Instance-Conditional Timescales of Decay for Non-Stationary Learning

Nishant Jain, Pradeep Shenoy

Google Research India
{nishantjn, shenoypradeep}@google.com

Abstract

Slow concept drift is a ubiquitous, yet under-studied problem in practical machine learning systems. In such settings, although recent data is more indicative of future data, naively prioritizing recent instances runs the risk of losing valuable information from the past. We propose an optimization-driven approach towards balancing instance importance over large training windows. First, we model instance relevance using a mixture of multiple timescales of decay, allowing us to capture rich temporal trends. Second, we learn an auxiliary *scorer model* that recovers the appropriate mixture of timescales as a function of the instance itself. Finally, we propose a nested optimization objective for learning the scorer, by which it maximizes forward transfer for the learned model. Experiments on a large real-world dataset of 39M photos over a 9 year period show upto 15% relative gains in accuracy compared to other robust learning baselines. We replicate our gains on two collections of real-world datasets for non-stationary learning, and extend our work to continual learning settings where, too, we beat SOTA methods by large margins.

1 Introduction

We study the problem of concept drift–a slow change in patterns of input data and label associations over time–in supervised models trained in an offline or batch-learning fashion (see e.g., (Yao et al. 2022)). Although concept drift is ubiquitous in user-facing AI applications, prevalent practice is still to train batch-learned models from scratch on newer data at regular intervals. By giving equal weight to each training instance, standard batch-learned models overemphasize past, irrelevant data. Alternative approaches favor sequential updates in a streaming data setting, including online learning (Hazan et al. 2016), and continual learning (Zeno et al. 2018; Aljundi, Kelchtermans, and Tuytelaars 2019; Delange et al. 2021). These approaches are myopic by design (i.e., only have access to the very latest data) and typically overemphasize recent data–in particular, they implicitly downweight past data exponentially as a function of instance age (see e.g., Hoeven, Erven, and Kotłowski (2018); Jones et al. (2023)).

We believe that modeling concept drift can benefit substantially from a richer representational language for importance weighting to improve forward transfer, and an optimization-driven approach towards effectively estimating the relevance of past information for future performance. Towards this end, we make 3 technical contributions: First, we model the age-dependent importance of an instance using a mixture of exponential basis functions, allowing significant flexibility in capturing temporal trends. Second, we customize importance weights for individual instances by learning an auxiliary instance-conditional *importance scor-*

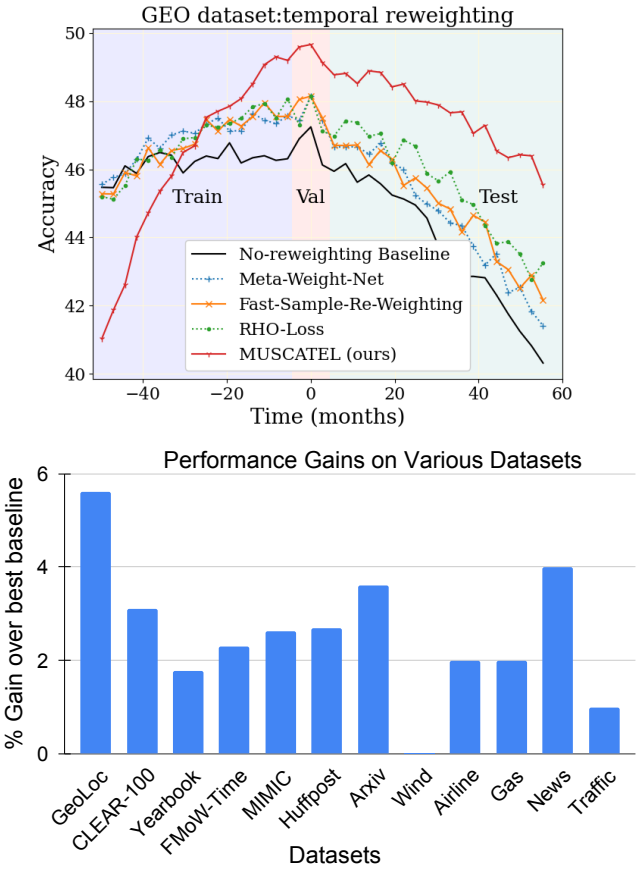

Figure 1: MUSCATEL: (a) Our approach significantly improves model performance on future data compared to best-performing baselines. (b) Gains are replicated across a wide range of real-world datasets. See text for details.

ing model–thus, instance importance is a function of both age and instance properties. Finally, we formulate and efficiently solve a nested optimization problem that jointly learns the supervised model and the scoring model with the mathematical objective of maximizing forward transfer of learned models.

We call our approach MUSCATEL: **Mu**lti-**Sca**le **Te**mporal **L**earning, and summarize our findings and results below:

- We show substantial gains (Fig. 1(a)) on a large, real-world dataset of 39M images over $\sim$ 10 years (Cai, Sener, and Koltun 2021)–we improve accuracy and reduce rate of model decay compared to baselines.
- We perform a comprehensive evaluation of MUSCATEL, combining real-world non-stationary learning datasets from 4 different sources (Cai, Sener, and Koltun 2021; Yao et al. 2022; Awasthi, Cortes, and Mohri 2023; Lin et al. 2021) for a total of 11 batch learning datasets, and 6 continual learning datasets. From an algorithmic perspective, too, we compare against SOTA algorithms in robust learning, meta-learning, and non-stationary learning. Fig. 1(b) summarizes our gains on a large number of real-world batch learning datasets over nearest SOTA for each dataset.
- We extend our approach to continual learning settings as well, by performing temporal reweighting within large minibatches, and compare against a range of online and continual learning baselines (see Jain and Shenoy (2023), appendix, for an extensive list of baselines). Across datasets and evaluation protocols, MUSCATEL shows consistent gains over and above all compared algorithms.
- We provide insight into the workings of our algorithm, and in particular our SCORER, showing that it focuses on relevant features and upweights instances in a manner that resonates with visual intuition in addition to being a key component in the substantial accuracy gains.

