TEST-TIME ADAPTATION USING TRANSPORTING FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation (TTA) considers a pretrained model on *source* domain and aims to adapt it to *target* domain without labels, where the *target* is subject to distribution-shift. Most existing TTA methods are based on entropy-minimization and feature alignment, where the former are classification-specific. While the latter can be extended to regression, they either naively align the entire feature space or use principal component analysis (PCA) to find dimensions with maximum variance and perform alignment along those dimensions. However, these maximum variance dimensions may not be those that are important for domain adaptation. A natural question then arises as to how we can identify features with maximal geometric displacements if they exhibit low variance. We use Optimal Transport and propose **T**ransporting Fl**ow** (TOW), a novel framework that actively optimizes for a k-dimensional subspace that incurs *maximal* displacement cost for feature alignment. TOW computes 2-Wasserstein distance on the pre-learned density of source features and the target activations. Through leveraging Normalizing Flows, it implicitly finds an optimal transport map that minimizes the transport cost during adaptation. Our proposed method can be extended to both continuous regression and classification tasks in standard settings with any loss function. We show that TOW outperforms baselines on four real-world datasets.

1 INTRODUCTION

Test-time adaptation (TTA) has emerged as a promising approach to restore pretrained model's performance on target domain without accessing target labels and source data in the adaptation process (Wang et al., 2021; Niu et al., 2023a; Lee et al., 2024; Adachi et al., 2025; Zhang et al., 2024; Jung et al., 2023; Zhang et al., 2022). TTA includes entropy minimization and feature alignment (FA) technique. The former are classification-specific and cannot be readily applied to scalar valued regression tasks (Liang et al., 2025). In supervised machine learning, one assumes that test or *target* data comes from the same process that generated the training or *source* data (Ben-David et al., 2006; Zhao et al., 2023). However, this assumption is often violated in reality due to *distribution*-shift. Factors like varying atmospheric pressure and lighting conditions can introduce a *discrepancy* in the target distribution that were not present in the source, which may degrade model performance (Liang et al., 2025).

Despite the growing adoption of prediction models in scientific applications (*e.g.,* predicting oxide concentration from Laser Induced Breakdown Spectroscopy (LIBS) spectrum; Noll (2011)), TTA for regression remains understudied. TTA methods for regression mainly employ FA techniques to align feature statistics of target with the pre-computed source, assuming this will reduce the *distribution* gap and improve performance on target domain (Eastwood et al., 2022; Nejjar et al., 2023). However, indiscriminate FA across all dimensions can result in sub-optimal adaptation as all features do not contribute equally, which could alter the input-target relationship learned by the pretrained model.

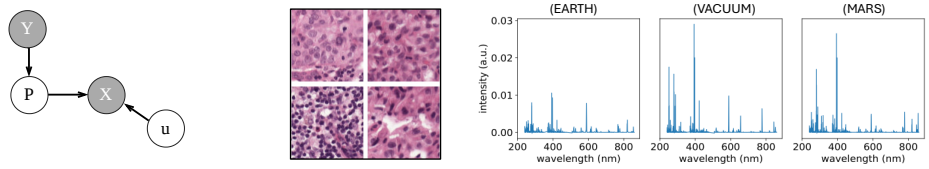
While SSA (Adachi et al., 2025) addresses this by performing FA in k-dimensional subspace based on principal component analysis (PCA), it assumes a Gaussian distribution on the source and target features. However, real-world data (e.g. LIBS spectra) is often arbitrarily distributed with multi-modal and skewed shapes. Therefore, imposing a restrictive unimodal Gaussian can destroy the information relevant for prediction. Moreover, Adachi et al. (2025) performs FA based on maximal principal components (PC), which can lead to sub-optimal outcomes if any of the selected PCs are associated with spurious correlations that do not cause the observed distribution shift, and other artifacts.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We ask the following question, *how can we identify features with maximal geometric displacements if they exhibit low variance?* For intuition, consider a multi-target setting where features associated with each target variable may not exhibit high variance under distribution shift, and therefore PCA-based subspace FA can harm TTA. Instead we need a principled way to identify the features that are most affected under distribution shift. This motivates us to consider Wasserstein metric – a geometric measure of dissimilarity between probability distributions. To this end, Optimal Transport (OT) theory naturally induces Wasserstein distance and has therefore recieved great interest in domain adaptation literature (Kerdoncuff et al., 2020; Khamis et al., 2024; Koç et al., 2025; Zhuang et al., 2024). Prior works including Courty et al. (2017) JDOT, Damodaran et al. (2018) DeepJDOT, Lee et al. (2019) SRW use OT to align joint distribution (features/label) of the target with the source; however, they are unsupervised domain adaptation methods where one has access to the target at the pre-training stage and thus cannot be applied to TTA. We apply OT to TTA settings using Normalizing Flows (NFs) and propose **T**ransporting Fl**ow** (TOW), a novel framework that actively seeks k-dimensional subspace where 2-Wasserstein distance between target and source features is maximal and *only* performs FA within that subspace. We leverage NFs for two tasks: 1) approximating the density of source features extracted from a pretrained model; 2) learn the optimal bijective function that maps target features to the pre-learned source density (detailed in Section 3). Our proposed method can be extended to both continuous regression and classification tasks. We conduct extensive experiments on four real-world datasets, where TOW consistently outperforms baselines with significant gains.

Our main contributions are:

- Develop an OT-TTA method that optimizes for the most discriminating k-dimensional subspace between target and source features. It avoids computational complexity of discrete OT solvers by leveraging Normalizing Flows and thus scales well to large datasets (*e.g.,* Camelyon17) while giving competitive results.
- We approximate the density of source features by training NFs on activations extracted from a pre-trained model, which we refer to as the **s**ource **d**ensity **e**stimator (SDE). During adaptation, we freeze the weights of the SDE. Then, TOW maps target features directly on the approximated source density while making no assumptions on the features distribution.
- We conduct extensive experiments on four datasets with tasks including both continuous regression and classification. TOW demonstrates the effectiveness of OT based objective function for FA through consistently outperforming baselines.



(a)`Distribution shift` (b)`Camelyon17` (c)`LIBS`

Figure 1: (a) Distribution shift induced by u in assumed anti-causal setting. Observed factors are shaded, latent factors are clear. (b) Tumor tissues with varying staining protocols. (c) LIBS spectral data measured with identical Y under different atmospheric conditions u.

2 Preliminaries

We consider a regression task with a predictor $f_\theta^s : \mathbf{X}_s \to \mathbb{R}$ trained on labeled *source* data $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{N^s}$ to predict a continuous target variable. In domain adaptation settings, the predictor is generally decomposed as $f_\theta^s = h \circ g_\theta^s$, where g_θ^s is the feature extractor and h a regressor head, then $\mathbf{z}_i^s = g_\theta^s(\mathbf{x}_i^s) \in \mathbb{R}^d$ are the learned feature activations. At test time, we observe data from the *target* domain under *distribution*-shift. Our goal is to adapt a pretrained feature extractor g_θ^s on $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{N^t}$ without accessing *target* labels and $\mathcal{D}_s$ during adaptation phase.

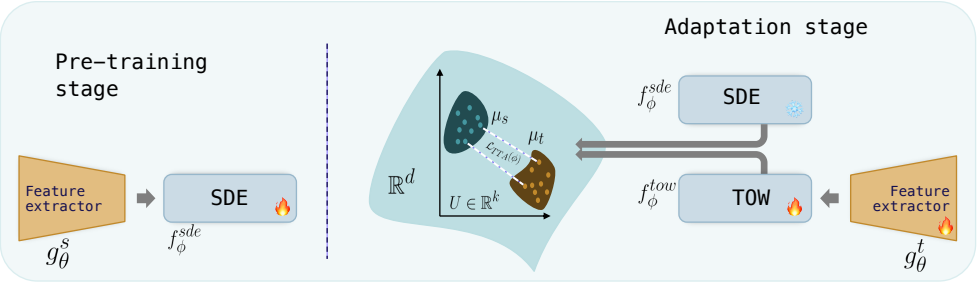


Figure 2: Schematic illustration of the Transporting Flow framework. Pre-training stage involves training the standard model, and SDE on source feature space. At adaptation stage, the pre-trained feature extractor is updated on target through a loss function designed for optimizing TOW.

Distribution-shift. In this work, we consider data generated from the structure $Y \to P \to X \leftarrow u$, where P is a latent state that carries information relevant to predict Y, X is an observed input and u a nuisance variable (Figure 1a). For instance, in the LIBS process u could be instrumental variables or environmental factors *e.g.,* atmospheric pressure, that causes variation in X as shown in Figure 1c. Another example is tumor tissue classification in pathological imaging, where staining protocols and equipment differences may induce distribution shift (Figure 1b). We assume $X = f(P, u)$ with $P \perp u$ and there exists a function g such that $P = g(X)$. We study distribution shift induced by u that introduces domain-specific variations in X, then $u_s \neq u_t$ and $p(X_s) \neq p(X_t)$. Under these assumptions, we aim to recover $p(g_\theta^s(X_s)) \approx p(g_\theta^t(X_t))$. This setting is similar to Eastwood et al. (2022), however, they tend to align entire feature space whereas our proposed approach optimizes for a subspace FA. We borrow taxonomy from Eastwood et al. (2022) and refer to the studied distribution shift as measurement-shift.

3 Proposed Method

In this section, we describe our proposed framework which is illustrated in Figure 2.

Notation. Let $\Omega \subseteq \mathbb{R}^d$ be a measurable space with Borel σ-algebra and $\mathcal{P}(\Omega)$ denote the set of Borel probability measures on Ω. We define $\mu_s, \mu_t \in \mathcal{P}(\Omega)$ as probability measures over the feature space of source $\mathbf{Z}_s = g_\theta^s(\mathbf{X}_s)$ and target $\mathbf{Z}_t = g_\theta^s(\mathbf{X}_t)$ domain, respectively. For $k \in [d]$, we consider the Grassmannian of k-dimensional subspaces U on $\mathbb{R}^d$.

