A Unified Framework for Comparing Distribution Matching Methods Across Trustworthy Machine Learning Tasks

Anonymous Author
Anonymous Institution

Abstract

Distribution matching (DM) is a fundamental tool in trustworthy machine learning (TML), with applications in fairness, calibration, and domain adaptation. While prior work advances individual DM methods based on information-theoretic and geometric divergences, a unified comparative framework remains lacking. We propose a framework integrating DM methods, metrics, and TML tasks to enable systematic comparisons. To our knowledge, this is the first work to compare latent spaces in TML while addressing scaling inconsistencies via PCA whitening. We empirically evaluate MMD, Sinkhorn and adversarial DM calibration methods across fairness, calibration, and domain adaptation. Our findings reveal: (1) simple NLL training objective with post-hoc calibration can outperforms other DM methods; (2) logit-based fairness methods outperform latent-based approaches; and (3) error and DM metrics show a U-shaped trend, and we connect this insight to theory Zhao et al. [2019b]. These insights inform the selection and refinement of DM algorithms for TML applications.

1 Introduction

Domain-invariant representation learning (DIRL) [Zhao et al., 2019a, 2022] aims to learn a representation function $g_\theta : \mathbb{X} \to \mathbb{Z}$, which map data from different domains into a shared latent space where their distributions align, enabling models to focus on task-relevant features while ignoring domain-specific variation as shown in **??**. Unlike representation learning for classification which seeeks to maximize the divergence between class distributions, DIRL seeks to minimize the divergence between the domain distribution. This approach is foundational to many trustworthy machine learning (TML) tasks—such as fair classification, domain adaptation, and uncertainty calibration—because it addresses the pervasive challenge of distribution shift that undermines model reliability in real-world applications.

Prior work on distribution matching has primarily been developed within specific TML tasks, often focusing on individual approaches rather than a comparative or unified framework [Han et al., 2023b, Reddy, 2022b, Tao et al., 2023, Gulrajani and Lopez-Paz, 2020, Marx et al., 2024b]. For instance, in uncertainty calibration [Marx et al., 2024b], DM has been explored using kernel-based approaches such as Maximum Mean Discrepancy (MMD) to align predicted and true confidence distributions. In contrast, domain adaptation methods typically rely on adversarial learning, where generative adversarial networks (GANs) or domain classifiers enforce domain-invariant representations [Ganin et al., 2016b]. In fairness, logit-based methods enforce fairness by directly constraining output distributions [Chung et al., 2024a], while latent space-based methods align intermediate feature distributions [Madras et al., 2018]. Consequently, there is limited understanding of which DM methods generalize best across tasks or how different alignment techniques trade off between computational efficiency, stability, and effectiveness.

To bridge this gap, we propose a unified framework for systematically comparing DM methods across multiple TML tasks. Our framework integrates representative DM methods including Maximum Mean Discrepancy (MMD) [Gretton et al., 2012], Sinkhorn divergence [Feydy et al., 2019a], and adversarial domain alignment. We evaluates their performance on three major TML tasks: fairness, calibration, and domain adaptation. While previous studies consider distribution matching (DM) in isolation for individual tasks, our framework enables cross-task comparison. In particu-

Preliminary work. Under review by AISTATS 2026. Do not distribute.

Table 1: Unified distribution-matching (DM) formalization across calibration (Calib), fairness (DP), and domain adaptation (DA).

Task	L_{task}	**Domains** d	**Matched object / matcher**	**DM constraint (type)**
Calib	$\mathbb{E}[\ell(f_\psi(g_\theta(x)), y)]$	$\{\text{FORECAST}, \text{TARGET}\}$	(y', q) with $q = \text{softmax}(f_\psi(g_\theta(x)))$	$D\big(p(y\,\vert\,q),\, p(\hat{y}\,\vert\,q)\big) \le \delta$ (conditional)
Fair (DP)	$\mathbb{E}[\ell(f_\psi(g_\theta(x,\varepsilon)), y)]$ / or $-I(x,\, z\,\vert\,d)$	sensitive groups ($d \in \{\text{A}, \text{B}\}$)	$z = g_\theta(x,\varepsilon)$ / or q	$D\big(p(z\,\vert\,d{=}1),\, p(z\,\vert\, d{=}2)\big) \le \delta$ (unconditional DP) *EO variant*: $D\big(p(z\,\vert\, d, y),\, p(z\,\vert\,y)\big) \le \delta$
DA	$\mathbb{E}[\ell(f_\psi(g_\theta(x,\varepsilon)), y)]$ (ERM on SRC)	$\{\text{SOURCE}, \text{TARGET}\}$	domain-invariant $z = g_\theta(x,\varepsilon)$	$D\big(p(z\,\vert\,\text{SRC}),\, p(z\,\vert\,\text{TGT})\big) \le \delta$ (unconditional)

Symbols: x input, y target, $z = g_\theta(\cdot)$ representation, $q = \text{softmax}(f_\psi(z))$; D can be MMD/Sinkhorn/adversarial.

lar, it reveals how intrinsic measures (MMD, Sinkhorn) align with task-specific objectives, including fairness and calibration metrics such as ECE [Guo et al., 2017b] and DP [Han et al., 2023b]. Additionally, we introduce a normalized divergence metric to control for latent space scaling, ensuring fair evaluation DM methods using non-parametric geometric divergences that can be computed directly from samples.

Our empirical results reveal key trends in DM effectiveness across tasks and highlight limitations in current methods that future research must address. First, we find that simple NLL training with post-hoc calibration can outperform other DM-based calibration methods. Second, although most fairness research has focused on representation learning (i.e., latent-based methods), we show that logit-based methods consistently outperform latent-based ones. Lastly, we demonstrate that strictly minimizing distributional discrepancy is not beneficial for domain adaptation by showing that both error and DM metrics exhibit a U-shaped trend. This suggest the existence of a optimal region that minimizes errors by controling DM metrics. Through this study, we aim to guide the selection of DM techniques for TML applications and inspire the development of more robust, generalizable DM algorithms. Our contributions can be summarized as follows:

1. We formalize a common theoretical framework that integrates DIRL and DM methods under a single umbrella, enabling systematic comparisons.
2. We propose a PCA whitened version of a distribution matching metric to be more fair when comparing methods in latent representations spaces.
3. Using the unified DM framework, we evaluate different DM methods across fairness, domain adaptation, and calibration tasks, highlighting their connection between intrinsic metric (e.g., MMD, Sinkhorn) and task specific metric (e.g., DP, ECE), and provide insightful guideline for practical usage.

2 Unified Framework for Distribution Matching and Trustworthy ML Tasks.

Notation. Let $\mathbf{x} \in \mathbb{X}$, $\mathrm{y} \in \mathbb{Y}$, and $\mathrm{d} \in \{1, 2, \ldots, k\}$ denote random variables corresponding to the input, target (optional), and domain label, respectively. Let $\mathbf{z} := g_\theta(\mathbf{x}, \mathrm{y}, \mathrm{d}, \epsilon) \sim p_\theta(\mathbf{z}|\mathrm{y}, \mathbf{x}, \mathrm{d})$ denote the latent representation of $\mathbf{x}$, and for logit based method, we denote $\hat{\mathbf{q}} := g_\theta(\mathbf{x}, \mathrm{y}, \mathrm{d}, \epsilon) \sim p_\theta(\hat{\mathbf{q}}|\mathrm{y}, \mathbf{x}, \mathrm{d})$ where g_θ is called the *matcher* with parameters θ that may optionally depend on the target variable y, the domain d, and exogenous noise ϵ to encompass stochastic aligners. If g does not depend on d and/or ϵ we will suppress notation w.r.t. these random variables for notational simplicity. Let $p_{\text{data}}(\mathbf{x}, \mathrm{y}, \mathrm{d})$ denote the true data distribution. Let ϕ denote parameters of variational models or distributions, e.g., $q_\phi(\mathbf{x}, \mathrm{y}, \mathrm{d})$ will denote a variational distribution and $h_\phi(\mathbf{z})$ will denote a variational discriminator for adversarial learning. Let ψ denote application-specific parameters, e.g., $\hat{\mathrm{y}} := f_\psi(\mathbf{z})$ will denote the predicted class based on the given classifier head in fair classification. Entropy, cross entropy, and mutual information will be denoted by $\mathrm{H}(\mathbf{x})$ and $\mathrm{H_c}(\mathbf{x}, \mathbf{z})$, and $\mathrm{I}(\mathbf{x}, \mathbf{z})$, respectively. Let $\mathrm{D}(p, q)$ denote a distribution divergence between p and q, e.g., $\mathrm{D_{KL}}$, $\mathrm{D_{JSD}}$, and $\mathrm{D_{W_\rho}}$ will denote KL, JSD, and Wasserstein-ρ divergences, respectively. Similarly, let $\widehat{\mathrm{D}}$, $\overline{\mathrm{D}}$, and $\underline{\mathrm{D}}$ denote an approximation, an upper bound, or a lower bound of a divergence respectively. Because DM involves mininizing a diver-

gence w.r.t. the matcher parameters θ, we will let $\mathrm{D}(\theta) := \mathrm{D}(p_\theta(\mathbf{z}|\mathrm{d}{=}1), p_\theta(\mathbf{z}|\mathrm{d}{=}2))$ with slight abuse of notation.

Distribution Matching Problem. The distribution matching problems we consider can be formulated as a task-specific objective plus a distribution matching constraint on the matched representation.

Definition 1. *(Distribution Matching Problem). A distribution matching problem minimizes a task objective* $\mathrm{L}_{\text{task}}(\tilde{f}_\psi, \tilde{g}_\theta)$*, where* $\tilde{f}_\psi$ *is a task-specific model and* $\tilde{g}_\theta$ *is the* matcher *model, subject to a DM constraint on the matched representation* $\tilde{\mathbf{z}} := \tilde{g}_\theta(\mathbf{x}, \mathrm{y}, \mathrm{d}, \epsilon)$*:*

$$\min_{\psi,\theta} \mathrm{L}_{\text{task}}(\tilde{f}_\psi, \tilde{g}_\theta) \quad s.t. \quad \mathrm{D}(p_\theta(\tilde{\mathbf{z}}|\mathrm{d}{=}1), p_\theta(\tilde{\mathbf{z}}|\mathrm{d}{=}2)) \leq \delta \tag{1}$$

where $\mathrm{D}(\cdot,\cdot)$ *is a distribution divergence and* δ *is the DM slackness hyperparameter.*

In practice, minimizing a distribution divergence is challenging when only samples are available. To address this, we employ information-theoretic divergences through parametric variational bounds and geometric divergences through non-parametric estimators, as detailed in section D.1. We will first review the main approaches for minimizing distributional divergence on different TML tasks.

2.1 Unified Formalization of Trustworthy ML Tasks as Distribution Matching

Many trustworthy ML tasks can be formulated as DM problems. In some cases, DM is fundamental to the trustworthy ML task (e.g., fairness or calibration), while in others, DM is one approach to the task (e.g., domain adaptation). For the tasks where DM is fundamental, the key question is: *What is the empirically achievable Pareto frontier between the task objective and the DM constraint (e.g., fairness-accuracy tradeoff)?* For the tasks where DM is an approach, the key question is: *Is DM performance correlated with the relevant task performance (e.g., does better DM yield better domain adaptation performance)?* In particular, we would like to disentangle the effect of the DM algorithm—which may be far from optimal—from the task performance. We conjecture that in some cases, the DM algorithm fails to achieve the DM objective even though the task objective may be reasonable.

Group Fair ML as Distribution Matching The goal of fair learning is to be as accurate as possible while satisfying a fairness constraint. Demographic parity (DP) (also known as statistical parity) is one common notion of group fairness that is satisfied if and only if $p(\hat{\mathrm{y}} = 1|\mathrm{d}{=}1) = p_\theta(\hat{\mathrm{y}} = 1|\mathrm{d}{=}2)$, i.e., these two distributions match. Fair classification seeks to directly learn predictions that are fair. Fair representation learning seeks to learn a representation such that all downstream tasks will be fair. We unify fair learning under our DM framework and notation below.

Proposition 1. *Fair learning [Madras et al., 2018, Song et al., 2019b] w.r.t. DP is a DM problem (1) with* $\tilde{g}_\theta(\mathbf{x}, \mathrm{y}, \mathrm{d}, \epsilon) = g_\theta(\mathbf{x}, \epsilon)$ *and* $\mathrm{L}_{\text{task}} = \mathbb{E}[\ell(f_\psi(g_\theta(\mathbf{x}, \epsilon)), \mathrm{y})]$ *for fair classification and* $\mathrm{L}_{\text{task}} = -\mathrm{I}(\mathbf{x}, \mathbf{z} = g_\theta(\mathbf{x}, \epsilon)|\mathrm{d})$ *for fair representation learning.*

In practice, both the classification and mutual information task objectives are often combined (e.g., [Madras et al., 2018, Gong et al., 2024] approximate mutual information via a VAE objective).