We believe that instance-specific temporal reweighting is a broadly applicable idea for learning under concept drift, and hope that our work contributes to vigorous discussion and development of new algorithms for this important, increasingly relevant problem.

2 Related Work

2.1 Data Drift and Adaptation

Early theoretical work on slow concept drift aimed to prove learning bounds under various models of drift such as label drift (Helmbold and Long 1994), and joint distribution drift (Bartlett 1992); see also Barve and Long (1996). Typically they advocated using only recent data for training (a windowed approach), although subsequent work (Mohri and Muñoz Medina 2012) introduced a notion of discrepancy that allowed effective use of older data. This discrepancy relies on measuring data-distribution change over time, rather than making overly restrictive assumptions. Recent work (Awasthi, Cortes, and Mohri 2023) built on this notion and proved more general performance bounds for arbitrary hypothesis function sets.

On the empirical side, multiple recent papers have proposed benchmark datasets for non-stationary learning, drawing from a wide range of applications (social media, medical records, satellite imagery, etc.) and spanning several years of data collection (Cai, Sener, and Koltun 2021; Yao et al. 2022; Lin et al. 2021) (see also (Awasthi, Cortes, and Mohri 2023) for additional benchmarks). We merge datasets from these disparate sources to perform a comprehensive, real-world evaluation of our approach applied to both batch-learning and continual learning settings.

2.2 Learning with Importance Weights

A number of recent proposals learn instance-specific parameters for training data in order to achieve certain secondary objectives; for instance, improved generalization (Ren et al. 2018; Shu et al. 2019; Mindermann et al. 2022), handling noisy labels (Vyas, Saxena, and Voice 2020), or implicit curricula for learning (Saxena, Tuzel, and DeCoste 2019). Typically the instance weights are free parameters (Ren et al. 2018) or functions of instance loss (Shu et al. 2019), and not the instance itself. Other work implicitly weights instances by sampling according to some loss criterion (Mindermann et al. 2022), in order to reduce training cost.

2.3 Continual & Online Learning

Continual Learning (CL) and online learning address scenarios where data is available in streaming fashion, and one instance (or a small buffer of instances) from the stream can be used to update the model before being discarded entirely. Settings include new task acquisition (Van de Ven and Tolias 2019; Delange et al. 2021; De Lange and Tuytelaars 2021), increased output range (Shmelkov, Schmid, and Alahari 2017; Rebuffi et al. 2017) or distribution shift whether discrete (domain-incremental CL (De Lange and Tuytelaars 2021; Delange et al. 2021)) or smooth (*concept drift* (Schlimmer and Granger 1986)). Other work in online learning aims at regret guarantees under certain models of environmental change (Herbster and Warmuth 1998), or combine an adaptive regret objective (Gradu, Hazan, and Minasyan 2020; Daniely, Gonen, and Shalev-Shwartz 2015; Duchi, Hazan, and Singer 2011; Hazan and Seshadhri 2009) with models of environment dynamics.

Due to two key reasons, we focus primarily on the batch learning problem (although we also extend our work to CL settings and compare in them): 1) CL and online learning implicitly downweight past data in a simplistic manner (exponential downweighting, see e.g., (Jones et al. 2023))–this is much less expressive than our instance- and age-dependent model of importance weights, 2) Due to inherent weaknesses (catastrophic forgetting (McCloskey and Cohen 1989), also related to previous point) and lack of access to past data for iterated learning, CL methods are vastly outperformed by batch learning except in specific settings (see e.g., our comparisons in Jain and Shenoy (2023), appendix).

3 MUSCATEL: Learning with Concept Drift

We study a learning problem where data is collected over a significantly long period of time, and the data distribution is

expected to continually evolve over time. Such concept drift is ubiquitous in practical machine learning systems, and is receiving significant attention in the academic literature recently, in benchmark development (Lin et al. 2021; Yao et al. 2022) as well as solution approaches (Cai, Sener, and Koltun 2021; Awasthi, Cortes, and Mohri 2023). We address batch learning in the face of such concept drift, and propose *instance reweighting schemes* that can effectively capture aspects of this temporal evolution, with the mathematical objective of maximizing forward transfer.

3.1 Preliminaries

Consider a supervised data stream with distribution D_t evolving with time, from which samples (x_t,y_t) are drawn at timesteps t. Here, x_t is an input instance with the corresponding label y_t. Given data collected upto time T from this stream, we aim to learn a model f_θ: $\mathcal{X} \rightarrow \mathcal{Y}$ where ($\mathcal{X}$, $\mathcal{Y}$) denote the (input, label) spaces and θ represents model parameters, in order to maximize the likelihood of future data upto some time $T + K$:

$$\prod_{k=\Delta t}^{K} \mathbb{P}\left(y_{T+k} \left| x_{T+k}, \{x_t, y_t\}_{t=1}^{T}, \theta\right.\right) \quad (1)$$

