{aligned}$$

Therefore the third term maximizes the dot product between the gradients of the batches for improved generalization, as in the original Reptile algorithm.

For the second and fourth terms, note that $(\theta_0 - W)$ is the direction of the gradient of the l^2sp, and hence can be interpreted similar to the first and third term, but with training gradients replaced by this l^2 gradient.

Hence, we have shown that the Reptile algorithm maximizing product of gradients for improving generalization holds true in our extension as well.

Corpus	Task	|Train|	|Dev|
BoolQ	QA	9427	3270
CB	NLI	250	57
COPA	QA	400	100
MultiRC	QA	5100	953
ReCoRD	QA	101K	10K
RTE	NLI	2500	278
WiC	WSD	6000	638

Table 1: Description of datasets in SuperGLUE.

Corpus	Model	|Train|	|Test|
MNIST	MLP	60K	10K
ImageNet	ResNet-50	1.2M	50K
SQuAD v2.0	Electra	130K	12K
Ubuntu Diag.	BERT	1M	18K
NewsQA	BERT-trans	97K	5.4K

Table 2: Description of NLP and image datasets. For SQuAD and ImageNet, column 4 refers to validation. Bert-trans is as described in Section 5.3.

4 Experimental Setup

4.1 Benchmark Datasets

SuperGLUE A popular NLP benchmark, which attempts to test various capabilities of language understanding. It itself consists of 8 datasets - Boolean Questions (Clark et al., 2019), Commitment Bank (De Marneffe et al., 2019), Choice of Possible Alternative (Gordon et al., 2012), Multi-Sentence Reading Comprehension (Khashabi et al., 2018), Reading Comprehension with Commonsense Reasoning (Zhang et al., 2018), Recognizing Textual Entailment (a combination of datasets from Dagan et al., 2005; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009; Poliak et al., 2018), Word-in-Context (Pilehvar and Camacho-Collados, 2019) and Winograd Schema Challenge (Levesque, 2011).

SQuAD v2.0 A popular span-style QA dataset, consisting of passages from Wikipedia, with questions and corresponding answer spans and unanswerable questions.

Ubuntu Dialog Corpus v2.0 A large-scale corpus of multi-turn conversations mined from Ubuntu IRC chat logs, and the task is to select the best response given a list of possible distractor responses.

NewsQA A span-style QA dataset, consisting of crowd-sourced questions and answers on CNN news articles, along with unanswerable questions.

Model	Params	Speed
Electra-large-cased	340M	8
BERT-large-cased	340M	8
BERT-base-uncased	110M	36
ResNet-50	23M	146
MLP (128H, 2L)	120K	~1M

Table 3: Models, number of network parameters, and training speeds in examples/second on a V100 GPU.

MNIST An image classification dataset of 28x28 scans of handwritten digits. While the dataset has long been *solved*, it nevertheless serves as a useful dataset to compare simpler architectures.

ImageNet A large-scale dataset for image classification, consisting of 1.2M training samples along with their corresponding class labels.

4.2 Models

BERT BERT is a transformer (Vaswani et al., 2017) model, and its derivative models are the backbone of most state-of-the-art models in NLP. We use the official implementation and pre-trained models of BERT-large-cased for SuperGLUE tasks, and BERT-base-uncased for Ubuntu Dialog Corpus, NewsQA, and for our ablation tests on SQuAD.

Electra Electra is a BERT-derived state-of-the-art model in many NLP tasks, with a discriminative pre-training task. We use the official pre-trained Electra-large model, and we implement our own classifier for SQuAD v2.0.

ResNet Residual blocks and their variants are the backbone of most image classification models today. We use Tensorflow Model Garden's implementation and pre-trained ResNet-50 for ImageNet.

MLP While models made of only simple Multi Layer Perceptrons have largely fallen out of favour, fully connected layers are often a part of larger architectures. We use an MLP with 2 Layers and 128 Hidden units as the model for MNIST.

4.3 Implementation Details

Fine-tuning on inference data is extremely quick as our method is first order, taking less than 15 minutes on a V100 for all datasets except ReCoRD and Ubuntu-Dialog, for which it takes a few hours. We use the Adam optimizer, and we disable our model's l^2 weight decay, if any. Batch-norm variables, if any, are also kept fixed.

