The Points You Didn't See: Continuous Domain Adaptation with Koopman Operator and Gaussian Mixture Optimal Transport

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we introduce a continuous domain adaptation method that learns domain-invariant representations by connecting optimal transport and transfer operators theory. We represent empirical source and target samples as Gaussian mixtures, solve an entropic GMM optimal transport problem between them, and use the resulting plan to construct a continuous weighted graph on which we define a Koopman operator. The key insight is that the infinite-dimensional Koopman eigenproblem admits an exact finite-dimensional reduction, yielding closed-form embeddings that we show are domain-invariant and enable out-of-sample mapping. Empirical validation on benchmarks spanning audio, signal processing, and visual adaptation modalities shows that our approach consistently outperforms OT-based, deep domain adaptation, and neural OT baselines, with a decisive advantage on out-of-sample generalization.

1 Introduction

Statistical learning theory is grounded in the assumption that training and test samples are drawn from the same underlying distribution. In practice, however, once a model is trained then deployed, the conditions under which inference-time data is acquired often diverge from those of the training set, inducing distributional shifts that alter the statistical structure of the data. In computer vision, for instance, changes in illumination, camera hardware, or capture conditions alter the appearance of the same visual content; in speech and signal processing, variations in sensor characteristics, recording environments, or background noise produce different representations of the same underlying phenomenon. Under such shifts, a model that performs well on the *source domain* may degrade unpredictably on the *target domain*, not because the model failed or even overfit the training data, but because the statistical structure it relied upon no longer holds. This problem lies at the core of domain adaptation (Pan & Yang, 2010), where the goal is to find a learning paradigm that remains robust to distributional shift by exploiting knowledge transferred from a domain where it was originally acquired (Redko et al., 2020). A widely adopted assumption in this setting is covariate shift (Shimodaira, 2000; Courty et al., 2017b), where the marginal distributions of inputs differ across domains, $P_s(x) \neq P_t(x)$, yet the conditional distribution of labels given inputs remains the same, $P_s(y \mid x) = P_t(y \mid x)$, *i.e.,* the underlying task is preserved. Ben-David et al. (2010) formalizes this intuition by showing that successful adaptation requires low error on the source domain, reduced discrepancy between $P_s(x)$ and $P_t(x)$, and a shared discriminative structure across both source and target domains, raising the question of *how such discrepancy should be measured and minimized meaningfully.*

To this end, Optimal Transport (OT) (Villani et al., 2009; Courty et al., 2017b) provides a principled and meaningful answer by offering a theoretically grounded way to compare and align probability distributions, seeking the minimum effort strategy required to transport samples from one distribution to another. It induces a transport plan that can be leveraged to map source samples toward the target distribution. However, the solution is sensitive to regularization (Cuturi, 2013), and mapping source samples to target domain via barycentric projection may produce biased domain alignment (Sad Saoud et al., 2026). Moreover, most shallow OT methods do not provide *out-of-sample mapping* (Seguy et al., 2018), in the sense that the transport plan is computed exclusively between the discrete measures (Montesuma et al., 2025), meaning any

newly observed point that was not part of the original optimization cannot be embedded or mapped without recomputing the OT plan from scratch, leaving the downstream classifier with no guarantee of correct predictions on such points at inference time. In this work, we address these limitations jointly by proposing a continuous domain adaptation framework that learns domain-invariant representations as eigenfunctions of the Koopman operator defined on an OT-coupled Gaussian mixture graph, providing a closed-form and continuous mapping that naturally extends to out-of-sample points without recomputing the transport plan.

Contributions. Our contributions are threefold. First, we formulate domain adaptation as a transfer operator problem combined with optimal transport, where source and target empirical samples are lifted to Gaussian mixtures and coupled through an entropic optimal transport plan, which induces a continuous random walk whose dynamics are given by a Koopman operator. Second, we prove that the resulting infinite-dimensional eigenproblem of this operator admits an exact finite-dimensional reduction, and derive a closed-form expression for the eigenfunctions through a block fixed-point system. Third, we interpret these eigenfunctions as domain-invariant representations and show that, because they extend to any new point in the input space, our approach delivers an out-of-sample mapping with a performance that consistently exceeds OT-based, deep domain adaptation, and neural OT baselines across benchmarks of varying modalities.