Calibration as DM Problem Vaicenavicius et al. [2019b] formalized canonical calibration for multiclass classification, where calibration means that the predicted probability vector for all classes coincides with the true underlying class probabilities:

$$p(\mathrm{y} = y|\hat{\mathbf{q}}) = p(\hat{\mathrm{y}} = y|\hat{\mathbf{q}}) := \mathrm{q}_y, \quad \forall y \in \mathbb{Y}, \hat{\mathbf{q}} \in \Delta^{|\mathbb{Y}|} \tag{2}$$

where $\hat{\mathbf{q}} := g_\theta(\mathbf{x})$ is the predicted class probabilities for k classes and $\Delta^{|\mathbb{Y}|}$ denotes the probability simplex. This calibration condition is a type of *conditional* distribution matching problem, i.e., match the marginal distribution of predictions to the true distribution *conditioned* on the model's output $\mathbf{q}$. In this case, the domain label is whether it is the real target variable or the predicted target variable. Marx et al. [2023] demonstrated that indeed many types of calibration including regression, classification, and decision calibration can be framed as conditional distribution matching problems. In fact, because the marginal distribution of the conditioning variables is the same regardless of the domain, the problem can be equivalently written as an unconditional DM problem. We now unify the results from Marx et al. [2023] using our framework below.

Proposition 2. *Calibration during training can be interpreted as a distribution matching problem. Let* $\hat{\mathrm{y}}'$*,* y'*, and* c *denote the forecast, target, and conditioning variables, respectively, as defined in Marx et al. [2023, Tables 1 and 2]. Here,* $\hat{\mathrm{y}}'$ *and* y' *are derived transformations of the raw forecast* $\hat{\mathrm{y}}$ *and the true outcome* y,[1] *calibration during training is a DM problem (1) with* $\tilde{g}_\theta(\mathbf{x}, \mathrm{y}, \mathrm{d}, \epsilon) = (\tilde{\mathrm{y}}', g_\theta(\mathbf{x}, \epsilon))$*, where* $\tilde{\mathrm{y}}' = \mathbb{1}(\mathrm{d}{=}1)\hat{\mathrm{y}}' + \mathbb{1}(\mathrm{d}{=}2)\mathrm{y}'$ *selects between the forecasted*

[1]Note that in many cases, $\hat{\mathrm{y}}' = \hat{\mathrm{y}}$ and similarly $\mathrm{y}' = \mathrm{y}$, but there are some cases from Marx et al. [2023] such as quantile calibration for regression or top-label calibration for classification that require using either the predicted CDF or indicator functions of $\hat{\mathrm{y}}$ and y.

and target variables, $c = g_\theta(\mathbf{x}, \epsilon)$ *represents the conditioning variable, and* $\mathrm{L}_{\text{task}} = \mathbb{E}[\ell(g_\theta(\mathbf{x}, \epsilon), \mathrm{y})]$ *is the standard negative log-likelihood empiricial risk minimization (ERM) objective.*

Domain Adaptation via Domain-Invariant Features Inspired by the bounds on domain adaptation generalization by Ben-David et al. [2006a], many domain adaptation papers aim to learn domain-invariant features (i.e., latent features whose distribution is independent of the domain labels). Specifically, Ben-David et al. [2006a] showed that the risk on the target domain could be bounded by the risk on the source domain plus the divergence between the feature distributions and a constant. A natural approach is to reduce the divergence between the feature distributions (i.e., distribution matching). Thus, domain-invariant domain adaptation can be unified under our framework.

Proposition 3 (Domain-invariant domain adaptation is DM)**.** *Domain-invariant domain adaptation is a DM problem (1) with* $\tilde{g}_\theta(\mathbf{x}, \mathrm{y}, \mathrm{d}, \epsilon) = g_\theta(\mathbf{x}, \epsilon)$ *and* $\mathrm{L}_{\text{task}} = \mathbb{E}[\ell(f_\psi(g_\theta(\mathbf{x}, \epsilon)), \mathrm{y})]$ *is the standard ERM objective where* f_ψ *is the classification head on top of the domain-invariant representation* $\mathbf{z} = g_\theta(\mathbf{x}, \epsilon)$*.*

2.2 Normalized Geometric Divergences for Evaluating DM Methods

One of the challenges with comparing DM methods is properly evaluating how well the DM constraint was satisfied. Ideally, we would measure the divergence of the latent domain distributions. However, even estimating distribution divergences is known to be a challenging problem in its own right. While the adversarial and VAE-based methods could provide bounds on information theoretic divergences, they would require training an auxiliary model at test time to evaluate each method. Thus, we focus on the non- parametric divergences MMD and Sinkhorn that can be estimated with only samples. However, there is one key challenge with these geometric divergences when comparing across diverse methods. The scale of the latent distribution can significantly affect the absolute MMD or Sinkhorn divergence estimate because geometric divergences are highly sensitive to scale. This is a problem if the latent space is learned since the latent space scale is arbitrary and generally distorted [Jing et al., 2021, Ermolov et al., 2021]. Thus, comparing methods using MMD or Sinkhorn directly would be unfair. To overcome this, we propose a simple approach based on applying principal component analysis (PCA) whitening [Kessy et al., 2018] of the latent space before measuring the divergence. This normalization eliminates the effect of scale in the latent distributions. We do not consider zero phase component analysis (ZCA) whitening, which applies rotation after PCA whitening since L2 based metrics are invariant under rotation. Simple proof can be found on section B. In contrast to latent space, logit space do not suffer from distortion problem since its axes are semantically tied to classes and trained under cross entropy, thus result into well-structured simplex equiangular tight frame (ETF) geometry at convergence also known as neural collapse [Papyan et al., 2020]. Therefore, we do not have to apply PCA whitening.

3 Experimental Setup

Given our unified DM framework for TML tasks, we aim to systematically compare both DM methods and DM evaluation metrics, where we evaluate DM methods both intrinsically (i.e., how well the distributions match) and extrinsically (i.e., how much the DM approach helps the TML task). In particular, we are interested in whether better DM corresponds to better task-specific metrics and vice versa or if the performance tradeoff is more complex—while most methods have theoretic grounding, it is unclear how it translates to empirical results. We are also interested in comparing methods for particular tasks and asking which method is the best method depending on the context. Finally, we explore a few other task-specific ideas along the way such as whether we should match latent representations or logits. To systematically and effectively do this, we fix the model architecture and the form of the objective function and only modify the implementation of the DM regularization to the problem. Specifically, we apply three representative DM methods (kernelized MMD, Sinkhorn, and adversarial) to three representative TML tasks (calibration, group fairness, and domain adaptation) across a variety of datasets. Ultimately we aim to elucidate some of the nuances involved in using DM methods for TML tasks and give practical guidelines for practitioners.

3.1 Experimental Setup

Datasets and Hyperparameter Tuning We consider ADULT dataset [Becker and Kohavi, 1996], COMPAS, and ACS-T datasets for calibration and fairness task. For the domain adaptation task, we use the MNIST and USPS dataset [Deng, 2012, Hull, 1994]. We tune the hyperparameters of each method using a TPE sampler [Bergstra et al., 2011] to find the best model and apply early stopping by tracking the validation loss. More detail can be found on section A.

Calibration Training We follow an individual calibration approach, as described in Marx et al. [2024a]. While prior work primarily used the Maximum Mean

Discrepancy (MMD) method, we extend the study by incorporating both the Sinkhorn divergence and an adversarial method. Since no prior work has applied adversarial techniques in this context, we implemented a GAN-based method designed to match the predicted distribution to the target ground-truth distribution. For post hoc calibration, we apply temperature scaling Hinton [2015].

Fair Classification Training Most fairness benchmark papers [Han et al., 2023a, Reddy, 2022a] focus on fair representation learning, which we refer to as latent-based methods. However, there is a lack of prior work studying logit-based approaches [Chung et al., 2024b]. In this paper, we compare distribution matching using both logit-based and latent-based methods using Sinkhorn, MMD, adversarial.

Domain Adaptation Training In this paper, we focus on unsupervised domain adaptation setting where we do not have access to the target label. [Wilson and Cook, 2020]. We use Sinkhorn based method [Courty et al., 2014], MMD based method [Tzeng et al., 2014], and adversarial based method [Ganin et al., 2016a].

Task Specific Metrics For all tasks, we use accuracy (or equivalently error) as a measure for the raw model performance (for domain adaptation, this is source accuracy). For each task, there is an additional task-specific metric. For the calibration task, we use the *Expected Calibration Error (ECE)* [Naeini et al., 2015a] based on measuring the discrepancy between the model confidence and accuracy: ECE =

$$\sum_{m=1}^{M} \frac{|B_m|}{n} \left| \mathbb{E}\left[\mathbb{I}\left(\hat{y} = y\right) \mid \hat{\mathbf{q}} \in B_m\right] - \mathbb{E}\left[\hat{\mathbf{q}} \mid \hat{\mathbf{q}} \in B_m\right] \right|, \tag{3}$$

where B_m denotes the set of samples in the m-th confidence bin, $|B_m|$ is the number of samples in bin m, n is the total number of samples, $\hat{\mathbf{q}}$ is predicted probability (confidence), and $\hat{y}$ is predicted label For group fairness, we use *Demographic Parity (DP)* difference that measures the discrepancy of the true positive rate between different domains:

$$\mathrm{DP} = |p_\theta(\hat{y} = 1 | d = 1) - p_\theta(\hat{y} = 1 | d = 2)| \tag{4}$$

For domain adaptation, we simply use the target accuracy (ACC).

4 Experimental Results Across Tasks

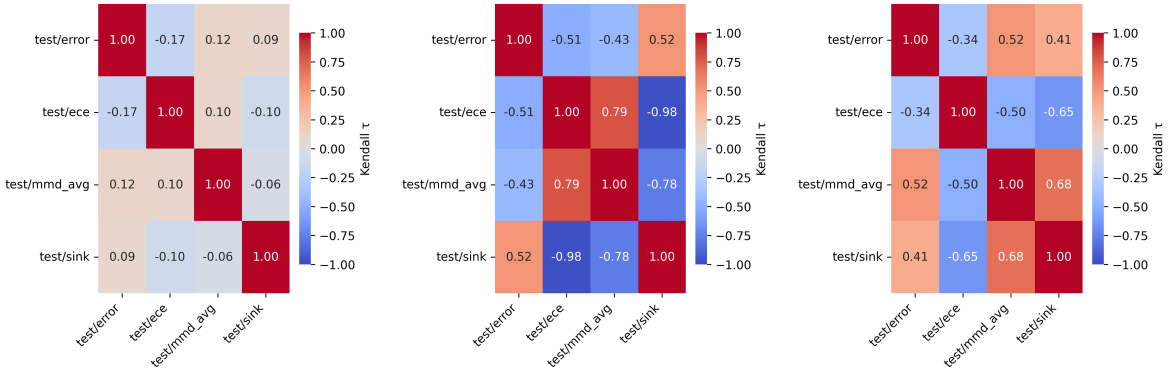
In each of the following sections, we highlight the most interesting results for each task but provide additional discussion in the appendix. For each task, we consider how well DM metrics correlate with accuracy and their corresponding task-specific metrics. In some cases, we find that, despite the theoretic grounding, lower DM metric does not imply a better task-specific metric—particularly for domain adaptation. The other key question is which method is most practical or useful for each task given the results. We also make several other observations that are specific to each task.

4.1 Calibration Task Results

The Sinkhorn divergence best captures distributional differences, whereas MMD fails to do so. As shown in Figure 1, the correlation between Sinkhorn divergence and ECE is mostly negative before temperature scaling (TS), consistent with the definition of calibration Equation (2). In contrast, the correlation between MMD and ECE is sometimes positive Figure 1b. This discrepancy arises from the stronger entropic regularization in MMD, which makes it a noisier estimator compared to the Sinkhorn divergence [Feydy et al., 2019a]. Moreover, the MMD regularizer fails to capture any correlation across metrics on large-scale datasets Figure 1a, even though it can still capture correlations on small-scale datasets. This failure can be explained by the fact that MMD is incomplete when the feature space X is not compact, which often occurs in large-scale datasets due to outliers [Simon-Gabriel et al., 2023, McCarty, 2025]. Lastly, as shown in Table 2, the MMD values across all methods and datasets are so small that they become indistinguishable.

There is a trade-off between Error and ECE (if post-hoc calibration is not used). Extensive research in fairness has investigated the trade-off between Demographic Parity (DP) and accuracy (ACC) [Han et al., 2023a, Plecko and Bareinboim, 2024, Gong et al., 2024]. However, the trade-off between ACC and ECE has not been thoroughly explored in the calibration domain. Recent work reveals a negative correlation between ECE and Error when varying weight decay strength [Wang and Zhang, 2024]. We also observe this trend: before applying Temperature Scaling (TS), Figure 1 shows a negative correlation between ECE and ACC, which indicates such a trade-off. We analyze the the case when post-hoc calibration is used in section C.1.

Which DM method should we use for Calibration task? Wang et al. [2021] show that applying TS to an unregularized cross-entropy model can outperform regularized alternatives in terms of ECE. Motivated by this, we include this baseline for comparison with other DM methods. Interestingly, training



(a) Calibration MMD (b) Calibration Sinkhorn Divergence (c) Calibration Adversarial

Figure 1: Kendall ranking correlation matrix of task specific metrics (Error and ECE), and DM metrics (MMD and Sinkhorn) across different calibration methods on ACS-T dataset.