Corpus	Metric	BERT	BERT + ours
BoolQ	Acc	76.4	**76.6** $\pm$ 0.01
CB	F1	88.1	**89.4** $\pm$ 0.01
	Acc	91.1	**92.9** $\pm$ 0.01
COPA	Acc	71.0	**72.0** $\pm$ 0.01
MultiRC	F1a	69.5	**70.0** $\pm$ 0.01
	EM	26.4	**26.8** $\pm$ 0.01
ReCoRD	F1	72.5	**73.0** $\pm$ 0.09
	EM	71.8	**72.4** $\pm$ 0.03
RTE	Acc	74.0	**75.1** $\pm$ 0.01
WiC	Acc	73.8	**74.3** $\pm$ 0.02

Table 4: Results on the validation set of SuperGLUE benchmark dataset, with Bert-large-cased model.

For each dataset, we train one model on the training set, followed by five runs on the pseudo-labeled thresholded inference set with varying seeds, and report the mean and standard deviation of the scores. As the test set for SuperGLUE and SQuAD are hidden, we provide results on the development set instead.

All default/official model hyper-parameters were used for each model/dataset, which can be found in their official source codes linked in the supplemental material, except we use $1e^{-5}$ as LR for Electra as we observed divergence with standard LR. We linearly decay LR except in the online case, where it is kept fixed. The hyper-parameters for Reptile and l^2sp are provided in the supplemental material. A reasonable set of hyper-params, that works across a range of datasets and models we tested, is 0.01 for LR_{outer}, 4 for $inner\_step$, and 0.1 for l^2sp, while LR_{inner} depends on the original model's LR. RTE, BoolQ, and WiC filter out f as 70% of data, while all other datasets filter 50%.

5 Results

5.1 Results on SuperGLUE benchmark

As shown in Table 4, our method consistently improves the performance on all the tasks in SuperGLUE, with very little extra compute, with upto 1.8 increase in accuracy. The gains tend to be larger on smaller datasets, but we observe significant improvement even with the largest task ReCoRD, with over 100K examples.

5.2 Results on other NLP datasets

Our method achieves gains of $0.68/0.72$ F1/EM on SQuAD v2.0 with BERT-base, as shown in Table 7. Even when using a state-of-the-art Electra model

Method	F1	EM
BERT	76.14	73.14
BERT + CBST (Zou et al., 2018)	76.22 ± 0.02	73.35 ± 0.02
BERT + Disagreement (Malach and Shalev-Shwartz, 2017)	76.23 ± 0.03	73.27 ± 0.06
BERT + Uncertainty Estimation (Zheng and Yang, 2020)	76.25 ± 0.05	73.29 ± 0.04
BERT + Mutual Mean-Teaching (Ge et al., 2020)	76.28 ± 0.04	73.23 ± 0.05
BERT + Co-teaching (Han et al., 2018)	76.28 ± 0.02	73.29 ± 0.04
BERT + Ours	**76.82** ± 0.04	**73.86** ± 0.04

Table 5: Comparison of our method to existing method, on SQuAD v2.0 corpus, using BERT-base-uncased.

Corpus	Metric	Base	Base + ours
MNIST	Acc	98.11	**98.38** ± 0.02
ImageNet	Acc	76.53	**76.69** ± 0.01
SQuAD v2.0	F1	90.13	**90.25** ± 0.01
	EM	87.44	**87.67** ± 0.01
Ubuntu Diag.	R10@1	76.79	**76.89** ± 0.01
NewsQA	F1	44.79	**49.36** ± 0.05
	EM	32.48	**38.71** ± 0.11
NewsQA (online)	F1	44.79	**47.49** ± 0.01
	EM	32.48	**34.11** ± 0.04

Table 6: Results on other NLP and Image datasets.

with SQuAD, we still observe consistent improvements in performance, as shown in Table 6. Even in the presence of large training-set sizes such as that of Ubuntu Dialog Corpus v2.0 with 1M training samples, we still observe consistent increase in performance with the BERT model.

5.3 Results in a Transfer Learning Setting

We also evaluate our approach in a transfer-learning setting on NewsQA, using a BERT-base-uncased model, which was pre-trained on SQuAD v2.0, by self-training on NewsQA train set, followed by evaluation on the test set. Our approach is especially effective in this setting, out-performing the original model by $4.57/6.23$ F1/EM respectively, as shown in Table 6. This experiment demonstrates that our approach is effective for unsupervised domain adaptation to a target domain even in the absence of source domain data.

5.4 Results on Image Classification

To demonstrate that our method also works in non-NLP domains, on ImageNet with ResNet-50, we report an increase in accuracy of 0.16. On MNIST dataset, the improvement in accuracy of our simple MLP model is 0.27.