Paper organization. The rest of the paper is organized as follows: Section 2 discusses related work on domain adaptation. Section 3 reviews the theoretical background on optimal transport, GMM optimal transport, and transfer operators. Section 4 details our proposed method **DoDyn**. Section 5 reports empirical results on benchmarks, and Section 6 examines the sensitivity of DoDyn to its key hyperparameters and design choices. Finally, We conclude in Section 7. All proofs, derivations, and additional experiments are provided in the appendix.

Notation. Scalars, vectors, and matrices are denoted by lowercase a, boldface lowercase $\mathbf{a}$, and boldface uppercase $\mathbf{A}$ respectively. For a matrix $\mathbf{A}$, $\mathbf{A}^\top$ denotes its transpose and $\mathbf{A}^{-\mathbf{1}}$ its inverse when $\mathbf{A}$ is invertible. The canonical inner product between two vectors or matrices is $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_i u_i v_i$, with the Frobenius inner product $\langle \mathbf{U}, \mathbf{V} \rangle_F = \mathrm{Tr}(\mathbf{U}^\top \mathbf{V})$ for matrices. We denote by $\|\cdot\|_2$ the Euclidean norm and $\|\cdot\|_F$ the Frobenius norm. The identity matrix of size n is $\mathbf{I_n}$. We write $\mathbf{1}_n = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$ for the vector of all ones, so that for $\mathbf{A} \in \mathbb{R}^{n \times m}$, $\mathbf{A}\mathbf{1}_m \in \mathbb{R}^n$ and $\mathbf{A}^\top \mathbf{1}_n \in \mathbb{R}^m$ denote its row and column sums respectively. δ_x denotes the Dirac measure at point x, such that a discrete measure μ can be written as $\mu = \sum_{i=1}^n a_i \delta_{x_i}$ with $x_i \in \mathbb{R}^d$, $a_i \geq 0$ and $\sum_{i=1}^n a_i = 1$. We denote by $\mathcal{M}_+^1(\mathcal{X})$ the set of probability distributions (positive Radon measures of unit mass) over a metric space $\mathcal{X}$. We use $\Pr(\cdot)$ to denote probabilities and $\mathbb{E}[\cdot]$ for expectations.

2 Related Work

Deep Domain Adaptation. Deep domain adaptation methods learn domain-invariant representations either by minimizing a statistical discrepancy between source and target features in a shared latent space via MMD-based adaptation layer and domain confusion loss (Tzeng et al., 2014), multi-kernel MMD across all task-specific layers embedded in a reproducing kernel Hilbert space (Long et al., 2015), and second-order statistic alignment (Sun et al., 2016), or via adversarial training methods, where a gradient reversal layer (Ganin et al., 2016), conditional domain discriminator and classifier predictions (Long et al., 2018), or two-classifier discrepancy maximization and a near support feature generator (Saito et al., 2018) are used to produce domain-invariant features. While effective, these methods fall outside the scope of this work, which focuses on geometrically explicit optimal transport based alignment.

Optimal Transport for Domain Adaptation. Optimal transport has been used both to design domain adaptation algorithms and to theoretically analyze the problem (Courty et al., 2017b; Redko et al., 2017). The foundational approach of Courty et al. (2017b) uses the Kantorovich relaxation to compute a regularized transport plan between source and target empirical distributions, defining a barycentric mapping of source samples toward the target domain. Perrot et al. (2016) jointly learned the coupling and an approximation of the transport map via a convex formulation, enabling out-of-sample generalization of the mapping. Flamary et al. (2019) studied linear Monge mapping estimation and derived concentration bounds and generalization guarantees for OT-based domain adaptation. JDOT (Courty et al., 2017a) and

DeepJDOT (Damodaran et al., 2018) extended the framework by simultaneously optimizing the transport coupling and a prediction function to align the full joint distribution $P(x, y)$ rather than input marginals alone. Beyond empirical distributions, Delon & Desolneux (2020) introduced a Wasserstein-type distance on the space of Gaussian mixture models by restricting the OT coupling to Gaussian mixtures, showing that this continuous formulation reduces to an efficient discrete problem between mixture components. Building on this, Montesuma et al. (2025) proposed two Unsupervised domain adaptation strategies: label propagation from source to target GMM components via the GMMOT plan, and sample mapping between domains. However, as acknowledged by the authors, this assumes data is multi-modal and well-separated into clusters, which is difficult to satisfy in high dimensions.

