Using Conjugate Pseudo-labels to Address Distribution Shifts in Machine Learning Models

Ansh Mujral
amujral@ucsd.edu

Ifunanya Okoroma
iokoroma@ucsd.edu

Keenan Serrao
kserrao@ucsd.edu

Nicholas Swetlin
nswetlin@ucsd.edu

Jun-Kun Wang
jkw005@ucsd.edu

Abstract

Machine learning models encounter the problem of distribution shifts, where the data used to train the model differs from the data used when the model is deployed at test time. When training new systems without access to the test data, it's important that these models still perform well under those circumstances. Without access to these labels at test time, an approach is to use conjugate pseudo-labels. In this approach, we construct pseudo-labels to simulate our testing environment and assess model performance. We do so through optimizing for a loss function. When replicating experiments from Goyal (2022), we optimize for both cross-entropy loss and polyloss.

Code: https://github.com/nickswetUCSD/Capstone-Checkpoint-A12

1 Introduction 2
2 Problem Statement 2
3 Methods 3
4 Results 6

1 Introduction

As artificial intelligence and machine learning systems intertwine their way into more real-world scenarios, it is crucial to make sure that these systems handle a broad range of environments. Deep neural networks perform remarkably well when the test data closely mirrors what they've been trained on. However, in the real world, models often encounter unfamiliar data from new domains.

This difference is known as the distributional shift between training and testing data and it can cause a sharp drop in performance – often calling into question how reliable these systems can be in prediction tasks. This challenge sparked the interest in a new method called test-time adaptation (TTA). This method helps models adjust to these distribution shifts using only the new, unlabeled data they encounter during testing.

TTA approaches often focus on fine-tuning the model's parameters through unsupervised objectives since there's no labeled data available during testing. Past research has shown some promising techniques, like entropy minimization. This is a technique that guides the model to adapt effectively to new data. However, choosing the right TTA loss function has mostly been a trial-and-error process. Without a consistent, theory-backed framework, TTA methods see mixed success depending on the type of data shift, limiting how well they can be applied in the real world.

2 Problem Statement

Current TTA methods face challenges because they do not generalize well across different domains. The main focus of this experiment is to address the issue of generalizability and to provide an adaptable loss function that can conduct TTA. Current loss functions are heuristic-based, which not only does not generalize to different distributions but also hinders the robustness of these models in real-world applications.

2.1 Inputs

The key inputs required for the TTA approach include using a pre-trained model initially optimized on a source domain, representing the data distribution used during training. When the domain shifts, the new input would be unlabeled test samples from a new target domain with a distribution that differs from the original source domain.

2.2 Outputs

The output of the TTA process is an adapted model that retains high accuracy on the target domain, despite the lack of labeled data in this new setting. Using an appropriate TTA loss, the model should ideally reach a level of accuracy close to a model trained specifically on target-domain data.

2.3 Goals

This experiment aims to develop a robust, theory-driven method for TTA. The study introduces "conjugate pseudo-labels," a framework for creating self-training pseudo-labels by using convex conjugate functions based on the original supervised loss. This adaptable approach allows for flexibility in TTA across different training losses, such as cross-entropy loss. By evaluating this method on a range of benchmarks, this experiment aims to show that conjugate pseudo-labeling can outperform traditional, heuristic-based TTA losses. Thus, this method is setting a higher standard for adaptive model performance in the face of domain shifts.

2.4 Motivation

Machine learning models often struggle to handle unpredictable, real-world conditions effectively, as the data they encounter differs significantly from their training scenarios. In critical fields like healthcare, finance, and autonomous systems, it is very important that models function reliably despite these distribution shifts. Conjugate pseudo-labeling is an advanced technique that can be used by offering a solid yet flexible way for models to adapt on their own. This approach helps models stay effective across changing conditions without relying on new labeled data, making it a broad and dependable tool to improve model reliability across various sectors.

3 Methods

Implementing the method of conjugate pseudo-labels helps us simulate a test environment when we do not have direct access to one. Experiments conducted in the paper that we have decided to replicate involve the augmentation (often called "corruption") of image data from widely-used datasets to simulate the environment of a distribution shift.

Due to disk space limitations, we have pursued replicating the experiments on only 50 percent of the dataset. This implementation is being performed through PyTorch on the CIFAR-10 and CIFAR-10-C datasets.

To replicate some of the results we've observed in the paper, we have decided to optimize the conjugate pseudo-labels with respect to (1) the cross-entropy loss and (2) polyloss with

$\epsilon = 1$. For each of these two losses, we compare the performance of conjugate pseudo-labeling against the performance of more traditional hard pseudo-labeling.