5.5 Comparison with Existing Methods

We compare our method with several existing approaches for Self-Training, Zou et al. (2018), Malach and Shalev-Shwartz (2017), Malach and Shalev-Shwartz (2017), Ge et al. (2020) and Han et al. (2018), on SQuAD v2.0 dataset.

As shown in Table 5 our method greatly outperforms the existing approaches, giving 4 to 5 times the relative improvement compared to other methods, improving performance by $0.68/0.82$ F1/EM compared to $0.14/0.15$ F1/EM of the best performing existing approach.

5.6 Online Variant

Our approach can also be used effectively without any modifications in an online setting, where the model keeps learning continuously as inference data is fed to the model. We use a trained model to make predictions on the input inference data, and at the same time, we use the model's predictions to finetune the model. For this kind of learning, we use a constant learning rate, as the total size of inference data is unavailable. As a baseline, we use BERT + CBST (trained on SQuAD-v2.0 data) with a constant learning rate. BERT + CBST + Reptile + l^2sp (Online) clearly outperforms BERT + CBST (Online) by $0.38/0.37$ F1/EM as shown in Table 7.

We also compare the performance of our method when running in online mode for a long time on NewsQA dataset, as shown in Table 6. The performance improvement is not as large as with decreasing LR, but still results in significant performance improvements of $2.70/1.63$ F1/EM, respectively.

6 Ablation Studies

We conduct extensive ablation studies to test the effectiveness of all parts of our approach. We perform these ablations on SQuAD v2.0 with BERT-base model.

Method	F1	EM
BERT	76.14	73.14
BERT + CBST	76.22 ± 0.02	73.35 ± 0.02
BERT + CBST + l^2sp	76.24 ± 0.04	73.49 ± 0.04
BERT + CBST + Reptile	76.61 ± 0.02	73.60 ± 0.03
BERT + Reptile + l^2sp	76.47 ± 0.07	73.63 ± 0.08
BERT + CBST + Reptile + l^2sp	**76.82** ± 0.04	**73.86** ± 0.04
BERT + CBST (Online)	76.20 ± 0.01	73.28 ± 0.01
BERT + CBST + Reptile + l^2sp (Online)	**76.58** ± 0.02	**73.65** ± 0.01

Table 7: Ablation Study of our method on SQuAD v2.0 corpus, using the BERT-base-uncased model.

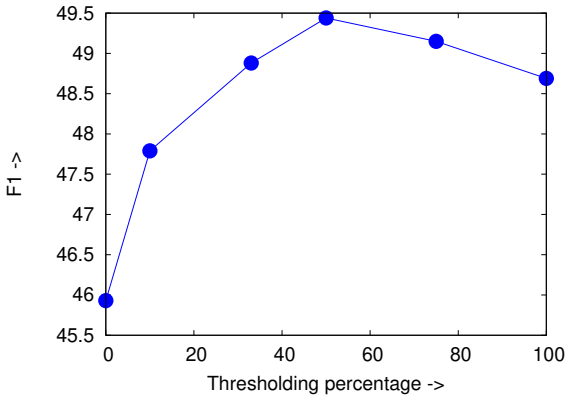

Figure 2: Ablation study of varying the thresholding percentage on NewsQA. The Y axis is F1 score, the X axis is the percentage of data left after thresholding.

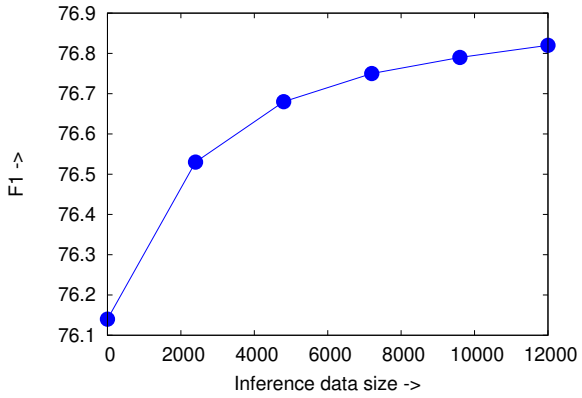

Figure 3: Effect of varying total size of Inference data on our method on SQuAD v2.0. The Y axis is F1 score, the X axis is the total amount of Inference data used.