Neural Optimal Transport. Beyond discrete solvers, neural OT methods learn continuous parametric transport maps. Seguy et al. (2018) proposed a two-step approach that first learns a stochastic dual OT plan then estimates a Monge map as a neural network approximating its barycentric projection, providing out-of-sample extension; Makkuva et al. (2020) parameterized the optimal transport map as the gradient of a convex function, and learns the optimal Kantorovich potential via minimax optimization over input-convex neural networks (ICNNs); which are also used in (Taghvaei & Jalali, 2019; Korotin et al., 2021). Korotin et al. (2023) proposed a unified framework learning continuous OT maps and plans for both strong and weak transport costs, with universal approximation guarantees. The latter methods are primarily designed for generative modeling and unpaired image-to-image translation rather than domain adaptation. While neural solvers provide out-of-sample extension, a major difference with our approach is that they operate in the primal feature space without extracting domain-invariant spectral structure, and their training relies on minimax optimization.

Spectral Domain Adaptation. A related line of work exploits spectral structures for domain adaptation, either via Laplacian-based spectral regularizer (Xiao et al., 2023) or label propagation (El Hamri et al., 2021). Closer to our work, Sad Saoud et al. (2026) construct a weighted adjacency matrix from OT plans between discrete measures and extract domain-invariant representations via spectral embedding with Arnoldi solver, but remain limited to a discrete formulation with no out-of-sample extension, and with empirical validation on *multi-source domain adaptation* setting only. To the best of our knowledge, no existing work combines Koopman operator theory with optimal transport to derive a continuous closed-form domain adaptation formulation that supports out-of-sample extension.

3 Background

This section provides the theoretical foundations underlying our approach, covering optimal transport (Section 3.1), its extension to Gaussian mixture models (Section 3.2), and Koopman operator theory (Section 3.3).

3.1 Optimal Transport

Optimal transport (OT) originates from the problem first posed by Monge (1781) of displacing mass at minimum total effort. Its modern measure-theoretic formulation, due to Kantorovich (1942), recasts this as an optimization problem over joint probability distributions, and has since become a powerful tool in machine learning (Villani et al., 2009; Peyré & Cuturi, 2019; Montesuma et al., 2024). In machine learning applications, the Kantorovich formulation is of particular interest as distributions are only accessible through finite sets of samples, making a probabilistic coupling between them both natural and tractable.

Definition 3.1 (Kantorovich Relaxation)**.** *Let μ and ν be two probability distributions over a metric space $\mathcal{X}$, and let $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ be a ground cost function measuring the effort of transporting a unit of mass from x to x'. The Kantorovich optimal transport problem seeks the coupling of minimum transport cost:*

$$\pi^* = \underset{\pi \in \Pi(\mu,\nu)}{\arg\inf} \int_{\mathcal{X}\times\mathcal{X}} c(x, x')\, d\pi(x, x'), \tag{1}$$

where $\Pi(\mu, \nu) = \left\{\pi \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{X}) \;\middle|\; \int_{\mathcal{X}} \pi(\cdot, x')\, dx' = \mu,\ \int_{\mathcal{X}} \pi(x, \cdot)\, dx = \nu\right\}$ is the set of joint distributions over $\mathcal{X} \times \mathcal{X}$ whose marginals are μ and ν respectively.