The data from T to $T + \Delta t$ is the validation data. To achieve this goal in presence of concept drift, we want to augment the loss on training instances using an importance function $\mathcal{I}mp(.)$, such that it more closely represents the loss on future data. This corresponds to decreasing the gap between the expected loss value (for instance cross entropy in classification) for the given and future data:

$$E_{(x,y)\sim D_t}[E_{t\sim p(t+T)}[l_{(x,y)}] - E_{t\sim p(t)}[\mathcal{I}mp \cdot l_{(x,y)}]] \quad (2)$$

where $l_{(x,y)} = l(y, f_\theta(x))$ is the loss function, $p(t) = \frac{1}{N}\sum_{i=1}^{N}\delta(t_i)$ where δ is the dirac-delta distribution, and t_i are the ordered timestamps at which any new data-samples were received. We operate in an offline/batch setting, where the standard learning strategy is to minimize the expected value of this desired loss over the data distribution D_t via *Empirical Risk Minimization* (ERM):

$$E_{t\sim p(t)}E_{(x,y)\sim D_t}[l(f_\theta(x), y)] \approx \frac{1}{T}\sum_{t=1}^{T} l(y_t, f_\theta(x_t)) \quad (3)$$

3.2 Modelling Temporal Drift

ERM generally performs well when the data distribution D is static. However, for evolving D_t, the approximation in eq. 3 is poor, leading to high error on test data from the future. In a real-world scenario, the relationship between $D_{t+\delta t}$ and $D_{1,\ldots,t}$ might be complex and difficult to model in a general manner. Instead, many recent works approximate the dependency using only the most recent data:

$$\mathbb{P}(D_{t+\delta t}|D_t, D_{t-\delta t_1}, ..., D_{t-\delta t_n}) \approx \mathbb{P}(D_{t+\delta t}|D_t) \quad (4)$$

With this myopic view of data evolution, an online learning approach may appear reasonable, as it implicitly places more emphasis on more recent data. In fact, previous work (Hoeven, Erven, and Kotłowski 2018) has drawn an equivalence between online learning and an exponential downweighting of past data. We therefore consider exponential downweighting of data in a batch learning setting as our first baseline. Specifically, instead of equal weights assigned to instances (Eq. (3)), we model importance as as exponentially decaying function resulting in $\hat{p}(t) = p(t)e^{-\lambda(T-t)}$ for some fixed λ which can be tuned using a validation dataset from the near future. We call this approach EXP. This modifies Eq. (3) as follows:

$$E_{t\sim\hat{p}(t)}E_{(x,y)\sim D_t}[l(f_\theta(x), y)] \approx \frac{1}{N}\sum_{i=1}^{T} \mathcal{I}mp(t) l(y_t, f_\theta(x_t)) \quad (5)$$

where the importance $\mathcal{I}mp(t) = e^{-\lambda(T-t)}$ is a function of instance age $(T - t)$. Note, in this proposed approach, we iterate multiple times over all weighted training instances; thus, it has significant advantages over online learning.

In the above model, using a single exponentially decaying function with a fixed decay rate may limit modeling flexibility. We instead broaden the definition of the importance function $\mathcal{I}mp(t)$ to a linear mixture of exponential basis functions:

$$\mathcal{I}mp(t) = \sum_k z_k e^{-a_k(T-t)} = \mathbf{z}^T e^{-\mathbf{a}(T-t)} \quad (6)$$

This increases the expressivity of $\mathcal{I}mp(t)$ by allowing us to model more heavy-tailed functions of time. Here, $\mathbf{a} = \{a_k\}_{k\in\{1,\ldots,K\}}$ are constants, and represent a basis set that capture different timescales of importance decay in the data. The K free parameters $\mathbf{z} = \{z_k\}_{k\in\{1,\ldots,K\}}$ (the mixing weights) assign relative importance to each of the timescales. We call this weighting approach MIXEXP. We make an additional design choice of setting $a_k = a_0^k$ for some fixed a_0 – this allows us to compactly represent a very wide range of timescales, while also reducing the number of free parameters in the formulation. Further, the choices of a_0, K are also not critical, as for moderate K, a very wide range of timescales are covered for any choice of a_0 ($K = 16, a_0 = 2$ in our experiments), to be mixed by the free parameters $\mathbf{z}$. Thus, we have $\mathbf{z}$ as the key hyperparameters to be tuned using the validation set.

3.3 Instance-Conditional Timescales

We come to the final, key component of our proposal for temporal importance weighting: instance-conditional timescales. In the discussions above, each instance receives an importance weight entirely controlled by its age. In practical settings, however, there are several latent variables that determine the rate of decay of importance. Consider, for instance, topics of discussion on social media–certain topics are reliably constant, while others are more short-lived. This suggests that a one-size-fits-all approach towards temporal reweighting may miss significant opportunities for optimization. To address this, we propose computing the parameters $\mathbf{z}$ as a function of the instance x, i.e.,

$$\mathcal{I}mp(x, t) = g(x)^T e^{-\mathbf{a}(T-t)} \quad (7)$$

where the function $g(x)$ now controls the scoring of instance importance in a compact, instance-conditional manner. In particular, we learn an auxiliary neural network (the

SCORER) with network parameters ϕ, i.e., $g_\phi(x)$. Note that we have deliberately chosen to separate instance-specific and time-specific components of $\mathcal{I}mp(x,t)$ in Eq. (7); indeed, experimenting with more general functions was not helpful due to the added complexity (see Jain and Shenoy (2023), appendix).

