A Probabilistic Framework for Lifelong Test-Time Adaptation

Dhanajit Brahma  
Indian Institute of Technology Kanpur  
dhanajit@cse.iitk.ac.in

Piyush Rai  
Indian Institute of Technology Kanpur  
piyush@cse.iitk.ac.in

Abstract

Test-time adaptation (TTA) is the problem of updating a pre-trained source model at inference time given test input(s) from a different target domain. Most existing TTA approaches assume the setting in which the target domain is stationary, i.e., all the test inputs come from a single target domain. However, in many practical settings, the test input distribution might exhibit a lifelong/continual shift over time. Moreover, existing TTA approaches also lack the ability to provide reliable uncertainty estimates, which is crucial when distribution shifts occur between the source and target domain. To address these issues, we present PETAL (Probabilistic lifelong Test-time Adaptation with self-training prior), which solves lifelong TTA using a probabilistic approach, and naturally results in (1) a student-teacher framework, where the teacher model is an exponential moving average of the student model, and (2) regularizing the model updates at inference time using the source model as a regularizer. To prevent model drift in the lifelong/continual TTA setting, we also propose a data-driven parameter restoration technique which contributes to reducing the error accumulation and maintaining the knowledge of recent domains by restoring only the irrelevant parameters. In terms of predictive error rate as well as uncertainty based metrics such as Brier score and negative log-likelihood, our method achieves better results than the current state-of-the-art for online lifelong test-time adaptation across various benchmarks, such as CIFAR-10C, CIFAR-100C, ImageNetC, and ImageNet3DCC datasets. The source code for our approach is accessible at https://github.com/dhanajitb/petal.

1. Introduction

Deep learning models exhibit excellent performance in settings where the model is evaluated on data from the same distribution as the training data. However, the performance of such models degrades drastically when the distribution of the test inputs at inference time is different from the distribution of the train data (source) [11, 16, 36]. Thus, there is a need to robustify the network to handle such scenarios. A particularly challenging setting is when we do not have any labeled target domain data to finetune the source model, and unsupervised adaptation must happen at test time when the unlabeled test inputs arrive. This problem is known as test-time adaptation (TTA) [28, 35, 38]. Moreover, due to the difficulty of training a single model to be robust to all potential distribution changes at test time, standard fine-tuning is infeasible, and TTA becomes necessary. Another challenge in TTA is that the source domain training data may no longer be available due to privacy/storage requirements, and we only have access to the source pre-trained model.

Current approaches addressing the problem of TTA [28, 35, 38, 41] are based on techniques like self-training based pseudo-labeling or entropy minimization in order to enhance performance under distribution shift during testing. One crucial challenge faced by existing TTA methods is that real-world machine learning systems work in non-stationary and continually changing environments. Even though the self-training based approaches perform well when test inputs are from a different domain but all still i.i.d., it has been found that the performance is unstable when target test inputs come from a continually changing environment [32]. Thus, it becomes necessary to perform test-time adaptation in a continual manner.

Such a setting is challenging because the continual adaptation of the model in the long term makes it more difficult to preserve knowledge about the source domain. Continually changing test distribution causes pseudo-labels to become noisier and miscalibrated [10] over time, leading to error accumulation [3] which is more likely to occur if early predictions are incorrect. When adapting to new test input, the model tends to forget source domain knowledge, triggering catastrophic forgetting [27, 31, 33]. Moreover, existing TTA methods do not account for model/predictive uncertainty, which can result in miscalibrated predictions.

Recently, [41] proposed CoTTA, an approach to address the continual/lifelong TTA setting using a stochastic parameter reset mechanism to prevent forgetting. Their reset mechanism however is based on randomly choosing a subset of weights to reset and is not data-driven. Moreover,
their method does not take into account model/predictive uncertainty and is therefore susceptible to overconfident and miscalibrated predictions.

To improve upon these challenges of continual/lifelong TTA, we propose a principled, probabilistic framework for lifelong TTA. Our framework (shown in Fig. 1 (Right)) constructs a posterior distribution over the source model weights and a data-dependent prior which results in a self-training based cross-entropy loss, with a regularizer term in the learning objective. This regularizer arises from terms corresponding to the posterior, which incorporates knowledge of source (training) domain data.