6.1 Thresholding

In Table 7, we compare using CBST thresholding of model outputs to fine-tune the model vs. using all the data. Using CBST + Reptile + l^2sp increases scores by $0.35/0.23$ F1/EM respectively compared to using all the pseudo-labels with Reptile + l^2sp.

We further study the effect of the thresholding fraction f used to select the subset of confident data. We use the pre-trained Bert-base-uncased model, self-trained on the training set of NewsQA data with pseudo-labels, while varying f, and then evaluate on the dev set. As can be seen in Fig.2, the optimal value for thresholding is around 50%, decreasing slowly as more data (but with less confident labels) is used, and decreasing more sharply as the total filtered data used decreases.

6.2 Reptile Algorithm

Compared to using just CBST, using the Reptile Algorithm to finetune results in more performance gains of $0.58/0.37$ F1/EM, as we can see in Table 7. This effect persists irrespective of whether l^2sp or the model's default weight decay towards 0 is used. This demonstrates that the increased generalization from Reptile's meta-gradients is indeed effective in increasing model performance and robustness.

We also conduct an ablation study on the choice of number of inner steps k on the performance of our model. As shows in Table 8, the number of inner updates does not have a major impact on the results, but we advise it be kept less than or equal to 4 as higher inner steps reduce the number of outer updates (as the total number of epochs is kept constant).

6.3 Inductive Bias towards pre-trained weights

We can also see in Table 7 that l^2sp is indeed effective, and by simply biasing the model towards the pre-trained weights, we can achieve better results. This effect becomes more pronounced when the Reptile algorithm is used, with $0.21/0.26$ F1/EM improvement of CBST + Reptile + l^2sp compared to CBST + Reptile.

We also conduct an ablation study on the choice of this bias, by transfer learning on NewsQA

Num Updates	F1	EM
Baseline	76.24 ± 0.04	73.49 ± 0.04
2	76.79 ± 0.02	73.87 ± 0.02
4	76.82 ± 0.01	73.86 ± 0.02
6	76.80 ± 0.01	73.74 ± 0.02
8	76.71 ± 0.01	73.68 ± 0.03

Table 8: Ablation of choice of hyper-parameter number of inner steps k for our method CBST + Reptile + l^2sp on SQuAD with BERT-base.

l^2sp decay	F1	EM
Baseline	76.14	73.14
0	76.38 ± 0.00	69.87 ± 0.21
6e-4	76.54 ± 0.02	70.74 ± 0.15
2e-3	76.10 ± 0.01	73.74 ± 0.12

Table 9: Performance of BERT-base on SQuAD, after self-training on NewsQA with transfer learning with our method, for varying choices of hyper-parameter decay for l^2sp.

dataset using our method with a model trained on SQuAD, and measuring the performance on SQuAD thereafter. As shows in Table 9, l^2sp prevents the model from forgetting its performance on SQuAD. However, higher values prevent it from improving its performance on the original squad by minimizing learning on NewsQA.

6.4 Effect of Inference Data Size

In Figure 3, we vary the amount of inference data available for our model to learn from, by training a BERT-base model on varying sizes of pseudo-labelled SQuAD v2.0 dev set, while keeping f fixed at 50%. The largest increase occurs early on in the training. However, even on using the full dev set, the performance keeps improving, giving an improvement in F1 of 0.68.

7 Related Works

7.1 Pseudo-labeling

Lee (2013) proposed a simple and efficient method of semi-supervised learning for deep neural networks, in which the proposed network is trained in a supervised fashion with labeled and unlabeled data simultaneously, using *pseudo-labels* created by selecting the classes which have the highest predicted probabilities as ground truth labels for unlabeled data. CBST (Zou et al., 2018) used different thresholds for pseudo-labels of different classes. Mutual Mean-teaching (Ge et al., 2020) used a moving average of two separate classifiers to refine pseudo-labels. Zheng and Yang (2020) used KL-divergence between two classifiers as a measure of classifier variance to filter incorrect pseudo-labels. Pseudo-labels and similar self-supervised techniques have grown increasingly popular, particularly when used in conjunction with extremely large unlabelled data, and was used by Noisy-Student (Xie et al., 2019) recently to achieve state-of-the-art performance on image classification.

7.2 Dynamic Evaluation

Adaptive language modelling has a long history, such as Kuhn (1988), and caching based models have resulted in improved performances over state-of-the-art, such as Merity et al. (2018). Krause et al. (2018) proposed to use dynamic evaluation adapted to recent history via a gradient descent based mechanism. However, their approach is limited to language modelling, where ground-truth labels are trivially available during inference, and does not generalize to standard classification setting.