3.1 Derivation Of Self-Training Loss For Cross-Entropy

The cross-entropy loss function is defined as:

$$L_{CE}(h(x), y) = -\sum_{i=1}^{c} y_i \log\left(\frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}\right)$$

where our model output $h(x) \in \mathbb{R}^c$ and y is the vector of true labels for our input data x. We want to express this loss function as the form:

$$L(h(x), y) = -y^T h(x) + f(h(x))$$

for some function $f(h)$.

Expanding the above, we get:

$$L_{CE}(h(x), y) = -\sum_{i=1}^{c} y_i \left(h_i - \log\left(\sum_{j=1}^{c} e^{h_j}\right)\right)$$

Breaking down further:

$$= -y^T h(x_0) + \sum_{i=1}^{c} y_i \log\left(\sum_{j=1}^{c} e^{h_j}\right)$$

Simplifying, we have:

$$= -y^T h(x_0) + \log\left(\sum_{j=1}^{c} e^{h_j}\right)\sum_{i=1}^{c} y_i$$

where $\sum_{i=1}^{c} y_i = 1$ due to normalization. It is now apparent that $f(h) = \log\left(\sum_{j=1}^{c} e^{h_j}\right)$.

The gradient of $f(h)$, from Goyal et.al., is the optimal psuedo-label vector $(\hat{y})$ for our self-training loss function.

The jth entry of $\nabla f(h)$ is given by:

$$\hat{y}_i = \nabla f(h)_i = \frac{e^{h_j}}{\sum_{i=1}^{c} e^{h_i}}$$

Substituting all true labels (y) with our pseudolabels $(\hat{y})$ in our original cross-entropy loss, we get a "self-training" loss function that does not at all depend on the true labels. Rather, there is only a dependence on $h(x)$, which is the output of our model:

$$\underbrace{L_{CE}^{\text{conj}}(h(x), \hat{y})}_{\text{Cross-Entropy Loss}} = -\sum_{i=1}^{c}\left(\frac{e^{h_j}}{\sum_{i=1}^{c} e^{h_i}}\right)\log\left(\frac{e^{h_j}}{\sum_{i=1}^{c} e^{h_i}}\right)$$

3.2 Derivation Of Self-Training Loss For Polyloss

PolyLoss is a loss of growing popularity in machine learning communities for its effectiveness. It is defined as:

$$L_{\text{poly}}(h_\theta(x), y) = L_{CE}(h_\theta(x), y) + \epsilon \cdot y^T \left(1 - \text{softmax}(h_\theta(x))\right),$$

where:

- $L_{CE}(h_\theta(x), y)$: The cross-entropy loss function.
- $\text{softmax}(h_\theta(x))_i = \frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}$: Predicted class probabilities.
- ϵ: A weighting parameter that adjusts the penalty on incorrect pseudo-labels.

—

We want to express this loss function in the form:

$$L_{\text{poly}}(h(x), y) = -y^T h(x) + f(h(x))$$

Making substitutions for $L_{\text{ce}}(h_\theta(x), y)$ we have:

$$L_{\text{poly}}(h_\theta(x), y) = \left[-y^T h_\theta(x) + \log\left(\sum_{j=1}^{c} e^{h_j}\right)\right] + \epsilon \cdot y^T \left(1 - \text{softmax}(h_\theta(x))\right).$$

Simplify the softmax adjustment term:

$$\epsilon \cdot y^T \left(1 - \text{softmax}(h_\theta(x))\right) = \epsilon \cdot \left(1 - \sum_{i=1}^{c} y_i \cdot \frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}\right).$$

Thus, PolyLoss becomes:

$$L_{\text{poly}}(h_\theta(x), y) = -y^T h_\theta(x) + \log\left(\sum_{j=1}^{c} e^{h_j}\right) + \epsilon \cdot \left(1 - \sum_{i=1}^{c} y_i \cdot \frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}\right).$$

We see that $L_{\text{poly}}(h_{\theta(x)}, y)$ is now expressed in the desired form:

$$L_{\text{poly}}(h_\theta(x), y) = f(h_\theta(x)) - y^T h_\theta(x),$$

where:

$$f(h_\theta(x)) = \log\left(\sum_{j=1}^{c} e^{h_j}\right) + \epsilon \cdot \left(1 - \sum_{i=1}^{c} \left(\frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}\right)^2\right).$$

—

From Goyal et.al., the optimal psuedo-label vector ($\hat{y}$) for the polyloss self-training loss function satisfies the following optimality condition (given by the Fenchel-Young inequality):

$$y_{\text{CPL}}(x) = (Dg(h_\theta(x)))^{-1} \cdot \nabla f(h_\theta(x)).$$

...where D is the Jacobian operator, and $g(h_\theta(x)) = h_\theta(x) - \epsilon \cdot (1 - \text{softmax}(h_\theta(x)))$.