Moreover, our framework also offers a probabilistic perspective and justification to the recently proposed CoTTA [41] approach, which arises as a special case of our probabilistic framework. In particular, only considering the data-driven prior in our approach without the regularizer term, corresponds to the student-teacher based cross-entropy loss used in CoTTA. Further, to improve upon the stochastic restore used by [41], we present a data-driven parameter restoration based on Fisher Information Matrix (FIM). In terms of improving accuracy and enhancing calibration during distribution shift, our approach surpasses existing approaches in various benchmarks.

Main Contributions

1. From a probabilistic perspective, we arrive at the student-teacher training framework in our proposed Probabilistic liFElong Test-time Adaptation with seLf-training prior (PETAL) approach. Inspired from the self-training framework [19, 42], the teacher model is the exponential moving average of the student model, as depicted in Fig. 1 (Right).

2. The student-teacher cross-entropy loss with a regularizer term corresponding to posterior of source domain data naturally emerges in the probabilistic formulation.

3. We propose a data-driven parameter restoration based on Fisher Information Matrix (FIM) to handle error accumulation and catastrophic forgetting.

2. Problem Setup and Background

In this section, we define the notation used, the problem setup of lifelong/continual TTA, and the basic probabilistic framework on which our approach is based.

2.1. Problem Setup

Let \( x \) denote the inputs sampled i.i.d from a generative model having parameters \( \psi \), and \( p(y|x, \theta) \), having parameters \( \theta \), be the conditional distribution from which the corresponding labels are sampled. We denote the prior distributions of \( \theta \) and \( \psi \) using \( p(\theta) \) and \( p(\psi) \), respectively.

A typical test-time adaptation setting is as follows: We have a model with parameters \( \theta_0 \) trained on the source training data \( \mathcal{X} = \{x_n, y_n\}_{n=1}^N \). The aim is to adapt \( \theta_0 \) and perform well on the test inputs \( \mathcal{U}^d = \{x_m^d\}_{m=1}^{M_d} \) from an unlabeled target domain \( d \). In case of multiple target domains, the adaptation happens for each target domain separately: \( \theta_0 \rightarrow \theta_d \).
In lifelong/continual test-time adaptation, unlabeled test inputs from \( D \) different target domains \( \{\mathcal{U}^d : d = 1, \ldots, D\} \) arrive continually, and, thus, the model can utilize only the data available for the current target domain. Note that there is no information available to the learner about the change in domain. At step \( t \), for a test input \( x_m \), we make predictions using \( p(y_m|x_m, \theta_t) \) as well as adapt the parameters for future steps, i.e., \( \theta_t \rightarrow \theta_{t+1} \). Note that there is a continual domain shift in the data distribution of \( x_m \). Moreover, the model evaluation is performed based on predictions obtained online. Fig. 1 (Left) depicts the online lifelong TTA problem setup.

2.2. The Underlying Probabilistic Framework

In this section, we review standard probabilistic discriminative models for supervised learning and semi-supervised learning (SSL), which incorporates unlabeled data via a partly data-dependent prior. Then we discuss a formulation of self-training based Bayesian SSL with partly data-driven cross-entropy prior. Further, we describe a modification to this Bayesian SSL formulation to handle the situation when the distribution of unlabeled inputs is different from the labeled inputs (covariate shift).

In Section 3, we describe how this Bayesian SSL formulation can be further extended for our problem, i.e., the lifelong/continual TTA setting where we learn a source model using only labeled data, and then the model has to predict labels of unlabeled test inputs coming from target domains with different distributions.

Bayesian Supervised Learning

The Bayesian setup for supervised learning typically assumes that we are given labeled data \( \mathcal{D} = \{x_n, y_n\}_{n=1}^{N} \), and we estimate \( \theta \) using its posterior distribution

\[
p(\theta | \mathcal{D}) \propto p(\theta) \prod_{n=1}^{N} p(y_n | x_n, \theta).
\]

Given a novel test input \( x_{N+1} \), we make predictions using posterior predictive distribution obtained by marginalizing over the posterior distribution \( p(y_{N+1}|x_{N+1}, \theta) = \int p(y_{N+1}|x_{N+1}, \theta)p(\theta | \mathcal{D})d\theta \).

Bayesian Semi-Supervised Learning

Here, we are provided with unlabeled data along with some labeled data. We denote the unlabeled data points as \( \mathcal{U} = \{x_m\}_{m=1}^{M} \). To circumvent the inability to use unlabeled data while inferring \( \theta \), one needs to make assumptions about the dependency between distributions of inputs and labels.