In the domain adaptation setting, μ and ν are referred to as the source and target measures μ_s and μ_t respectively, and we adopt this notation throughout. The formulation in equation 1 is an infinite-dimensional problem (Peyré & Cuturi, 2019; Montesuma et al., 2025). In practice, one only has access to finite sets of source and target observations $\boldsymbol{X_s} = \{\mathbf{x_s}^i\}_{i=1}^{n_s}$ and $\boldsymbol{X_t} = \{\mathbf{x_t}^j\}_{j=1}^{n_t}$, respectively. A natural way to represent these observations as probability distributions is through their associated empirical measures $\mu_s = \frac{1}{n_s}\sum_{i=1}^{n_s}\delta_{\mathbf{x_s}^i}$ and $\mu_t = \frac{1}{n_t}\sum_{j=1}^{n_t}\delta_{\mathbf{x_t}^j}$, where μ_s and μ_t are supported on $\boldsymbol{X_s}$ and $\boldsymbol{X_t}$ respectively. Under this discretizations, equation 1 reduces to the following finite linear program:

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi}\in\Pi(\mu_s,\mu_t)}{\arg\min}\ \langle \boldsymbol{\pi}, \boldsymbol{C}\rangle_F = \underset{\pi\in\Pi(\mu_s,\mu_t)}{\arg\min} \sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\pi_{ij}\, C_{ij}, \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{n_s\times n_t}$ is the cost matrix with entries $C_{ij} = c(\mathbf{x_s}^i, \mathbf{x_t}^j) = \|\mathbf{x_s}^i - \mathbf{x_t}^j\|_2^2$ *i.e, square euclidean distance*, and $\Pi(\mu_s,\mu_t) = \left\{\boldsymbol{\pi}\in\mathbb{R}_+^{n_s\times n_t}\ \middle|\ \boldsymbol{\pi}\mathbf{1}_{n_t} = \mu_s,\ \boldsymbol{\pi}^\top\mathbf{1}_{n_s} = \mu_t\right\}$. This problem is convex in $\boldsymbol{\pi}$ and admits at least one solution, yet solving it exactly via linear programming (Dantzig, 2016) scales in $\mathcal{O}(n^3\log n)$ when $n_s = n_t = n$, rendering it computationally expensive at the scales typical of machine learning applications (Peyré & Cuturi, 2019). Cuturi (2013) proposed adding an entropic regularization term to improve computational tractability. In its continuous form, the *entropic optimal transport problem* reads:

$$\pi_\varepsilon^* \;=\; \underset{\pi\in\Pi(\mu_s,\mu_t)}{\arg\inf} \int_{\mathcal{X}\times\mathcal{X}} c(x,x')\, d\pi(x,x') \;+\; \varepsilon \underbrace{\int_{\mathcal{X}\times\mathcal{X}} \log\left(\frac{d\pi(x,x')}{d(\mu_s\otimes\mu_t)(x,x')}\right) d\pi(x,x')}_{\mathrm{KL}(\pi\,||\mu_s\otimes\mu_t)}, \tag{3}$$

where $\varepsilon > 0$ controls the regularization strength, and $\mathrm{KL}(\pi\ ||\ \mu_s\otimes\mu_t)$ is the Kullback-Leibler divergence between the candidate coupling π and the product measure $\mu_s\otimes\mu_t$. One recovers equation 1 in the limit $\varepsilon\to 0$. Its discrete counterpart reads:

$$\boldsymbol{\pi}_\varepsilon^* \;=\; \underset{\boldsymbol{\pi}\in\Pi(\mu_s,\mu_t)}{\arg\min}\ \langle \boldsymbol{\pi}, \mathbf{C}\rangle_F \;-\; \varepsilon\, H(\boldsymbol{\pi}), \qquad H(\boldsymbol{\pi}) \overset{\text{def.}}{=} -\sum_{i,j}\pi_{ij}\big(\log\pi_{ij} - 1\big), \tag{4}$$

where $H(\boldsymbol{\pi})$ is the Shannon entropy of the coupling. The appeal of equation 4 is that the entropic term makes the objective strictly convex in $\boldsymbol{\pi}$, ensuring a unique minimizer that can be obtained through the iterative matrix-scaling procedure of Sinkhorn & Knopp (1967). The optimal coupling takes the scaling form $\pi^*_{\varepsilon,ij} = u_i\, K_{ij}\, v_j$, where $K_{ij} = \exp(-C_{ij}/\varepsilon)$ is the *Gibbs kernel* and $\mathbf{u}\in\mathbb{R}_+^{n_s}$, $\mathbf{v}\in\mathbb{R}_+^{n_t}$ enforce the marginal constraints. Starting from $\mathbf{v}^{(0)} = \mathbf{1}_{n_t}$, Sinkhorn's algorithm alternates:

$$\mathbf{u}^{(\ell+1)} \;=\; \frac{\boldsymbol{\mu_s}}{\mathbf{K}\mathbf{v}^{(\ell)}}, \qquad \mathbf{v}^{(\ell+1)} \;=\; \frac{\boldsymbol{\mu_t}}{\mathbf{K}^\top\mathbf{u}^{(\ell+1)}}, \tag{5}$$

