Enhancing MLOps efficiency through Distributionally Invariant Models

Sanyog Dewani, Sambit Das, Aman Gulati, Purav Aggarwal, Vikalp Gajbhiye, and Srujana Merugu

Amazon

Abstract. MLOps has emerged as a critical challenge in the field of artificial intelligence, due to the necessity for continual model updates. This requirement arises from common occurrences such as model degradation, shifts in input data distribution and changes in incidence rates. A significant bottleneck in these automated updates is the drift in the output score distribution that requires incremental effort from model consumers to change their operating points or refresh their downstream models, a cost that must be borne with each model update. Existing works address these issues via score calibration or learning an independent mapping function between the new distribution and the old one. We propose to generate *distributionally invariant models* at the time of model training using additional constraint of distribution matching in the loss function. We showcase the superiority of using Wasserstein distance for distribution matching over the commonly employed KL divergence. Experimental results across datasets demonstrate that the proposed approach maintains the positive prediction counts with an average recall lift of 100 bps and is versatile under challenging situations of high distribution drifts, imbalanced datasets and diverse model architectures.

1 Introduction

MLOps (Machine Learning Operations) is crucial for managing large-scale machine learning models in production, involving continuous monitoring, re-training, evaluation, and deployment. Model performances decay over time due to changes in input features or incidence rates, necessitating regular updates. For example, in the domain of fraud detection, a skewed operating point may be used for automation purposes where the precision requirements are high while a relaxed operating point is preferred for manual investigation. Figure 1 highlights how a change in model score distribution (highlighted as previous and new in the figure) can severely impact downstream consumers requiring costly and risky adjustments to operating points, rule sets, and dependent models. Therefore, it is crucial to ensure that re-training does not lead to a change in the model score distribution and that the new model remains "distributionally invariant".

Existing approaches focus on calibrating model scores or performing post-hoc distribution matching through transformation functions. Calibration ensures that model outputs accurately represent outcome probabilities, but calibrated models can still experience distribution shifts due to changing incidence

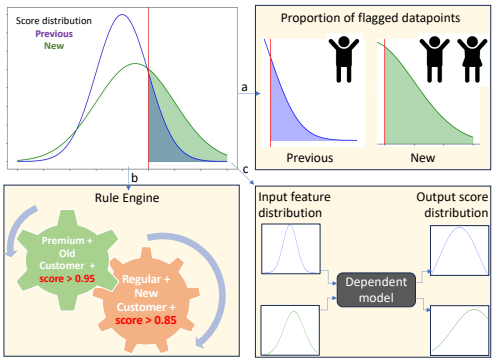


Fig. 1: Downstream impact of distribution drift in model score distribution. Top-Left: showcases the drift in model scores distribution due to model re-training. Here a, b and c showcase the impact of using new (vs previous) score distribution. Top-right (a): showcases an inadvertent impact on a larger number of customers at a specific operating point due to change in output score distribution. Bottom-left (b): are the dependent rules on the model scores, fine-tuned at different operating points represented by high/low risk in red but are now invalid. Bottom-right (c): presents the cascading effect that this can have on downstream models consuming the model scores as an input feature.

rates. Additionally, calibration is often overlooked in production due to time constraints, lack of awareness, and perceived adequacy of un-calibrated models. Learning post-hoc transformation functions can help maintain performance but the distribution matching might be sub-optimal especially on skewed operating points and datasets with high distribution shifts. Point-to-point learning captures only the local patterns and thus cannot assure a global level distribution matching as observed for one of our datasets in Figure 2. Also, the approach has additional maintenance overhead owing to the extra transformation model.

In this paper, we propose a novel Distributionally Invariant re-training Methodology (DIM) that enhances model performance without altering the output score distribution of previously deployed models. DIM demonstrates superior backward compatibility and comparable performance to existing methods such as post-hoc learning and score calibration on multiple real-world datasets. DIM maintains overall positive prediction counts with an average recall improvement of 100bps on public datasets, potentially saving $\sim$30% bandwidth per model refresh in production, leading to significant monetary savings.

We discuss the advantages of the Wasserstein metric over KL divergence for distribution matching constraints under significant incidence rate drifts and skewed operating points. While presented for binary class datasets with tabular features, the approach is adaptable to multi-class scenarios and classification based language models. We demonstrate that DIM is agnostic to model archi-

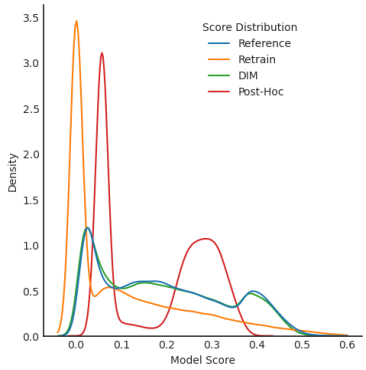


Fig. 2: Comparison of output score distribution of various retraining methods with reference model score distribution for `DELINQUENCY` dataset. Vanilla retrained model generates significantly different score distribution vs reference model. DIM is able to closely map the reference model score distribution while post-hoc transformation introduces significant deviation in the transformed score distribution.

tecture, enabling upgrades to newer generation architectures like transformers (eg: BERT) without compromising distributional invariance.