—

We find $\nabla f(h_\theta(x))$ and $Dg(h_\theta(x))$ to be:

- $$\nabla f(h_\theta(x)) = z, \quad z_i = \frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}, \quad i = 1, \ldots, c.$$
- $$Dg(h_\theta(x)) = I + \epsilon \cdot \text{diag}(z) - \epsilon \cdot zz^T, \quad z = \text{softmax}(h_\theta(x)).$$

Substituting $Dg(h_\theta(x))$ and $\nabla f(h_\theta(x))$ into our optimality condition yields a formula for the appropriate conjugate pseudolabel :

$$y_{\text{CPL}}(x) = \left(I + \epsilon \cdot \text{diag}(z) - \epsilon \cdot zz^T\right)^{-1} z, \quad z = \text{softmax}(h_\theta(x))$$

We can then substitute $y_{\text{CPL}}(x)$ in for y in our original polyloss formula to get the self-training loss function for polyloss.:

$$L_{\text{poly}}^{\text{conj}}(h_\theta(x)) = f(h_\theta(x)) - \left(\left(I + \epsilon \cdot \text{diag}(z) - \epsilon \cdot zz^T\right)^{-1} z\right)^T h_\theta(x).$$

Expanding, this becomes:

$$\boxed{L_{\text{poly}}^{\text{conj}}(h_\theta(x)) = \log\left(\sum_{j=1}^{c} e^{h_j}\right) + \epsilon \cdot \left(1 - \sum_{i=1}^{c} z_i^2\right) - \sum_{i=1}^{c} \frac{\tilde{z}_i \cdot h_i}{1 + \epsilon \cdot z_i - \epsilon \cdot \sum_{j=1}^{c} z_j z_i},}$$

where:

$$z_i = \frac{e^{h_i}}{\sum_{j=1}^{c} e^{h_j}}, \quad \tilde{z} = \left(I + \epsilon \cdot \text{diag}(z) - \epsilon \cdot zz^T\right)^{-1} z.$$

—

This is the final formula for the self-training loss function derived from PolyLoss.

4 Results

After testing the error of hard pseudo-label and conjugate pseudo-label test-time adaptation in an .ipynb file, here are the results we obtained.

As seen in Tables 1 and 2, hard pseudo-labeling for a source classifier trained with cross entropy loss achieves the correct label about 10-11% of the time, and whereas conjugate pseudo-labels achieve the correct label about 60-80% of the time. This echoes the results of Goyal (2022).


Overall Error Rate Per Epoch
Cross Entropy
Hard Pseudo-Labels
Conjugate Pseudo-Labels
Error Rate
Epoch
0.35
0.34
0.33
0.32
0.31
0.30
0.29
0.28
0
5
10
15
20
25

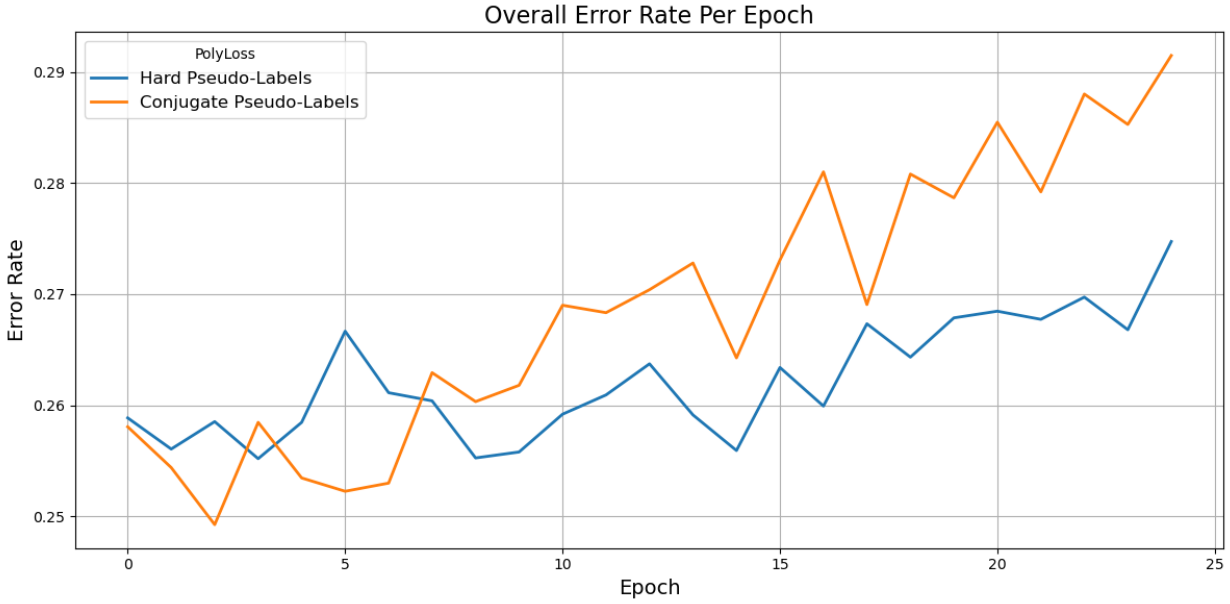
Overall Error Rate Per Epoch
PolyLoss
Hard Pseudo-Labels
Conjugate Pseudo-Labels
Error Rate
Epoch
0.29
0.28
0.27
0.26
0.25
0
5
10
15
20
25