Table 2: Experimental results for tabular classification tasks for calibration. We display test metrics for each training procedure, with and without post-hoc calibration [Guo et al., 2017b]. n is the number of examples; d is the number of features. We repeat all the experiments across 10 random seeds and report the mean and standard deviation for each metric. We bold top 2 methods if average values are tie.

Dataset	Training Objective	ACC ↑	ECE ↓	SINK ↓	MMD ↓
ADULT	NLL	0.8435 ± 0.002	0.0163 ± 0.003	0.1894 ± 0.002	0.0000 ± 0.000
n= 30162	NLL + MMD	0.8438 ± 0.002	0.0175 ± 0.002	0.1887 ± 0.002	0.0000 ± 0.000
d= 102	NLL + Sink (Ours)	0.8429 ± 0.003	0.0166 ± 0.003	0.1889 ± 0.003	0.0000 ± 0.000
	NLL + Adv (Ours)	$\mathbf{0.8446 \pm 0.002}$	0.0155 ± 0.003	$\mathbf{0.1884 \pm 0.001}$	0.0000 ± 0.000
COMPAS	NLL	0.6481 ± 0.033	0.0402 ± 0.011	0.2936 ± 0.009	0.0001 ± 0.000
n= 6172	NLL + MMD	0.6372 ± 0.033	0.0407 ± 0.014	0.2956 ± 0.010	0.0000 ± 0.000
d= 401	NLL + Sink (Ours)	0.6361 ± 0.034	$\mathbf{0.0400 \pm 0.010}$	0.2984 ± 0.008	0.0000 ± 0.000
	NLL + Adv (Ours)	$\mathbf{0.6579 \pm 0.012}$	$\mathbf{0.0402 \pm 0.009}$	$\mathbf{0.2903 \pm 0.007}$	0.0001 ± 0.000
ACS-T	NLL	0.6502 ± 0.003	0.0397 ± 0.004	0.4087 ± 0.002	0.0000 ± 0.000
n= 172508	NLL + MMD	0.6496 ± 0.003	0.0430 ± 0.004	0.4081 ± 0.002	0.0000 ± 0.000
d= 1567	NLL + Sink (Ours)	$\mathbf{0.6585 \pm 0.001}$	0.1411 ± 0.002	$\mathbf{0.3722 \pm 0.001}$	0.0000 ± 0.000
	NLL + Adv (Ours)	0.6497 ± 0.003	0.0414 ± 0.004	0.4089 ± 0.002	0.0000 ± 0.000

with the *NLL* objective followed by TS often achieves ECE comparable to that of other DM methods, consistent with our explanation in section C.1. Even without TS, *NLL* alone yields competitive ECE on small-scale datasets table 2. While *NLL + Adv* sometimes achieves the lowest ECE (e.g., on Compas), it also introduces instability when the feature dimension is small, as in the German dataset section C.1, leading to high variance in both ACC and ECE.

Takeaway for Calibration

We recommend that practitioners use **NLL + post-hoc calibration** for large-scale datasets, while opting for **NLL** without post-hoc calibration on small-scale datasets, since post-hoc calibration can lead to miscalibration when the validation set is too small.

4.2 Fair Classification Results

Latent-based methods fail to capture consistent correlations due to noise. By definition of demographic parity (DP) in eq. (4), we expect distribution-matching (DM) metrics to exhibit a positive correlation with DP and a negative correlation with error, reflecting the well-known trade-off between error and fairness [Plecko and Bareinboim, 2024]. However, as shown in fig. 8b, both Sink and MMD display the opposite pattern: they are positively correlated with error and negatively correlated with DP. Moreover, 8c indicates that latent-based adversarial methods yield weak or inconsistent correlations between DM and task metrics.

In contrast, the logit-based results (Figure 9) reveal a clearer trend: DM metrics are negatively correlated with error and positively correlated with DP, with the exception of NLL + Sink. Interestingly, NLL + Sink

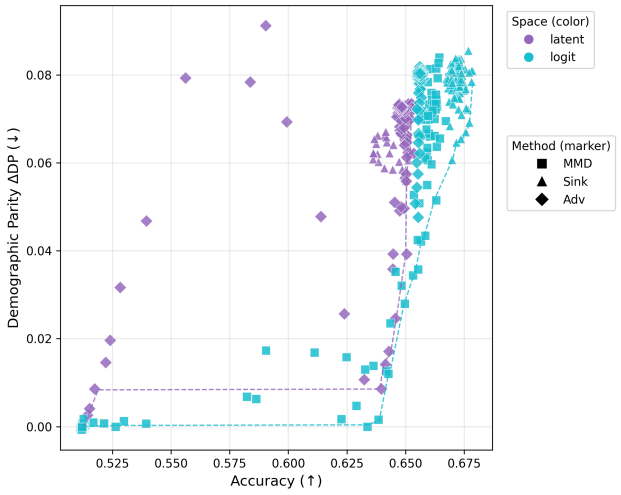


Figure 2: Fairness–accuracy trade-off comparison across methods on ACS-T dataset, with each method distinguished by a unique **shape** and each representation type (latent vs. logit) by a distinct **color**. We can clearly observe that logit shows better trade off by attaining comparable or higher accuracy at substantially lower Δ DP near the Pareto optimal frontier

appears to break the trade-off entirely, showing almost no relationship between error and DP—as visualized in Figure 6, where its curve forms an almost vertical line in the upper-right corner.

Sinkhorn Divergence can excessively shrink latent representations and validates the need for PCA whitening for a proper DM metric. We observe that *NLL + Sink* often causes extensive distortion of the representation distribution, leading to extremely low DP values (e.g., 0.0025 on ACS-T, see table 7). However, this benefit comes at a substantial cost: accuracy is reduced by nearly 10% compared to other methods. Furthermore, the Sinkhorn distance between distributions is relatively small before applying PCA, which may lead to an unfair comparison across metrics (Table 7). This effect is well known that the learned latent space can have arbitrary scaling and geometric distortion [Jing et al., 2021, Ermolov et al., 2021]. This issue is evident in the SINK values reported in Table 7: before applying PCA, the *NLL + Sink* method often shrinks the distribution excessively, leading to artificially low SINK values. In contrast, logit-based methods do not exhibit this problem. Applying PCA whitening that we introduce mitigates this issue by normalizing the latent space, and we clearly observe that the resulting scales become comparable across methods.

Which DM method should we use for Fairness task We observe that logit-based methods capture a more consistent correlation between DM metrics and task-specific performance (section 4.2).

This consistency enables better control of the DP–ACC trade-off, as shown in fig. 2. Logit-based methods concentrate along the Pareto-optimal frontier of the fairness–utility trade-off (i.e., the lower-right region), and encounter the fairness–utility cliff only in the highest-accuracy regime.

In addition, we note that MMD-based methods can effectively manage the trade-off by producing a wide spread of points across different DP values (fig. 6). Sinkhorn-based methods are primarily concentrated in the high-accuracy region, but they are also associated with elevated DP values. In contrast, adversarial methods exhibit high variance: for latent-based regularization, points scatter widely across the plot, while for logit-based regularization, DP remains within a narrow range but accuracy is consistently the lowest among the three methods.

Takeaway for Fairness

We recommend practitioners to adopt logit-based methods over latent-based ones, as they provide a more favorable DP–ACC trade-off. Sinkhorn-based methods are preferable when achieving high accuracy is the primary goal, even at the cost of moderate DP gaps. In contrast, MMD-based methods are recommended when minimizing DP disparity is the priority, as they offer greater flexibility with respect to the accuracy–fairness trade-off.

4.3 Domain Adaptation Results

U-shaped Trend in Error In domain adaptation, Ben-David et al. [2006b] provide a useful bound on the target error in terms of the source error via the $\mathcal{H}$-divergence, suggesting that a good representation should achieve both low source error and low $\mathcal{H}$-divergence between source and target distributions. However, computing the $\mathcal{H}$-divergence is often impractical in practice, so we instead employ geometric divergences to approximate the distance between distributions. Interestingly, our experimental results deviate from the direct implication of this theoretical bound, a phenomenon also noted by Zhao et al. [2019b]. As shown in section C.3, low geometric divergence does not necessarily correspond to low target error. In fact, as geometric divergence increases up to a certain point, target error decreases, but beyond that point, the target error begins to rise again, forming a U-shaped trend.

Table 3: Experimental results for image classification tasks for domain adaptation. n: target examples; d: target features. We repeat all the experiments across 10 random seeds and report the mean and standard deviation for each metric. We bold top 2 methods if average values are tie.

Dataset	Training Objective	Source ACC ↑	Target ACC ↑	SINK ↓	SINK PCA ↓	MMD ↓	MMD PCA ↓
MNIST→USPS	NLL + MMD	$\mathbf{0.9583 \pm 0.006}$	0.6162 ± 0.047	135.9770 ± 17.334	746.6720 ± 0.802	0.1312 ± 0.016	0.0033 ± 0.000
n= 9298	NLL + Sink	0.9461 ± 0.007	$\mathbf{0.6895 \pm 0.029}$	$\mathbf{2.0890 \pm 0.126}$	$\mathbf{741.1450 \pm 0.253}$	$\mathbf{0.0112 \pm 0.002}$	$\mathbf{0.0021 \pm 0.000}$
d= 256	NLL + Adv	0.9412 ± 0.017	0.6053 ± 0.052	192.9870 ± 19.096	746.7900 ± 0.594	0.1315 ± 0.010	0.0033 ± 0.000
USPS→MNIST	NLL + MMD	0.9015 ± 0.025	$\mathbf{0.5647 \pm 0.060}$	78.6320 ± 11.157	742.5810 ± 0.557	0.0269 ± 0.004	0.0023 ± 0.000
n= 70000	NLL + Sink	$\mathbf{0.9106 \pm 0.018}$	0.5451 ± 0.031	$\mathbf{4.6300 \pm 0.260}$	$\mathbf{741.6220 \pm 0.380}$	$\mathbf{0.0040 \pm 0.001}$	$\mathbf{0.0022 \pm 0.000}$
d= 784	NLL + Adv	0.8859 ± 0.015	0.5010 ± 0.012	139.5100 ± 27.899	741.8970 ± 0.599	0.0343 ± 0.008	0.0022 ± 0.000

By leveraging motivating empirical result above, we can explain the information theoretic lower bound on [Zhao et al., 2019b] where source error is $\varepsilon_S(h \circ g) = \mathbb{E}_{x \sim D_S}[|h(g(x)) - f_S(x)|]$, $d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y)$ represent Jensen Shannon divergence (JSD) between marginal label distribution, and $d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$ represent JSD between latent distribution.

Theorem 1 (Restatement of Theorem 4.3 in Zhao et al. [2019b])**.** *Suppose the condition in Lemma 4.8 holds and* $d_{JS}(\mathcal{D}_S^Y, \mathcal{D}_T^Y) \geq d_{JS}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$*, then:*

$$\varepsilon_S(h \circ g) + \varepsilon_T(h \circ g) \geq \frac{1}{2}\left(d_{JS}(\mathcal{D}_S^Y, \mathcal{D}_T^Y) - d_{JS}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)\right)^2.$$

We can treat $d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y)$ as constants because they remain fixed while $d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$ varies. Since $d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y)$ is non-negative, we can minimize the target error and source erorr by making $d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$ close to $d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y)$. This trend can also be observed in section C.3. Both errors decrease down to a certain minimum point as increase of DM metrics, which implies that at minimum point i.e. $d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z) = d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y)$.

One downside of this theorem is that it does not explain the error increase once $d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$ exceeds $d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y)$, because theorem 1 requires $d_{\mathrm{JS}}(\mathcal{D}_S^Y, \mathcal{D}_T^Y) \geq d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$. Nevertheless, it still provides meaningful insight that we can achieve optimal target accuracy by controlling $d_{\mathrm{JS}}(\mathcal{D}_S^Z, \mathcal{D}_T^Z)$ via varying the regularization weight on distribution divergence.

Which DM method should we use for domain adaptation tasks?

As shown in section C.3, Sinkhorn exhibits a clear correlation with both error and DM metrics, whereas other measures appear noisy. This indicates that optimizing with Sinkhorn can effectively control the error. Furthermore, as presented in table 3, Sinkhorn achieves the highest target ACC on *MNIST→USPS* and the second-highest target ACC, which is very close to the best result.

Takeaway for Domain Adaptation

We recommend practitioners use Sinkhorn-based methods, as they can effectively control error by regulating DM metrics. Moreover, the U-shaped trend between error and DM metrics can guide practitioners to tune hyperparameters toward the optimal region.