3.4 Learning a SCORER for Instance Importance

Our goal is to optimize for forward transfer, i.e., the performance of the learned classifier $f_\theta(\cdot)$ on future instances, through the use of the SCORER. This naturally leads to the following objective for the SCORER:

$$\mathcal{L}^v = \frac{1}{M}\sum_{t=T+1}^{T+\Delta t} l(y_t^v, f_\theta(x_t^v)) \tag{8}$$

i.e., learn a SCORER that minimizes loss on a small amount of data immediately following the training data (denoted as $(x_t^v,\ y_t^v)$). This loss implicitly depends on the primary model's parameters θ, which in turn depends on the SCORER's parameters. More specifically, θ is learned as:

$$\theta^*(\phi) = \arg\min_\theta \frac{1}{T}\sum_{t=1}^{T} g_\phi(x_t)\cdot e^{-\mathbf{a}(T-t)} l(y_t, f_\theta(x_t)) \tag{9}$$

with the choice of θ^* being a function of the SCORER's parameters ϕ. Thus, optimizing the SCORER objective in Eq. (8) can be written as *an outer optimization* over the above optimization for θ^*:

$$\phi^* = \arg\min_\phi \frac{1}{M}\sum_{t=T+1}^{T+\Delta t} l(y_t^v, f_{\theta^*}(x_t^v)) \tag{10}$$

where, again, the objective is implicitly a function of ϕ through the dependence on $\theta^*(\phi)$.

Optimization: The above *bilevel optimization* structure has been used by previous work, e.g., in gradient-based based hyperparameter optimization methods (Lorraine, Vicol, and Duvenaud 2020; Blondel et al. 2021), and in reweighting for mitigating label noise (Shu et al. 2019; Zhang and Pfister 2021). The former works use implicit gradients for the outer optimization, while the latter used an approximation of a one-step unroll of the inner optimization, and alternating updates to (θ, ϕ). These two approaches have complementary strengths; implicit gradients allow for more precise optimization, while also incurring additional computation costs involving calculating the Hessian. We discuss both below.

Given the objective of the SCORER in eq. 10 and θ being a function of ϕ (eq. 9), the aggregated objective of the SCORER can be written as $\mathcal{L}(v) = G^v(\theta^*)$ and its gradient w.r.t. ϕ can be calculated using chain rule as follows:

$$\frac{\partial G^v(\theta^*)}{\partial \phi} = \frac{\partial G^v(\theta^*)}{\partial \theta}\frac{\partial \theta^*}{\partial \phi} \tag{11}$$

Calculating the second term in the above equation requires the gradient of optimal classifier parameters with respect to SCORER parameters and can be calculated as follows (refer Jain and Shenoy (2023), appendix for more details) :

$$\frac{\partial \theta^*}{\partial \phi} = -\left[\frac{\partial^2 \mathcal{L}_{tr}}{\partial\theta\partial\theta^T}\right]^{-1} \times \frac{\partial^2 \mathcal{L}_{tr}}{\partial\theta\partial\phi^T}\bigg|_{\theta^*,\phi} \tag{12}$$

where $\mathcal{L}_{tr}$ denotes the training objective from eq. 9. The first term corresponds to the inverse of a Hessian and the second is a second-order term involving gradient w.r.t. SCORER followed by target network parameters. We followed recent work on implicit differentiation to approximate the inverse of the Hessian term (Lorraine, Vicol, and Duvenaud 2020). This leads to a nested optimization setting where we update the SCORER for every L updates to the classifier, and the classifier parameters after these L updates are denoted as $\hat{\theta^*}$, approximating the optimal classifier parameters in Eq. (10). For more details regarding the implicit gradient and approximation, please refer Jain and Shenoy (2023), appendix.

Alternating updates: As discussed above, some works (Shu et al. 2019; Ren et al. 2018) avoid this term and use an online approximation to arrive at a single optimization loop with alternating updates to θ and ϕ. We also implemented this variant for our formulation; our findings were qualitatively unchanged, with a modest quantitative trade-off between runtime and accuracy between the two options (see Jain and Shenoy (2023), appendix). The final update for θ, ϕ, comes out to be as follows:

$$\begin{aligned}
\phi_{b+1} &= \phi_b + \alpha\beta\frac{1}{MN}\sum_{i=1}^{T}\sum_{j=T+1}^{T+\Delta t}\frac{\partial}{\partial\theta} l(y_j^v, f_\theta(x_j^v))\bigg|_{\theta_b} \\
&\quad \cdot \frac{\partial}{\partial\phi} g_\phi(x_{i_1})\bigg|_{\phi_b} \cdot e^{-\mathbf{a}t_i}\frac{\partial}{\partial\theta} l(y_i, f_\theta(x_i))\bigg|_{\theta_b} \\
\theta_{b+1} &= \theta_b - \beta\cdot g_{\phi_b}(x_i)\cdot e^{-\mathbf{a}t_i}\frac{1}{N}\sum_{i=1}^{T}\frac{\partial}{\partial\theta} l(y_i, f_\theta(x_i))\bigg|_{\theta_b}
\end{aligned} \tag{13}$$

where $M = \Delta t$ and b is the number of epochs. A detailed derivation of the optimization process along with the above approximations is provided in Jain and Shenoy (2023).

4 Experiment Setup

4.1 Datasets

Geolocalization. We experiment extensively with the geolocalization (GEO) dataset proposed by (Cai, Sener, and Koltun 2021): 39M images from YFCC100M (Thomee et al. 2016), spanning 2004-2014 and containing natural distribution shifts due to changes in image contents. The task is classification of image region-of-origin (712 geolocations spanning the globe). Images timestamps are used to conduct temporal learning & evaluation experiments. The authors (Cai, Sener, and Koltun 2021) show evidence of gradual distribution shift through a range of experiments in a CL setting.