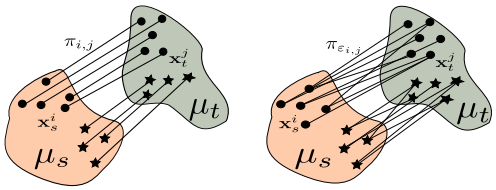

Figure 1: Illustration of optimal transport problem: (left) **Exact OT**; (right) **Entropic OT**.

with entry-wise divisions, and returns $\widehat{\boldsymbol{\pi}}_\varepsilon^L = \mathrm{diag}(\mathbf{u}^{(L)})\,\mathbf{K}\,\mathrm{diag}(\mathbf{v}^{(L)})$ after L iterations, recovering a solution of equation 2 as $\varepsilon\to 0$ and $L\to\infty$. Each iteration reduces to matrix-vector products that parallelize naturally on GPUs and integrate with differentiable pipelines (Genevay et al., 2018), bringing the per-iteration cost down to $\mathcal{O}(n^2)$ when $n_s = n_t = n$ (Peyré & Cuturi, 2019). Additionally, another property of the entropic OT is the smoothing it induces on the transport plan: as ε grows, each source point spreads its mass across several nearby target candidates rather than committing to a one-to-one mapping as in the Exact OT(see Figure 1), a property that we leverage in our approach.

3.2 Gaussian Mixture Optimal Transport

When both μ_s and μ_t are Gaussian, i.e., $\mu_s = \mathcal{N}(\mathbf{m}^s, \boldsymbol{\Sigma}^s)$ and $\mu_t = \mathcal{N}(\mathbf{m}^t, \boldsymbol{\Sigma}^t)$, where $\mathbf{m}^s, \boldsymbol{\Sigma}^s, \mathbf{m}^t, \boldsymbol{\Sigma}^t$ denote the mean and the covariance of source and target Gaussian distributions, respectively. The OT

problem equation 1 with quadratic cost admits a closed-form solution, where the optimal transport cost takes the Bures form (Takatsu, 2011):

$$\mathcal{W}_2^2(\mu_s, \mu_t) \;=\; \|\mathbf{m}^s - \mathbf{m}^t\|_2^2 \;+\; \mathrm{tr}\Big(\boldsymbol{\Sigma}^s + \boldsymbol{\Sigma}^t - 2\big((\boldsymbol{\Sigma}^s)^{1/2}\,\boldsymbol{\Sigma}^t\,(\boldsymbol{\Sigma}^s)^{1/2}\big)^{1/2}\Big). \tag{6}$$

Furthermore, provided Σ^s is non-singular, the corresponding optimal transport map is uniquely determined and takes the affine form (Delon & Desolneux, 2020):

$$T(\mathbf{x}_s) \;=\; \mathbf{m}^t \;+\; (\boldsymbol{\Sigma}^s)^{-1/2}\big(\boldsymbol{\Sigma}^s\,\boldsymbol{\Sigma}^t\big)^{1/2}(\mathbf{x}_s - \mathbf{m}^s). \tag{7}$$