Table 1: Meta Test Error Rates for Various Noise and Distortion Types at Severity Level 5 using Hard Pseudo-Labels and Conjugate Pseudo-Labels with Cross-Entropy Loss. Lowest errors are **bolded**.

Noise/Distortion Type	Hard PL Error (%)	Conjugate PL Error (%)
Gaussian Noise	88.86	**32.43**
Shot Noise	89.00	**31.74**
Impulse Noise	89.48	**39.23**
Defocus Blur	88.98	**25.93**
Glass Blur	89.04	**35.61**
Motion Blur	88.90	**30.82**
Zoom Blur	89.12	**22.81**
Snow	89.70	**32.41**
Frost	89.76	**29.99**
Fog	89.76	**38.44**
Brightness	89.38	**25.64**
Contrast	88.52	**36.21**
Elastic Transform	89.26	**31.30**
Pixelate	88.66	**25.58**
JPEG Compression	89.12	**29.55**

Table 2: Meta Test Error Rates for Various Noise and Distortion Types at Severity Level 5 using Hard Pseudo-Labels and Conjugate Pseudo-Labels with PolyLoss. Lowest errors are **bolded**.

Noise/Distortion Type	Hard PL Error (%)	Conjugate PL Error (%)
Gaussian Noise	**68.62**	78.04
Shot Noise	**65.42**	71.72
Impulse Noise	**85.7**	87.94
Defocus Blur	18.36	**17.1**
Glass Blur	**42.2**	42.52
Motion Blur	32.22	**31.6**
Zoom Blur	18.48	**17.06**
Snow	26.58	**26.34**
Frost	28.26	**27.9**
Fog	**53.26**	66.36
Brightness	18.92	**18.2**
Contrast	**79.68**	87.44
Elastic Transform	27.44	**26.92**
Pixelate	24.56	**20.7**
JPEG Compression	**26.22**	26.44

Overview of Results

The findings illustrate that conjugate pseudolabeling (CPL) almost consistently outperforms hard pseudo-labeling (HPL) under various scenarios of noise and distortion. This trend is

particularly evident when cross-entropy loss is utilized. As summarized in Table 1, CPL significantly reduces meta test error rates compared to HPL. For instance, under Gaussian noise, the error rate drops from 88.86% to 32.43%, and for defocus blur, it decreases from 88.98% to 25.93%. CPL also excels in handling complex distortions such as frost and brightness.

Table 2 presents an analysis of performance under the PolyLoss framework, where CPL exhibits competitive or superior outcomes in most cases. However, for distortions like impulse noise and contrast, HPL demonstrates a slight edge. These results might be attributed to specific dynamics of PolyLoss optimization, which could affect CPL's efficacy under these particular conditions.

Interpretation of Findings

These results reinforce the advantages of CPL in mitigating distribution shifts, especially in contexts where HPL encounters difficulties. The consistent error rate reductions underscore CPL's flexibility and resilience, establishing it as a powerful approach for test-time adaptation. The gap in performance between CPL and HPL emphasizes the significance of employing a sound theoretical basis for pseudo-labeling during evaluation phases.

The variation in results between cross-entropy loss and PolyLoss highlights the crucial role of loss function selection in shaping the success of test-time adaptation strategies. While PolyLoss offers increased versatility, the findings suggest that additional fine-tuning may be necessary to maximize CPL's potential within this framework. The usage of Cross Entropy Loss within CPL is a key contributor to the framework that makes CPL so effective within TTA.

References

Goyal, Raghunathan Kolter, Sun. 2022. "Test-Time Adaptation via Conjugate Pseudo-labels." In *Neural Information Processing Systems (NeurIPS)*. [Link]