Wild-Time Benchmark. This collection of 5 datasets captures real-world concept drift (Yao et al. 2022)– Yearbook, FMoW-time, MIMIC, Huffpost, and arxiv –and spans multiple years. The data covers multiple domains (American high school yearbook photos, satellite images, medical records, news headlines, arxiv preprints). A quick

summary is in Jain and Shenoy (2023), appendix; please also refer to the source paper for more details.

CLEAR Benchmark. We also experiment with the CLEAR-100 Benchmark (Lin et al. 2021), designed to evaluate continual algorithms under realistic distribution shifts and a single-task setting. Like GEO, CLEAR is also sampled from YFCC100M (8M images, sorted in order of timestamps), and aims to capture slow drift in the visual appearance of objects such as laptops, cameras, *etc.*. The authors divide the data into 10 temporal buckets and suggest evaluation schemes for testing CL models.

Other Real-World drifting datasets. We further compare on more real-world datasets inheriting concept drift based on a recent paper (Awasthi, Cortes, and Mohri 2023). We compare on both regression and classification tasks following the paper. For the regression task Wind, Airline, Gas, News, Traffic datasets are used. We follow the same setup as the (Awasthi, Cortes, and Mohri 2023) paper, using a completely different source as well as time-segment for testing. The datasets capture the temporal variation ranging from 1 day to 26 years. For more details on these and for description of the classification datasets, please refer Jain and Shenoy (2023).

4.2 Batch Training & Baselines

Our primary focus is on conventional supervised learning where a model is trained to convergence by iterating over a single, large training dataset. For GEO dataset, no baselines for this setup were proposed in the paper, so we propose the following list. Although they were not designed for concept drift, they all (like our approach) use a specialized validation set as "target" for optimizing model performance, and indeed do outperform the ERM baseline:

MetaWeightNet: Reweights training instances as a function of instance loss, to minimize loss on a given meta-dataset (Shu et al. 2019).

Fast sample reweighting: Loss-based reweighting of instances similar to MetaWeightNet, with automatic selection and updates to the meta-set (Zhang and Pfister 2021).

RHO-loss: Ranking training instances for each batch based on minimizing loss on a hold-out set, and selecting top-ranked instances for training (Mindermann et al. 2022).

For the WILD-TIME datasets, we report on all the baselines reported in that paper; likewise, we compare against the already published benchmarks reported in Awasthi, Cortes, and Mohri (2023) for the additional datasets contained there (please refer to the respective papers for details).

We split each dataset into train, validation, and test sets in temporal order, reusing existing partitions where available. Thus, validation data is more recent than train data. For the GEO dataset (see Fig. 1(a)), these sets contain 18M, 2M, 19M images covering time periods of around 54,6,60 months respectively. In addition, for this dataset we also held out small data samples from the train and validation period too, so we could compare train/test period accuracy in an unbiased manner. All methods are trained to converge, and performance on the test period is reported as metric (classification accuracy, or RMSE for regression).

Method \ Year	1	2	3	4	5
ERM	45.9	45.4	44.1	42.7	41.0
MWNet	46.8	46.3	44.8	43.7	42.0
FSR	46.9	46.1	45.2	44.1	42.6
RhoLoss	47.2	46.7	46.0	44.6	43.3
MUSCATEL(ours)	49.1	48.7	48.4	47.6	47.2
%gain vs ERM	**6.7**	**7.1**	**9.8**	**11.5**	**15**
%gain vs SOTA	**3.9**	**4.1**	**5.1**	**6.8**	**8.8**

Table 1: Year-wise error rate for our test set from Geolocalization Dataset. Our method outperform all the other baselines by significant margins.

4.3 Continual Learning & Baselines

We also evaluate our approach in the *offline continual learning setting*, where data is presented to the model in small buckets in sequence, and each bucket is processed and discarded before the next bucket is made available.

We compare against a suite of recently proposed popular CL baselines including Reservoir Sampling (Kim, Jeong, and Kim 2020), MiR (Aljundi et al. 2019), GDUMB (Prabhu, Torr, and Dokania 2020); see Jain and Shenoy (2023), appendix for additional comparisons namely ER (Chaudhry et al. 2019), LwF (Li and Hoiem 2017), EwC (Kirkpatrick et al. 2017), AGEM (Chaudhry et al. 2018) and other recent methods. In addition, we compare against two recent task free-continual learning methods proposed specifically for non-stationary data streams– Continual Prototype Evaluation(De Lange and Tuytelaars 2021) and SVGD (Wang et al. 2022). Finally, we compare against representative online learning algorithms designed for non-stationary data–Ader (Zhang, Lu, and Zhou 2018) and Scream (Zhao, Wang, and Zhou 2022). Please see Jain and Shenoy (2023) for detailed descriptions of the various baselines.

We split GEO into 39 buckets of 1M images each, with one bucket roughly spanning 3 months. For CLEAR, we use the provided temporal data splits (10 buckets of 160k images each plus a corresponding test set). For adapting MUSCATEL to this continual setting, we use the most recent 10% of each bucket as validation data. Other baselines use the whole bucket (including our validation) as training data. Our temporal reweighting operates only over the time span of the bucket (3 months), since older data are not available in the continual learning setup. All algorithms iterate over each bucket of data k times (k=5 for GEO, and as per the benchmark proposal for CLEAR100).

5 Results

5.1 MUSCATEL Maximizes Forward Transfer

We first illustrate the complexities of non-stationary learning, and the substantial benefits of our approach, on a large-scale real-world dataset (GEO, 39M photos spanning a 9 year period) (Fig. 1(a)), followed by quantitative analysis on a range of benchmark datasets (Tab. 1,Tab. 2,Tab. 3).