To this end, [9] proposed a prior that is partly data-dependent via the inputs \( x \): \( p(\theta|\psi) \propto p(\theta) \exp(-\lambda H_{\theta,\psi}(y|x)) \), where \( p(\theta) \) is the prior in Eq. 1, and \( H \) is the conditional entropy of the class label. Here, partly data-dependent prior refers to a prior defined using only the input \( x \) treated as given, not the label \( y \) which is treated as a random variable.

Bayesian Semi-Supervised Learning with Self-Training

The self-training framework [19, 42] has demonstrated significant success in semi-supervised learning. Our proposed framework is also based on self-training wherein we use an exponential moving average of the parameters \( \theta \) of the student model \( p(y|x, \theta) \) (which is initialized with the source pre-trained model parameters \( \theta_0 \)), and refer to the averaged model as the teacher model \( \{\theta_i\}_i \) where \( \pi \) is the smoothing factor. For brevity, we will omit the time step index \( t \) from here onwards.

In the semi-supervised setting, the teacher model can be utilized to obtain augmentation-averaged pseudo-labels \( \hat{y} \) corresponding to an unlabeled input \( x \). To prevent error accumulation, augmentation is only used when the domain difference is substantial. Defining \( \hat{y}' = p(y|x, \theta_0) \); \( y' = \frac{1}{K} \sum_{i=1}^{K} p(y|\alpha_i(x), \theta_0) \), the pseudo-label is defined as

\[
\hat{y}' = \begin{cases} y', & \text{if } C(p(y|x, \theta_0) \geq \tau) \\ \hat{y}', & \text{otherwise}. \end{cases}
\]

Here, \( K \) is number of times augmentation is applied, \( \alpha_i() \) is augmentation function, \( C() \) gives the confidence of the prediction, and \( \tau \) is threshold for selecting confident predictions. The prediction confidence of current input using source domain pre-trained model \( \theta_0 \) gives us an estimate of the domain difference between source and target domain.

Using these pseudo labels, we formulate the following partly data-driven cross-entropy prior

\[
p(\theta|\psi) \propto p(\theta) \exp(-\lambda H_{\theta,\psi}(\hat{y}', y|x))
\]

\[
= p(\theta) \exp(\lambda \mathbb{E}_{x \sim p(x,y'), y' \sim p(y|x,\theta')}[\log p(y|x, \theta)]).
\]

Here, \( y = p(y|x, \theta) \) is the prediction of the student model and \( H^\psi \) is the conditional cross-entropy of labels conditioned on the inputs. This cross-entropy term leverages the knowledge from the teacher model as it is incorporated into the partly data-driven prior.

Bayesian Semi-Supervised Learning with Unlabeled Data Distribution Shift

The above semi-supervised learning formulation assumes that the unlabeled inputs come from the same distribution as the labeled inputs. To handle the situation when the unlabeled inputs come from a different distribution, we introduce additional generative parameters \( \bar{\psi} \). Here, \( \bar{\psi} = \mathbb{E}_{x \sim p(x, \theta')}[\log p(y|x, \theta')] \) is the conditional entropy of labels conditioned on the inputs. This cross-entropy term leverages the knowledge from the teacher model as it is incorporated into the partly data-driven prior.

Bayesian Semi-Supervised Learning with Self-Training

In Eq. 3, the prior becomes

\[
p(\theta|\psi, \bar{\psi}) \propto p(\theta) \exp(-\lambda H_{\theta,\psi}(y', y|x))
\]

\[
\exp(-\lambda H_{\theta,\psi}(\hat{y}', y|x)).
\]
3. Probabilistic Test-Time Adaptation

Our formulation for TTA is similar to the formulation we described above for the problem of semi-supervised learning with unlabeled data distribution shift, with a key difference. In contrast to Bayesian SSL in which labeled and unlabeled data are available during training, in test-time adaptation, we need to predict labels of the test inputs from a different domain, but we only have access to the source model weights without the source domain training data. In our probabilistic approach, we assume that the source model is given in form of the approximate posterior distributions of the network parameters at test-time. Let $\bar{x}$ represent the source domain data.