Rahman et al. (2019) also used pseudo-labels during inference to learn, but differently from our paper, they primarily focus on a transductive zero-shot detection, and do not use our proposed meta-learning and inductive bias. Kim et al. (2019) also proposed to use pseudo-labels to learn during evaluation, but require changes to the model's training phase. Su et al. (2016) also used pseudo-labels on inference data to improve model performance, but their contributions are primarily focused on adapting Self-Training to unbalanced classes. Dynamic evaluation can be considered a form of Fast-weights (Ba et al., 2016), which unlike our approach, requires changes during the training phase.

7.3 Generic Methods for Noisy Labels

Loss correction methods such as Patrini et al. (2017) model the noise transition matrix. Other approaches try to directly correct the noisy labels, such as Veit et al. (2017), but require access to a clean set. Others directly modify the loss function to make it more stable to noisy labels, such as Generalized Cross Entropy (Zhang and Sabuncu, 2018). Other approaches, most related to our approach, refine the training strategy, such as Co-teaching (Han et al., 2018) or Mutual Mean-teaching, using two classifiers to select the data for each other.

7.4 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation methods often use Adversarial Methods, such as Jiang et al. (2020), to distinguish between source and target domains. Distance based methods, such as Chen et al. (2019), aim to minimize the distribution discrepancy across different domains. Other methods such as Courty et al. (2017) rely on optimal transport between source and target domains. These methods often need access to source domain data, or modify the original model or training procedure.

7.5 Meta-Learning for Transfer Learning

Algorithms that rely on Fisher/Hessian matrices have been proposed to improve transfer learning, such as Kirkpatrick et al. (2016). Nichol et al. (2018) proposed using batched FO-MAML during training to learn better weight initialization values. Often these algorithms also use some form of Experience Replay, where saved/cached examples from previous tasks are replayed to prevent the model from forgetting. Riemer et al. (2018) proposed Meta-Experience Replay (MER), exploiting a trade-off between transfer and interference by enforcing gradient alignment across examples.

8 Conclusion

We propose a method for self-supervised learning for any classifier model during inference using the model's own predictions, adapting Reptile algorithm from meta-learning and an inductive bias for maintaining generalization while improving performance. We demonstrate the effectiveness of our method on a wide range of tasks, including SuperGLUE benchmark, question answering on SQuAD v2.0 and NewsQA, response selection on Ubuntu Dialog Corpus v2.0, and image classification on ImageNet and MNIST. Our approach consistently improves the performance of standard backbones such as BERT, Electra, and ResNet. Our method is effective for improving the performance of neural models without any changes to the underlying models, their training, or access to training data, requires minimum extra compute, and is also effective in online and transfer-learning settings.

Acknowledgments

We want to thank Mr. Wonho Ryu and Mr. Haejun Lee of Samsung Research, Seoul, Korea, for their guidance and leadership. We would also like to thank all the reviewers for their valuable feedback and suggestions.

References

Jimmy Ba, Geoffrey E. Hinton, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4331–4339.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

Satrajit Chatterjee. 2020. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. 2019. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3296–3303. AAAI Press.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3730–3739.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23-2, pages 107–124.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Li Deng. 2012. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.*, 29(6):141–142.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2185–2194. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13042–13054.

Stanislav Fort, Pawel Krzysztof Nowak, and Srini Narayanan. 2019. Stiffness: A new perspective on generalization in neural networks. *CoRR*, abs/1901.09491.

Yixiao Ge, Dapeng Chen, and Hongsheng Li. 2020. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June 28-29, 2007*, pages 1–9. Association for Computational Linguistics.

Micah Goldblum, Jonas Geiping, Avi Schwarzschild, Michael Moeller, and Tom Goldstein. 2020. Truth or backpropaganda? an empirical investigation of deep learning theory. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Andrew S. Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 394–398. The Association for Computer Linguistics.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montr 'e al, Canada*, pages 8536–8546.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. 2020. Implicit class-conditioned do-

main alignment for unsupervised domain adaptation. *CoRR*, abs/2006.04996.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 252–262. Association for Computational Linguistics.

Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. 2019. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6091–6100. IEEE.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2018. Dynamic evaluation of neural sequence models. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2771–2780. PMLR.

Roland Kuhn. 1988. Speech recognition and the frequency of recently used words: a modified markov model for natural language. In *Proceedings of the 12th International Conference on Computational Linguistics, COLING '88, Budapest, Hungary, August 22-27, 1988*, pages 348–350. John von Neumann Society for Computing Sciences, Budapest.

Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2.

Hector J. Levesque. 2011. The winograd schema challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI.

Xuhong Li, Yves Grandvalet, and Franck Davoine. 2018. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2830–2839. PMLR.

Ryan Thomas Lowe, Nissan Pow, Iulian Vlad Serban, Laurent Charlin, Chia-Wei Liu, and Joelle Pineau. 2017. Training end-to-end dialogue systems with the ubuntu dialogue corpus. *Dialogue Discourse*, 8(1):31–65.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling "when to update" from "how to update". In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 960–970.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2233–2241. IEEE Computer Society.

Mohammad Taher Pilehvar and José Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1267–1273. Association for Computational Linguistics.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 67–81. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper.pdf*.

Shafin Rahman, Salman Khan, and Nick Barnes. 2019. Transductive learning for zero-shot object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6081–6090. IEEE.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *CoRR*, abs/1810.11910.

Kyungmin Su, W. David Hairston, and Kay A. Robbins. 2016. Adaptive thresholding and reweighting to improve domain transfer learning for unbalanced data with applications to EEG imbalance. In *15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, Anaheim, CA, USA, December 18-20, 2016*, pages 320–325. IEEE Computer Society.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6575–6583. IEEE Computer Society.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 3261–3275.

Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Self-training with noisy student improves imagenet classification. *CoRR*, abs/1911.04252.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 8792–8802.

Zhedong Zheng and Yi Yang. 2020. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *CoRR*, abs / 2003.03773.

Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 297–313. Springer.

A Statistical Significance of Results

For each dataset, we train one model on the training set, followed by five runs on the pseudo-labeled thresholded test set with varying seeds, and report the mean and standard deviation of the scores. Smaller datasets in SuperGLUE are known to have significant variation between multiple runs when fine-tuning BERT model, however, most of this variation comes from random initialization of the classification layer. In our experiments, as the model has already been fine-tuned on the train set, the only variation between runs is the order of input data. This results in an extremely small variation in score between different runs, much smaller than the performance gains observed, making the improvements statistically significant.

A.1 Significance tests

We provide below in Table 10 and Table 12 the P-values for one-sample T-test for the Table 6 and Table 7, with the null hypothesis that the scores of our results have the same mean as the baseline. Our results are significant at 99% confidence in all settings.

B Improved Generalization

The NewsQA results in Table 6 are scores on the test set, while the model was self-trained on the train set. The scores indicate that, the model does not over-fit while self-training as our approach significantly improves the scores on the test set.

Corpus	$p-value$
SQuAD v2.0 Electra (F1)	3e-5
Ubuntu Dialog v2.0	5e-5
NewsQA (F1)	1e-9
ImageNet	4e-7

Table 10: P-values for one-sample T-test with the null hypothesis for Table 6.

As a further test of improved generalization, we split the squad dev set in two equal halves, performed our self-training on one half, and evaluated on the other half. Scores in Table 11 show, self-training on one half improved generalization on the other.

Model/Approach	$F1$	EM
BERT	76.28	73.32
BERT+ours	76.40	73.45

Table 11: Results on one half of the squad-dev set.

C Links to Source code

For SuperGLUE, we use the Official Implementation for BERT available at https://github.com/nyu-mll/jiant, along with the default pre-trained models. For ablation tests on SQuAD, we used the official implementation and pre-trained models at https://github.com/google-research/bert. For Ubuntu Dialog, we used the same pre-trained models, but we implement our own classifier. For Electra, we used the pre-trained models from https://github.com/google-research/electra. For ResNet-50, we used Tensorflow Model Garden's official implementation as well as pre-trained model on ImageNet at https://github.com/tensorflow/models/tree/r1.13.0/official/resnet. For MNIST, we implemented our own MLP following https://www.tensorflow.org/datasets/keras_example. The Reptile+l^2sp Optimizer is trivial to implement in all of the above models following the pseudo-code from the main paper, by modifying the *Optimizer* class used for each of the models.

D Links to download data

SuperGLUE can be downloaded from https://super.gluebenchmark.com/. SQuAD v2.0 can be downloaded from https://rajpurkar.github.io/SQuAD-explorer/. Ubuntu Dialog Corpus can be generated using https://github.com/rkadlec/ubuntu-ranking-dataset-creator. ImageNet can be downloaded from http://image-net.org. MNIST can be downloaded from http://yann.lecun.com/exdb/mnist/. NewsQA can be downloaded from https://www.microsoft.com/en-us/research/project/newsqa-dataset/.