Domain adaptation as transport problem. The Monge problem (Cayley, 1882) seeks to find a map $\mathcal{T}_0 : \Omega \to \Omega$ for moving μ_s towards μ_t as:

$$\mathcal{T}_0 = \inf_{\mathcal{T}_{\#}\mu_s = \mu_t} \int_\Omega \mathrm{d}(\mathbf{z}, \mathcal{T}(\mathbf{z}))\mathrm{d}\mu_s(\mathbf{z}) \tag{1}$$

where $\mathcal{T}_{\#}\mu_s$ is the pushforward operator of μ_s under $\mathcal{T}$ and d denotes a metric. If solution to $\mathcal{T}_0$ exists, it is called optimal transport map; however, in many settings it does not exists (*e.g.,* consider moving one Dirac from μ_s towards μ_t with two Dirac measures). A relaxation to this problem was proposed by Kantorovich (2006) as:

$$\gamma^* = \underset{\gamma \in \Pi(\mu_s, \mu_t)}{\arg\min} \int_{\Omega \times \Omega} d(\mathbf{z}_s, \mathbf{z}_t)\mathrm{d}\gamma(\mathbf{z}_s, \mathbf{z}_t) \tag{2}$$

where $\gamma(\mu_s, \mu_t)$ defines a set of couplings over the joint probability measures $\gamma \in \mathcal{P}(\Omega \times \Omega)$ between the marginal distributions μ_s and μ_t. The optimal solutions to this problem are called transport plans. The Kantorovich relaxation can be seen as a generalization of the Monge problem with relaxed assumptions on the existence and uniqueness of the solution Visentin and Cheridito (2025).

3.1 USING DISPLACEMENT COST IN SUBSPACE OT TO DETECT MAXIMALLY DISPLACED FEATURES UNDER DISTRIBUTION-SHIFT

The Wasserstein distance leverages the underlying geometric properties of the probability distributions and remains well-defined even when two distributions do not overlap, making it an appealing discrepancy measure (Arjovsky et al., 2017). These advantages of Wasserstein distance motivate us to extend the concept of OT on k-dimensional subspaces for the FA task in TTA settings. We utilize 2-Wasserstein distance to measure the discrepancy between source and target features. Computing $\mathcal{S}_k(\mu_s, \mu_t)$ as in 3 requires solving OT plan γ^*, a linear programming problem which is inefficient for continuous distributions. Moreover, we found reg-OT solvers proposed in Paty and Cuturi (2019) to be prohibitively expensive. Following Paty and Cuturi (2019), we define:

Definition 1. *For $k \in [d]$, the k-dimensional subspace 2-Wasserstein distance between μ_s and μ_t as*:

$$\mathcal{S}_k(\mu_s, \mu_t) = \inf_{\gamma \in \Pi(\mu_s, \mu_t)} \sup_{U \in \mathcal{G}_k} \left[\int ||P_U(\mathbf{z}_s - \mathbf{z}_t)||^2 d\gamma(\mathbf{z}_s, \mathbf{z}_t) \right]^{1/2} \tag{3}$$

Eq. 3 can be reformulated as "min-max" problem by taking the maximal sum of k-largest eigenvalues of the second-order moment matrix of displacement costs given a fixed γ (see Lemma 2 in Paty and Cuturi (2019)). Intuitively, feature dimensions with maximum displacement between domains will be most discriminative for the adaptation task. From Paty and Cuturi (2019), 3 can be rewritten as:

$$S_k^2(\mu_s, \mu_t) = \min_{\gamma \in \Pi(\mu_s, \mu_t)} \sum_{l=1}^{k} \lambda_l(M_\gamma) \tag{4}$$

where, M_γ is the second-order moment matrix of displacement costs defined over the entire feature space.

Approximating OT map via Normalizing Flows. Recently, Normalizing Flows (NF) have become a natural choice for approximating bijective pushforward maps in OT theory (Visentin and Cheridito, 2025), given that they are inherently invertible functions. In our proposed framework, we leverage NFs to approximate OT map with parameterized invertible transformations $f_{\phi^*} \approx T_0$ and therefore we call it Transporting Flow. Following, Visentin and Cheridito (2025), we use affine coupling layers to learn a bijective mapping from target features to source; however, we do not explicitly define a base distribution. Instead we compose the coupling layers with a pre-learned NF on source, creating a direct mapping to source feature space without imposing any restriction on the features distribution. Following Brenier's theorem,

Lemma 2.1. *Let $\mu_s, \mu_t \in \mathcal{P}(\mathbb{R}^d)$. Then, the Kantorovich problem 2 under quadratic cost admits a unique solution $\gamma \in \Pi(\mu_s, \mu_t)$ and there exists $\gamma = (id, \mathcal{T})_{\#}\mu_s$, where $\mathcal{T} \in B(\mu_s, \mu_t)$ is μ_s-almost unique everywhere that satisfies the Monge's map 1.*

We define our objective function by rewriting 4 as:

$$\mathcal{L}_{TTA}(\phi) = \text{argmin}_{\phi \in \Phi} \int U_k ||\mathbf{z} - f_\phi(\mathbf{z})||^2 d\mu_t(\mathbf{z}) \tag{5}$$

where, $(f_{\phi^*})_{\#}\mu_t = \mu_s$. Optimizing f_ϕ will implicitly result in learning optimal transport map. In 4, the transport plan γ and the top-k eigenspace are optimized alternatively, with the eigenspace obtained via spectral decomposition of M_γ. However, in 5, when optimizing (implicitly) over U_k in an end-to-end manner, the eigendecomposition part becomes numerically instable due to gradient flow through the decomposition operation. We address this by explicitly optimizing U_k on the Stiefel manifold with Riemannian updates.

3.2 TEST-TIME ADAPTATION USING TRANSPORTING FLOW

Since we are interested in TTA, where one does not have access to source data, we need an approximation of μ_s to learn a transformation $\mathcal{T}$ between μ_s and μ_t. For this purpose, we use NFs to estimate the density of the low-dimensional source features, which are extracted from g_θ^s after training f_θ^s.

Note that we do not use source input samples; instead we estimate density of pre-learned feature activations $\mathbf{Z}_s$. One can simply save the weights of learned density estimator and use it at adaptation time. We refer to it as the **s**ource **d**ensity **e**stimator (SDE) and denote by f_ϕ^{sde}.

During the adaptation phase, given a mini-batch of target samples, we extract $\mathbf{z}_i^t$ from g_θ^s pretrained feature extractor and load f_ϕ^{sde} SDE; however, we freeze its parameters. Next, we initialize f_ϕ^{tow} TOW and obtain $\mathbf{z}_i^{tow} = f_\phi^{tow}(\mathbf{z}_i^t)$, then we take $\mathbf{z}_i^{tow}$ and compute its density under pre-learned source feature space as $\mathbf{z}_i^{sde} = f_\phi^{sde}(\mathbf{z}_i^{tow})$. We use $\mathbf{z}_i^t$ and $\mathbf{z}_i^{tow} = f_\phi^{tow}(\mathbf{z}_i^t)$ to compute $\mathcal{L}_{TTA}$ and learn a feature extractor g_θ^t on target by updating either all layers or the affine parameters of BatchNorm layers in g_θ^s. We define the procedure of TOW in Algorithm 1.

Algorithm 1 Transporting Flow (TOW).

Require: Target dataset $\mathcal{D}_t = \{x_i^t\}_{i=1}^N$, and pre-learned source density estimator f_ϕ^{sde}
  for $t = 1, \ldots, N$ **do**
    Extract $\mathbf{z}_i^t = g_\theta^s(\mathbf{x}_i^t)$
    Compute $\mathbf{z}_i^{tow} = (f_\phi^{sde} \circ f_\phi^{tow})(\mathbf{z}_i^t)$
    Compute $\mathcal{L}_{TTA}(\mathbf{z}_i^t, \mathbf{z}_i^{tow})$
    Update affine parameters of BatchNorm layers in g_θ^s to learn g_θ^t on $\mathcal{D}_t$
  end for

4 Experiments

In this section, we provide empirical results on four datasets from diverse applications including continuous regression and classification tasks. Notably, one of the regression tasks involves a real-world scientific application; predicting chemical oxide concentrations from LIBS spectrum.

4.1 Datasets

LIBS. We used LIBS spectral data from Dyar et al. (2024) measured under different conditions induced by u that simulate EARTH, VACUUM and MARS-like environments. The task is to predict continuous values of chemical oxide concentration from a given spectrum. Following Kontolati et al. (2021), we train a univariate model for each oxide independently. We used EARTH as source, and VACUUM and MARS as target domains. LIBS is inherently anti-causal in its data-generating process. Further details are provided in Appendix C.1.

Camelyon17. We used Camelyon17 data from Bandi et al. (2018). It is a large-scale pathological imaging dataset from five different hospitals. The distribution-shift arises through different tissue staining techniques and equipment difference across hospitals. The task involves classifying the image with tumor tissue ($Y = 1$) or normal tissue ($Y = 0$). We used $h3$ as source domain and $h0, h1, h2$, and $h4$ as target domains. This problem has anti-causal structure, with tumor Y causing specific patterns in the observed scan of a tissue.

Birdcalls. We used Birdcalls data from Gao et al. (2023). It records audio clips of birds singing to identify their species. It contains Mel spectograms of each clip to enable classification task. Distribution-shift arises from variation in gain settings of audio recording devices, habitat acoustics (background sounds) and bird population density. Following Liu et al. (2025), we use test set with 724 audio clips.

California housing. We used California housing data from Nugent (2017). It is a tabular dataset and the task is to predict housing prices based on demographic covariates. Following Adachi et al. (2025), we split the data with Near-Bay as source domain and rest as a single target domain.