Replacing $p(x|\psi)$ and $p(x|\bar{\psi})$ with the empirical distributions of $x$ and $\bar{x}$, we get

$$p(\theta|\psi, \bar{\psi}) \propto p(\theta) \exp \left( -\frac{\lambda}{N} \sum_{n=1}^{N} H^{<}(y'_n, y_n|x_n, \theta) \right) \exp \left( -\frac{\lambda}{M} \sum_{m=1}^{M} H^{<}(y'_m, y_m|x_m, \bar{\theta}) \right). \quad (5)$$

Using Eq. 5 as the prior in Eq. 1 and taking logarithm on both sides (ignoring additive normalization constants), the new posterior distribution of the network parameters becomes:

$$\log p(\theta|D, U) = \log p(\theta) + \sum_{n=1}^{N} \log p(y_n|x_n, \theta) - \frac{\lambda}{N} \sum_{n=1}^{N} H^{<}(y'_n, y_n|x_n, \theta) - \frac{\lambda}{M} \sum_{m=1}^{M} H^{<}(y'_m, y_m|x_m, \bar{\theta}). \quad (6)$$

3.1. Parameter Restoration

Stochastic Restoration

In lifelong TTA, in order to reduce the error accumulation over the long term in self-training and handle catastrophic forgetting, [41] proposed stochastic restoration of weights by additionally updating the parameters. Let $\theta^f$ denote the flattened parameter $\theta$ of the student model, and $D$ be the dimension of $\theta^f$. After the gradient update at time step $t$, stochastic restore further updates the parameters:

$$m \sim \text{Bernoulli}(\rho), \quad \theta^f_{t+1} = m \odot \theta^f_{t} + (1-m) \odot \theta^f_{t+1}. \quad (10)$$

Here, $\odot$ is element-wise multiplication, and $\rho$ is stochastic restore probability. $m$ is mask to determine which parameters within $\theta^f_{t+1}$ to restore to original source weight $\theta^f_{t}$. 

Fisher Information Based Restoration

To improve upon stochastic restoration, we propose a data-driven parameter restoration. Fisher Information Matrix (FIM) is widely used as a metric of parameter importance for a given data [15]. Thus, we use FIM, $F$, of the student model parameterized by $\theta$ as a measure of the importance of the parameters. For a given time step $t$ with unlabeled test inputs batch $B = \{x_m\}_{m=1}^{B\text{size}}$, we consider the following diagonal approximation of FIM:

$$F = \text{Diag} (\nabla_{\theta} \mathcal{L} (\nabla_{\theta} \mathcal{L})^T), \quad \mathcal{L} = \log q(\theta) - \frac{\lambda}{|B|} \sum_{m=1}^{B\text{size}} H^{<}(y'_m, y_m|x_m). \quad (11)$$

Here, $y'_m$ and $y_m$ are the teacher and student model predictions, respectively. Note that $F$ has the same dimension as $\theta$. Thus, upon using FIM based restore, the parameter restoration in Eq. 6 becomes

$$m_p = \begin{cases} 1, & \text{if } F_p < \gamma \land p = 1, \ldots, P, \\ 0, & \text{otherwise}. \end{cases} \quad (12)$$

Here, $\gamma = \text{quantile}(F, \delta)$ is the threshold value which is the $\delta$-quantile of $F$. Thus, the elements in $m$ corresponding to FIM value less than $\gamma$ would be 1, implying that the corresponding parameters would be restored to original source weight $\theta^f_{t}$. The algorithm for PETAL is given in Algorithm 1. Fig. 1 (Right) provides an overview of our approach PETAL.
Algorithm 1 Proposed Approach PETAL

Input: Training dataset $\mathcal{X} = \{x_n, y_n\}_{n=1}^N$;
Test domain data $\mathcal{X}_d = \{x_d\}_{d=1}^D$.

Output: Prediction $\hat{y}_d$

Training:
1: Compute $q(\theta) \approx p(\theta|\mathcal{X})$

Continual Adaptation:
2: Initialize time step $t = 0$, prediction set $\hat{y}_d = \emptyset$
3: Initialize student model $\theta_0 = \max_{d} \log q(\theta)$
4: Initialize teacher model $\theta_0'$ = $\theta_0$
5: for $d = 1, \cdots, D$ do
6: for each batch $B \in \mathcal{X}_d$ do
7: $\theta_{t+1}', \theta_{t+1}, B_y = \text{Adapt}(\theta_{t} ', \theta, \theta_0, B, q(\theta), t)$
8: $\hat{y}_d = \hat{y}_d \cup B_y$
9: $t = t + 1$
10: end for
11: end for

4. Related Work

4.1. Unsupervised Domain Adaptation

The goal of unsupervised domain adaptation (UDA) [6, 24, 30, 40] is to enhance the performance of the learning model when there is a change in distribution between the training data domain and the test data domain. UDA approaches often assume that the source and (unlabeled) target domain data are accessible simultaneously. Most existing methods address UDA by ensuring that the feature distributions [6, 24, 37] or the input spaces [12, 44] of the source and the target domains are brought closer.