5 Conclusion

We introduced a unified framework that casts calibration, group fairness, and domain adaptation as distribution matching (DM) problems, enabling fair comparisons of MMD, Sinkhorn, and adversarial approaches across tasks. Normalizing latent spaces via PCA whitening makes geometric DM metrics comparable and reveals consistent trends. Empirically, (i) plain NLL with post-hoc temperature scaling is a strong alibration baseline; (ii) logit-based fairness methods better navigate the DP–ACC trade-off than latent-based ones; and (iii) in domain adaptation, error follows a U-shaped curve versus DM strength, with Sinkhorn providing the most controllable knob. Our findings offer practical guidance and suggest future research directions. (1) Better DM-based calibration methods are needed, since a simple NLL baseline often outperforms most DM approaches. (2) Future work should develop logit-based fairness methods, as most existing algorithms remain latent-based.

References

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

Shun-ichi Amari. Divergence, optimization and geometry. In *International conference on neural information processing*, pages 185–193. Springer, 2009.

Shun-ichi Amari and Andrzej Cichocki. Information geometry of divergence functions. *Bulletin of the polish academy of sciences. Technical sciences*, 58 (1):183–195, 2010.

Barry Becker and Ronny Kohavi. Adult. *UCI Machine Learning Repository*, 10:C5XW20, 1996.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006a. URL `https://proceedings.neurips.cc/paper_files/paper/2006/file/b1b0432ceafb0ce714426e9114852ac7-Paper.pdf`.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006b.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Meta-calibration: Learning of model calibration using differentiable expected calibration error. *arXiv preprint arXiv:2106.09613*, 2021.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper_files/paper/2016/file/45fbc6d3e05ebd93369ce542e8f2322d-Paper.pdf`.

Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519, 2009.

Mingliang Chen and Min Wu. Towards threshold invariant fair classification. In *Conference on Uncertainty in Artificial Intelligence*, pages 560–569. PMLR, 2020.

Ching-Hao Chiu, Yu-Jen Chen, Yawen Wu, Yiyu Shi, and Tsung-Yi Ho. Achieve fairness without demographics for dermatological disease diagnosis. *Medical Image Analysis*, 95:103188, 2024.

Wonwoong Cho, Ziyu Gong, and David I. Inouye. Cooperative distribution alignment via jsd upper bound. In *Neural Information Processing Systems (NeurIPS)*, dec 2022.

Hao-Wei Chung, Ching-Hao Chiu, Yu-Jen Chen, Yiyu Shi, and Tsung-Yi Ho. Toward fairness via maximum mean discrepancy regularization on logits space. *arXiv preprint arXiv:2402.13061*, 2024a.

Hao-Wei Chung, Ching-Hao Chiu, Yu-Jen Chen, Yiyu Shi, and Tsung-Yi Ho. Toward fairness via maximum mean discrepancy regularization on logits space. *arXiv preprint arXiv:2402.13061*, 2024b.

Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 274–289. Springer, 2014.

Elliot Creager, David Madras, Joern-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1436–1445. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/creager19a.html`.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1095–1108, 2021.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International conference on machine learning*, pages 3015–3024. PMLR, 2021.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019a.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019b.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016a.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016b.

Tilmann Gneiting and Roopesh Ranjan. Combining predictive distributions. 2013.

Ziyu Gong, Ben Usman, Han Zhao, and David I. Inouye. Towards practical non-adversarial distribution matching. In **International Conference on Artificial Intelligence and Statistics (AISTATS)**, May 2024.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017a.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017b.

Xiaotian Han, Jianfeng Chi, Yu Chen, Qifan Wang, Han Zhao, Na Zou, and Xia Hu. Ffb: A fair fairness benchmark for in-processing group fairness methods. *arXiv preprint arXiv:2306.09468*, 2023a.

Xiaotian Han, Jianfeng Chi, Yu Chen, Qifan Wang, Han Zhao, Na Zou, and Xia Hu. Ffb: A fair fairness benchmark for in-processing group fairness methods. *arXiv preprint arXiv:2306.09468*, 2023b.

Yan Han, Ailin Hu, Qingqing Huang, Yan Zhang, Zhichao Lin, and Jinghua Ma. Sinkhorn divergence-based contrast domain adaptation for remaining useful life prediction of rolling bearings under multiple operating conditions. *Reliability Engineering & System Safety*, 253:110557, 2025.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019.

Archit Karandikar, Nicholas Cain, Dustin Tran, Balaji Lakshminarayanan, Jonathon Shlens, Michael C Mozer, and Becca Roelofs. Soft calibration objectives for neural networks. *Advances in Neural Information Processing Systems*, 34:29768–29779, 2021.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pages 2796–2804. PMLR, 2018.

Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial intelligence and statistics*, pages 623–631. PMLR, 2017.

Yanis Lalou, Théo Gnassounou, Antoine Collas, Antoine de Mathelin, Oleksii Kachaiev, Ambroise Odonnat, Alexandre Gramfort, Thomas Moreau, and Rémi Flamary. Skada-bench: Benchmarking unsupervised domain adaptation methods with realistic validation on diverse modalities, 2025. URL https://arxiv.org/abs/2407.11676.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 97–105, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/long15.html.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks, 2017. URL https://arxiv.org/abs/1605.06636.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing. *Logic, language, and security: essays dedicated to Andre Scedrov on the occasion of his 65th birthday*, pages 189–202, 2020.

Rachel Luo, Aadyot Bhatnagar, Yu Bai, Shengjia Zhao, Huan Wang, Caiming Xiong, Silvio Savarese, Stefano Ermon, Edward Schmerling, and Marco Pavone. Local calibration: metrics and recalibration. In *Uncertainty in Artificial Intelligence*, pages 1286–1295. PMLR, 2022.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018.

Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by distribution matching: Trainable kernel calibration metrics. *Advances in Neural Information Processing Systems*, 2023.

Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by distribution matching: trainable kernel calibration metrics. *Advances in Neural Information Processing Systems*, 36, 2024a.

Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by distribution matching: trainable kernel calibration metrics. *Advances in Neural Information Processing Systems*, 36, 2024b.

Rayan Mazouz, Karan Muvvala, Akash Ratheesh Babu, Luca Laurenti, and Morteza Lahijanian. Safety guarantees for neural network dynamic systems via stochastic barrier functions. *Advances in Neural Information Processing Systems*, 35:9672–9686, 2022.

Logan S McCarty. Quantum-inspired probability metrics define a complete, universal space for statistical learning. *arXiv preprint arXiv:2508.21086*, 2025.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015a.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015b.

Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6(1):405–431, 2019.

Sneh Pandya, Purvik Patel, Brian D. Nord, Mike Walmsley, and Aleksandra Ćiprijanović. Sidda: Sinkhorn dynamic domain adaptation for image classification with equivariant neural networks, 2025. URL https://arxiv.org/abs/2501.14048.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999a.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999b.

Drago Plecko and Elias Bareinboim. Fairness-accuracy trade-offs: A causal perspective. *arXiv preprint arXiv:2405.15443*, 2024.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Yu Qiao and Nobuaki Minematsu. f-divergence is a generalized invariant measure between distributions. In *INTERSPEECH*, pages 1349–1352, 2008.

Charan Reddy. Benchmarking bias mitigation algorithms in representation learning through fairness metrics. 2022a.

Charan Reddy. Benchmarking bias mitigation algorithms in representation learning through fairness metrics. 2022b.

Roshni Sahoo, Shengjia Zhao, Alyssa Chen, and Stefano Ermon. Reliable decisions with threshold calibration. *Advances in Neural Information Processing Systems*, 34:1831–1844, 2021.

Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11): 5973–6006, 2016.

Thibault Séjourné, Gabriel Peyré, and François-Xavier Vialard. Unbalanced optimal transport, from theory to numerics. *Handbook of Numerical Analysis*, 24: 407–471, 2023.

Carl-Johann Simon-Gabriel, Alessandro Barp, Bernhard Schölkopf, and Lester Mackey. Metrizing weak convergence with maximum mean discrepancies. *Journal of Machine Learning Research*, 24 (184):1–20, 2023.

Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pages 5897–5906. PMLR, 2019a.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2164–2173. PMLR, 2019b.

Linwei Tao, Younan Zhu, Haolan Guo, Minjing Dong, and Chang Xu. A benchmark study on calibration. *arXiv preprint arXiv:2308.11838*, 2023.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *The 22nd international conference on artificial intelligence and statistics*, pages 3459–3467. PMLR, 2019a.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *The 22nd international conference on artificial intelligence and statistics*, pages 3459–3467. PMLR, 2019b.

Deng-Bao Wang and Min-Ling Zhang. Calibration bottleneck: Over-compressed representations are less calibratable. In *Forty-first International Conference on Machine Learning*, 2024.

Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR, 2019a.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR, 2019b.

Han Zhao, Chen Dan, Bryon Aragam, Tommi S Jaakkola, Geoffrey J Gordon, and Pradeep Ravikumar. Fundamental limits and tradeoffs in invariant representation learning. *Journal of machine learning research*, 23(340):1–49, 2022.

Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. In *International Conference on Machine Learning*, pages 11387–11397. PMLR, 2020.

Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34:22313–22324, 2021.

Johanna F Ziegel and Tilmann Gneiting. Copula calibration. 2014.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Experiment Setup

We select the best model based on validation accuracy on a held-out set. To better understand the trade off between accuracy and task-specific metrics as well as to understand the sensitivity and variance of the DM methods, we then train multiple models by sweeping a range of the DM regularization parameter λ. Specifically, we choose a range of 100 total evenly log spaced interval for $\{\lambda^* \cdot 10^{-3}, \lambda^* \cdot 10^3\}$ where λ^* indicate best performing DM loss weight to capture overall effect of DM loss. This allows us to see the DM method's performance across a range of DM regularization values. From this we can compute the pairwise Kendall's Tau correlation matrix across accuracy, DM metrics (MMD and Sinkhorn), and task-specific metrics (ECE, DP, and target accuracy). We also use this setup for generating trade off plots. In addition, we report the peak GPU memory allocated (GiB) and average runtime. These metrics were obtained by running the best hyperparameter configuration across 10 random seeds, the same setup used for the main results for each tasks table 2, table 7, table 3.

B Proof of L2 distances invariance under PCA and ZCA whitening

Lemma 1 (Orthogonal invariance of the Euclidean norm)**.** *Let $R \in \mathbb{R}^{d\times d}$ be orthogonal (i.e. $R^\top R = I$). Then for all $z \in \mathbb{R}^d$,*

$$\|Rz\|_2^2 = z^\top R^\top Rz = z^\top z = \|z\|_2^2.$$

Hence $\|Rz - Rz'\|_2 = \|z - z'\|_2$ for all $z, z' \in \mathbb{R}^d$.

Proposition 4 (L2 distances are identical under ZCA vs. PCA whitening)**.** *Let a zero-mean random vector $X \in \mathbb{R}^d$ have covariance $\Sigma \succ 0$ with eigendecomposition $\Sigma = U\Lambda U^\top$ where U is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d) \succ 0$. Define the PCA and ZCA whitening maps*

$$W_{\mathrm{PCA}} := \Lambda^{-1/2}U^\top,$$
$$W_{\mathrm{ZCA}} := U\Lambda^{-1/2}U^\top.$$

Then for any $x, y \in \mathbb{R}^d$,

$$\left\|W_{\mathrm{ZCA}}x - W_{\mathrm{ZCA}}y\right\|_2 = \left\|W_{\mathrm{PCA}}x - W_{\mathrm{PCA}}y\right\|_2.$$

Proof. Observe that $W_{\mathrm{ZCA}} = U\, W_{\mathrm{PCA}}$. Set

$$z_{\mathrm{PCA}} := W_{\mathrm{PCA}}x, \qquad z'_{\mathrm{PCA}} := W_{\mathrm{PCA}}y,$$
$$z_{\mathrm{ZCA}} := W_{\mathrm{ZCA}}x = Uz_{\mathrm{PCA}}, \quad z'_{\mathrm{ZCA}} := W_{\mathrm{ZCA}}y = Uz'_{\mathrm{PCA}}.$$

By the lemma, since U is orthogonal,

$$\|z_{\mathrm{ZCA}} - z'_{\mathrm{ZCA}}\|_2 = \|U(z_{\mathrm{PCA}} - z'_{\mathrm{PCA}})\|_2 = \|z_{\mathrm{PCA}} - z'_{\mathrm{PCA}}\|_2.$$

□

C Additional Results

C.1 Calibration

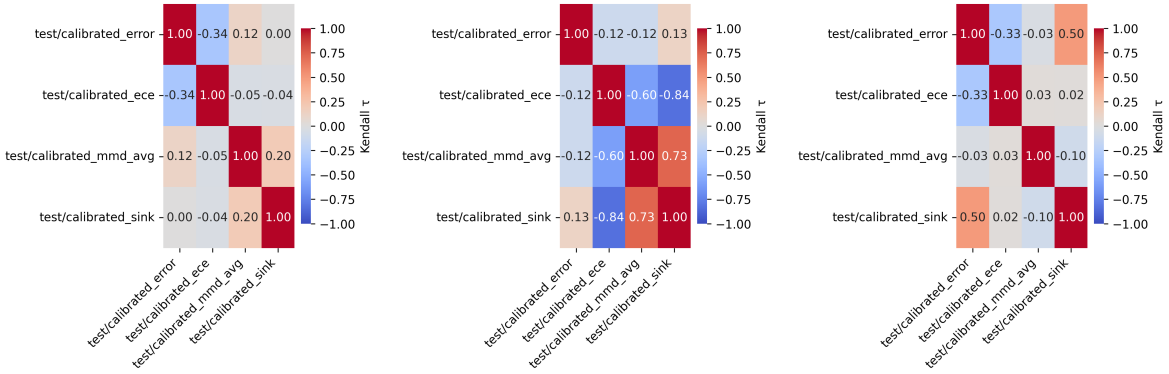
Post-hoc calibration benefits under-calibrated models more than already well-calibrated models. The magnitude of ECE reduction varies by training method. For example, on the ACS-T dataset, the decrease in ECE after temperature scaling (TS) under *cross-entropy loss/negative log-likelihood (NLL) + Sink* objective is the largest among all methods, starting from the highest initial ECE value (Table 2). Moreover, *NLL + Sink* shows the strongest trade-off between ECE and error, indicating that it is a poorly calibrated model Figure 1b. After applying TS, however, this trade-off disappears Figure 4b, while other methods that exhibit only moderate trade-offs continue to show them. This pattern is consistent with Wang et al. [2021], which reports that applying TS to an overconfident (poorly calibrated) model yields a larger ECE reduction than applying TS to an already well-calibrated model. Intuitively, stronger regularization compresses the distribution of maximum logits and softens per-example probabilities, leaving little headroom for post-hoc calibration.