2 Related Work

Score Calibration: In classification tasks, significant work has focused on score calibration to obtain well-calibrated class probability estimates from model outputs. Non-parametric approaches for score calibration include Isotonic Regression [20], Gaussian Process-based calibration [18], Spline calibration [5] and Bayesian binning into quantiles [10]. Parametric approaches consist of Platt scaling [12] and Temperature scaling [4]. However, these techniques are unable to ensure distribution invariance in case the previous model is un-calibrated. Also, retrained calibrated models are not robust to distribution drift due to changes in incidence rate.

Backward Compatible Training: In contrast to extensive work on score calibration, limited research has been done to ensure distribution invariance during model training. Some approaches [11,19,13] focus on similarity of features between new and old models. Trauble et. al. [17] propose a Bayesian approach to update classifier predictions while minimizing negative flips. Hu et. al. [7] suggest backward compatible embeddings using an additional loss term. Additionally, post-hoc transformations such as Support Vector Regression (SVR) [6] can also be applied to retrained model scores to achieve backward compatibility.

Distributionally Invariant Model: Our work focuses on joint training by adding a distribution matching constraint to the loss function. Unlike previous works that primarily focus on embedding compatibility or decision preservation

of reference model, we address distribution invariance directly by integrating divergence estimation metrics such as Kullback-Leibler (KL) divergence [8] and Wasserstein distance, which has been popularized by Generative Adversarial Networks (GANs) [1,3].

Distribution Invariance Metrics: Evaluating model efficacy is crucial for ensuring distributional invariance relative to previous models. Srivastava et al. [14] propose two key metrics for backward compatibility: (a) Backward Trust Compatibility (BTC), which measures the fraction of correct predictions after a model update, and (b) Backward Error Compatibility (BEC), which indicates the probability that an incorrect prediction post-update is not new.

3 Problem Setting

We have a reference model M^{ref} trained and validated on dataset D^{ref}. Each dataset contains two sample sets, one for model training and other for model validation. For the sake of brevity, we do not refer to train and validation sets in notations separately. The model performance of M^{ref} may deteriorate on a recent time period dataset D^{new} due to change in incidence rate and thus necessitate retraining. Let $M^{retrain}$ denote a model retrained on data D^{new}. Both M^{ref} and $M^{retrain}$ are trained with a binary cross entropy BCE loss.

Let $\hat{Y}^{retrain}_{new}$ and $\hat{Y}^{ref}_{new}$ denote the vectors of positive class probabilities from the models $M^{retrain}$ and M^{ref} on data D^{new}. These two vectors are likely to diverge significantly due to change in incidence rate. Using $\hat{Y}^{retrain}_{new}$ in downstream systems can lead to adverse impact like degradation in performance of downstream models consuming these scores, widely different precision/recall values of business rules using fixed operating points. To alleviate this issue we propose, in section 4, Distributionally Invariant Models M^{DIM} with two objectives (i) similar distribution of model predictions in comparison to predictions from reference model (ii) minimal loss in model performance metrics such that $AUC(Y_{new}, \hat{Y}^{DIM}_{new}) \sim AUC(Y_{new}, \hat{Y}^{retrain}_{new})$ where Y_{new} are the actual labels. While we limit our exploration to binary classification use cases where there is a decay in model performance due to distribution shift, the formulation is general and can be to extended to other use like multi-class classification and model architecture changes to enhance performance.

4 Methodological Framework

We present our proposed approach to build Distributionally Invariant Models (DIM) so that any model refresh need not lead to a re-evaluation exercise for each of the downstream systems consuming the model scores. We use dataset D^{new} to train the new model M^{DIM}. Our model architecture comprises of a loss function with two components. In addition to the vanilla classification loss (BCE), we add a distribution matching constraint to match the score distributions of $\hat{Y}^{ref}_{new}$ and $\hat{Y}^{\theta}_{new}$ where θ are the parameters of M^{DIM}. This augmented training not only enables learning of the enriched representations specific to assigned task

but also the appropriate transformation to match the new score distribution to the reference distribution with minimum loss of information. We also introduce a trade-off hyperparameter λ and optimize it to achieve lowest possible divergence loss while maintaining model performance. M^{DIM} is obtained by optimising the parameters θ in Eq. 1.