4.2. Test-Time Adaptation

Some works also refer to TTA as source-free domain adaptation. Recent works explore source-free domain [18, 20, 23] setting in which training data is unavailable and only unlabeled data is available during adaptation. Test entropy minimization (TENT) [38] starts from a source pre-trained model and updates only the batch-norm (BN) parameters by minimizing entropy in test predictions. [34] address TTA by updating the source domain BN statistics using test input statistics. Continual Test-Time Adaptation (CoTTA) [41] addresses online lifelong TTA by employing weight averaging and augmentation averaging, and random parameter restoration back to source pre-trained model parameters. [8] adapts to continually changing target domains by utilizing a normalization layer to handle the out-of-distribution examples and balanced reservoir sampling to store the simulated i.i.d. data in the memory. It would be an interesting future work to extend PETAL for the temporally correlated test stream setting proposed by [8]. EATA [29] is another related work that looks at preventing forgetting in the context of TTA; however, EATA mainly focuses on preventing the forgetting of the source task model and is not designed to handle forgetting in a lifelong TTA setting. Our work PETAL is a principled probabilistic approach for lifelong test-time adaptation that uses an approximate posterior during test-time adaptation obtained from source domain data. PETAL also offers a probabilistic perspective and justification to CoTTA which arises a special case of PETAL.

Bayesian Adaptation for Covariate Shift (BACS) [46] proposes a Bayesian perspective for TENT, which naturally gives rise to the entropy term along with a regularizer that captures knowledge from posterior density obtained from training data. However, BACS only addresses standard TTA setting, and regularized entropy minimization lacks the ability to handle error accumulation and catastrophic forgetting encountered in lifelong TTA.

Algorithm 2 Adapt

Input: $\theta_{t} ', \theta_0, B, q(\theta), t$

Require: Number of augmentations $K$; learning rate $\eta$;
Threshold for confident predictions $\tau$;
Confidence function $C$; Cross-entropy weight $\lambda$;
Quantile of FIM $\delta$; Augmentation function $\alpha$; Smoothing factor $\pi$

Output: $\theta_{t+1}, \theta_{t+1}, B_y$
1: Initialize: $H = 0, B_y = \emptyset$
2: for each test input $\bar{x} \in B$ do
3: Teacher model prediction: $\bar{y}' = p(y|\bar{x}, \theta')$
4: Augmented Average: $\tilde{y}' = \frac{1}{K} \sum_{t=1}^{K} p(y|\alpha_t(\bar{x}), \theta')$
5: Augment based on domain gap:
6: Using student model, predict: $y = p(y|x, \theta)$
7: Update: $H = H + Hr(y, y|\bar{x})$
8: $B_y = B_y \cup \{y\}'$
9: end for
10: Compute: $L = \left( \log q(\theta) - \frac{1}{\lambda} H \right)$
11: Adapt student model: $\theta_{t+1} = \theta_{t+1} + \eta \nabla_{\theta} L$
12: Update teacher model: $\theta_{t+1}' = \pi \theta_{t+1}' + (1 - \pi) \theta_{t+1}$
13: Compute FIM: $F = \text{Diag} \left( \nabla_{\theta} L \right)$
14: Compute mask $m$ for resetting:
15: Reset updated student model back to source model:

$$\theta_{t+1} = m \odot \theta_0 + (1 - m) \odot \theta_{t+1}$$


**Table 1.** Experimental results for CIFAR10-to-CIFAR10C online lifelong test-time adaptation task. The numbers denote the classification error (%) obtained with the highest corruption of severity level 5. TENT-online uses domain information, denoted using +.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gaussian</th>
<th>shot</th>
<th>impulse</th>
<th>defocus</th>
<th>glass</th>
<th>motion</th>
<th>zoom</th>
<th>snow</th>
<th>frost</th>
<th>fog</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>72.33</td>
<td>65.71</td>
<td>72.92</td>
<td>46.94</td>
<td>54.12</td>
<td>34.75</td>
<td>42.02</td>
<td>25.07</td>
<td>41.30</td>
<td>26.01</td>
<td>9.30</td>
</tr>
<tr>
<td>Pseudo-label</td>
<td>26.70</td>
<td>22.10</td>
<td>32.00</td>
<td>13.80</td>
<td>32.20</td>
<td>15.30</td>
<td>12.70</td>
<td>17.30</td>
<td>17.30</td>
<td>16.50</td>
<td>10.10</td>
</tr>
<tr>
<td>TENT-continual</td>
<td>24.80</td>
<td>20.60</td>
<td>28.60</td>
<td>14.40</td>
<td>31.10</td>
<td>16.50</td>
<td>14.10</td>
<td>19.10</td>
<td>18.60</td>
<td>18.60</td>
<td>12.20</td>
</tr>
</tbody>
</table>