Post-hoc calibration does not reduce ECE on small scale dataset. Interestingly, on the Compas, all methods exhibit an *increase* in ECE after applying TS while ECE stays same on German dataset. There are two potential explanations for this. First, as noted in Section C.1, the methods might already be well calibrated, leaving little room for improvement. However, this explanation can be ruled out: the standard deviation of ECE on Compas is substantially larger than on other datasets (ex, 0.003 on Adult vs. 0.053 on German; see Table 4), indicating that the models are not already well calibrated. A more plausible explanation is that TS relies on a validation set to determine the optimal temperature. When the validation set is small, TS can overfit to it for small feature dimension data (German) or under-fit on large feature dimension data (Compas) , leading to miscalibration [Guo et al., 2017b]. **Therefore, we do not recommend to use TS on small scale dataset.**

Which DM method is most sensitive to hyperparameter tuning for the Calibration task? Multi-task learning is often sensitive to the weights on auxiliary losses. Practitioners therefore prefer methods that are robust to the distribution-matching (DM) regularization weight λ. To quantify robustness, we compute the standard deviation (STD) of each key metric while varying only λ over 100 values evenly spaced on a log scale, holding all other hyperparameters fixed at the best-performing setting from Sec-

Table 4: Experimental results for tabular classification tasks for calibration. We display test metrics for each training procedure, with and without post-hoc calibration [Guo et al., 2017b]. n is the number of examples; d is the number of features. We repeat all the experiments across 10 random seeds and report the mean and standard deviation for each metric. We bold top 2 methods if average values are tie.

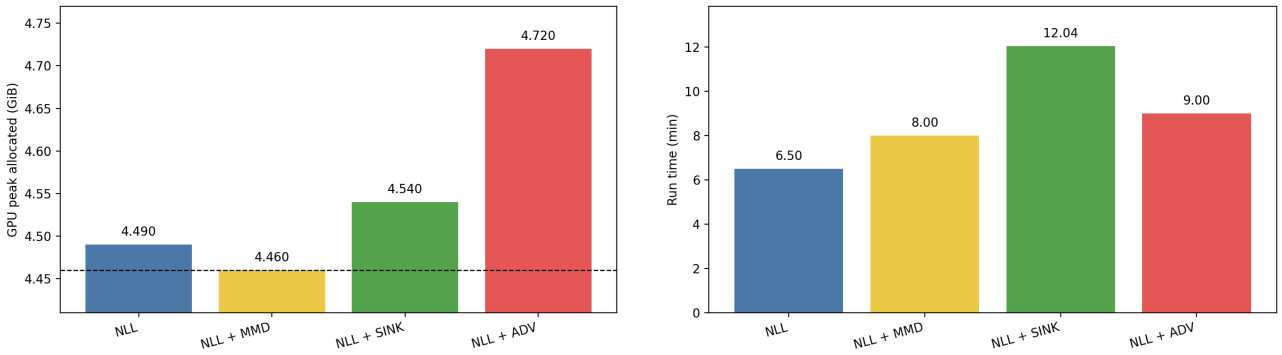
Dataset	Training Objective	ACC ↑	ECE ↓	SINK ↓	MMD ↓
ADULT	NLL	0.8435 ± 0.002	0.0163 ± 0.003	0.1894 ± 0.002	0.0000 ± 0.000
n= 30162	+ *Post-hoc*	0.8435 ± 0.002	0.0123 ± 0.002	0.1975 ± 0.001	0.0000 ± 0.000
d= 102	NLL + MMD	0.8438 ± 0.002	0.0175 ± 0.002	0.1887 ± 0.002	0.0000 ± 0.000
	+ *Post-hoc*	0.8438 ± 0.002	0.0120 ± 0.003	0.1973 ± 0.001	0.0000 ± 0.000
	NLL + Sink (Ours)	0.8429 ± 0.003	0.0166 ± 0.003	0.1889 ± 0.003	0.0000 ± 0.000
	+ *Post-hoc*	0.8429 ± 0.003	0.0122 ± 0.003	0.1972 ± 0.001	0.0000 ± 0.000
	NLL + Adv (Ours)	$\mathbf{0.8446 \pm 0.002}$	0.0155 ± 0.003	$\mathbf{0.1884 \pm 0.001}$	0.0000 ± 0.000
	+ *Post-hoc*	$\mathbf{0.8446 \pm 0.002}$	$\mathbf{0.0114 \pm 0.002}$	0.1971 ± 0.001	0.0000 ± 0.000
COMPAS	NLL	0.6481 ± 0.033	0.0402 ± 0.011	0.2936 ± 0.009	0.0001 ± 0.000
n= 6172	+ *Post-hoc*	0.6481 ± 0.033	0.0506 ± 0.013	0.2995 ± 0.008	0.0001 ± 0.000
d= 401	NLL + MMD	0.6372 ± 0.033	0.0407 ± 0.014	0.2956 ± 0.010	0.0000 ± 0.000
	+ *Post-hoc*	0.6372 ± 0.033	0.0540 ± 0.015	0.3039 ± 0.007	0.0001 ± 0.000
	NLL + Sink (Ours)	0.6361 ± 0.034	$\mathbf{0.0400 \pm 0.010}$	0.2984 ± 0.008	0.0000 ± 0.000
	+ *Post-hoc*	0.6361 ± 0.034	0.0546 ± 0.017	0.3061 ± 0.007	0.0000 ± 0.000
	NLL + Adv (Ours)	$\mathbf{0.6579 \pm 0.012}$	$\mathbf{0.0402 \pm 0.009}$	$\mathbf{0.2903 \pm 0.007}$	0.0001 ± 0.000
	+ *Post-hoc*	$\mathbf{0.6579 \pm 0.012}$	0.0555 ± 0.012	0.2991 ± 0.006	0.0001 ± 0.001
GERMAN	NLL	$\mathbf{0.6605 \pm 0.024}$	$\mathbf{0.1021 \pm 0.023}$	0.3551 ± 0.013	0.0001 ± 0.000
n= 1000	+ *Post-hoc*	0.6605 ± 0.024	0.1044 ± 0.021	0.3560 ± 0.009	0.0001 ± 0.000
d= 58	NLL + MMD	$\mathbf{0.6620 \pm 0.024}$	0.1084 ± 0.014	$\mathbf{0.3523 \pm 0.006}$	0.0001 ± 0.000
	+ *Post-hoc*	$\mathbf{0.6620 \pm 0.024}$	0.1058 ± 0.020	0.3560 ± 0.004	0.0001 ± 0.000
	NLL + Sink (Ours)	0.6610 ± 0.020	0.1188 ± 0.017	0.3542 ± 0.015	0.0001 ± 0.000
	+ *Post-hoc*	0.6610 ± 0.020	0.1099 ± 0.016	0.3606 ± 0.013	0.0001 ± 0.000
	NLL + Adv (Ours)	0.5075 ± 0.118	0.1104 ± 0.053	0.4906 ± 0.027	0.0001 ± 0.000
	+ *Post-hoc*	0.5075 ± 0.118	0.1098 ± 0.041	0.4819 ± 0.029	0.0001 ± 0.000
ACS-T	NLL	0.6502 ± 0.003	0.0397 ± 0.004	0.4087 ± 0.002	0.0000 ± 0.000
n= 172508	+ *Post-hoc*	0.6502 ± 0.003	$\mathbf{0.0153 \pm 0.003}$	0.4213 ± 0.002	0.0000 ± 0.000
d= 1567	NLL + MMD	0.6496 ± 0.003	0.0430 ± 0.004	0.4081 ± 0.002	0.0000 ± 0.000
	+ *Post-hoc*	0.6496 ± 0.003	0.0156 ± 0.005	0.4213 ± 0.001	0.0000 ± 0.000
	NLL + Sink (Ours)	$\mathbf{0.6585 \pm 0.001}$	0.1411 ± 0.002	$\mathbf{0.3722 \pm 0.001}$	0.0000 ± 0.000
	+ *Post-hoc*	$\mathbf{0.6585 \pm 0.001}$	0.0192 ± 0.002	0.4186 ± 0.001	0.0000 ± 0.000
	NLL + Adv (Ours)	0.6497 ± 0.003	0.0414 ± 0.004	0.4089 ± 0.002	0.0000 ± 0.000
	+ *Post-hoc*	0.6497 ± 0.003	$\mathbf{0.0154 \pm 0.004}$	0.4217 ± 0.001	0.0000 ± 0.000



(a) Calibration MMD (b) Calibration Sinkhorn Divergence (c) Calibration Adversarial

Figure 4: **After temperature scaling** Kendall ranking correlation matrix of task specific metric (ECE, Error), and DM metric (MMD, Sinkhorn Divergence) across different calibration methods on ACS-T dataset

tion 4.1. As shown in Table 5, Calib (Sink) attains the lowest STD for ACC, whereas Calib (MMD) exhibits superior robustness on the other metrics. This aligns with the different smoothness and variance proper-



(a) Peak GPU memory allocated (GiB).

(b) Runtime (minutes).

Figure 5: Compute profile for calibration methods on ACS-T: (a) peak GPU memory (GiB) and (b) runtime (min). Each point is averaged over the same 10 runs used for Table 2.

ties of the regularizers: kernel MMD is bounded and yields Lipschitz-smooth, low-variance minibatch estimates, so scaling λ perturbs optimization more gently [Gretton et al., 2012]. By contrast, Sinkhorn divergences, while enjoying fast convergence, produce sharper transport-aware gradients than MMD (see Fig. 4 in [Feydy et al., 2019b]), making performance more sensitive to λ. Finally, adversarial objectives introduce additional min–max stochasticity, which typically amplifies sensitivity to λ [Ganin et al., 2016b]. Therefore, **NLL + MMD)** is the most robustness method in terms of λ sensitivity.

Table 5: Standard deviation of metrics across calibration methods when varying λ with 100 evenly log-spaced values (lower is better) on the best-performing model on ACS-T dataset.

Method	ACC	ECE	Sink	MMD
NLL + MMD	$\mathbf{1.475 \times 10^{-4}}$	$\mathbf{2.030 \times 10^{-4}}$	$\mathbf{1.172 \times 10^{-5}}$	1.47×10^{-1}
NLL + SINK	2.938×10^{-3}	9.485×10^{-2}	2.720×10^{-2}	3.66×10^{-7}
NLL + ADV	1.036×10^{-3}	1.135×10^{-2}	3.907×10^{-3}	$\mathbf{3.51 \times 10^{-9}}$

Which DM method is most computationally efficient? We report peak GPU memory allocated (GiB) and average run time by running the best hyperparameter configuration across 10 random seeds, as shown in Table 2. As illustrated in Figure 10a, *NLL+Adv* requires the largest GPU memory footprint since it introduces a discriminator network. However, overall, there is no significant difference within method. In terms of run time, *NLL+Sink* is the most expensive due to the iterative nature of optimal transport. Overall, both in memory and run time, plain *NLL* remains the most efficient method, which also achives comparable result table 2.

C.2 Fairness

Comparison between different methods We note that MMD-based methods can effectively manage the trade-off by producing a wide spread of points across different DP values fig. 6. Sinkhorn-based methods are primarily concentrated in the high-accuracy region, but they are also asso- ciated with elevated DP values. In contrast, adver- sarial methods exhibit high variance: for latent-based regularization, points scatter widely across the plot, while for logit-based regularization, DP remains within a narrow range but accuracy is consistently the lowest among the three methods.