$$M^{DIM} = \arg\min_{\theta} \Big(BCE(\hat{Y}^{\theta}_{new}, Y_{new}) + \lambda \cdot d(\hat{Y}^{\theta}_{new}, \hat{Y}^{ref}_{new}) \Big) \tag{1}$$

Choice of Divergence Function: Divergences that are utilised for distribution matching between two distributions P and Q can be divided into two broad categories: a) Integral probability-based divergences ($D_{\mathbb{H}}(\cdot,\cdot)$) - a function that optimises the difference in expectations as in Eq. 2, b) ϕ-divergences ($D_{\phi}(\cdot,\cdot)$) - a function that considers the expectation of the ratios of the two distributions as in Eq. 3. We experiment with Wasserstein metric from the former and KL divergence from the latter and showcase the performance of each for distribution matching. We denote DIM with Wasserstein distance and KL divergence loss as as M^{DIM}_{Wass} and M^{DIM}_{KL} respectively.

$$D_{\mathbb{H}}(P,Q) = \sup_{g \in \mathbb{H}} |E_{X\sim P} g(X) - E_{Y\sim Q} g(Y)| \tag{2}$$

$$D\phi(P,Q) = \int_{\chi} q(x)\phi(\frac{P(x)}{Q(x)})dx \tag{3}$$

Wasserstein distance and KL divergence capture different aspects of distribution matching. While Wasserstein distance is sensitive to differences in the tails of the distributions, KL divergence emphasizes on differences in the densities around the mode (reverse KL) or mean (forward KL). In cases of high distribution shift, where overlap between $P(x)$ and $Q(x)$ is minimal, Wasserstein remains finite by measuring transportation cost between supports (Eq. 4) [16,9], while KL divergence $\to \infty$ when $\text{supp}(P) \cap \text{supp}(Q) \to \emptyset$ (Eq. 5). For general non-overlapping distributions wasserstein distance can be estimated with:

$$W_p(P,Q) = \left(\inf_{\gamma \in \Gamma(P,Q)} \int d(x,y)^p d\gamma(x,y) \right)^{1/p} < \infty \tag{4}$$

where $\Gamma(P,Q)$ denotes the set of all joint distributions (couplings) with marginals P and Q, and $d(x,y)$ is the ground metric defining distances in the underlying space X, whereas for KL divergence:

$$D_{KL}(P \parallel Q) = +\infty \quad \text{when } P \not\ll Q \tag{5}$$

Additionally, Wasserstein's gradient (Eq. 6) stability [15] derives from its formulation as an expectation over the optimal coupling γ^* rather than probability ratios [2]. The log-ratio term in KL divergence's gradient (Eq. 7) creates an unbounded loss landscape which can lead to vanishing gradients under support mismatch ($\lim_{P_\theta \perp Q} |\nabla_\theta D_{KL}| = 0$) or exploding variance that are sensitive to distributional shifts.

$$\nabla_\theta W_p(P_\theta, Q) = \mathbb{E}(x,y) \sim \gamma^* \left[\nabla\theta c(x,y)\right] \tag{6}$$

where γ^* is the optimal transport plan minimizing $\mathbb{E}_{(x,y)\sim\gamma}[d(x,y)^p]$

$$\nabla_\theta D_{KL}(P_\theta|Q) = \mathbb{E}x \sim P\theta \left[\nabla_\theta \log P_\theta(x) \left(\log \frac{P_\theta(x)}{Q(x)} + 1\right)\right] \tag{7}$$

We critically evaluate the relevance of DIM by probing the proposed methodology along the following research questions which we explore in detail in section 5.3.

- **RQ1**: How do model score distributions of DIM, reference model and benchmark techniques compare against each other?
- **RQ2**: How consistent are the predictions of DIM and benchmark techniques against the reference model using evaluation metrics such as BTC?
- **RQ3**: How does DIM model performance fare against retrained model and benchmark techniques of distribution matching?
- **RQ4**: How to select the optimal value of regularization hyper-parameter (λ)?

5 Experiment Results

5.1 Datasets

To showcase the efficacy of DIM on different settings, we selected two public datasets with varying degrees of distribution drift in the target variable. Additionally, we use an internal dataset to test the applicability across changes in model architecture.

External datasets: We have used two external datasets - (a) Australia rain prediction (`RAINFALL` - Low Distribution Drift): This dataset contains 10 years (2007-2017) of daily weather observations from several locations across Australia with the objective of predicting whether it rained the next day. The data is split into two time periods T1 (2007-2013) and T2 (2013-2017) and we observe a rainfall distribution shift of 180 bps from T1 to T2. (b) Loan Delinquency Prediction (`DELINQUENCY` - High Distribution Drift): This dataset contains loans approved by lending club from 2007-2018 and whether the loan was written off (went delinquent). The objective is to predict whether the loan went delinquent using the loan level aggregate features. We leveraged a subset of 75 features for our modelling exercise. The data is split into two time periods T1 (2007-2013) and T2 (2014-2018) and we have randomly selected 200K records from each time-period. The metrics reported are an average of 10 runs for each experiment. Data set details, split of D^{ref}, D^{new} data is present in Table 5.