4.3. Continual Learning

The objective of Continual Learning (CL) is to learn from a sequential series of tasks, enabling the model to retain previously acquired knowledge while learning a new task, preventing catastrophic forgetting [27, 31, 33]. Elastic weight consolidation (EWC) [15] is a regularization-based technique that penalizes parameter changes having a significant impact on prediction. [22] proposed learning without forgetting (LwF), which preserves knowledge of previous tasks using knowledge distillation. Gradient episodic memory (GEM) [25] maintains a limited number of samples to retrain while constraining fresh task updates from interfering with prior task knowledge. [1, 39] address the continual semi-supervised learning problem where continually arriving tasks consist of labeled and unlabeled data.

5. Experiments

We thoroughly evaluate PETAL and compare it to other state-of-the-art approaches on image classification lifelong test-time adaptation benchmark tasks: CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, ImageNet-to-ImageNetC, and ImageNet-to-Imagenet3DCC.

5.1. Benchmark Datasets

[11] developed CIFAR10C, CIFAR100C, and ImageNetC datasets to serve as benchmarks for the robustness of classification models. In each dataset, there are 15 different types of corruption and five different levels of severity. These corruptions are applied to test images of original CIFAR10 and CIFAR100 [17] datasets and validation images of original ImageNet [5] dataset. Further, we experiment with Imagenet 3D Common Corruptions (Imagenet3DCC) dataset, recently proposed by [14], which utilizes the geometry of the scene in transformations, leading to more realistic corruptions. Imagenet3DCC dataset consists of 12 different types of corruptions, each with five levels of severity. Refer to Appendix for details.

In online lifelong TTA, we begin with a network trained on CIFAR10, CIFAR100, and ImageNet clean training set for the respective experiments. At the time of testing, the model gets corrupted images online.

Following CoTTA, we continually adjust the source pre-trained model to each corruption type as they sequentially arrive, as opposed to conventional TTA in which the pre-trained model is separately adapted to each corruption type. We evaluate the model using online predictions obtained immediately as the data is encountered. We follow the online lifelong test-time adaptation setting for all the experiments. For ImageNet-to-ImageNetC experiments, we evaluate using 10 different sequences of corruptions.

5.2. Model Architectures and Hyperparameters

Following TENT [38] and CoTTA [41], we adopt pre-trained WideResNet-28 [45] model for CIFAR10-to-CIFAR10C, pre-trained ResNeXt-29 [43] model for CIFAR100-to-CIFAR100C, and standard pre-trained ResNet-50 model for both ImageNet-to-ImagenetC and ImageNet-to-Imagenet3DCC experiments from RobustBench [4].

We utilize SWAG-D [26] to approximate the posterior density from the source domain training data. SWAG-D approximates the posterior using a Gaussian distribution with diagonal covariance from the SGD trajectory. Before adapting the model, we initialize it with the maximum a posteriori (MAP) of the approximate posterior that corresponds to the solution obtained by Stochastic Weight Averaging [15]. This is effectively the source domain pre-trained model.

We update all the trainable parameters in all experiments. We use $K = 32$ number of augmentations. We adopt the same augmentation confidence threshold described in [41]. For FIM based parameter restoration, we set the quantile value $\delta = 0.03$. We refer the readers to the Appendix for more details on the hyperparameters.

5.3. Baselines and Compared Approaches

To evaluate the efficacy of PETAL, we compare the PETAL with CoTTA and five other methods in online lifelong test-time adaptation. Source denotes the baseline pre-trained model that has not been adapted to test inputs. In $BN$
Adapt [21, 34], the network parameters are kept frozen, and only Batch Normalization statistics are adapted to produce predictions for test inputs. Pseudo-label updates the BatchNorm trainable parameters with hard pseudo-labels [19]. TENT-online denotes the TENT [38] approach in this setting, but it has access to extra information about the change in the domain and, thus, resets itself to the original pre-trained model upon encountering test inputs from the new domain and adapts afresh. But this additional knowledge is unavailable in real-life scenarios. TENT-continual has no extra information about domain change. CoTTA [41] uses weight averaging and augmentation averaging, along with randomly restoring parameters to the original pre-trained model. However, it lacks explicit uncertainty modeling and data-driven parameter restoration.