E Corresponding Validation set results for Test Set

Our only reported test scores are on MNIST, NewsQA, and Ubuntu Dialog Corpus. For MNIST, there is no official validation set. For Ubuntu Dialog Corpus, the validation score of our model is 76.44 Recall10@1, and for NewsQA, it is 49.44±0.04 F1 and 39.26±0.15 EM, respectively.

F Hyper-parameters of our approach

The hyper-parameter search bounds were chosen based on heuristic manual estimates, primarily considering the product of the LR_{inner} and LR_{outer}, compared to the model's native LR when the fraction of training steps left equals the ratio of the size of the training set to the size of the filtered inference set. Each set of hyper-parameters was run three times, and the hyper-parameter search was run in a grid. We list the hyper-parameters of our $Reptile + l^2sp$ approach in Table 13.

G Dataset descriptions

G.1 SuperGLUE

BoolQ Boolean Questions, a Question Answering (QA) dataset with short passages and yes/no questions, with data from Wikipedia and Google search engine queries.

CB Commitment Bank, consisting of passages with labels for *commitment* of speakers of clauses to said clause, framed as three-class NLI, with data from WSJ, British National Corpus and Switchboard. Evaluated with unweighted average F1 and accuracy.

COPA Choice of Possible Alternative, a dataset to classify the cause/effect of a given premise from two alternatives, with fully handcrafted data.

MultiRC Multi-Sentence Reading Comprehension, a QA dataset, with a list of multiple-choice possible answers for each question to a

Approaches (F1)	$p-value$
CBST+Reptile-vs-CBST	3e-7
CBST+Reptile+l2sp-vs-CBST+Reptile	1e-5
CBST+Reptile+l2sp(online)-vs-CBST(online)	2e-6
CBST+Reptile+l2sp-vs-Baseline(One-sample)	1e-6

Table 12: P-values for one-sample T-test with the null hypothesis for Table 7.

Corpus	LR_{outer}	LR_{inner}	$inner\_steps$	l^2sp
SQuAD v2.0 Bert	[**1e-2**, 3e-3]	1e-5	4	1e-2
SQuAD v2.0 Electra	3e-3	[**1e-5**, 1e-6]	4	[1e-2, **5e-2**]
Ubuntu Dialog v2.0	[1e-2, **3e-3**]	[**1e-5**, 1e-6]	4	[5e-2, **1e-1**]
NewsQA	1e-2	1e-5	4	[**2e-3**, 1e-2]
ImageNet	[3e-2, **1e-2**]	1e-4	4	[**1e-1**, 5e-1]
MNIST	1e-1	1e-4	4	1e-1
BoolQ	1e-2	1e-6	4	0.4
CB	1e-2	1e-6	4	0.4
COPA	1e-2	1e-6	2	0.4
MultiRC	1e-1	1e-4	2	0.4
ReCoRD	1e-2	1e-6	4	0.4
RTE	1e-2	1e-6	2	0.4
WiC	1e-2	1e-6	2	0.4

Table 13: Hyper-parameters for all Datasets. Best performing parameters are in **bold**.

paragraph. Evaluated with F1 over all answer-options($F1_a$), and exact match of each question's set of answers(EM).

ReCoRD Reading Comprehension with Commonsense Reasoning, a QA dataset consisting of articles and Cloze-style questions with a masked entity, scored on predicting the masked entity from the entities in the article, with data from CNN and Daily Mail. Scored with token-level F1 and EM.

RTE Recognizing Textual Entailment, as binary classification of *entailment* or *not entailment*, with data from Wikipedia and news.

WiC Word-in-Context, a word sense disambiguation (WSD) dataset, tasked with binary classification of sentence pairs based on the sense of a common polysemous word. Data is from WordNet and Wiktionary.

WSC Winograd Schema Challenge, a coreference resolution task on resolving pronouns to a list of noun phrases. As the models we tested only predicted the majority class, we omit this dataset.

G.2 SQuAD v2.0

The Stanford Question Answering Dataset v2.0 is a popular span-style QA dataset, consisting of passages from Wikipedia, labelled by annotators for questions on the passages and corresponding answer spans, along with unanswerable questions as well. This dataset is evaluated with F1 and EM scores of predicted answer spans.