Internal dataset (`LOSS-PREDICTION`): Lending products enable customers to make purchases on credit and repay through equated monthly installments (EMIs). While such credit products can increase affordability and customer engagement on e-commerce stores, non-repayment of these loans, result in losses for the companies. The objective is to train ML models to identify loans which would end up in losses and take necessary corrective actions. While the existing model is built using aggregated features, we observed a lift in model performance using sequential customer transaction data which necessitated model architecture change. Hence, we use this dataset to showcase the applicability of DIM to

match distributions under model architecture change. It also represents a high distribution shift use case due to significant change (2.3X increase) in default rate from D^{ref} to D^{new}. Owing to sensitive nature of the use case, we cannot disclose additional data details.

5.2 Experimental Setup

Model Architecture: For model retraining, we use a 3-layer feed forward architecture with *ReLU* activation and batch normalization applied on intermediate layer neurons. The model output is generated with *Sigmoid* activation. For the experiments conducted on the applicability of DIM with model architecture change as described in section 5.3, we use transformer encoder blocks to generate representations from sequential transaction data which is subsequently passed to feed forward network along with the aggregated features. For building the DIM models, we employ a loss function that is a linear combination of distribution matching penalty (Wasserstein, KL) and binary cross entropy loss.

Techniques to Benchmark Performance: We have experimented with post-hoc transformation functions to benchmark performance using model $M^{retrain}$. We learn a mapping function between $\hat{Y}_{new}^{retrain}$ and $\hat{Y}_{new}^{ref}$ and represent it by F_{map}. We experiment with Support Vector Regression (SVR) and linear regression (using logits of the model scores) as the F_{map} functions. Isotonic regression is constrained for monotonic relationships only and is thus limited to quantile matching or probability calibration and cannot be used for distribution matching in general. Additionally, we have also benchmarked performance by using KL divergence as distribution matching constraint in the loss function (instead of Wasserstein) of M^{DIM} model.

Metrics to Measure Performance: To evaluate distribution matching with reference model, we have showcased density plots of model scores and used the fraction of positive class predictions, recall, Backward Trust Compatibility (BTC) at fixed operating points as metrics. To measure model performance we have used ROC-AUC and PR-AUC metrics. All the results are discussed in detailed in section 5.3 and are reported for 10 model runs on validation set of D^{new}.

Compute Infrastructure: Model re-training using DIM does not demand any significant increase in compute and the infrastructure is thus dependent on the size of the dataset, model architecture, and hyper-parameters. For our experiments, we have have used an EC2 r5.xlarge instance which has 32 GB memory.

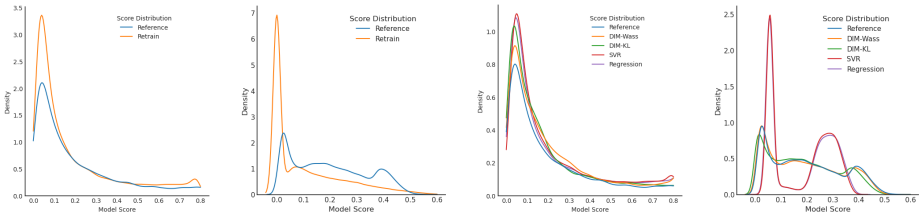
5.3 Experiment Results

ML models experience performance decay over time due to shifts in input feature distributions or changes in the event rates of the dependent variable. This necessitates frequent retraining to improve performance. Table 1 elicits the degradation in model performance (in terms of ROC-AUC) of M^{ref} model for `RAINFALL` and `DELINQUENCY` datasets in comparison to retrained model $M^{retrain}$. However,

Table 1: Reference and Retrained Model Performance

Model	**Data**	`RAINFALL`		`DELINQUENCY`	
		Event Rate (%)	AUC (%)	Event Rate (%)	AUC (%)
M^{ref}	D^{ref}	22.9	84.9	16.6	67.2
M^{ref}	D^{new}	21.1	84.4	10.7	84.5
$M^{retrain}$	D^{new}	21.1	85.3	10.7	87.3

figures 3a and 3b depict the score distributions of the reference model M^{ref} and the retrained model $M^{retrain}$ pointing to a clear distributional drift despite the performance improvement due to retraining. Deploying $M^{retrain}$ in production would require change of all dependent operating points by the consumers of the model scores and dependent downstream models will need refresh.