### 5.4. Evaluation Metrics

We evaluate our model using the error rate in predictions. To evaluate the uncertainty estimation, we use negative log-likelihood (NLL) and Brier score [2]. Both NLL and Brier are proper scoring rules [7], and they are minimized if and only if the predicted distribution becomes identical to the actual distribution. In Table 1, 2 and 4, the number within brackets is the standard deviation over 5 runs. Refer to the Appendix for more details on evaluation metrics.

Table 3. CIFAR10-to-CIFAR10C results for gradually changing severity level before changing corruption types. The numbers are averaged over all 15 corruption types. The number after ± is the standard deviation over 10 random corruption sequences. Our method surpasses all baselines in the depicted settings.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error (%)</th>
<th>Brier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>29.34</td>
<td>0.408</td>
</tr>
<tr>
<td>BN Adapt</td>
<td>34.18</td>
<td>0.575</td>
</tr>
<tr>
<td>TENT</td>
<td>40.37</td>
<td>0.159</td>
</tr>
<tr>
<td>CoTTA</td>
<td>34.78</td>
<td>0.003</td>
</tr>
</tbody>
</table>

### 5.5. CIFAR10-to-CIFAR10C Results

In Table 1, we observe that directly using the pre-trained model (Source) leads to poor performance with an average error rate of 43.51%, suggesting the necessity of adaptation. Adapting the Batch Normalization (BN) statistics improves the average error rate to 20.44%. Using hard pseudo-labels and updating only the BN parameters further improves the performance to 19.8%. TENT-online reduces the error rate to 18.58% using extra information about domain change, but access to such information is mostly unavailable in real-world scenarios. As expected, the error rate of TENT-continual increases to 20.7% without access to domain change information. Further, CoTTA improves the average error rate to 16.29%. Our proposed approach consistently outperforms other approaches for most individual corruption types, reducing the average error rate to 15.95%. Moreover, our proposed approach demonstrates no performance degradation in the long term.

To investigate the contribution of FIM based parameter restore, we show the results of our approach with stochastic restore in place of FIM based restore, denoted using S-Res. We observe that FIM based restore performs better than S-Res for most of the corruption types, highlighting the effectiveness of FIM based restore. Further, the improved performance of PETAL (S-Res) and PETAL (FIM) over various baselines, including CoTTA which is specifically designed for continual TTA, demonstrates the effectiveness of our probabilistic framework where the source model’s posterior induces a regularizer and the data-driven resetting helps make an informed selection of weights to reset/keep. Moreover, the contribution of the regularizer term (induced by the source model’s posterior) is evident from the superior performance of PETAL over CoTTA, since CoTTA is a special case of PETAL without the regularizer term.

Gradually changing corruptions: Following [41], we evaluate PETAL in the setting where the severity of corruption changes gradually before the change in corruption type. When the corruption type changes, the severity level is lowest, and thus, domain shift is gradual. In addition, distribution shifts within each corruption type are also gradual. Refer to the Appendix for details.

We report the average error and average Brier score over 10 randomly shuffled orders of corruptions. We can observe in Table 3 that our approach PETAL performs better than other approaches in terms of error and Brier score.
Table 4. CIFAR100-to-CIFAR100C results with the most severe level of corruption, 5, averaged over all corruption types. Our method surpasses all baselines in terms of NLL and Brier uncertainty estimation measures.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Source</th>
<th>BN Adapt</th>
<th>TENT</th>
<th>CoTTA</th>
<th>PETAL (FIM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLL</td>
<td>2.4945</td>
<td>1.3937</td>
<td>7.3789</td>
<td>1.2767 (0.0006)</td>
<td>1.2286 (0.0011)</td>
</tr>
<tr>
<td>Error</td>
<td>0.6704</td>
<td>0.4744</td>
<td>1.1015</td>
<td>0.4430 (0.0003)</td>
<td>0.4317 (0.0003)</td>
</tr>
</tbody>
</table>

5.6. CIFAR100-to-CIFAR100C Results

To illustrate the efficacy of the proposed approach, we conduct an evaluation on the more challenging CIFAR100-to-CIFAR100C task. In Table 2, we compare the results with Source, BN Adapt, Pseudo-label, and TENT-continual approaches. We observe that TENT-continual performs better initially, but as new domains arrive continually, the performance degrades drastically in the long term. With an average error rate of 31.46%, our proposed approach PETAL (FIM) consistently outperforms the other approaches.