Method	YB		FMoW		MIMIC		HP		Arxiv	
	ID	OOD	ID	OOD	ID	OOD	ID	OOD	ID	OOD
ERM	97.99	79.50	58.07	54.07	71.30	61.33	79.40	70.42	53.78	45.94
GroupDRO-T	96.04	77.06	46.57	43.87	69.70	56.12	78.04	69.53	49.78	39.06
mixup	96.42	76.72	56.93	53.67	70.08	58.82	80.15	71.18	50.72	47.82
LISA	96.56	83.65	55.10	52.33	70.52	56.90	78.20	69.99	50.72	47.82
CORAL-T	98.19	77.53	52.60	49.43	70.18	57.31	78.19	70.05	53.25	42.32
IRM-T	97.02	80.46	46.60	45.00	72.33	56.53	78.38	70.21	46.30	35.75
MUSCATEL	**98.13**	**82.78**	**59.87**	**56.95**	**72.80**	**64.12**	**81.56**	**73.12**	**55.89**	**49.57**

Table 2: Average accuracy on various datasets of the Wild-Time benchmark for both In-Dist. and OOD due to temporal shift setting. Our method is able to outperform all other methods on both In-Dist. and OOD settings, even though it downweights past. This shows it can robustly learn time-invariant features.

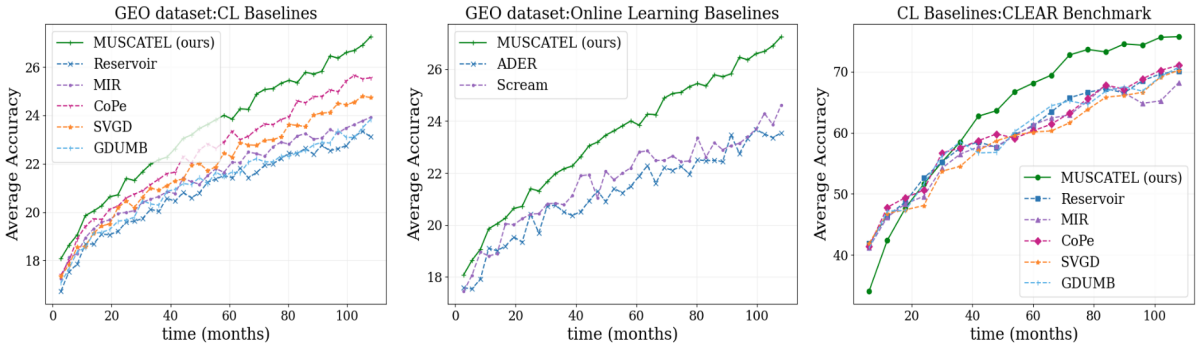

Figure 2: Comparison of MUSCATEL against various continual learning baselines, on the GEO dataset, with CL baselines (a) and non-stationary online learning methods (b). Panel (c) shows comparison on the CLEAR-100 concept drift benchmark for continual learning, where, too, we show clear gains.

Method	DM	MM	EXP	BSTS	SDRIFT	Ours
Wind	1.12	1.19	0.98	0.98	**0.95**	**0.95**
Airline	1.78	1.41	0.98	0.94	0.94	**0.92**
Gas	0.42	0.47	0.94	1.02	0.40	**0.38**
News	1.13	1.10	0.98	1.00	0.97	**0.93**
Traffic	2.20	0.99	0.996	0.98	0.96	**0.95**

Table 3: Comparison with drift datasets and baselines from the SDRIFT paper (Awasthi, Cortes, and Mohri 2023). Our method consistently performs best across all datasets.

Figure 1(a) compares various baselines for batch training on the GEO dataset, over the training period (past data) as well as the testing period (future data). First, ERM has flat performance over the training window, but decays rapidly over time in the test period, confirming the non-iid nature of the data. Most methods beat ERM on the test period, strongly supporting the value of using a *temporally appropriate validation set* as a "target set" in concept drift scenarios. MUSCATEL outperforms other methods by a significant margin over the test period (upto 15% relative), showcasing the power of our temporal importance weighting approach. Note that MUSCATEL clearly trades off past accuracy in favor of more recent data; this is due to the meta-objective of maximizing validation accuracy, which is more recent and more representative of future (test) data.

5.2 Quantitative, Cross-Dataset Gains

We first quantified the gains of our approach as a function of time since the models were trained; Tab. 1 compiles these findings (cf. Fig. 1(a)). We note a steep decay in performance for all algorithms; however, MUSCATEL not only beats the baselines consistently but by a widening margin as time passes, with upto 15% relative gains over ERM, and 8.8% relative over the nearest baseline.

Next, we replicate our findings on a range of datasets included in the recent Wild-Time concept drift benchmark suite (Yao et al. 2022)–5 real-world datasets spanning social media, satellite imagery, and medical records over long time spans. Here, older data is used for training and "in-domain (ID)" testing, and newer data for "out-of-domain (OOD)" testing. Tab. 2 shows model accuracy over training and test periods (past & future respectively) for a range of baselines included in the benchmark. Note the following trends: a) large gaps between ID & OOD accuracies, highlighting the data drift, b) No baseline consistently outperforms the others across datasets, and c) MUSCATEL handily beats all baselines in both in-domain and out-of-domain accuracy.