4.2 Results

For all experiments, we report mean and 1 standard deviation over five random runs.

Baselines. Source-only, AdaBN (Benz et al., 2021), SSA (Adachi et al., 2025), FR (Eastwood et al., 2022), RSD (Chen et al., 2021), TACT (Liu et al., 2025). PL (Lee et al., 2013), SHOT(Liang et al.,

2020) , TSD (Wang et al., 2023), Tent (Upchurch et al., 2017), SAR (Niu et al., 2023b). Note that RSD is not a true TTA method as it requires access to source data during adaptation. Some of the baselines are classification-specific and therefore we cannot extend those to scalar valued regression tasks. We provide further details on baselines in Appendix C.3.

Pretrained-source model. For LIBS and California housing, we trained MLP using standard MSE loss function, with $\mathbf{z}_i \in \mathbb{R}^{100}$. We trained ResNet18 with $\mathbf{z}_i \in \mathbb{R}^{512}$ on Camelyon17 and Birdcalls datasets. We provide training details in Appendix C.2.

Test-time adaptation with TOW (ours). We set $k = 25$ as the default subspace dimension for LIBS and California housing datasets. For Camelyon17 and Birdcalls, we tried k in $\{50, 55, 60, \ldots, 85, 90\}$ and achieved best results within $\{50, 55\}$ for Camelyon17 and Birdcalls with $k = 77$. For LIBS, California housing and Birdcalls only updating the affine parameters of BN layers obtained optimal results. For Camelyon17, we updated all layers in the feature extractor.

We evaluate regression tasks (LIBS, California housing) in terms of RMSE and R^2, report Accuracy for Camelyon17 and F1-recall score for Birdcalls.

Table 1: MARS RMSE ↓ values across 8 target variables. Best values are **Bolded**.

Method	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
Source-only	$16.40_{\pm 8.71}$	$1.13_{\pm 0.15}$	$8.19_{\pm 1.92}$	$5.48_{\pm 1.30}$	$2.72_{\pm 0.61}$	$3.24_{\pm 2.36}$	$1.17_{\pm 0.06}$	$2.47_{\pm 0.38}$
AdaBN	$7.14_{\pm 0.71}$	$0.67_{\pm 0.10}$	$4.13_{\pm 0.68}$	$3.90_{\pm 1.73}$	$2.93_{\pm 0.44}$	$2.68_{\pm 0.67}$	$1.12_{\pm 0.11}$	$1.51_{\pm 0.21}$
RSD	$7.55_{\pm 0.81}$	$0.94_{\pm 0.44}$	$4.54_{\pm 0.66}$	$3.94_{\pm 2.13}$	$3.03_{\pm 0.74}$	$2.65_{\pm 1.13}$	$1.20_{\pm 0.13}$	$1.86_{\pm 0.24}$
SSA	$13.83_{\pm 5.47}$	$1.03_{\pm 0.08}$	$3.78_{\pm 0.27}$	$4.46_{\pm 0.41}$	$6.38_{\pm 0.67}$	$4.11_{\pm 0.87}$	$1.21_{\pm 0.03}$	$2.17_{\pm 0.04}$
FR	$8.27_{\pm 0.97}$	$0.69_{\pm 0.08}$	$3.90_{\pm 0.54}$	$3.30_{\pm 0.74}$	$3.74_{\pm 0.86}$	$2.45_{\pm 0.28}$	$1.02_{\pm 0.11}$	$1.44_{\pm 0.20}$
TOW	$\mathbf{6.06_{\pm 0.16}}$	$\mathbf{0.59_{\pm 0.04}}$	$\mathbf{3.28_{\pm 0.47}}$	$\mathbf{3.28_{\pm 0.72}}$	$\mathbf{2.56_{\pm 0.53}}$	$\mathbf{1.97_{\pm 0.36}}$	$\mathbf{0.92_{\pm 0.04}}$	$\mathbf{1.21_{\pm 0.04}}$
Oracle	$4.39_{\pm 0.20}$	$0.36_{\pm 0.02}$	$1.86_{\pm 0.10}$	$1.91_{\pm 0.07}$	$1.68_{\pm 0.09}$	$1.27_{\pm 0.12}$	$0.59_{\pm 0.02}$	$0.74_{\pm 0.02}$

Table 2: VACUUM RMSE ↓ values across 8 target variables. Best values are **Bolded**.

Method	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
Source-only	$12.95_{\pm 2.21}$	$0.98_{\pm 0.08}$	$6.80_{\pm 1.24}$	$5.27_{\pm 1.03}$	$4.47_{\pm 1.33}$	$5.72_{\pm 0.24}$	$1.68_{\pm 0.39}$	$2.47_{\pm 0.16}$
AdaBN	$8.07_{\pm 1.05}$	$0.81_{\pm 0.10}$	$3.19_{\pm 0.23}$	$3.35_{\pm 0.80}$	$3.06_{\pm 0.71}$	$2.48_{\pm 0.34}$	$1.02_{\pm 0.11}$	$1.56_{\pm 0.07}$
RSD	$9.44_{\pm 1.82}$	$0.95_{\pm 0.10}$	$3.65_{\pm 0.30}$	$3.69_{\pm 0.81}$	$3.43_{\pm 0.61}$	$3.01_{\pm 0.67}$	$1.19_{\pm 0.20}$	$1.82_{\pm 0.31}$
SSA	$12.55_{\pm 1.11}$	$1.17_{\pm 0.12}$	$4.09_{\pm 0.18}$	$4.94_{\pm 0.36}$	$6.52_{\pm 0.15}$	$5.19_{\pm 0.59}$	$1.34_{\pm 0.07}$	$2.36_{\pm 0.10}$
FR	$8.45_{\pm 0.73}$	$0.80_{\pm 0.06}$	$3.29_{\pm 0.15}$	$\mathbf{3.15_{\pm 0.22}}$	$4.18_{\pm 0.94}$	$2.95_{\pm 0.28}$	$1.10_{\pm 0.10}$	$2.01_{\pm 0.15}$
TOW	$\mathbf{7.22_{\pm 0.52}}$	$\mathbf{0.74_{\pm 0.03}}$	$\mathbf{2.82_{\pm 0.24}}$	$3.21_{\pm 0.75}$	$\mathbf{3.01_{\pm 0.70}}$	$\mathbf{2.46_{\pm 0.34}}$	$\mathbf{0.88_{\pm 0.05}}$	$\mathbf{1.38_{\pm 0.05}}$
Oracle	$4.22_{\pm 0.10}$	$0.51_{\pm 0.03}$	$1.75_{\pm 0.02}$	$1.85_{\pm 0.04}$	$1.72_{\pm 0.07}$	$1.58_{\pm 0.03}$	$0.53_{\pm 0.02}$	$0.82_{\pm 0.04}$

For LIBS data, Tables 1, 2 report RMSE values for the MARS and VACUUM target domain, respectively. Corresponding R^2 results are provided in Tables 7, 8 (see Appendix A.1). TOW outperforms baselines on all oxides except for one in the case of VACUUM, where FR has better results. Notably, TOW achieves significant gains in both the mean and standard deviation on both metrics. In our experiments, SSA consistently performs poorly and since FR aligns the entire feature space – it could potentially lead to sub-optimal adaptation. Whereas TOW leverages 2-Wasserstein distance on source and target feature space to find feature dimensions that get maximally displaced under measurement-shift. We believe TOW preserves geometric structure of feature space during adaptation through optimal transport geometric properties. We constructed semi-synthetic target domains using source EARTH to stress-test TOW's robustness to spurious correlations, with results provided in Appendix B.2. **On California housing**, TOW consistently gives better results compared to FR, SSA and AdaBN, which are strong baselines for TTA in regression tasks (see Table 4).

Table 3: Camelon17 Accuracy ↑. Best values are **Bolded**.

Method	h0	h1	h2	h4	Avg
Source-only	$73.25_{\pm 3.10}$	$66.62_{\pm 2.69}$	$32.55_{\pm 2.29}$	$47.48_{\pm 2.94}$	54.97
AdaBN	$90.04_{\pm 0.76}$	$84.53_{\pm 0.76}$	$75.85_{\pm 1.55}$	$82.75_{\pm 1.57}$	83.29
PL	$95.16_{\pm 0.14}$	$82.64_{\pm 0.26}$	$89.47_{\pm 2.93}$	$96.28_{\pm 0.63}$	90.88
SHOT	$\mathbf{95.73}_{\pm 0.25}$	$78.29_{\pm 0.43}$	$88.70_{\pm 0.28}$	$\mathbf{97.12}_{\pm 0.16}$	89.96
FR	$94.89_{\pm 0.44}$	$80.14_{\pm 2.43}$	$88.53_{\pm 0.59}$	$96.06_{\pm 0.45}$	89.91
TOW	$92.74_{\pm 0.49}$	$\mathbf{85.05}_{\pm 0.42}$	$\mathbf{90.31}_{\pm 0.57}$	$85.74_{\pm 4.87}$	88.46

Table 4: California results.

Method	R2 ↑	RMSE ↓
Source-only	$0.412_{\pm 0.05}$	$0.766_{\pm 0.03}$
AdaBN	$0.537_{\pm 0.01}$	$0.680_{\pm 0.01}$
RSD	$0.533_{\pm 0.02}$	$0.683_{\pm 0.02}$
SSA	$0.509_{\pm 0.02}$	$0.701_{\pm 0.02}$
FR	$0.525_{\pm 0.02}$	$0.689_{\pm 0.01}$
TOW	$\mathbf{0.547}_{\pm 0.01}$	$\mathbf{0.673}_{\pm 0.01}$
Oracle	$0.789_{\pm 0.00}$	$0.460_{\pm 0.00}$

On **Camelyon17** classification task Table 3 reports accuracy, we can see TOW convincingly achieves higher accuracy on two hospital indices $(h1, h2)$ and gives significantly better average accuracy than AdaBN. While optimizing over k-dimensional subspace may further improve results on $h0$ and $h4$, these results demonstrate that TOW scales effectively to large datasets and generalizes across both continuous regression and classification tasks. TOW also achieves superior F1-recall score on **Birdcalls**, with results provided in Appendix A.2.