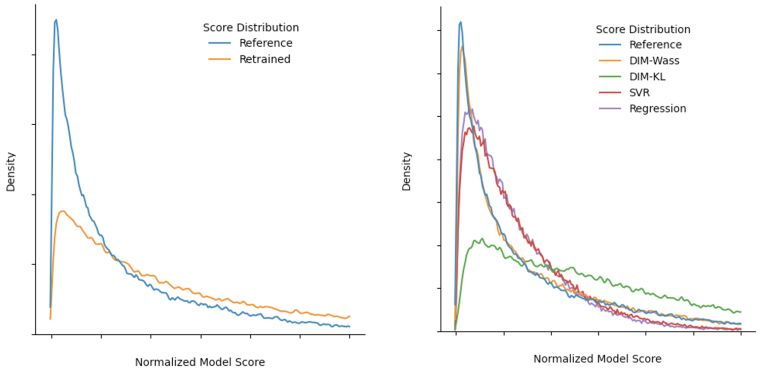


(a) (b) (c) (d)

Fig. 3: (a) and (b) depict the comparison of reference score distribution with retrained model score distribution for `RAINFALL` and `DELINQUENCY` respectively. In (c) and (d), we compare the score distributions of Reference model, DIM models and benchmark techniques for `RAINFALL` and `DELINQUENCY` respectively.

RQ1: Distribution comparison *(a) Score density distributions:* In figures 3c and 3d we observe that DIM methods (DIM-Wass and DIM-KL) exhibit superior performance by better mimicking the reference model score distribution compared to SVR and Linear Regression (Reg). Specifically, DIM-Wass aligns closely with the reference model score distribution, underscoring its effectiveness. For low distribution shift data (`RAINFALL`), the density plots of post hoc methods are relatively close to the reference model for the majority of the score range. However, these post hoc methods perform poorly for high distribution drift data, exemplified by the `DELINQUENCY` dataset. This indicates the effectiveness of in-training distribution matching constraints utilized by DIM methods vs post hoc methods.
(b) Applicability with model architecture change: Here, we build a new model (retrained) on the `LOSS-PREDICTION` dataset by feeding sequential transaction data through transformer layers added to the feed forward layers. In Figure

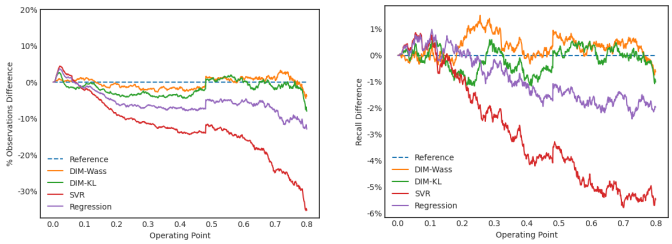
4, we observe that the model distribution of the retrained model is significantly different from reference model due to difference in underlying model architecture. Interestingly, DIM-Wass has a closer distribution match as compared to DIM-KL and benchmark techniques. Due to the sensitive nature of this use case, we use normalised model scores to present the results. This showcases the applicability of DIM to newer generation architectures and can be leveraged across different use cases like NLP based classification.



(a) (b)

Fig. 4: (a) Comparison of (normalized) score distributions of reference model vs retrained model with architecture change for `LOSS-PREDICTION`. (b) Reference distribution mapping with DIM-Wass and benchmark techniques. Due to the sensitive nature of this use case, we have hidden the x and y axis values.

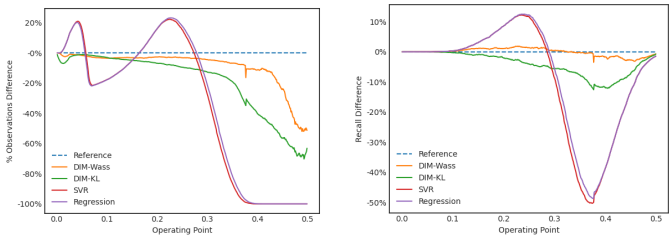
(c) Recall and positive class fraction drift: While the density plots of DIM-Wass are similar to reference model, we inspect them further by evaluating the impact of distribution shift on metrics that matter the most to customer such as the fraction of positive class predictions and recall across different operating points. In Figure 5a, we plot the percentage difference in predicted rainfall days (P) by different models in comparison to reference model for the `RAINFALL` dataset. It is obtained by $(P_\tau^{model} - P_\tau^{ref})/P_\tau^{ref}$ where P_τ^{model} and P_τ^{ref} are rainfall predicted days flagged by the specific model (DIM-Wass, SVR, Reg) and reference model respectively at operating point τ. Using this metric we can observe whether the predicted class observations are similar to reference model and the downstream systems can work seamlessly with the new model (vs reference model). Additionally, we generate a similar plot in Figure 5b to compare recall values of different models vs reference model, $Recall_\tau^{model} - Recall_\tau^{ref}$. The number of rainfall days predicted by DIM-Wass and DIM-KL are similar to reference, but SVR and Reg techniques predict significantly lower number of rainfall days across different operating points. Additionally, M_{Wass}^{DIM} generates higher recall than SVR, Reg and M^{ref} and marginally higher than DIM-KL at