To measure the ability of uncertainty estimation of our approach, we compare with the other approaches in Table 4 in terms of NLL and Brier score. We obtain both NLL and Brier score for corruption with a severity level of 5 and average over all corruption types. Our approach performs better than all other approaches in terms of average NLL and average Brier, demonstrating the ability of our approach to improve the uncertainty estimation. Moreover, the FIM based restore outperforms stochastic restore for most corruption types, illustrating the utility of data-driven resetting.

Figure 2. ImageNet-to-ImageNetC results averaged over 10 different corruption orders with level 5 corruption severity.

5.7. ImageNet-to-ImageNetC Results

In Table 5, we investigate the performance of our proposed approach for ImageNet-to-ImageNetC task with 10 different sequences of corruption. We obtain the performance scores by averaging over all corruption types and all corruption orders. In terms of average error, average NLL, and average Brier, our approach performs better than the other approaches. In Fig. 2, we compare the performance by averaging over 10 different corruption orders in terms of error rate and Brier score. For most corruption types, our approach performs better than other existing approaches.

Table 5. ImageNet-to-ImageNetC results averaged over all corruption types and over 10 diverse corruption orders (highest corruption severity level 5).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Source</th>
<th>BN Adapt</th>
<th>TENT</th>
<th>CoTTA</th>
<th>PETAL (FIM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLL</td>
<td>69.21</td>
<td>69.32</td>
<td>95.93</td>
<td>59.91</td>
<td>59.64</td>
</tr>
<tr>
<td>Error</td>
<td>5.0701</td>
<td>3.9956</td>
<td>3.6076</td>
<td>3.3425</td>
<td>3.3252</td>
</tr>
<tr>
<td>Brier</td>
<td>0.9459</td>
<td>0.8345</td>
<td>0.9222</td>
<td>0.7681</td>
<td>0.7663</td>
</tr>
</tbody>
</table>

5.8. ImageNet-to-ImageNet3DCC Results

For ImageNet-to-ImageNet3DCC dataset, we experiment with 10 different random sequences of corruptions. We provide the results in Table 6 by averaging over 10 random sequences of corruptions and 12 corruption types at severity level 5. PETAL consistently outperforms all other approaches in terms of error rate, NLL, and Brier score.

Table 6. ImageNet-to-ImageNet3DCC results averaged over all corruption types and over 10 diverse corruption orders (highest corruption severity level 5).

<table>
<thead>
<tr>
<th>Metric</th>
<th>Source</th>
<th>BN Adapt</th>
<th>TENT</th>
<th>CoTTA</th>
<th>PETAL (FIM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>69.21</td>
<td>69.32</td>
<td>95.93</td>
<td>59.91</td>
<td>59.64</td>
</tr>
<tr>
<td>NLL</td>
<td>5.0701</td>
<td>3.9956</td>
<td>3.6076</td>
<td>3.3425</td>
<td>3.3252</td>
</tr>
<tr>
<td>Brier</td>
<td>0.9459</td>
<td>0.8345</td>
<td>0.9222</td>
<td>0.7681</td>
<td>0.7663</td>
</tr>
</tbody>
</table>

6. Conclusion

We proposed a probabilistic framework for lifelong TTA using a partly data-driven prior. Addressing the problem via the probabilistic perspective naturally gives rise to the student-teacher framework along with a regularizer that captures the source domain knowledge. In lifelong TTA, we have demonstrated that our principled use of an approximate training posterior surpasses prior heuristic approaches. Our proposed approach also provides more reliable uncertainty estimates demonstrated with better NLL and Brier score. Further, we developed a Fisher information matrix based parameter restoration, which is driven by the data to improve upon existing stochastic restore. In terms of error rate, NLL and Brier score, PETAL yields state-of-the-art results across CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, ImageNet-to-ImageNetC, and ImageNet-to-ImageNet3DCC benchmark tasks.

Acknowledgments: The authors acknowledge support from the Qualcomm Innovation Fellowship and travel grant support from the Research-I Foundation at IIT Kanpur.
References