5 Ablative Study

5.1 Effect of subspace dimension

In this section, we investigate the performance of TOW with different values for subspace dimension. Since LIBS models are trained using a shallow MLP with $\mathbf{z}_i \in \mathbb{R}^{100}$, we selected $k \in \{5, 10, 15, 20, 25\}$. Table 5 shows RMSE values for MARS, with results improving for all target variables as the value of k increases, though we see diminishing returns after $k = 25$. We also found that the performance depends on the choice of learning rate (LR) used for updating pre-trained feature extractor's g_θ^t parameters and optimizing the OT map in TOW. We conducted extensive experiments on the sensitivity to both LRs with $k \in \{5, 15, 19, 22, 25\}$ on MARS and VACCUM. See Appendix B.1 for detailed results.

Table 5: MARS RMSE ↓ results across 8 target variables.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	$7.14_{\pm 0.56}$	$0.67_{\pm 0.11}$	$3.98_{\pm 0.69}$	$3.83_{\pm 1.66}$	$3.47_{\pm 0.55}$	$2.61_{\pm 0.60}$	$1.06_{\pm 0.11}$	$1.32_{\pm 0.16}$
k10	$7.10_{\pm 0.50}$	$0.66_{\pm 0.11}$	$3.97_{\pm 0.64}$	$3.85_{\pm 1.75}$	$3.51_{\pm 0.53}$	$2.59_{\pm 0.60}$	$1.04_{\pm 0.10}$	$1.29_{\pm 0.14}$
k15	$6.92_{\pm 0.57}$	$0.68_{\pm 0.13}$	$3.93_{\pm 0.67}$	$3.76_{\pm 1.60}$	$3.43_{\pm 0.55}$	$2.58_{\pm 0.59}$	$1.04_{\pm 0.10}$	$1.28_{\pm 0.13}$
k20	$6.92_{\pm 0.61}$	$0.67_{\pm 0.11}$	$3.93_{\pm 0.64}$	$3.84_{\pm 1.84}$	$3.42_{\pm 0.55}$	$2.60_{\pm 0.64}$	$1.03_{\pm 0.11}$	$1.28_{\pm 0.14}$
k25	$6.91_{\pm 0.54}$	$0.64_{\pm 0.11}$	$3.92_{\pm 0.66}$	$3.65_{\pm 1.46}$	$3.39_{\pm 0.53}$	$2.63_{\pm 0.69}$	$1.02_{\pm 0.10}$	$1.27_{\pm 0.14}$

Table 6: Camelyon17 Accuracy ↑ resutls.

	h0	h1	h2	h4
k50	$91.77_{\pm 0.95}$	$85.05_{\pm 0.42}$	$90.21_{\pm 0.53}$	$84.62_{\pm 7.81}$
k55	$92.37_{\pm 0.77}$	$84.56_{\pm 0.83}$	$90.31_{\pm 0.57}$	$85.64_{\pm 4.85}$
k90	$92.74_{\pm 0.49}$	$84.43_{\pm 0.82}$	$74.97_{\pm 3.34}$	$77.81_{\pm 3.55}$

We performed similar experiments on **Camelyon17** with $k \in [50, 90]$, but only report $k \in \{50, 55, 90\}$ due to space constraints. Note that, in Table 6, TOW with only $k = 50$ and $k = 55$ achieves higher accuracy on indices $h1$ and $h2$ compared to other baselines (in Table 3). This suggests that only *fewer* task-relevant dimensions get maximally displaced across domains under measurement-shift and FA along such dimensions can substantially improve pretrained model's performance on *target*. We provide results on LR sensitivity in Appendix B.1.2.

6 Related work

Unsupervised Domain Adaptation (DA). Conventionally, unsupervised DA has been studied to improve model performance on a target domain. Following Ben-David et al. (2006), many UDA methods are developed to minimize the discrepancy across domains in feature space (Ganin et al., 2016; Nguyen et al., 2022; Tzeng et al., 2017; Shu et al., 2018). In regression settings, Chen et al. (2021) and Nejjar et al. (2023) show that naive feature alignment can alter feature scale and impede domain adaptation. To address this issue, Chen et al. (2021) proposes RSD to align source and target feature subspaces to reduce the distribution gap without explicitly altering feature scale. Despite success, the application of these approaches is hindered as they require target data during the training process.

Test-time Adaptation. TTA setting does not require access to source data and target labels at the adaptation phase. TTA methods work with off-the-shelf trained models in standard settings. We categorize TTA approaches into three types based on their adaptation objectives:

Entropy minimization (Entropic-TTA). Wang et al. (2021) (Tent) formulates entropy as an objective function to improve accuracy on the target domain. Following Wang et al. (2021) Tent, Zhou and Levine (2021) BACS, Zhang et al. (2022) MEMO, Niu et al. (2023a) SAR and Niu et al. (2022) EATA modifies entropy objective with different strategies to improve robustness and adaptation performance. Zhang et al. (2024) COME regularizes the entropy objective to conservatively update model parameters to mitigate overconfident predictions during adaptation. These are classification-specific methods and cannot be applied to scalar valued regression tasks. Moreover, they heavily rely on pseudo-labeling and augmentations based strategies. In contrast, our proposed approach works for both tasks with standard pretrained models on source domain.

Feature alignment (FA-TTA). Feature alignment adapts the target domain with source by *aligning* their feature statistics. Since source data is not accessible at test time, cheap source statistics are precomputed using a pretrained model and used during adaptation. Benz et al. (2021) AdaBN updates the running statistics of BatchNorm layer on the target domain and results in strong performance. Recent works including Eastwood et al. (2022) BUFR, Adachi et al. (2023) CAFe, and Jung et al. (2023) CAFA explore aligning distributions in low-dimensional feature space. Adachi et al. (2025) builds SSA on the idea that feature space in regression tasks is diverse, only few dimensions contribute to final prediction. SSA assumes features are Gaussian distributed and selects eigenspace with maximum variance. Our approach uses Wasserstein distance to measure discrepancy between source and target, and makes no assumption on the features distribution.

7 Conclusion

We propose Optimal Transport based TTA method that defines Wasserstein distance on source and target feature space to find a subspace exhibiting maximum domain discrepancy. Our proposed method is a task-agnostic method, therefore it can be used with any loss function in standard settings. We conduct experiments on four real-world datasets with tasks including both continuous regression and classification, where it outperforms baselines with significant gains. We show it scales well to large datasets as demonstrated by our experiments on Camelyon17.

AI use statement. We used Claude AI for grammar corrections, which were subsequently reviewed by authors.

Reproducibility statement. We provide details on the datasets and experimental settings in Section 4.1, 4.2 and Appendix C. We plan to release code upon acceptance.

REFERENCES

Kazuki Adachi, Shin'Ya Yamaguchi, and Atsutoshi Kumagai. Covariance-aware feature alignment with pre-computed source statistics for test-time adaptation to multiple image corruptions. In *2023 IEEE International Conference on Image Processing (ICIP)*, page 800–804. IEEE, October 2023. doi: 10.1109/icip49359.2023.10222901. URL http://dx.doi.org/10.1109/ICIP49359.2023.10222901.

Kazuki Adachi, Shin'ya Yamaguchi, Atsutoshi Kumagai, and Tomoki Hamagami. Test-time adaptation for regression by subspace alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SXtl7NRyE5.

Ryan B Anderson, Olivier Forni, Agnes Cousin, Roger C Wiens, Samuel M Clegg, Jens Frydenvang, Travis SJ Gabriel, Ann Ollila, Susanne Schröder, Olivier Beyssac, et al. Post-landing major element quantification using supercam laser induced breakdown spectroscopy. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 188:106347, 2022.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. Pmlr, 2017.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

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

Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness, 2021. URL https://arxiv.org/abs/2010.03630.

Cayley. On monge's "mémoire sur la théorie des déblais et des remblais.". *Proceedings of the London Mathematical Society*, 1(1):139–143, 1882.

Xinyang Chen, Sinan Wang, Jianmin Wang, and Mingsheng Long. Representation subspace distance for domain adaptation regression. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1749–1759. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/chen21u.html.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation, 2017. URL https://arxiv.org/abs/1705.08848.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation, 2018. URL https://arxiv.org/abs/1803.10081.

MD Dyar, CR Ytsma, and K Lepore. Geochemistry by laser-induced breakdown spectroscopy on the moon: Accuracy, detection limits, and realistic constraints on interpretations. *Earth and Space Science*, 11(10):e2024EA003635, 2024.

Cian Eastwood, Ian Mason, Christopher K. I. Williams, and Bernhard Schölkopf. Source-free adaptation to measurement shift via bottom-up feature restoration, 2022. URL https://arxiv.org/abs/2107.05446.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016. URL https://arxiv.org/abs/1505.07818.

Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-domain robustness via targeted augmentations. In *International conference on machine learning*, pages 10800–10834. PMLR, 2023.

Sanghun Jung, Jungsoo Lee, Nanhee Kim, Amirreza Shaban, Byron Boots, and Jaegul Choo. Cafa: Class-aware feature alignment for test-time adaptation, 2023. URL https://arxiv.org/abs/2206.00205.

Lev Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133:1381–1382, 2006. URL https://api.semanticscholar.org/CorpusID:122853046.

Tanguy Kerdoncuff, Rémi Emonet, and Marc Sebban. Metric learning in optimal transport for domain adaptation. In *International joint conference on artificial intelligence*, pages 2162–2168. IJCAI, 2020.