(a) (b)

Fig. 5: Difference in fraction of positive class and recall for DIM-Wass and benchmark techniques w.r.t to Reference model for `RAINFALL` in (a) and (b).

different operating points. When the distribution shift is higher (`DELINQUENCY` dataset), we notice in Figures 6a and 6b that even DIM-KL underperforms (in addition to SVR & Reg) in comparison to DIM-Wass for both observation and recall difference.



(a) (b)

Fig. 6: In (a) and (b), comparison of difference in fraction of positive class and recall with respect to Reference model respectively for DIM-Wass and benchmark techniques for `DELINQUENCY` data

Tables 2 and 3 showcase observations and recall difference across different operating points: (a): Liberal: Towards the lower end of the score spectrum where higher recall is prioritized, (b) Balanced: Maintaining a trade-off between low and high operating points and (c) High: Skewed operating point towards distribution tail. We notice that DIM-Wass works for different operating points across datasets by identifying similar fraction of positive class observations (w.r.t to reference model) and higher recall in comparison to DIM-KL and benchmark techniques. Only for a single operating point (liberal) for `DELINQUENCY` dataset, SVR and Reg generate higher recall due to larger fraction of the population is classified as positive than DIM-Wass and the reference model.

Table 2: Difference of Positive Class Predictions and Recall of DIM, benchmark techniques with reference model at key operating points (OP) of `RAINFALL` dataset. Liberal, balanced, and high operating points refer to 0.3, 0.5, and 0.7 score values. Results are represented as Mean ± SD over 10 model runs.

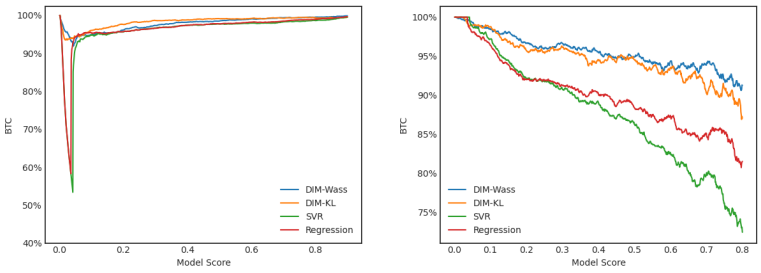
Metric (Diff)	**OP**	**DIM-Wass (%)**	**DIM-KL (%)**	**SVR (%)**	**Reg (%)**
Positive Class	Liberal	**-1.24 ± 0.47**	-2.65 ± 1.22	-11.14 ± 2.83	-6.74 ± 2.49
	Balanced	1.13 ± 0.61	**0.52 ± 0.24**	-10.71 ± 3.24	-5.55 ± 2.13
	High	**0.33 ± 0.21**	-0.35 ± 0.12	-22.29 ± 7.84	-8.41 ± 3.37
Recall	Liberal	**1.06 ± 0.11**	0.54 ± 0.32	-2.35 ± 1.14	-0.54 ± 0.22
	Balanced	**0.98 ± 0.25**	0.21 ± 0.08	-3.2 ± 1.35	-1.27 ± 0.50
	High	**0.73 ± 0.24**	-0.16 ± 0.09	-5.81 ± 2.15	-1.32 ± 0.48

Table 3: Difference of Positive Class Predictions and Recall of DIM, benchmark techniques with reference model at key operating points of `DELINQUENCY` dataset. Liberal, balanced and high operating points refer to 0.2, 0.3 and 0.4 score values. Results are represented as Mean ± SD over 10 model runs

Metric (Diff)	**OP**	**DIM-Wass (%)**	**DIM-KL (%)**	**SVR (%)**	**Reg (%)**
Positive Class	Liberal	**-2.95 ± 2.42**	-6.24 ± 1.91	15.35 ± 0.98	15.13 ± 1.11
	Balanced	**-3.30 ± 2.60**	-10.06 ± 1.68	-31.19 ± 5.01	-28.29 ± 5.94
	High	**-9.15 ± 3.10**	-41.16 ± 10.4	-99.89 ± 0.23	-99.78 ± 0.41
Recall	Liberal	1.22 ± 0.90	-1.73 ± 0.57	**8.56 ± 0.07**	8.55 ± 0.08
	Balanced	**1.25 ± 1.18**	-4.09 ± 0.75	-10.76 ± 2.98	-9.23 ± 3.25
	High	**-1.43 ± 0.93**	-12.41 ± 3.62	-38.57 ± 0.18	-38.49 ± 0.28