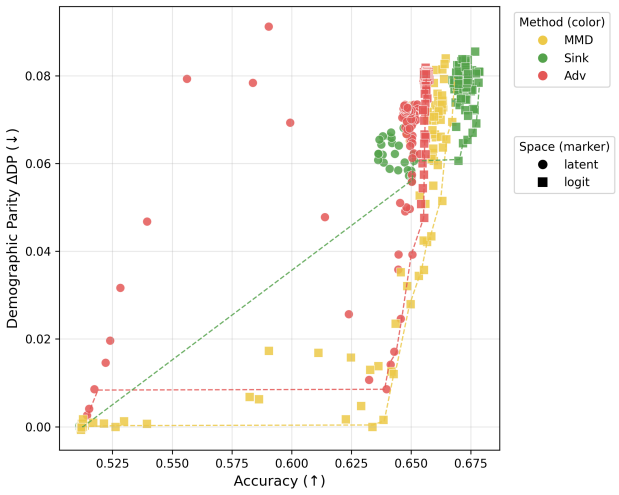


Figure 6: Fairness–accuracy trade-off comparison across methods, with each method distinguished by a unique **color** and each representation type (latent vs. logit) by a distinct marker **shape**.

Which DM method is most sensitive to hyperparameter tuning for the Fairness task? We adopt the same setting as in section C.1 to obtain these results. We expected that *NLL + Adv* would have the largest deviation due to the nature of min-max optimization. For the latent-based methods, this matches our observation, but interestingly, for the logit-based methods, it shows the lowest standard deviation (STD) among the other logit-based methods in table 6. We can observe this trend in **??**, where the latent-based method shows scattered points across various regions (red circles), while the logit-based method's points are concentrated in a specific region (red squares). For the latent-based methods, *NLL + MMD* shows the lowest STD. We also observe that applying PCA before evaluating divergence effectively decreases the STD, which implies a more stable metric, as shown in table 6.

Which DM method is most computationally efficient? We followed the same setup as in section C.1 to measure the average runtime and peak GPU memory allocation. The logit-based methods are more efficient in terms of runtime while maintaining comparable peak GPU memory consumption. Therefore, we recommend using logit-based methods over their latent-based counterparts, as they not only provide a better DP-ACC tradeoff (fig. 2) but also offer greater efficiency.

C.3 Domain Adaptation

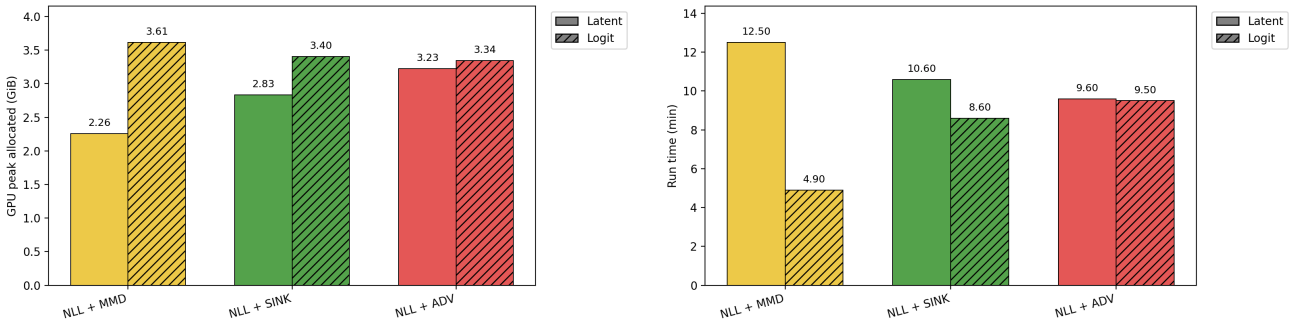
U-shape trend over Error and DM metrics. Following the setup in section C.1, we observe a U-shaped trend for error versus DM metrics when varying the strength of λ. This supports the finding that strict distribution matching is not always beneficial [Zhao et al., 2019b]. We identify a Pareto front containing models with the lowest error, and the U-shaped trend implies these models can be reached by adjusting the strength of the DM loss weight figs. 11 and 12. Notably, this trend is absent for *NLL+Adv*, which does not directly minimize geometric divergence.

Which DM method is most sensitive to hyperparameter tuning for the Domain Adaptation task? As expected, *NLL + ADV* exhibits a low standard deviation (STD). This is because its performance shows little sensitivity to the value of λ (fig. 11), a finding confirmed by its low STD scores across most metrics in table 8. However, this stability can be a drawback, as it also means that it is difficult to optimize performance by sweeping the λ hyperparameter.

Table 8: Standard deviation of metrics across domain adaptation methods when varying λ with 100 evenly log-spaced values (lower is better) on the best-performing model on MNIST → USPS dataset.

Method	ACC	Target ACC	Sink	Sink PCA	MMD	MMD PCA
NLL + MMD	$\mathbf{5.299 \times 10^{-3}}$	4.324×10^{-2}	**3.461**	3.968×10^{-1}	3.027×10^{-2}	6.093×10^{-5}
NLL + SINK	2.232×10^{-2}	5.817×10^{-2}	3.581	1.156	3.071×10^{-2}	2.644×10^{-4}
NLL + ADV	1.270×10^{-2}	$\mathbf{1.704 \times 10^{-2}}$	11.699	$\mathbf{3.496 \times 10^{-1}}$	$\mathbf{2.237 \times 10^{-2}}$	$\mathbf{5.293 \times 10^{-5}}$

Which DM method is most computationally efficient? We expect that geometric divergence-based methods would have high computational costs, an assumption supported by the peak GPU memory consumption shown in fig. 10. Both MMD and Sinkhorn exhibit higher GPU memory usage. However, regarding runtime, these geometric methods are actually faster due to quicker convergence, whereas the adversarial approach *NLL + Adv* requires a longer training duration.



(a) Peak GPU memory allocated (GiB).

(b) Runtime (minutes).

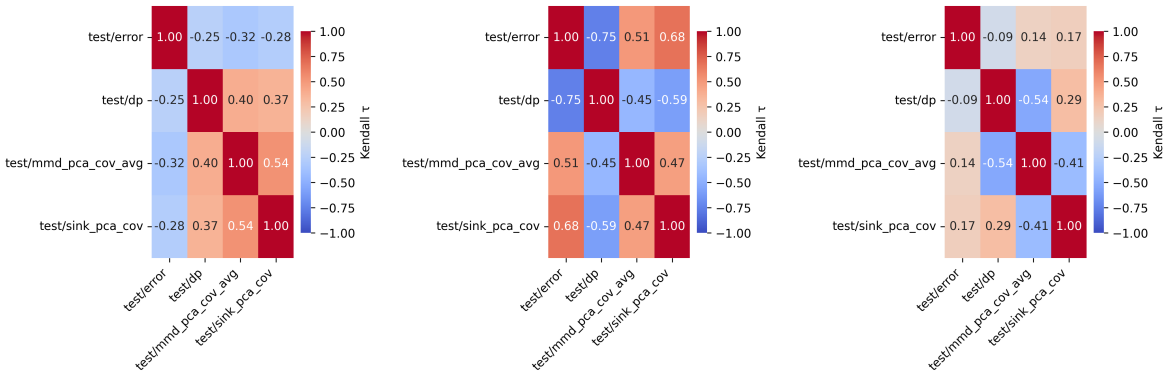
Figure 7: Compute profile for fairness methods on ACS-T: (a) peak GPU memory (GiB) and (b) runtime (min). Each point is averaged over the same 10 runs used for Table 7.

Table 6: Standard deviation of metrics across fairness both **logit and latent based** methods when varying λ with 100 evenly log- spaced values (lower is better) on the best-performing model on ACS-T dataset.

Method	ACC	DP	Sink	Sink PCA	MMD	MMD PCA
NLL + MMD (logit)	6.626×10^{-2}	3.285×10^{-2}	6.181×10^{-2}	–	4.502×10^{-4}	–
NLL + SINK (logit)	2.382×10^{-3}	$\mathbf{4.238 \times 10^{-3}}$	9.751×10^{-2}	–	1.468×10^{-4}	–
NLL + ADV (logit)	$\mathbf{4.178 \times 10^{-4}}$	6.641×10^{-3}	$\mathbf{2.495 \times 10^{-3}}$	–	$\mathbf{5.241 \times 10^{-5}}$	–
NLL + MMD (latent)	$\mathbf{4.196 \times 10^{-4}}$	$\mathbf{1.438 \times 10^{-3}}$	1.123	$\mathbf{8.507 \times 10^{-2}}$	$\mathbf{3.700 \times 10^{-5}}$	$\mathbf{2.304 \times 10^{-6}}$
NLL + SINK (latent)	5.706×10^{-2}	2.701×10^{-2}	$\mathbf{5.215 \times 10^{-2}}$	4.999	7.553×10^{-5}	1.126×10^{-5}
NLL + ADV (latent)	3.423×10^{-2}	1.947×10^{-2}	4.844	1.852	6.713×10^{-3}	2.677×10^{-4}

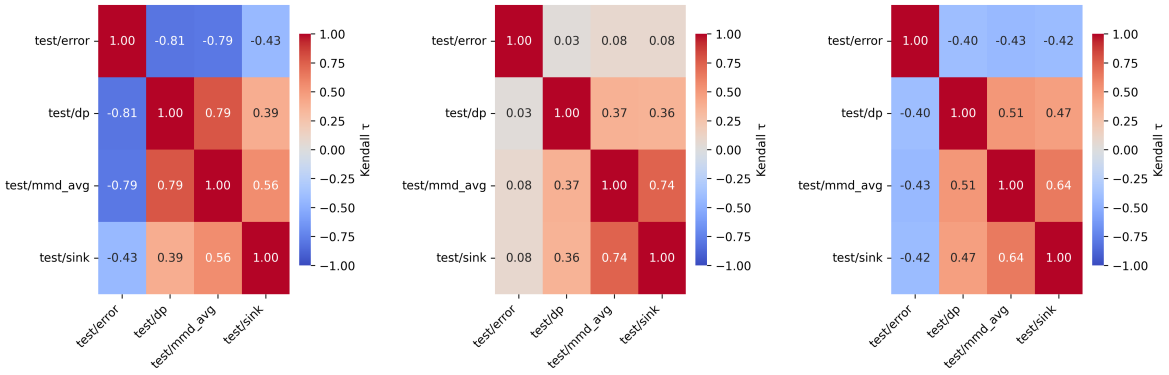
Table 7: Experimental results for tabular classification tasks (latent vs. logit). n: examples; d: features. We repeat all the experiments across 10 random seeds and report the mean and standard deviation for each metric. We bold top 2 methods if average values are tie.

Dataset	Training Objective	ACC ↑	DP ↓	SINK ↓	SINK PCA ↓	MMD ↓	MMD PCA ↓
Adult	NLL + MMD (latent)	0.8438 ± 0.001	0.1898 ± 0.008	11.2550 ± 0.562	99.0750 ± 0.499	0.0113 ± 0.001	$\mathbf{0.0031 \pm 0.000}$
n= 30162	NLL + Sink (latent)	$\mathbf{0.8450 \pm 0.002}$	0.1886 ± 0.005	$\mathbf{0.0462 \pm 0.009}$	$\mathbf{98.8210 \pm 0.783}$	$\mathbf{0.0015 \pm 0.000}$	$\mathbf{0.0031 \pm 0.000}$
d= 102	NLL + Adv (latent)	0.8436 ± 0.002	0.1885 ± 0.005	11.7600 ± 0.709	99.1190 ± 0.489	0.0117 ± 0.001	$\mathbf{0.0031 \pm 0.000}$
	NLL + MMD (logit)	0.8439 ± 0.002	0.1906 ± 0.008	1.6260 ± 0.104	—	0.0245 ± 0.002	—
	NLL + Sink (logit)	0.8435 ± 0.002	$\mathbf{0.1874 \pm 0.006}$	1.2580 ± 0.055	—	0.0215 ± 0.002	—
	NLL + Adv (logit)	0.8429 ± 0.002	0.1921 ± 0.007	1.5520 ± 0.120	—	0.0245 ± 0.001	—
Compas	NLL + MMD (latent)	0.6433 ± 0.035	0.1153 ± 0.036	19.0260 ± 1.060	96.8740 ± 0.539	0.0024 ± 0.000	$\mathbf{0.0028 \pm 0.000}$
n= 6172	NLL + Sink (latent)	$\mathbf{0.6639 \pm 0.007}$	0.1252 ± 0.013	0.9221 ± 0.115	$\mathbf{95.3320 \pm 0.875}$	$\mathbf{0.0013 \pm 0.000}$	$\mathbf{0.0028 \pm 0.000}$
d= 401	NLL + Adv (latent)	0.6571 ± 0.007	0.1241 ± 0.014	19.2330 ± 1.032	97.0250 ± 0.473	0.0023 ± 0.000	$\mathbf{0.0028 \pm 0.000}$
	NLL + MMD (logit)	0.6518 ± 0.034	0.1144 ± 0.037	0.0669 ± 0.022	—	0.0040 ± 0.001	—
	NLL + Sink (logit)	0.5883 ± 0.054	$\mathbf{0.0545 \pm 0.051}$	$\mathbf{0.0369 \pm 0.039}$	—	0.0030 ± 0.001	—
	NLL + Adv (logit)	0.6583 ± 0.009	0.1230 ± 0.018	0.0712 ± 0.022	—	0.0041 ± 0.001	—
ACS-T	NLL + MMD (latent)	0.6493 ± 0.002	0.0738 ± 0.004	30.6040 ± 1.314	97.9590 ± 0.358	0.0022 ± 0.000	$\mathbf{0.0013 \pm 0.000}$
n= 172508	NLL + Sink (latent)	0.5103 ± 0.007	$\mathbf{0.0025 \pm 0.005}$	0.0455 ± 0.014	99.7710 ± 0.229	$\mathbf{0.0003 \pm 0.000}$	$\mathbf{0.0013 \pm 0.000}$
d= 1567	NLL + Adv (latent)	0.6508 ± 0.003	0.0758 ± 0.005	25.2920 ± 2.395	$\mathbf{94.9840 \pm 2.629}$	0.0028 ± 0.001	0.0014 ± 0.000
	NLL + MMD (logit)	0.6464 ± 0.004	0.0392 ± 0.006	0.1066 ± 0.038	—	0.0008 ± 0.000	—
	NLL + Sink (logit)	$\mathbf{0.6665 \pm 0.002}$	0.0820 ± 0.005	0.2171 ± 0.107	—	0.0013 ± 0.000	—
	NLL + Adv (logit)	0.6511 ± 0.002	0.0776 ± 0.004	$\mathbf{0.0418 \pm 0.007}$	—	0.0010 ± 0.000	—