Abdelwahed Khamis, Russell Tsuchida, Mohamed Tarek, Vivien Rolland, and Lars Petersson. Scalable optimal transport methods in machine learning: A contemporary survey. *IEEE transactions on pattern analysis and machine intelligence*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Okan Koç, Alexander Soen, Chao-Kai Chiang, and Masashi Sugiyama. Domain adaptation and entanglement: an optimal transport perspective. *arXiv preprint arXiv:2503.08155*, 2025.

Katiana Kontolati, Natalie Klein, Nishant Panda, and Diane Oyen. Neural density estimation and uncertainty quantification for laser induced breakdown spectroscopy spectra. *arXiv preprint arXiv:2108.08709*, 2021.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation, 2019. URL https://arxiv.org/abs/1903.04064.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors, 2024. URL https://arxiv.org/abs/2403.07366.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025.

Yingnan Liu, Rui Qiao, Mong Li Lee, and Wynne Hsu. Test-time adaptation by causal trimming, 2025. URL https://arxiv.org/abs/2510.11133.

Ismail Nejjar, Qin Wang, and Olga Fink. Dare-gram : Unsupervised domain adaptation regression by aligning inverse gram matrices, 2023. URL https://arxiv.org/abs/2303.13325.

A. Tuan Nguyen, Toan Tran, Yarin Gal, Philip H. S. Torr, and Atılım Güneş Baydin. Kl guided domain adaptation, 2022. URL https://arxiv.org/abs/2106.07780.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world, 2023a. URL https://arxiv.org/abs/2302.12400.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023b.

Reinhard Noll. Laser-induced breakdown spectroscopy. In *Laser-Induced Breakdown Spectroscopy: Fundamentals and Applications*, pages 7–15. Springer, 2011.

Cam Nugent. The california housing prices dataset. Kaggle, 2017. URL https://www.kaggle.com/datasets/camnugent/california-housing-prices.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

François-Pierre Paty and Marco Cuturi. Subspace robust wasserstein distances, 2019. URL https://arxiv.org/abs/1901.08949.

Rui Shu, Hung H. Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation, 2018. URL https://arxiv.org/abs/1802.08735.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation, 2017. URL https://arxiv.org/abs/1702.05464.

Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6090–6099. IEEE, 2017.

Gabriele Visentin and Patrick Cheridito. Computing optimal transport maps and wasserstein barycenters using conditional normalizing flows, 2025. URL https://arxiv.org/abs/2505.22364.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization, 2021. URL https://arxiv.org/abs/2006.10726.

Shuai Wang, Daoan Zhang, Zipei Yan, Jianguo Zhang, and Rui Li. Feature alignment and uniformity for test time adaptation. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 20050–20060. IEEE, 2023.

RC Wiens, S Maurice, J Lasue, O Forni, RB Anderson, S Clegg, S Bender, D Blaney, BL Barraclough, A Cousin, et al. Pre-flight calibration and initial data processing for the chemcam laser-induced breakdown spectroscopy instrument on the mars science laboratory rover. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 82:1–27, 2013.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

Qingyang Zhang, Yatao Bian, Xinke Kong, Peilin Zhao, and Changqing Zhang. Come: Test-time adaption by conservatively minimizing entropy, 2024. URL https://arxiv.org/abs/2410.10894.

Bowen Zhao, Chen Chen, and Shu-Tao Xia. Delta: degradation-free fully test-time adaptation, 2023. URL https://arxiv.org/abs/2301.13018.

Aurick Zhou and Sergey Levine. Bayesian adaptation for covariate shift. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=15HPeY8MGQ.

Zhan Zhuang, Yu Zhang, and Ying Wei. Gradual domain adaptation via gradient flow. In *International Conference on Learning Representations*, volume 2024, pages 26953–26978, 2024.

APPENDIX

A ADDITIONAL MAIN RESULTS

A.1 R^2 RESULTS ON MARS AND VACCUM

Tables 7 and 8 report R^2 scores on MARS and VACUUM. TOW achieves consistent performance over all baselines: 9%+ gain for SiO2 on both domains, 13% on MARS and 9% on VACUUM for K2O and the strongest gains on Al2O3 with 28% on MARS and 11% on VACUUM. Overall, TOW outperforms all baselines across all target oxides on both domains.

Table 7: MARS $R^2 \uparrow$ values across 8 target variables. Best values are **Bolded**.

Method	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
Source-only	$-0.97_{\pm 2.05}$	$-0.39_{\pm 0.36}$	$-3.62_{\pm 1.85}$	$-0.28_{\pm 0.63}$	$0.82_{\pm 0.07}$	$0.41_{\pm 0.85}$	$0.23_{\pm 0.08}$	$-0.25_{\pm 0.38}$
AdaBN	$0.69_{\pm 0.06}$	$0.51_{\pm 0.15}$	$-0.15_{\pm 0.34}$	$0.28_{\pm 0.61}$	$0.80_{\pm 0.06}$	$0.70_{\pm 0.14}$	$0.28_{\pm 0.14}$	$0.53_{\pm 0.13}$
RSD	$0.66_{\pm 0.07}$	$-0.11_{\pm 1.02}$	$-0.38_{\pm 0.40}$	$0.22_{\pm 0.90}$	$0.78_{\pm 0.10}$	$0.69_{\pm 0.27}$	$0.18_{\pm 0.19}$	$0.30_{\pm 0.19}$
SSA	$-0.29_{\pm 1.12}$	$-0.15_{\pm 0.16}$	$0.05_{\pm 0.14}$	$0.18_{\pm 0.15}$	$0.06_{\pm 0.20}$	$0.31_{\pm 0.31}$	$0.17_{\pm 0.05}$	$0.05_{\pm 0.03}$
FR	$0.59_{\pm 0.09}$	$0.49_{\pm 0.11}$	$-0.02_{\pm 0.26}$	$0.53_{\pm 0.21}$	$0.67_{\pm 0.16}$	$0.76_{\pm 0.06}$	$0.41_{\pm 0.12}$	$0.58_{\pm 0.12}$
TOW	$\mathbf{0.78_{\pm 0.01}}$	$\mathbf{0.63_{\pm 0.05}}$	$\mathbf{0.28_{\pm 0.20}}$	$\mathbf{0.54_{\pm 0.20}}$	$\mathbf{0.85_{\pm 0.07}}$	$\mathbf{0.84_{\pm 0.05}}$	$\mathbf{0.52_{\pm 0.04}}$	$\mathbf{0.71_{\pm 0.02}}$
Oracle	$0.88_{\pm 0.01}$	$0.86_{\pm 0.02}$	$0.77_{\pm 0.02}$	$0.85_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.81_{\pm 0.01}$	$0.89_{\pm 0.00}$

Table 8: VACUUM $R^2 \uparrow$ values across 8 target variables. Best values are **Bolded**.

Method	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
Source-only	$0.05_{\pm 0.31}$	$0.19_{\pm 0.13}$	$-1.38_{\pm 0.87}$	$-0.05_{\pm 0.45}$	$0.53_{\pm 0.27}$	$-0.09_{\pm 0.09}$	$-0.54_{\pm 0.76}$	$-0.17_{\pm 0.16}$
AdaBN	$0.64_{\pm 0.09}$	$0.44_{\pm 0.15}$	$0.49_{\pm 0.07}$	$0.57_{\pm 0.20}$	$\mathbf{0.79_{\pm 0.11}}$	$0.79_{\pm 0.06}$	$0.45_{\pm 0.12}$	$0.54_{\pm 0.04}$
RSD	$0.49_{\pm 0.19}$	$0.23_{\pm 0.16}$	$0.33_{\pm 0.11}$	$0.48_{\pm 0.24}$	$0.74_{\pm 0.08}$	$0.69_{\pm 0.15}$	$0.24_{\pm 0.24}$	$0.35_{\pm 0.22}$
SSA	$0.13_{\pm 0.16}$	$-0.16_{\pm 0.24}$	$0.16_{\pm 0.08}$	$0.10_{\pm 0.13}$	$0.07_{\pm 0.04}$	$0.09_{\pm 0.20}$	$0.05_{\pm 0.10}$	$-0.06_{\pm 0.09}$
FR	$0.60_{\pm 0.07}$	$0.45_{\pm 0.08}$	$0.46_{\pm 0.05}$	$\mathbf{0.63_{\pm 0.05}}$	$0.61_{\pm 0.19}$	$0.71_{\pm 0.06}$	$0.35_{\pm 0.12}$	$0.22_{\pm 0.11}$
TOW	$\mathbf{0.71_{\pm 0.04}}$	$\mathbf{0.54_{\pm 0.04}}$	$\mathbf{0.60_{\pm 0.07}}$	$0.61_{\pm 0.20}$	$\mathbf{0.79_{\pm 0.11}}$	$\mathbf{0.80_{\pm 0.06}}$	$\mathbf{0.59_{\pm 0.05}}$	$\mathbf{0.63_{\pm 0.03}}$
Oracle	$0.90_{\pm 0.00}$	$0.78_{\pm 0.02}$	$0.85_{\pm 0.00}$	$0.87_{\pm 0.01}$	$0.94_{\pm 0.01}$	$0.92_{\pm 0.00}$	$0.85_{\pm 0.01}$	$0.87_{\pm 0.01}$

A.2 F1-RECALL SCORES ON BIRDCALLS

Table 9: Birdcalls results. Best values are **Bolded**.

Method	F1 recall ↑
TSD	$0.238_{\pm 0.008}$
TACT	$0.238_{\pm 0.008}$
SHOT	$0.240_{\pm 0.011}$
Tent	$0.263_{\pm 0.026}$
SAR	$0.263_{\pm 0.026}$
TOW	$\mathbf{0.274}_{\pm 0.046}$

B ABLATION STUDIES

B.1 ABLATION ON THE SENSITIVITY OF LEARNING RATE (LR)

This section investigates the sensitivity of downstream performance to the LR used for 1) updating the affine parameters of pre-trained feature extractor on target domain and 2) optimizing the optimal transport map in TOW. First, we vary the LR used to update the parameters of feature extractor while keeping TOW's fixed. We then fix the LR for feature extractor at its best-performing value and vary the one used for optimizing TOW. We report RMSE results with different subspace dimensions in all settings.