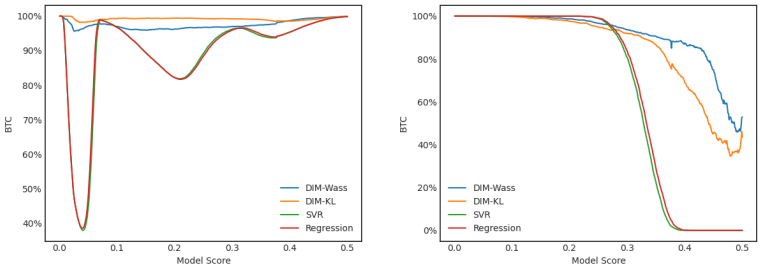
RQ2: Consistency of predictions To measure consistency of predictions, we evaluate BTC metric of DIM. BTC is defined as the fraction of predictions identified correctly by the new model like DIM, SVR in comparison to the reference model. Figures 7a and 7b show the results on `RAINFALL`. While the overall BTC values are high and similar for all models, DIM-Wass outperforms DIM-KL, SVR & Reg on the BTC values specific to positive class (rainfall predicted days). We observe similar results for `DELINQUENCY` (Figures 8a and 8b).

RQ3: Model performance comparison By incorporating a distance metric in the model loss function, we foresee a decrease in AUC of DIM methods compared to $M^{retrain}$. However, our objective is to achieve superior distribution matching and ensure that the performance drop is minimal. Models trained on recent data ($M^{retrain}$, DIM, SVR) significantly outperform the reference model across datasets (refer 4). Notably, the DIM-Wass model achieves comparable performance to the benchmark methods for `RAINFALL` and exhibits only a marginally lower AUC for `DELINQUENCY`.



(a) (b)

Fig. 7: For `RAINFALL`, overall BTC values in (a) and BTC values when the target label=1, the positive class, i.e., rainfall day in (b).



(a) (b)

Fig. 8: For `DELINQUENCY`, overall BTC values for in (a) and BTC values when the target label=1, the positive class, i.e., delinquent loan in (b).

The `DELINQUENCY` dataset, however, represents a high distribution-shift scenario. In such cases, while the benchmark methods may demonstrate slightly higher performance, they perform significantly worse in terms of distribution matching compared to the DIM-Wass model (see 5.3). Please note that the performance metrics of SVR and Reg is expected to be similar to $M^{retrain}$ as they learn a point to point mapping of observations and have similar rank ordering.

RQ4: Optimal selection of regularization hyper-parameter (λ) As shown in Eq. 1, the DIM loss function comprises two different loss terms and hence, it is important to select optimal value of the regularization hyperparameter (λ). In Figure 9, we observe that as we increase λ, there is a decrease in the model AUC and the Wasserstein loss, as expected. To select the optimal range of λ, we choose the operating point where Wasserstein loss is close to its minimum value and the loss in model performance (AUC) is low. For DIM-Wass, we operate in the range of 450 to 500. In practice, λ should be chosen basis business specific trade off between distribution matching and model performance.

Table 4: Model performance comparison. The goal here is to ensure that the proposed solution can maintain similar performance as compared to $M^{retrain}$. Results are represented as Mean ± SD over 10 model runs

Model	RAINFALL		DELINQUENCY	
	ROC-AUC (%)	PR-AUC (%)	ROC-AUC (%)	PR-AUC (%)
Reference	84.39 ± 0.28	64.50 ± 0.32	84.47 ± 0.13	36.70 ± 0.54
Retrained	85.28 ± 0.11	66.53 ± 0.28	87.39 ± 0.10	40.30 ± 0.62
SVR	85.01 ± 0.09	66.01 ± 0.35	87.39 ± 0.10	40.30 ± 0.62
Reg	85.03 ± 0.10	66.02 ± 0.37	87.37 ± 0.10	40.29 ± 0.62
DIM-KL	84.75 ± 0.09	65.33 ± 0.35	84.56 ± 0.05	36.85 ± 0.14
DIM-Wass	85.07 ± 0.15	66.17 ± 0.34	85.90 ± 0.18	38.47 ± 0.16

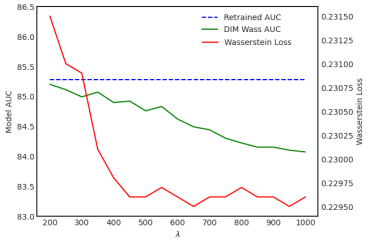


Fig. 9: Model AUC and Wasserstein loss values on changing λ hyper-parameter values for the `RAINFALL` dataset. Notice that Wasserstein loss becomes stagnant post λ value of 450.