We then evaluated another set of non-stationary regression datasets (see e.g., Awasthi, Cortes, and Mohri (2023) for dataset details & baselines). Again, MUSCATEL consistently beats other baselines by noticeable margins in the test

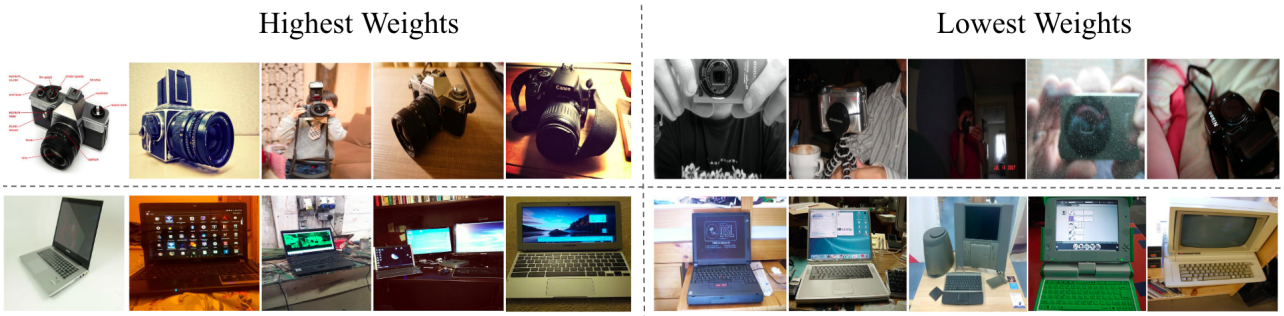

Figure 3: Visualizing the highest weighted examples, using our SCORER on the CLEAR-100 benchmark.

period (Tab. 3), confirming the value of our approach in non-stationary offline learning. Fig. 1(b) summarizes relative % gain of our approach across datasets, over and above nearest baseline in each datasets.

5.3 Adapting MUSCATEL for Continual Learning

Although our focus is on importance weighting in the widely-prevalent batch learning scenario, our broader goal is to address learning when data distributions change slowly over time. Since continual learning is an alternate approach for non-stationary learning, we adapted our reweighting scheme for *offline continual learning*, and compared against SOTA techniques in this domain (see Section 4.3 for more details). MUSCATEL clearly outperforms these baselines on the GEO dataset (Fig. 2a), and many others (see Jain and Shenoy (2023), appendix), as well as *online learning* algorithms designed with theoretical guarantees for non-stationary learning (Fig. 2(b)). Finally, we also replicated these findings on the recently proposed CLEAR benchmark (Fig. 2(c)), designed explicitly to test CL algorithms against concept drift.

5.4 Understanding the SCORER's Role

We summarize various analyses showcasing the value of our auxiliary SCORER model in producing gains, as well as some insights into its workings. Detailed data is presented in the extended version (Jain and Shenoy 2023) due to space considerations; we only summarize findings here.

Value add: We compared temporal scoring options on the GEO dataset, and found a strict ordering in performance: ERM $<$ EXP$<$ MIXEXP$<$ INSTMIXEXP, suggesting that each of the design choices in our approach provided noticeable additional gains.

Interpretability: (1) Post-hoc analysis of INSTMIXEXP weights shows a shallower dropoff on average as a function of time compared to exponential fit; however, in any given temporal bucket, the instance weights varied substantially. This supports the idea that the SCORER plays a large role in instance-specific customization of weights (cf. Eq. (7)). (2) Figure 3 shows the most and least SCORER-weighted images on the CLEAR dataset (a collection of object images through time), clearly showcasing that the SCORER emphasizes more modern-looking instances, and conform to our intuitions & expectations. GradCam analysis (Selvaraju et al. 2017) of the SCORER's most relevant features confirms that the network focuses on the primary object in the image, and associated relevant features for determining its importance for forward transfer (see Jain and Shenoy (2023), appendix).

Tuning: Our results are relatively insensitive to the choice of the parameter a_0. Scaling a_0 by 0.5,2, or 4 showed minimal influence on the results. Similarly, given the geometric spacing of the a_i's , increasing K beyond a point made no difference. In short, our results are stable over a very broad range of settings for these hyperparameters, by design and in practice (see Jain and Shenoy (2023) for details).

Cost: 1) All numbers presented in this work used a simple 4-layer CNN for the SCORER; increasing SCORER complexity improved accuracy marginally, up to a point. 2) Per epoch running time for MUSCATEL is consistently 1.2x-1.5x that of the ERM baseline, and convergence is roughly similar.

Please see supplementary materials in Jain and Shenoy (2023) for a range of additional experiments including a more general SCORER $g(x, t)$ that is not constrained to a mixture of exponentials, other modifications to the temporal reweighting functions, a head-to-head comparison of CL and batch learning, etc., and implementation details omitted due to lack of space.

6 Conclusion

We address the problem of supervised learning in the face of slow concept drift, and propose MUSCATEL–instance-specific multi-scale temporal importance weighting–as an approach towards addressing this issue. We propose a SCORER to compactly represent instance-specific importance weights, and a nested optimization objective with an efficient implementation for learning the SCORER. Extensive experiments on a suite of real-world datasets confirm substantial concept drift, and show MUSCATEL providing significant performance gains in both offline/ batch training & continued training settings. Interestingly, MUSCATEL improves both backward and forward transfer performance compared to baseline, despite its use of weighted discounting for past data. Our results also show that under non-stationary data conditions, standard CL approaches are at a significant disadvantage compared to batch training, which should be preferred where practicable.

References

Aljundi, R.; Caccia, L.; Belilovsky, E.; Caccia, M.; Lin, M.; Charlin, L.; and Tuytelaars, T. 2019. Online Continual Learning with Maximally Interfered Retrieval.