B.1.1 LIBS

Results after varying LR for feature extractor (FE).

We fix TOW's LR to `1e-4` and vary the FE's in `{1e-4, 2e-4, 3e-4, 5e-4}`. Due to space constraints, we only report for { `(1e-4)` × `(1e-4, 5e-4)` } in Tables 10, 11, 12 and 13. We observe improved RMSE scores across all target oxides at a LR of `5e-4` on both MARS and VACUUM. For MARS, the results further improved when setting TOW's LR to `1e-5` (Table 14) whereas best values on VACUUM are obtained when TOW's LR is `1e-4`. We obtained best values using Adam optimizer for both FE and TOW. We suggest FE's LR within `{1e-4, 2e-4, 3e-4, 4e-4, 5e-4}` and TOW's in `{1e-4, 1e-5}`.

Table 10: MARS RMSE ↓ results across 8 target variables. FE LR `1e-4`, TOW LR `1e-4`.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	$7.14_{\pm 0.56}$	$0.67_{\pm 0.11}$	$3.98_{\pm 0.69}$	$3.83_{\pm 1.66}$	$3.47_{\pm 0.55}$	$2.61_{\pm 0.60}$	$1.06_{\pm 0.11}$	$1.32_{\pm 0.16}$
k15	$6.92_{\pm 0.57}$	$0.68_{\pm 0.13}$	$3.93_{\pm 0.67}$	$3.76_{\pm 1.60}$	$3.43_{\pm 0.55}$	$2.58_{\pm 0.59}$	$1.04_{\pm 0.10}$	$1.28_{\pm 0.13}$
k19	$\mathbf{6.86_{\pm 0.53}}$	$0.67_{\pm 0.11}$	$3.93_{\pm 0.68}$	$3.84_{\pm 1.85}$	$3.41_{\pm 0.56}$	$2.61_{\pm 0.66}$	$1.03_{\pm 0.10}$	$\mathbf{1.27_{\pm 0.13}}$
k22	$6.92_{\pm 0.56}$	$0.66_{\pm 0.11}$	$3.95_{\pm 0.69}$	$3.72_{\pm 1.54}$	$3.41_{\pm 0.56}$	$\mathbf{2.55_{\pm 0.60}}$	$1.03_{\pm 0.10}$	$\mathbf{1.27_{\pm 0.14}}$
k25	$6.91_{\pm 0.54}$	$\mathbf{0.64_{\pm 0.11}}$	$\mathbf{3.92_{\pm 0.66}}$	$\mathbf{3.65_{\pm 1.46}}$	$\mathbf{3.39_{\pm 0.53}}$	$2.63_{\pm 0.69}$	$\mathbf{1.02_{\pm 0.10}}$	$\mathbf{1.27_{\pm 0.14}}$

Table 11: MARS RMSE ↓ results across 8 target variables. FE LR `5e-4`, TOW LR `1e-4`.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	$6.47_{\pm 0.38}$	$0.66_{\pm 0.11}$	$3.79_{\pm 0.79}$	$3.77_{\pm 2.05}$	$3.34_{\pm 0.52}$	$2.38_{\pm 0.63}$	$1.01_{\pm 0.11}$	$1.29_{\pm 0.13}$
k15	$\mathbf{6.06_{\pm 0.16}}$	$0.66_{\pm 0.14}$	$3.64_{\pm 0.67}$	$3.33_{\pm 1.16}$	$3.26_{\pm 0.51}$	$2.15_{\pm 0.48}$	$0.96_{\pm 0.10}$	$1.25_{\pm 0.07}$
k19	$6.17_{\pm 0.31}$	$\mathbf{0.63_{\pm 0.03}}$	$3.66_{\pm 0.64}$	$3.52_{\pm 1.70}$	$3.12_{\pm 0.48}$	$2.30_{\pm 0.65}$	$0.94_{\pm 0.08}$	$1.22_{\pm 0.07}$
k22	$6.32_{\pm 0.28}$	$\mathbf{0.63_{\pm 0.12}}$	$3.64_{\pm 0.67}$	$3.33_{\pm 1.17}$	$3.17_{\pm 0.45}$	$\mathbf{2.13_{\pm 0.47}}$	$0.94_{\pm 0.09}$	$\mathbf{1.21_{\pm 0.06}}$
k25	$6.36_{\pm 0.18}$	$\mathbf{0.63_{\pm 0.13}}$	$\mathbf{3.47_{\pm 0.52}}$	$\mathbf{3.31_{\pm 1.21}}$	$\mathbf{3.02_{\pm 0.51}}$	$2.46_{\pm 0.78}$	$\mathbf{0.92_{\pm 0.08}}$	$\mathbf{1.21_{\pm 0.04}}$

Table 12: VACUUM RMSE ↓ results across 8 target variables. FE LR `1e-4`, TOW LR `1e-4`.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	$7.69_{\pm 0.91}$	$0.80_{\pm 0.09}$	$3.17_{\pm 0.22}$	$3.34_{\pm 0.90}$	$3.06_{\pm 0.70}$	$\mathbf{2.58_{\pm 0.42}}$	$0.99_{\pm 0.11}$	$1.49_{\pm 0.06}$
k15	$7.62_{\pm 0.96}$	$0.79_{\pm 0.09}$	$\mathbf{3.12_{\pm 0.17}}$	$3.37_{\pm 0.97}$	$3.04_{\pm 0.69}$	$\mathbf{2.58_{\pm 0.41}}$	$0.97_{\pm 0.11}$	$1.45_{\pm 0.06}$
k19	$7.63_{\pm 0.96}$	$0.80_{\pm 0.09}$	$\mathbf{3.12_{\pm 0.21}}$	$3.37_{\pm 1.02}$	$3.03_{\pm 0.69}$	$2.61_{\pm 0.48}$	$\mathbf{0.95_{\pm 0.11}}$	$1.44_{\pm 0.06}$
k22	$7.63_{\pm 0.87}$	$0.80_{\pm 0.09}$	$\mathbf{3.12_{\pm 0.21}}$	$3.29_{\pm 0.90}$	$\mathbf{3.01_{\pm 0.70}}$	$2.60_{\pm 0.46}$	$0.96_{\pm 0.10}$	$1.44_{\pm 0.07}$
k25	$\mathbf{7.55_{\pm 0.82}}$	$\mathbf{0.78_{\pm 0.09}}$	$\mathbf{3.12_{\pm 0.17}}$	$\mathbf{3.27_{\pm 0.82}}$	$3.02_{\pm 0.69}$	$2.63_{\pm 0.55}$	$\mathbf{0.95_{\pm 0.10}}$	$\mathbf{1.43_{\pm 0.06}}$

Table 13: VACUUM RMSE ↓ results across 8 target variables. FE LR `5e-4`, TOW LR `1e-4`.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	$\mathbf{7.24_{\pm 0.52}}$	$0.77_{\pm 0.06}$	$3.12_{\pm 0.31}$	$3.32_{\pm 1.09}$	$3.09_{\pm 0.71}$	$\mathbf{2.53_{\pm 0.45}}$	$0.92_{\pm 0.09}$	$1.39_{\pm 0.04}$
k15	$7.36_{\pm 0.79}$	$\mathbf{0.74_{\pm 0.03}}$	$3.01_{\pm 0.20}$	$3.34_{\pm 0.94}$	$3.14_{\pm 0.65}$	$2.55_{\pm 0.41}$	$0.91_{\pm 0.07}$	$1.41_{\pm 0.04}$
k19	$7.62_{\pm 0.49}$	$0.76_{\pm 0.04}$	$3.10_{\pm 0.37}$	$3.36_{\pm 0.95}$	$3.07_{\pm 0.64}$	$2.65_{\pm 0.62}$	$\mathbf{0.88_{\pm 0.05}}$	$1.41_{\pm 0.10}$
k22	$7.80_{\pm 0.44}$	$0.77_{\pm 0.07}$	$\mathbf{2.96_{\pm 0.27}}$	$\mathbf{3.21_{\pm 0.75}}$	$\mathbf{3.05_{\pm 0.66}}$	$2.63_{\pm 0.46}$	$0.89_{\pm 0.04}$	$\mathbf{1.38_{\pm 0.05}}$
k25	$7.53_{\pm 0.50}$	$0.76_{\pm 0.05}$	$3.07_{\pm 0.25}$	$3.33_{\pm 0.80}$	$3.12_{\pm 0.57}$	$2.84_{\pm 0.80}$	$0.90_{\pm 0.04}$	$1.39_{\pm 0.06}$

Results after varying LR for TOW.

Tables 14 and 15 show results with varying LR for TOW. We vary TOW's LR to `1e-5` and fix the feature extractor's at its best performing value of `1e-5`. Note that RMSE scores substantially vary across different subspace dimensions in Tables 11 and 14 for MARS and Tables 13 and 15 for VACUUM. These results suggest that TOW actively optimizes for a subspace with maximum displacement and at different LRs, it may convergence to different manifold structures in the feature space.