Delon & Desolneux (2020) extended this to the setting where both μ_s and μ_t are Gaussian mixtures, namely $\mu_s^{GM} = \sum_{k_1=1}^{K_s} \alpha_{sk_1}\, P_{Sk_1}$ with $P_{Sk_1} = \mathcal{N}(\mathbf{m}_{k_1}^s, \boldsymbol{\Sigma}_{k_1}^s)$ and $\mu_t^{GM} = \sum_{k_2=1}^{K_t} \alpha_{tk_2}\, P_{Tk_2}$ with $P_{Tk_2} = \mathcal{N}(\mathbf{m}_{k_2}^t, \boldsymbol{\Sigma}_{k_2}^t)$, with mixing weights $\boldsymbol{\alpha}_s \in \Delta^{K_s}$ and $\boldsymbol{\alpha}_t \in \Delta^{K_t}$, where K_s and K_t denote the number of Gaussian components in the source and target mixtures respectively, and $\Delta^K = \{\alpha \in \mathbb{R}_+^K \mid \sum_{k=1}^K \alpha_k = 1\}$ is the probability simplex. A key theoretical observation therein is that the optimal coupling between two Gaussian mixtures under the problem in equation 1 is generally not a Gaussian mixture itself. To this end, they proposed restricting the feasible set $\Pi(\mu_s^{GM}, \mu_t^{GM})$ to couplings in GMM_{2d}, the space of Gaussian mixtures on $\mathbb{R}^{2d}$. Under this restriction, the infimum is attained, and the optimal coupling necessarily factorizes as $\pi^*(\mathbf{x}_s, \mathbf{x}_t) = \sum_{k_1,k_2} \pi_{k_1,k_2}\, P_{Sk_1}(\mathbf{x}_s)\, P_{Tk_2}(\mathbf{x}_t)$. Furthermore, Delon & Desolneux (2020) show that this continuous infinite-dimensional problem *reduces exactly to* a finite discrete OT problem between the Gaussian components:

$$\boldsymbol{\pi}^* \;=\; \underset{\boldsymbol{\pi}\in\Pi(\boldsymbol{\alpha}_s,\boldsymbol{\alpha}_t)}{\arg\min} \sum_{k_1=1}^{K_s}\sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, \mathcal{W}_2^2(P_{Sk_1}, P_{Tk_2}), \tag{8}$$

with $\Pi(\alpha_s, \alpha_t) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{K_s\times K_t} \mid \boldsymbol{\pi}\mathbf{1}_{K_t} = \boldsymbol{\alpha_s},\ \boldsymbol{\pi}^\top \mathbf{1}_{K_s} = \boldsymbol{\alpha}_t\}$, and $\mathcal{W}_2^2$ given in closed form by equation 6. This reduction collapses the problem from an infinite-dimensional search to a discrete OT of size $K_s \times K_t$, far smaller than the $n_s \times n_t$ empirical problem equation 2, while retaining the full Gaussian mixture structure of the marginals and producing a coupling that is analytically tractable by construction (Delon & Desolneux, 2020; Montesuma et al., 2025).

Remark 3.2. *Since equation 8 is a discrete OT problem of size $K_s \times K_t$, entropic regularization applies directly, yielding:*

$$\boldsymbol{\pi}_\varepsilon^* \;=\; \underset{\boldsymbol{\pi}\in\Pi(\boldsymbol{\alpha}_s,\boldsymbol{\alpha}_t)}{\arg\min} \sum_{k_1=1}^{K_s}\sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, \mathcal{W}_2^2(P_{Sk_1}, P_{Tk_2}) \;+\; \varepsilon \sum_{k_1=1}^{K_s}\sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\big(\log \pi_{k_1,k_2} - 1\big), \tag{9}$$

solvable via Sinkhorn's algorithm on the $K_s \times K_t$ cost matrix with entries $\mathcal{W}_2^2(P_{Sk_1}, P_{Tk_2})$.

3.3 Transfer Operators and Spectral Embedding

Spectral embedding is a classical technique that represents the vertices of a graph in a low-dimensional Euclidean space where geometric proximity reflects the connectivity structure of the graph. In its discrete formulation, given a finite weighted undirected graph $\mathcal{G} = (V, E, \mathcal{A})$ with adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and degree matrix $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1}_n)$, one constructs the random-walk normalized Laplacian $\mathbf{L}_{\mathrm{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ and takes its eigenvectors associated with the smallest eigenvalues as coordinates, which encode the cluster structure of the graph. For a complete treatment of the discrete formulation and its connections with normalized cuts and graph Laplacians, we refer the reader to (Aggarwal & Reddy, 2014). Since our framework relies on a continuous formulation, we adopt the equivalent perspective of *transfer operators*, which extend spectral embedding to random-walk processes on continuous state spaces and provide a principled way to study their long-time behavior (Klus & Trower, 2024).