Aljundi, R.; Kelchtermans, K.; and Tuytelaars, T. 2019. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11254–11263.

Awasthi, P.; Cortes, C.; and Mohri, C. 2023. Theory and algorithm for batch distribution drift problems. In *International Conference on Artificial Intelligence and Statistics*, 9826–9851. PMLR.

Bartlett, P. L. 1992. Learning with a slowly changing distribution. In *Proceedings of the fifth annual workshop on Computational learning theory*, 243–252.

Barve, R. D.; and Long, P. M. 1996. On the complexity of learning from drifting distributions. In *Proceedings of the ninth annual conference on Computational learning theory*, 122–130.

Blondel, M.; Berthet, Q.; Cuturi, M.; Frostig, R.; Hoyer, S.; Llinares-López, F.; Pedregosa, F.; and Vert, J.-P. 2021. Efficient and modular implicit differentiation. *arXiv preprint arXiv:2105.15183*.

Cai, Z.; Sener, O.; and Koltun, V. 2021. Online continual learning with natural distribution shifts: An empirical study with visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8281–8290.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H.; and Ranzato, M. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Daniely, A.; Gonen, A.; and Shalev-Shwartz, S. 2015. Strongly adaptive online learning. In *International Conference on Machine Learning*, 1405–1411. PMLR.

De Lange, M.; and Tuytelaars, T. 2021. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8250–8259.

Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Gradu, P.; Hazan, E.; and Minasyan, E. 2020. Adaptive regret for control of time-varying dynamics. *arXiv preprint arXiv:2007.04393*.

Hazan, E.; and Seshadhri, C. 2009. Efficient learning algorithms for changing environments. In *Proceedings of the 26th annual international conference on machine learning*, 393–400.

Hazan, E.; et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4): 157–325.

Helmbold, D. P.; and Long, P. M. 1994. Tracking drifting concepts by minimizing disagreements. *Machine learning*, 14: 27–45.

Herbster, M.; and Warmuth, M. K. 1998. Tracking the best expert. *Machine learning*, 32(2): 151–178.

Hoeven, D.; Erven, T.; and Kotłowski, W. 2018. The many faces of exponential weights in online learning. In *Conference On Learning Theory*, 2067–2092. PMLR.

Jain, N.; and Shenoy, P. 2023. Instance-Conditional Timescales of Decay for Non-Stationary Learning. arXiv:2212.05908.

Jones, M.; Scott, T. R.; Ren, M.; ElSayed, G.; Hermann, K.; Mayo, D.; and Mozer, M. 2023. Learning in temporally structured environments. *ICLR*.

Kim, C. D.; Jeong, J.; and Kim, G. 2020. Imbalanced continual learning with partitioning reservoir sampling. In *European Conference on Computer Vision*, 411–428. Springer.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

Lin, Z.; Shi, J.; Pathak, D.; and Ramanan, D. 2021. The CLEAR Benchmark: Continual LEArning on Real-World Imagery. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Lorraine, J.; Vicol, P.; and Duvenaud, D. 2020. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, 1540–1552. PMLR.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Mindermann, S.; Brauner, J. M.; Razzak, M. T.; Sharma, M.; Kirsch, A.; Xu, W.; Höltgen, B.; Gomez, A. N.; Morisot, A.; Farquhar, S.; et al. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, 15630–15649. PMLR.

Mohri, M.; and Muñoz Medina, A. 2012. New analysis and algorithm for learning with drifting distributions. In *Algorithmic Learning Theory: 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings 23*, 124–138. Springer.

Prabhu, A.; Torr, P. H.; and Dokania, P. K. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, 524–540. Springer.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, 4334–4343. PMLR.

Saxena, S.; Tuzel, O.; and DeCoste, D. 2019. Data parameters: A new family of parameters for learning a differentiable curriculum. *Advances in Neural Information Processing Systems*, 32.

Schlimmer, J. C.; and Granger, R. H. 1986. Beyond incremental processing: tracking concept drift. In *AAAI*, 502–507.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shmelkov, K.; Schmid, C.; and Alahari, K. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, 3400–3409.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2): 64–73.

Van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.

Vyas, N.; Saxena, S.; and Voice, T. 2020. Learning soft labels via meta learning. *arXiv preprint arXiv:2009.09496*.

Wang, Z.; Shen, L.; Fang, L.; Suo, Q.; Duan, T.; and Gao, M. 2022. Improving task-free continual learning by distributionally robust memory evolution. In *International Conference on Machine Learning*, 22985–22998. PMLR.

Yao, H.; Choi, C.; Cao, B.; Lee, Y.; Koh, P. W. W.; and Finn, C. 2022. Wild-time: A benchmark of in-the-wild distribution shift over time. *Advances in Neural Information Processing Systems*, 35: 10309–10324.

Zeno, C.; Golan, I.; Hoffer, E.; and Soudry, D. 2018. Task agnostic continual learning using online variational bayes. *arXiv preprint arXiv:1803.10123*.

Zhang, L.; Lu, S.; and Zhou, Z.-H. 2018. Adaptive online learning in dynamic environments. *Advances in neural information processing systems*, 31.

Zhang, Z.; and Pfister, T. 2021. Learning fast sample re-weighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 725–734.

Zhao, P.; Wang, Y.-X.; and Zhou, Z.-H. 2022. Non-stationary online learning with memory and non-stochastic control. In *International Conference on Artificial Intelligence and Statistics*, 2101–2133. PMLR.