Table 14: MARS RMSE ↓ results across 8 target variables. TOW LR `1e-5`. FE LR `5e-4`.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	**6.43**$_{\pm 0.60}$	0.67$_{\pm 0.08}$	3.62$_{\pm 0.61}$	3.52$_{\pm 1.74}$	3.14$_{\pm 0.50}$	2.09$_{\pm 0.64}$	0.97$_{\pm 0.07}$	**1.30**$_{\pm 0.12}$
k15	6.52$_{\pm 0.58}$	0.72$_{\pm 0.11}$	3.61$_{\pm 0.56}$	**3.28**$_{\pm 0.72}$	2.98$_{\pm 0.56}$	**1.97**$_{\pm 0.36}$	0.95$_{\pm 0.07}$	1.41$_{\pm 0.09}$
k19	7.48$_{\pm 0.97}$	0.64$_{\pm 0.09}$	3.50$_{\pm 0.60}$	3.47$_{\pm 1.94}$	2.66$_{\pm 0.30}$	2.00$_{\pm 0.40}$	0.93$_{\pm 0.05}$	1.32$_{\pm 0.13}$
k22	7.14$_{\pm 0.38}$	**0.59**$_{\pm 0.04}$	3.44$_{\pm 0.58}$	3.50$_{\pm 0.96}$	3.20$_{\pm 0.88}$	2.10$_{\pm 0.35}$	**0.92**$_{\pm 0.04}$	1.36$_{\pm 0.17}$
k25	7.47$_{\pm 0.57}$	0.62$_{\pm 0.07}$	**3.28**$_{\pm 0.47}$	3.29$_{\pm 0.83}$	**2.56**$_{\pm 0.53}$	2.24$_{\pm 0.45}$	0.93$_{\pm 0.05}$	1.38$_{\pm 0.09}$

Table 15: VACUUM RMSE ↓ results across 8 target variables. TOW LR `1e-5`. FE LR `5e-4`.

	SiO2	TiO2	Al2O3	Fe2O3	MgO	CaO	Na2O	K2O
k5	**7.22**$_{\pm 0.52}$	**0.77**$_{\pm 0.05}$	2.93$_{\pm 0.24}$	**3.43**$_{\pm 1.12}$	**3.13**$_{\pm 0.60}$	**2.46**$_{\pm 0.34}$	**0.91**$_{\pm 0.07}$	**1.44**$_{\pm 0.05}$
k15	8.32$_{\pm 0.71}$	0.80$_{\pm 0.08}$	2.91$_{\pm 0.09}$	3.60$_{\pm 0.91}$	3.31$_{\pm 0.52}$	2.72$_{\pm 0.42}$	0.99$_{\pm 0.06}$	1.66$_{\pm 0.06}$
k19	8.93$_{\pm 0.93}$	0.81$_{\pm 0.11}$	3.00$_{\pm 0.38}$	3.46$_{\pm 0.94}$	3.51$_{\pm 0.32}$	2.76$_{\pm 0.59}$	0.95$_{\pm 0.04}$	1.61$_{\pm 0.13}$
k22	9.17$_{\pm 1.08}$	0.80$_{\pm 0.12}$	2.99$_{\pm 0.27}$	3.56$_{\pm 0.70}$	3.61$_{\pm 0.75}$	2.84$_{\pm 0.33}$	0.96$_{\pm 0.05}$	1.61$_{\pm 0.16}$
k25	9.16$_{\pm 1.04}$	0.84$_{\pm 0.11}$	**2.82**$_{\pm 0.24}$	3.56$_{\pm 0.67}$	3.37$_{\pm 0.48}$	3.06$_{\pm 0.61}$	1.04$_{\pm 0.05}$	1.61$_{\pm 0.09}$

B.1.2 CAMELYON17

On Camelyon17, we fix TOW LR to `1e-5` and vary FR LR within `{1e-5, 1e-6}`. We report accuracy with different subspace dimensions. We observe that updating all layers in feature extractor significantly improves accuracy on $h2$ compared to only BN layers. We attribute this to the greater complexity of task, which may benefit from adapting the early convolutional layers of ResNet as they tend to capture low-level features. This finding is consistent with the observation reported in (Eastwood et al., 2022). when updating only BN layers. We observed SGD optimizer works better for indices like $h0$ and $h1$ and Adam for $h2$. Here, we use Adam optimizer for both FE and TOW, set TOW weight decay to `1e-4`z. Table 16 report results when all layers are updated whereas Table 17

Table 16: Accuracy ↑ results with Adam optimizer. When updating all layers in FE.

	FR LR `1e-5`				FE LR `1e-6`			
	h0	h1	h2	h4	h0	h1	h2	h4
k50	82.73$_{\pm 6.59}$	71.10$_{\pm 2.37}$	90.04$_{\pm 0.97}$	75.85$_{\pm 6.16}$	91.30$_{\pm 1.01}$	82.15$_{\pm 0.21}$	87.43$_{\pm 2.68}$	84.03$_{\pm 8.82}$
k55	85.64$_{\pm 3.76}$	69.72$_{\pm 3.36}$	84.56$_{\pm 11.47}$	72.31$_{\pm 2.64}$	91.97$_{\pm 0.48}$	82.43$_{\pm 1.17}$	86.24$_{\pm 1.66}$	79.96$_{\pm 8.32}$
k90	88.21$_{\pm 1.88}$	70.07$_{\pm 3.69}$	72.22$_{\pm 10.63}$	58.26$_{\pm 11.62}$	91.92$_{\pm 0.36}$	82.11$_{\pm 0.98}$	86.49$_{\pm 2.24}$	81.88$_{\pm 7.69}$

Table 17: Accuracy ↑ results with Adam optimizer. When updating BN layers in FE.

	FE LR `1e-5`				FE LR `1e-6`			
	h0	h1	h2	h4	h0	h1	h2	h4
k50	90.47$_{\pm 0.62}$	84.48$_{\pm 0.85}$	77.02$_{\pm 1.94}$	84.02$_{\pm 1.13}$	90.04$_{\pm 0.68}$	84.55$_{\pm 0.87}$	76.43$_{\pm 1.96}$	82.70$_{\pm 1.53}$
k55	90.67$_{\pm 0.63}$	84.55$_{\pm 0.87}$	76.34$_{\pm 1.72}$	84.42$_{\pm 1.06}$	90.25$_{\pm 0.62}$	84.63$_{\pm 0.81}$	75.74$_{\pm 1.65}$	83.01$_{\pm 1.38}$
k90	90.62$_{\pm 0.65}$	84.39$_{\pm 0.90}$	76.57$_{\pm 1.94}$	84.50$_{\pm 1.15}$	90.12$_{\pm 0.66}$	84.45$_{\pm 0.79}$	75.99$_{\pm 1.95}$	83.10$_{\pm 1.46}$

B.2 EXPERIMENTS ON LIBS SEMI-SYNTHETIC TARGET DOMAIN

To stress-test TOW's robustness, we create semi-synthetic target domains by randomly selecting dead peaks and perturbing them with Gaussian noise whose intensity (height) is correlated with the target variable. Specifically, we generate spurious noise with $z = \rho \cdot y_{\text{std}} + \sqrt{1-\rho^2} \cdot \epsilon_{\text{std}}$ for correlation strength ρ, where y_{std} is computed from all measurements for each target variable. We set $\rho = 0.9$ for all target variables. We apply truncation to bound perturbed peaks within physically plausible ranges. We use *source* EARTH to construct two target domains by (a) varying the number (size) of selected dead peaks while keeping their intensity (height) fixed, and (b) varying the intensity while keeping the size of selected peaks fixed. For domain (a), we vary the size of perturbed peaks within $\{15, 60, 90\} \times t$, where t is the number of target variables and scale the heights to $1.0\times$ the maximum peak intensity of original spectrum (Figure 3). For domain (b), we fix the number of perturbed peaks

to 15 and vary the heights in $\{2.0, 4.0, 8.0\}$ relative to the maximum peak intensity of the original spectrum (Figure 4).

Tables 18, 20 and 19, 21 report RMSE values for baselines and TOW with different subspace dimensions, respectively. We observe that as the size of perturbed peaks (Table 18 and19) and the scaling factor of height (Table 20 and 21) increases, the pretrained model's degraded performance suggests that it may be relying on spurious correlations and/or is sensitive to feature scale. RSD outperforms FR potentially because it leverages a geometry-aware discrepancy measure to reduce domain gap. TOW achieves best RMSE values with $k \in \{2, 5\}$, suggesting that only a few dimensions experienced domain shift. Since we linearly added perturbed peaks in the source EARTH to create semi-synthetic target domains, only a few dimensions may be affected – those relevant for prediction and thus lower value of k obtained better results. Whereas MARS and VACUUM might have more dimensions affected through non-linear process and thereby $k > 5$ gave best results in Tables 1, 2. Surprisingly, AdaBN performs better than TOW on both semi-synthetic domains, though we do not fully understand why it outperforms TOW under these settings but not on MARS and VACUUM. We leave this for future work.

Domain (a), with varying peak sizes.

Table 18: RMSE ↓ results. The performance of all baselines degrades as the size of perturbed peaks increases, except for AdaBN.

Method	peaks15-height1.0					peaks60-height1.0					peaks90-height1.0				
	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO
Source-only	4.03	0.40	1.63	1.26	2.29	6.75	0.50	2.30	1.80	1.92	12.47	0.59	2.33	2.08	2.61
AdaBN	4.01	0.43	1.78	1.43	1.84	4.13	0.45	1.81	1.41	1.86	5.21	0.47	1.83	1.49	1.87
RSD	4.62	0.51	1.83	2.01	2.60	5.17	0.79	2.06	2.49	1.84	8.19	0.59	3.67	3.73	3.71
SSA	11.53	1.12	3.91	4.94	5.71	12.17	1.11	3.97	5.11	5.54	12.38	1.11	4.18	5.07	5.98
FR	6.91	0.74	2.48	2.41	3.91	9.19	0.87	2.57	2.19	3.55	9.83	0.88	3.06	5.31	4.94

Table 19: TOW achieves RMSE ↓ results competitive to AdaBN.

	peaks15-height1.0					peaks60-height1.0					peaks90-height1.0				
	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO
k2	4.41	0.46	1.72	1.45	2.09	4.56	0.48	1.78	1.54	1.97	5.57	0.49	1.89	1.48	1.94
k5	4.32	0.45	1.78	1.64	2.14	4.42	0.47	1.86	1.47	2.05	5.51	0.48	1.84	1.67	2.02
k15	4.61	0.50	1.91	1.75	2.19	4.67	0.51	1.94	1.77	2.06	5.42	0.51	1.96	1.94	2.00
k25	5.38	0.54	1.94	2.13	2.23	5.36	0.55	1.92	1.75	2.07	5.98	0.55	1.95	1.91	2.14

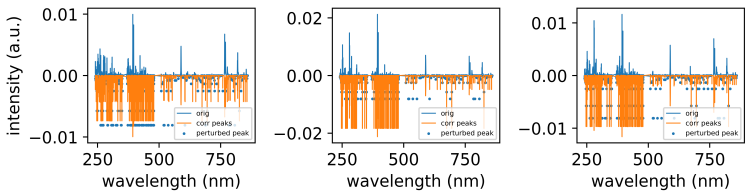


Figure 3: Randomly selected dead peaks with perturbed correlated noise (size60_height1.0).