6 Model launch and Business Impact

DIM saves significant time and effort spent on downstream changes (such as model refresh, threshold updates) and leads to faster model deployment. DIM-Wass model has been tested as a pilot for 12 weeks for the `LOSS-PREDICTION` task in two geographies. We estimate to save ∼30% of bandwidth for each model refresh and significant financial savings due to faster model deployment since changes to downstream systems will no longer be required.

7 Conclusion and Future Work

Building distributionally invariant models is critical for production ML pipelines due to frequent model refreshes. DIM enhances training loss with a distribution matching penalty. We demonstrate its efficacy via two use cases with varying distribution shifts, consistency with the reference model (BTC), and impact on model performance. Future work involves extending DIM to a) partial distribution matching, applying constraints to relevant distribution segments, and b) incorporating the Wasserstein constraint into tree-based models like XGBoost.

References

1. Ishani Deshpande, Ziyu Zhang, and Alexander G. Schwing. Generative modeling using the sliced wasserstein distance. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3483–3491, 2018.
2. Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss, 2015.
3. Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017.
4. Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.
5. Kartik Gupta, Amir M. Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard I. Hartley. Calibration of neural networks using splines. *ArXiv*, abs/2006.12800, 2020.
6. M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.
7. Weihua Hu, Rajas Bansal, Kaidi Cao, Nikhil Rao, Karthik Subbian, and Jure Leskovec. Learning backward compatible embeddings. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22. ACM, August 2022.
8. S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.
9. Jiaming Lv, Haoyuan Yang, and Peihua Li. Wasserstein distance rivals kullback-leibler divergence for knowledge distillation, 2024.
10. Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 2015.
11. Tan Pan, Furong Xu, Xudong Yang, Sifeng He, Cheng peng Jiang, Qingpei Guo, Fengyi Zhang, Yuan Cheng, Lei Yang, and Wei Chu. Boundary-aware backward-compatible representation via adversarial learning in image retrieval. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15201–15210, 2023.
12. John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. 1999.
13. Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. 2021.
14. Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. An empirical analysis of backward compatibility in machine learning systems. 2020.
15. Haoran Sun, Hanjun Dai, Bo Dai, Haomin Zhou, and Dale Schuurmans. Discrete langevin sampler via wasserstein gradient flow, 2023.
16. Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders, 2019.
17. Frederik Träuble, Julius von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter Gehler. Backward-compatible prediction updates: A probabilistic approach. In *Neural Information Processing Systems*, 2021.
18. Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, 2019.

19. Shengsen Wu, Liang Chen, Yihang Lou, Yan Bai, Tao Bai, Minghua Deng, and Ling yu Duan. Neighborhood consensus contrastive learning for backward-compatible representation. In *AAAI Conference on Artificial Intelligence*, 2021.
20. Bianca Zadrozny and Charles Peter Elkan. Transforming classifier scores into accurate multiclass probability estimates. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.

A Appendix: Public Data set Details

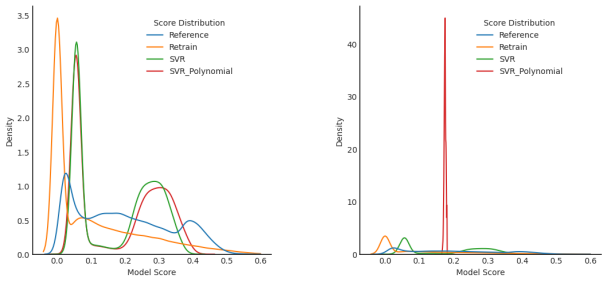
Dataset details and download links are present in Table 5.

Table 5: Dataset Statistics and download links

Dataset	**Data Size**	**# Cat**	**# Cont**	**Event Rate**
`RAINFALL`	145K	5	18	21.9%
`DELINQUENCY`	2,260K	2	73	11.9%

B Appendix: SVR Parameter Details

We have experimented with different kernels like sigmoid, polynomial with different degrees (especially for polynomial) to build simple to complex mapping functions using SVR. We observe in Figure 10a that polynomial kernel (SVR Polynomial) with degree as 1 has a similar distribution to sigmoid kernel (SVR) however as we build a complex model with higher degrees (degree as 2) in polynomial kernel in Figure 10b, the score distributions are constrained in a tight range. Hence, distribution matching with reference does not work with complex mapping function models. The final results added in the paper are using sigmoid as kernel for SVR.



(a) (b)

Fig. 10: SVR trained with 'poly' as kernel with degree=1 in (a) and degree=2 in (b) for the `DELINQUENCY` dataset