G.3 Ubuntu Dialog Corpus v2.0

The Ubuntu Dialog Corpus is a large-scale corpus of multi-turn real human conversations mined from Ubuntu IRC chat logs, with only two participants per conversation. Each conversation is annotated with the next utterance (response) following the conversation, and the task is to select the best response given a list of possible distractor responses. The dataset is evaluated with Recall score of picking the correct response out of 10 possible responses, $Recall10@1$.

G.4 NewsQA

NewsQA is a span-style QA dataset, consisting of crowd-sourced questions on CNN news articles and their corresponding answer spans, along with unanswerable questions. This datasets is evaluated with F1 and EM scores of predicted answer spans.

G.5 MNIST

MNIST is a popular image classification dataset, consisting of normalized and anti-aliased 28x28 scans of handwritten numerical digits. While the dataset has long been *solved*, it nevertheless serves as a useful dataset to compare simpler architectures.

G.6 ImageNet

ImageNet is a large-scale dataset for image classification, consisting of 1.2M training samples along with their corresponding class labels. It is often the de-facto dataset when comparing Image Classification models.

H Expected validation performance

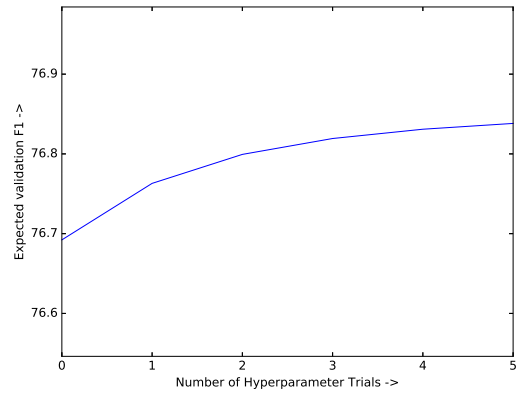

Figure 4: Expected Validation performance of our hyper-parameter searches, for SQuAD dataset with BERT-base model.

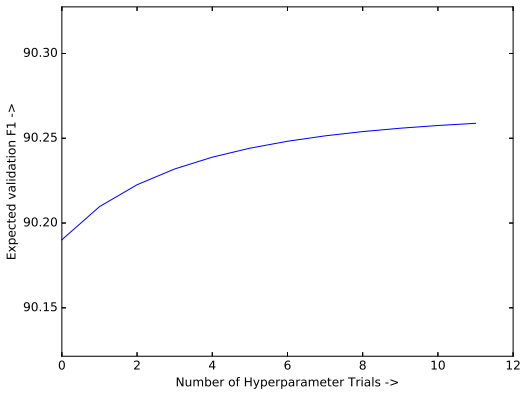

Figure 5: Expected Validation performance of our hyper-parameter searches, for SQuAD dataset with Electra-large model.

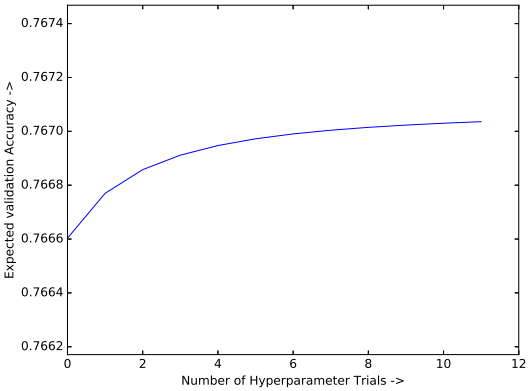

Figure 6: Expected Validation performance of our hyper-parameter searches, for ImageNet dataset with ResNet-50 model.

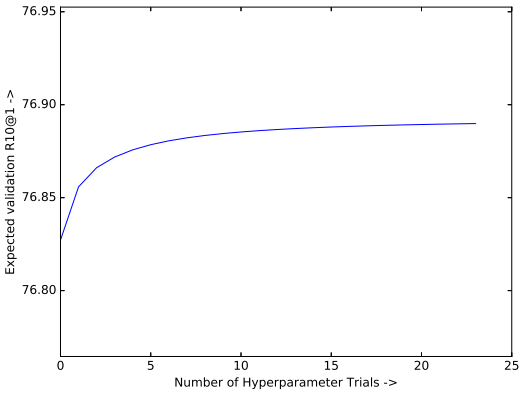

Figure 7: Expected Validation performance of our hyper-parameter searches, for Ubuntu dialog corpus with BERT-base model.

We provide the expected validation performance for all the datasets we ran hyper-parameter searches on, as described in (Dodge et al., 2019).