Domain (b), with varying peak heights.

Table 20: RMSE ↓ results. The performance of all baselines degrades as the height of perturbed peaks increases, except for AdaBN.

Method	peaks15-height2.0					peaks15-height4.0					peaks15-height8.0				
	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO
Source-only	4.59	0.44	1.56	1.35	1.91	9.44	0.53	2.51	3.24	3.17	8.27	1.21	3.86	3.02	3.33
AdaBN	4.03	0.46	1.78	1.45	1.88	4.49	0.46	1.75	1.64	1.78	5.76	0.47	2.09	2.43	2.24
RSD	4.41	0.48	2.56	1.59	2.92	7.66	0.52	2.57	3.16	2.17	12.04	0.57	3.13	3.61	3.22
SSA	11.26	1.12	3.97	4.97	5.71	12.21	1.11	3.90	5.32	6.08	12.79	1.10	3.76	5.30	6.05
FR	7.30	0.66	2.44	3.68	4.11	8.84	0.71	3.22	4.11	3.93	11.62	0.91	3.18	5.50	4.13

Table 21: TOW achieves RMSE ↓ results competitive to AdaBN.

	peaks15-height2.0					peaks15-height4.0					peaks15-height8.0				
	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO	SiO2	TiO2	Al2O3	Fe2O3	MgO
k2	4.48	0.48	1.74	1.46	2.04	4.82	0.48	1.75	1.72	2.14	5.62	0.52	2.22	2.40	2.33
k5	4.40	0.47	1.75	1.72	2.09	4.88	0.46	1.77	2.04	2.07	5.43	0.49	2.26	2.71	2.34
k15	4.66	0.51	1.89	2.29	2.18	5.17	0.49	1.82	2.13	2.01	5.77	0.51	2.30	2.54	2.39
k25	5.41	0.56	1.89	1.45	2.20	5.89	0.54	1.85	2.59	2.03	6.23	0.54	2.12	2.43	2.46

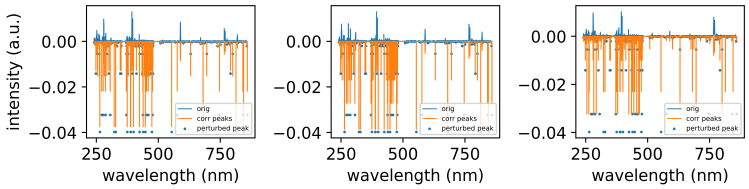


Figure 4: Randomly selected dead peaks with perturbed correlated noise (size15_height4.0).

C Experimental Details

We used PyTorch (Paszke et al., 2019) in the implementation of all experiments including source model pre-training, baselines and our proposed method. We conducted experiments using NVIDIA Geforce RTX 5070 GPU.

C.1 Datasets

LIBS (Dyar et al., 2024). EARTH dataset contains 6,500 samples, and MARS and VACUUM consist of 7,500 and 8,500 measurements respectively. The data processing process is described in (Wiens et al., 2013; Anderson et al., 2022). We download EARTH, VACUUM from [1] and MARS at [2]. The datasets are publicly available.
California housing (Nugent, 2017). It is a commonly used tabular benchmark dataset for distribution shift. We download at Kaggle [3].
Camelyon17 (Bandi et al., 2018). It consists of pathological tumor tissue scans from five hospitals. Following (Eastwood et al., 2022), we use index h3 as source domain, however, we report accuracy over each target index individually and compute the average. The dataset is publicly available via `wilds` [4] library.
Birdcalls (Gao et al., 2023). It is a multi-classification task dataset. For consistency, we follow the dataset split from (Liu et al., 2025). The dataset is publicly available via `wilds` [4] library.

C.2 Source domain pre-training

LIBS. Following (Adachi et al., 2025), we train a 5-layer MLP on EARTH. Since each measurement (spectrum) has 8 target chemical oxides, we construct a univariate model for each oxide independently. We trained models using standard MSE loss function with Adam (Kingma and Ba, 2017) optimizer, learning rate of `1e-03`, 128 batch size and 200 epochs. We employ early stopping with a patience of 20 epochs to prevent overfitting. We used 70% data for training and 30% for validation.

California housing. We adopt the data split, architecture and optimization settings from LIBS.

Camelyon17. Following (Eastwood et al., 2022), we trained ResNet18 from scratch on hospital index $h3$ using SGD optimizer. We set batch size to 200, learning rate as `1e-3` and epochs to 20.

[1] https://zenodo.org/records/10287455
[2] https://zenodo.org/records/7566042
[3] https://www.kaggle.com/datasets/camnugent/california-housing-prices
[4] https://github.com/p-lambda/wilds/

Birdcalls. We adopt the architecture from Camelyon17; however, we use Adam optimizer and set batch size to 200, learning rate as `1e-3` and train for 100 epochs.

C.3 Test-time Adaptation

Source-only. We test pre-trained source model on target domain in `model.eval()` mode.

AdaBN (Benz et al., 2021). Updates running stats of BatchNorm layers with target domain by testing the model in `model.train()` mode.

Oracle We fine-tune the pre-trained source model on target domain to get an upper bound of the performance.

Representation subspace distance (RSD) (Chen et al., 2021). RSD is an unsupervised domain adaptation method designed for regression tasks. Since conitnuous regression is not robust to feature scale (unlike classification), RSD uses SVD to define a geometrical distance over orthogonal bases to reduce domain discrepancy. Following Adachi et al. (2025), we pre-compute SVD on full source features and perform adaptation with mini-batch target features using RSD loss function.

Feature restoration (FR) (Eastwood et al., 2022). FR uses histogram to approximate the marginal distribution of source features and logits. During adaptation, it computes same statistics from target features and aligns with the precomputed source ones. We use scalar valued outputs instead of logits in the case of continuous regression tasks. We tested learning rate $\{\texttt{1e-3}, \texttt{1e-4}, \texttt{1e-5}\}$ and achieved best values at 0.0001. Following Eastwood et al. (2022), we set the number of histogram bins to 8 and soft-binning temperature τ to 0.001.

Subspace Significant Alignment (SSA) (Adachi et al., 2025). SSA uses PCA to identify source feature subspace with maximum variance, it then computes its second-order statistics and principal basis. During adaptation,it projects target features on source principal basis and aligns the projected traget features with the source ones. We optimized the number of feature dimensions k over $\{5, 10, 15, 20, 25, 30, 35\}$ and report best results.

TACT (Liu et al., 2025). TACT is a test-time adaptation method that applies a number of data augmentations on target features. It then uses PCA to identify non-causal dimensions and removes their influence on the representations used for final prediction. TACT cannot be applied directly as it requires carefully designed domain-specific augmentations to preserve the underlying causal features and not suitable for regression tasks. We optimize parameters in the range suggested by Liu et al. (2025).

Test-time Self Distillation (TSD) (Wang et al., 2023). It is a test-time adaptation method that uses class-wise pseudo-prototypes to promote a uniform distribution of features and maintains a feature memory bank to enable local feature alignment for new test samples. Applying it to a regression task is non-trivial as its components are designed primarily for classification.

SAR (Niu et al., 2023a). SAR investigates the impact of BN layer statistics on TTA performance under wild test samples. It proposes sharpness-aware entropy minimization objective that removes samples' with large gradients and encourages model confidence on remaining (reliable) test samples. SAR is a classification-specific method due to its entropy based objective function and thus cannot be applied to regression.

Tent (Wang et al., 2021). Tent is a test-time adaptation method that uses prediction entropy as an objective function to improve pretrained model's confidence on target domain. It updates the channel-wise affine parameters in BN layers. However, it cannot be extended to scalar valued regression tasks as computing entropy requires a distributional output.

SHOT (Liang et al., 2020). SHOT is an unsupervised, source-free domain adaptation method that uses information-maximization and class-wise prototype-based pseudo-labeling to adapt the feature extractor to target while keeping the pre-trained source classifier fixed. Therefore, it cannot be applied to regression tasks as it contains classification-specific components (e.g., pseudo-labeling).

Pseudo-Labels (PL) (Lee et al., 2013). It assigns pseudo-labels to unlabeled target samples based on their predicted class probabilities and therefore cannot be extended to continuous scalar valued regression tasks.

Transporting Flow (ours). Based on our experiments on four datasets with diverse tasks, we recommend optimizing g_θ^t feature extractor's learning rate in $\{\texttt{1e-4}, \texttt{1e-5}, \texttt{1e-6}\}$ and f_ϕ^{tow} TOW in $\{\texttt{1e-4}, \texttt{1e-5}\}$. For continuous regression, we suggest $k \in \{15, 20, 25, 30\}$ with shallow networks. For classification, assuming only a few feature dimensions will get substantially displaced under measurement-shift, we suggest $kin\{50, 60, 70, 80, 90\}$.