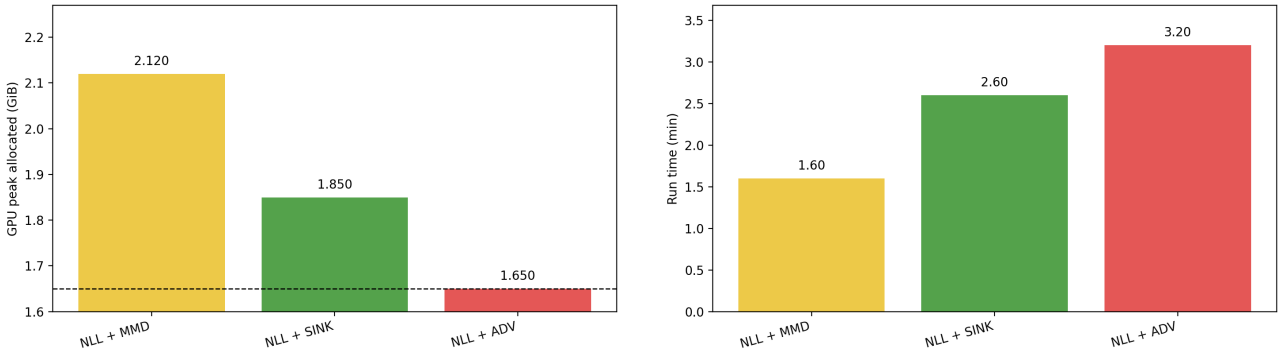
(a) Fairness MMD Latent based (b) Fairness Sinkhorn Divergence Latent based (c) Fairness Adversarial Latent based

Figure 8: **Latent based.** Kendall ranking correlation matrix of task specific metric (DP ,Error), and DM metric (MMD, Sinkhorn Divergence) across different fairness methods on ACS-T dataset



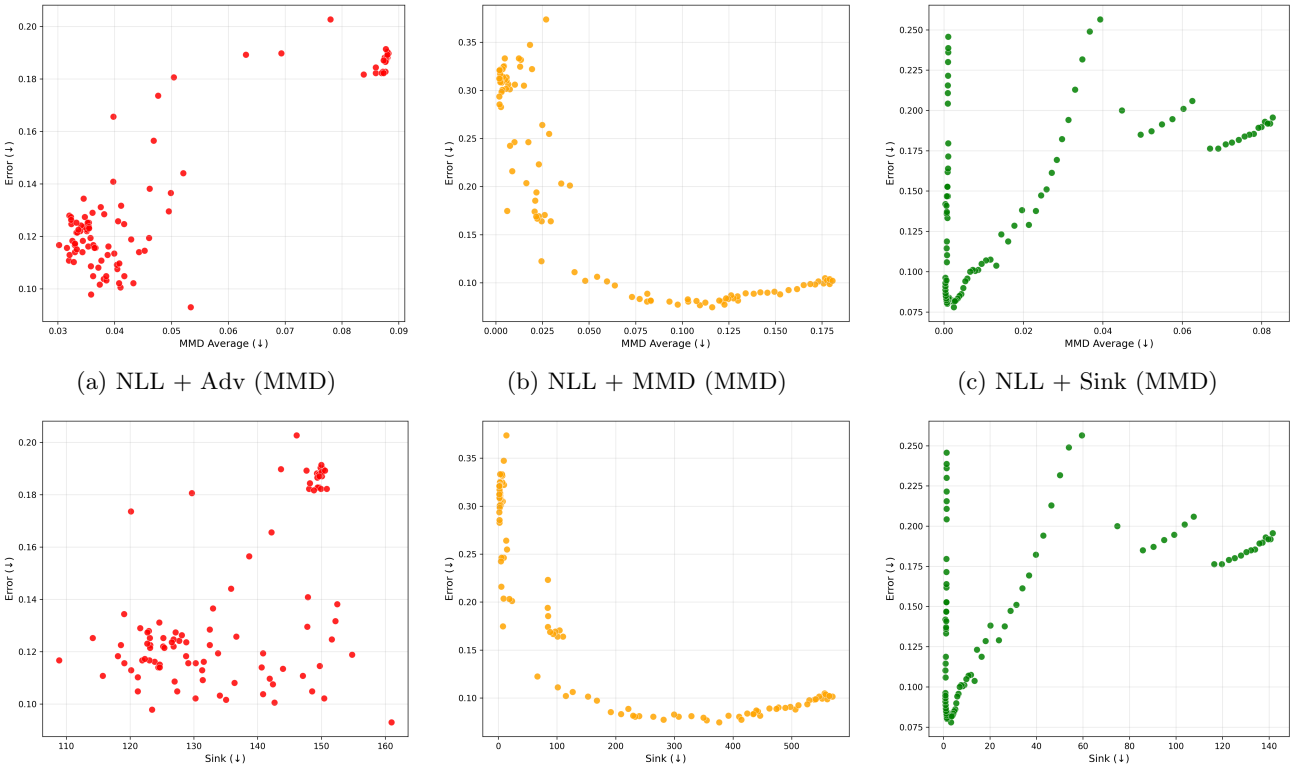
(a) Fairness MMD Logit based (b) Fairness Sinkhorn Divergence Logit based (c) Fairness Adversarial

Figure 9: **Logit based** Kendall ranking correlation matrix of task specific metric (DP ,Error), and DM metric (MMD, Sinkhorn Divergence) across different fairness methods on ACS-T dataset



(a) Peak GPU memory allocated (GiB). (b) Runtime (minutes).

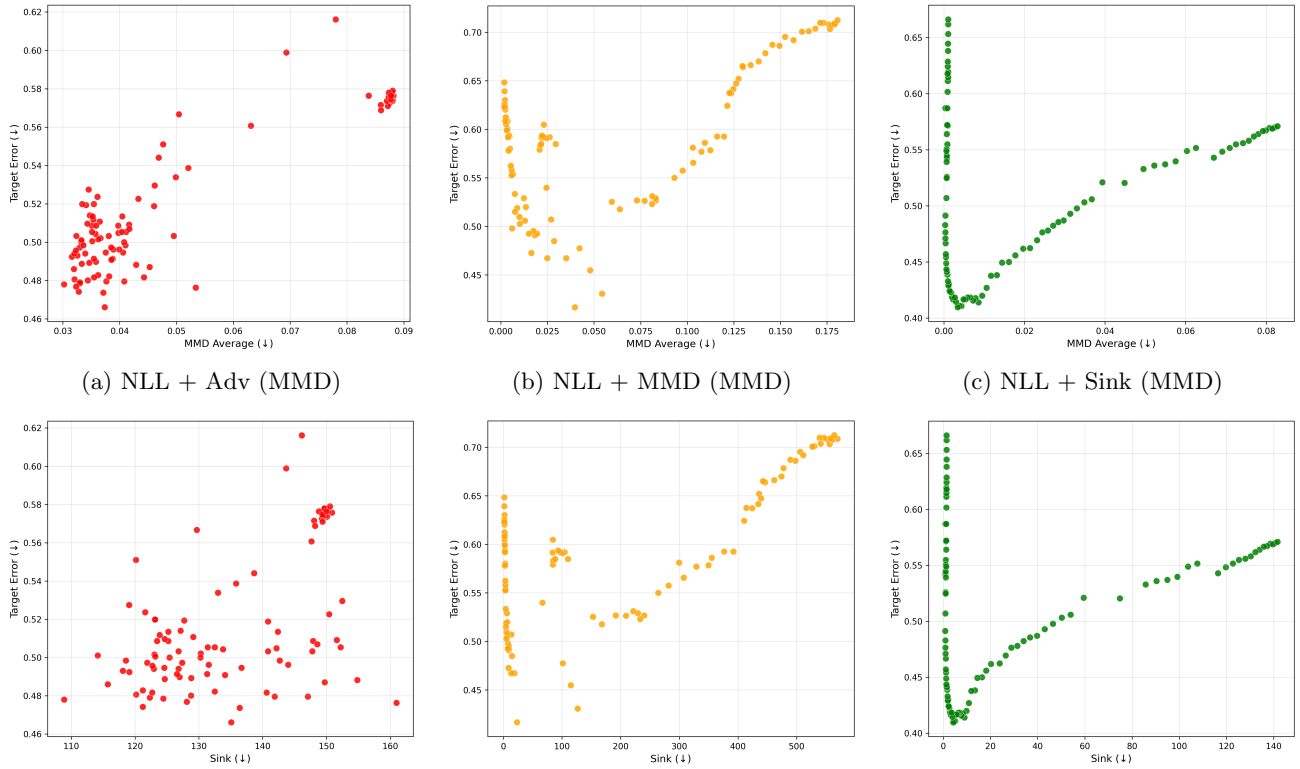
Figure 10: Compute profile for domain adaptation methods on MNIST→USPS: (a) peak GPU memory (GiB) and (b) runtime (min). Each point is averaged over the same 10 runs used for Table 3.



(a) NLL + Adv (MMD) (b) NLL + MMD (MMD) (c) NLL + Sink (MMD)

(d) NLL + Adv (Sink) (e) NLL + MMD (Sink) (f) NLL + Sink (Sink)

Figure 11: **Error vs. MMD and Sinkhorn.** Top: error vs. MMD. Bottom: error vs. Sinkhorn. U-shaped trends are visible for MMD/Sink sweeps, while Adv shows weak correlation.



(a) NLL + Adv (MMD) (b) NLL + MMD (MMD) (c) NLL + Sink (MMD)

(d) NLL + Adv (Sink) (e) NLL + MMD (Sink) (f) NLL + Sink (Sink)

Figure 12: **Target Error vs. MMD and Sinkhorn.** Top: target error vs. MMD. Bottom: target error vs. Sinkhorn. U-shaped trends are visible for MMD/Sink sweeps; Adv shows weak correlation.

D Related Work

Calibration Even though many deep learning models achieve high predictive performance, they often produce unreliable predictions due to a lack of calibration. Most deep learning models tend to be overconfident, as indicated by spiking posterior distributions Guo et al. [2017a]. Several factors contribute to this issue, including over-parameterized networks, insufficient regularization, limited data, and imbalanced label distributions Guo et al. [2017a]. There has been extensive research on calibration in both classification Bröcker [2009], Kull et al. [2017], Naeini et al. [2015b], Platt et al. [1999b], Dwork et al. [2021], Hébert-Johnson et al. [2018], Pleiss et al. [2017] and regression tasks Ziegel and Gneiting [2014], Kuleshov et al. [2018], Gneiting and Ranjan [2013], Song et al. [2019a], Zhao et al. [2020]. However, much of the community's focus has been on binary classification settings Karandikar et al. [2021], Vaicenavicius et al. [2019a], Bohdal et al. [2021], Platt et al. [1999a], Guo et al. [2017a]. Recently, Marx et al. [2024a] extended calibration into the distribution matching framework by leveraging the Maximum Mean Discrepancy (MMD)-based metric. This work unified recent advances in calibration across classification and regression tasks Kuleshov et al. [2018], Sahoo et al. [2021], Gneiting and Ranjan [2013], Zhao et al. [2021], Pessach and Shmueli [2022], Song et al. [2019a], Zhao et al. [2020], Luo et al. [2022]. Among the various calibration methods, our work focuses on individual calibration Zhao et al. [2020] conditioned on the variable $\mathbf{x}$.

Fairness Fairness in machine learning has garnered significant attention from the research community, with the primary goal of ensuring that machine learning models do not exhibit bias toward specific groups or individuals. Fairness algorithms are broadly categorized into two types: group fairness and individual fairness. Group fairness emphasizes equitable treatment across predefined demographic groups (e.g., male and female), while individual fairness ensures that similar individuals are treated similarly. To mitigate bias in machine learning models, researchers have proposed three primary strategies: preprocessing Creager et al. [2019], Lu et al. [2020], in-processing Chen and Wu [2020], Chiu et al. [2024], and post-processing Dwork et al. [2012], Hardt et al. [2016]. Preprocessing techniques modify the data before training, such as through normalization, relabeling, or reweighting. Post-processing methods adjust model outputs after training, typically at test time. In contrast, in-processing approaches impose fairness constraints during the training phase and have gained significant attention due to their ability to directly influence model behavior.

Our work focuses on in-processing methods, which are particularly relevant for enforcing fairness constraints during training. Prior studies in this area have primarily concentrated on specific applications or methods, often restricting their analysis to either latent space or logit space techniques. For instance, recent benchmark efforts have predominantly explored latent space approaches without extending their analysis to logit space methods Han et al. [2023b]. Additionally, these works often fail to provide a comprehensive comparison across different fairness methods or applications. In contrast, our study systematically evaluates in-processing methods by leveraging fairness techniques in both latent and logit spaces. We incorporate distribution-matching constraints and then evaluate their effectiveness using both information-theoretic and geometric divergence metrics. Consequently, we have a more holistic understanding of the trade-offs between different fairness methods. By addressing these gaps, our work provides a more comprehensive benchmark for group fairness methods compared to existing literature.

Domain Adaptation Domain adaptation seeks to enhance model generalization on out-of-distribution data. In this work, we focus on closed-set unsupervised domain adaptation, where the source and target domains share the same label space, but only the source domain is labeled.

Early methods aligned source and target feature distributions using statistical losses—for example, integrating a multi-kernel Maximum Mean Discrepancy (MMD) loss into deep neural networks Long et al. [2015]. Subsequent works refined these techniques [Long et al., 2017, Bousmalis et al., 2016] or introduced related MMD variants [**?**Zellinger et al., 2017, Kang et al., 2019]. In parallel, adversarial approaches have gained traction due to its flexibility and effectiveness. By incorporating a domain discriminator that distinguishes between source and target features, feature extractors can be trained to deceive the discriminator, thereby promoting domain-invariant representations [Ajakan et al., 2014, Ganin and Lempitsky, 2015, Ganin et al., 2016b, Tzeng et al., 2017]. Although less common, recent studies have also leveraged Sinkhorn divergences for domain adaptation [Pandya et al., 2025, Han et al., 2025], offering a promising alternative that efficiently aligns latent spaces via regularized optimal transport.

Many previous domain adaptation benchmarks evaluate models with dedicated designs that are intrinsically tied to specific divergence measures and task formulations [Lalou et al., 2025, **?**]. In contrast, our work introduces a unified distribution matching frame-

work that employs a generalized network architecture across all experiments. By keeping the architecture fixed, we interchange different divergence measures (e.g., Sinkhorn, adversarial, MMD, and variational methods) and systematically assess their relationship with domain adaptation performance under uniform experimental conditions.

D.1 Unified Training Objectives for Distribution Matching

D.1.1 Comparison between Information-Theoretic Divergence vs Geometric Divergence

We broadly categorize differentiable divergences into information-theoretic and geometric. Information-theoretic divergences Amari and Cichocki [2010] are usually estimated using a variational approximation. Information-theoretic divergences have the elegant property of being invariant under invertible transformations [Qiao and Minematsu, 2008] and thus are very useful when operating in latent spaces where the scale is irrelevant. Moreover, it can be computed with $O(N)$ for discrete measure Séjourné et al. [2023]. The drawbacks are information-theoretic divergences usually require learning an auxiliary variational model, which may be challenging itself and it is sensitive to support mismatch Séjourné et al. [2023]. Geometric divergences on the other hand use distances between points in the space and thus vary with scale Amari [2009]. This makes it more challenging to apply geometric-based divergences in latent space as simple scaling transformations drastically change these divergence measures. However, the two most common geometric divergences, Wasserstein and MMD, can be non-parametrically approximated using only a batch of samples from both domains without the need to train an auxiliary model. Also, geometric divergence metrize weak* topology that is $\alpha_n \rightharpoonup \alpha \Leftrightarrow L(\alpha_n, \alpha) \to 0$, which implies that a lower loss corresponds to closer distribution matching Feydy et al. [2019b].

D.1.2 Information Theoretic Divergences via Parametric Variational Bounds

Most differentiable approximations to information-theoretic divergences are bounds that involve training a variational model h_ϕ, such that the bound is tight if optimized perfectly but otherwise remains a bound. Adversarial GAN-based approaches form a variational *lower* bound on a divergence. The standard GAN-based loss bounds the JS divergence and trains a classifier with cross entropy loss ℓ_{CE} to predict the domain label:

$$\underline{\mathrm{D}}_{\text{ADV}}(\theta) := \max_{\phi} \mathbb{E}_p[-\ell_{\text{CE}}(h_\phi \circ g_\theta(\mathbf{x}), \mathrm{d})] \leq \mathrm{D}_{\text{JSD}}(\theta)\,. \tag{5}$$

Adversarial objectives for all f-divergences Sason and Verdú [2016] and even Wasserstein distance Panaretos and Zemel [2019] (a geometric divergence) can be formulated. Notice that the DM problem involves minimizing this approximation and thus it forms a min-max, i.e., adversarial problem, hence the name.

In contrast to adversarial lower bounds, there have been multiple approaches to form variational *upper* bounds. One of the more common bounds is based on a variational autoencoder (VAE) structure. Recently, [Gong et al., 2024] generalized previous VAE-based approaches into a self-contained loss similar to the adversarial loss above that upper bounds the JSD:

$$\overline{\mathrm{D}}_{\text{VAUB}}(\theta) := \min_{\phi} \mathbb{E}_p\Big[-\log\Big(\frac{q_\phi(\mathbf{x}|\mathbf{z},\mathrm{d})}{p_\theta(\mathbf{z}|\mathbf{x},\mathrm{d})}\cdot q_\phi(\mathbf{z})\Big)\Big] + C \tag{6}$$

$$\geq \mathrm{D}_{\text{JSD}}(\theta)\,, \tag{7}$$

where $g_\theta(\mathbf{x};\mathrm{d},\epsilon)$ is a *stochastic* encoder using the reparameterization trick where $\epsilon \sim \mathcal{N}(0, I)$, $q_\phi(\mathbf{x},\mathbf{z}|\mathrm{d}) := q_\phi(\mathbf{z})q_\phi(\mathbf{x}|\mathbf{z},\mathrm{d})$ is a decoder distribution where $q_\phi(\mathbf{z})$ is a learnable prior distribution, and C is a constant that is independent of θ and ϕ. If q_ϕ is minimized perfectly *including* the learnable prior distribution, then the bound becomes equal to the JS divergence. Note that this has a similar form to the adversarial approach except that it is a min problem and thus forms a min-min problem. A flow-based variant [Cho et al., 2022] provides an upper bound that only depends on optimizing the prior.

D.1.3 Non-Parametric Geometric Divergences

Geometric divergences (e.g., Wasserstein, Sinkhorn or MMD) vary with invertible transformations of the space. Intuitively, they depend on the distances in the space rather than ratios of densities as in information-theoretic divergences. One natural approach is to compute the distance between the domain distribution means. However, the means having a distance of zero is only necessary but not sufficient condition for the distributions to be equal. The maximum mean discrepancy (MMD) finds a function of random variables that maximizes the expectation between the domain distributions. While the function class could be a set of neural networks as in MMD-GAN Li et al. [2017], the most commonly used class of functions is a reproducing kernel hilbert space (RKHS) Gretton et al. [2012]. The MMD can be solved exactly when comparing empirical distributions, i.e., batches of samples from each domain. Thus, this empirical MMD can be

used as a plug-in estimator of the distribution-level MMD:

$$\mathrm{D}^2_{\mathrm{MMD}}(\theta) \approx \widehat{\mathrm{D}}^2_{\mathrm{MMD}}(\theta)^{:} = \|\hat{\mu}_1 - \hat{\mu}_2\|^2_{\mathcal{H}} \tag{8}$$

$$= \widehat{\mathbb{E}}[\mathcal{K}(\mathbf{z}_1, \mathbf{z}_1)] - 2\widehat{\mathbb{E}}[\mathcal{K}(\mathbf{z}_1, \mathbf{z}_2)] + \widehat{\mathbb{E}}[\mathcal{K}(\mathbf{z}_2, \mathbf{z}_2)]\,, \tag{9}$$

where $\mathcal{H}$ is an RKHS with kernel $\mathcal{K}$, $\hat{\mu}_1$ and $\hat{\mu}_2$ are the empirical (sample-based) means of domain 1 and 2 respectively in $\mathcal{H}$, and the expectations are based on unbiased sample averages [Gretton et al., 2012]. This can be seen as a generalization of comparing the empirical mean of the two distributions but using the implicit infinite dimensional space of a RKHS. One of the challenges is that this scales quadratically in the number of samples in the batch and thus cannot be computed for very large batches. Additionally, the performance can be sensitive to the kernel bandwidth parameter, which can be non-trivial to select in practice.

Another geometric divergence is based on Wasserstein distance. The Wasserstein-1 distance Panaretos and Zemel [2019] is defined optimal transport cost between the domain distributions using the the cost function $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$. The Wasserstein-1 between two empirical distributions (i.e., samples) can be computed by solving a linear program. Recently, linear program neural network layers have been proposed, which could be used to approximate it Mazouz et al. [2022]. However, solving a linear program for every batch of training samples is likely too expensive. In practice, an approximation to the Wasserstein distance based on an entropy-regularized optimal transport problem is often used. For this approximation, the Sinkhorn algorithm Cuturi [2013], which only requires matrix-vector multiplications, is often used since it has a complexity of $O(m^2 N_{\text{iter}})$ where m is the dimensionality and N_{iter} is the max number of Sinkhorn iterations. This approximation can be written as a regularized optimization problem Cuturi [2013]:

$$\hat{D}_{\mathrm{SINK}}(\theta) := \Big(\mathbb{E}_{\hat{\pi}_\lambda}\big[c(\mathbf{z}_1, \mathbf{z}_2)\big] \quad \text{s.t. } \hat{\pi}_\lambda := \arg\min_{\hat{\pi}\in\Pi} \mathbb{E}_{\hat{\pi}}\big[\|\mathbf{z}_1 - \mathbf{z}_2\|_2\big] + \lambda \mathsf{H}(\hat{\pi})\Big)^{\lambda\to 0} \approx D_{\mathrm{W}_1} \tag{10}$$

where $\hat{\pi}_\lambda := \arg\min_{\hat{\pi}\in\Pi} \mathbb{E}_{\hat{\pi}}[\|\mathbf{z}_1 - \mathbf{z}_2\|_2] + \lambda \mathrm{H}(\hat{\pi})$ and where $\hat{\pi}(\mathbf{z}_1, \mathbf{z}_2) \in \Pi$ is the empirical coupling distribution between samples from each domain, Π corresponds to the set of joint discrete probability distributions over $\mathbf{z}_1$ and $\mathbf{z}_2$ whose marginals are $p_\theta(\mathbf{z}|\mathrm{d}{=}1)$ and $p_\theta(\mathbf{z}|\mathrm{d}=2)$, respectively, and $\overset{\lambda\to 0}{\approx}$ means that it approaches the true Wasserstein-1 distance as λ goes to zero. Note that this has two approximations. First, it compares a batch samples from each domain rather than the population-level distributions. Second, if $\lambda > 0$, then it forms an approximation to the Wasserstein-1 distance. While the Sinkhorn algorithm improves the computational complexity significantly, the algorithm is still at least quadratic in the number of samples in the batch and thus, like MMD, is difficult to apply for a large number of samples.

Previously mentioned geometric divergences have some problem, first MMD suffers from flat geometry, which eventually result into vanishing gradient Feydy et al. [2019b]. Also, vanilla OT causes dimension collapse on source mapping due to $\hat{D}_{SINK}(z_1, z_1) \neq 0$ by entropic regularization, thus it will introduce bias solution. Therefore, Sinkhorn divergence addresses above problem by interpolating between MMD and Sinkhorn with additional auto correlation term to prevent bias.

$$\hat{D}_{SINKD}(\theta) \overset{\text{def.}}{=} \hat{D}_{SINK}(z_1, z_2) - \frac{1}{2}\hat{D}_{SINK}(z_1, z_1) - \frac{1}{2}\hat{D}_{SINK}(z_2, z_2) \tag{11}$$

$$\hat{D}_{W_1} \xleftarrow{\varepsilon\to 0} \hat{D}_{SINKD}(\theta) \xrightarrow{\varepsilon\to+\infty} \hat{D}^2_{MMD}(\theta) \tag{12}$$

E Computation Requirements

All experiments were run on a single node with 4× NVIDIA RTX A5000 GPUs (24 GiB each; 96 GiB total), NVIDIA driver 535.261.03, and CUDA 12.2. The node uses a 1-socket AMD EPYC 7352 24-Core Processor, 48 hardware threads, 128 MiB (8 instances) L3 cache, and 257G RAM, running Ubuntu 22.04 (Linux 5.15.0-152-generic)