Definition 3.3 (Koopman Operator)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an observable and let $p(x, y)$ be the transition kernel of a Markov process on $\mathbb{R}^d$. The Koopman operator $\mathcal{K}$ is the linear operator $\mathcal{K} : L^2(\mathbb{R}^d) \longrightarrow L^2(\mathbb{R}^d)$ defined by*

$$(\mathcal{K}f)(x) = \int_{\mathbb{R}^d} p(x,y)\, f(y)\, dy. \tag{10}$$

The operator $\mathcal{K}$ describes the evolution of observables under the underlying stochastic dynamics, equivalently along the random walk on the associated graph $\mathcal{G}$.

Lemma 3.4 (Spectral properties under reversibility)**.** *Let $p(x,y)$ be the transition kernel of a Markov process on $\mathbb{R}^d$ that is reversible with respect to an invariant density μ, i.e., $\mu(x)\,p(x,y) = \mu(y)\,p(y,x)$ for almost every $(x,y) \in \mathbb{R}^d \times \mathbb{R}^d$. Then the Koopman operator $\mathcal{K} : L^2(\mathbb{R}^d,\mu) \to L^2(\mathbb{R}^d,\mu)$ defined in equation 10 is self-adjoint with real-valued spectrum contained in $[-1,1]$ (Klus & Trower, 2024, Lemma 2.17 and Proposition A.1). The eigenvalue $\lambda = 1$ is always attained with the constant function $\mathbf{1}$ as eigenfunction, and its multiplicity equals the number of connected components of the graph induced by p (Levin & Peres, 2017).*

Under Lemma 3.4, the eigenfunctions of $\mathcal{K}$ encode the long-time behavior of the random walk: subdominant eigenvalues $\lambda_\ell \approx 1$ correspond to eigenfunctions f_ℓ that are nearly invariant under the dynamics, $\mathcal{K}f_\ell \approx f_\ell$. These near-invariant eigenfunctions identify <u>*metastable sets*</u> of the walk, *i.e.,* regions of the state space within which a random walker remains for long periods before transitioning out, which are the continuous analogue of clusters in the discrete case. The number of dominant eigenvalues close to 1, separated from the rest of the spectrum by a clear spectral gap, indicates the number of such metastable sets, and their associated eigenfunctions serve as the spectral embedding of the underlying process (Klus & Trower, 2024).

4 DoDyn

In this section, we present our method, **Domain Dynamics (DoDyn)**, for learning representations of samples that are invariant to distributional shifts via Koopman operator and Gaussian mixture OT.

4.1 Lifting Empirical Distributions to Gaussian Mixtures

Rather than working with the discrete empirical measures $\mu_s = \frac{1}{n_s}\sum_i \delta_{\mathbf{x}_s^{k_1}}$ and $\mu_t = \frac{1}{n_t}\sum_j \delta_{\mathbf{x}_t^{k_2}}$, which place all their mass at isolated points, we lift each sample to a Gaussian density centered at the sample location. Formally, let $\mathbf{X}_s = (\mathbf{x}_s^1, \ldots, \mathbf{x}_s^{n_s}) \in \mathbb{R}^{n_s \times d}$ and $\mathbf{X}_t = (\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{n_t}) \in \mathbb{R}^{n_t \times d}$ denote the source and target empirical samples. Each sample $\mathbf{x}_s^{k_1}$ is associated with a Gaussian component $P_{Sk_1} = \mathcal{N}(\mathbf{x}_s^{k_1}, \mathbf{\Sigma})$ (resp. P_{Tk_2} for $\mathbf{x}_t^{k_2}$), where $\mathbf{\Sigma}$ is a diagonal covariance initialized as a constant. We take $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_d$ shared across all components. With a common covariance the Bures term cancels and the cost in equation 6 reduces to $\|\mathbf{x}_s^{k_1} - \mathbf{x}_t^{k_2}\|_2^2$, independent of σ^2. We also standardize both domains coordinate-wise beforehand so that disparate feature scales do not bias this Euclidean cost. We report numerical evidence about the choice of using a shared isotropic covariance rather than learned anisotropic GMMs in Section 6. Intuitively, every sample spreads a fraction of its mass over its local neighborhood, producing a smooth continuous density rather than a sum of Dirac masses. The resulting source and target densities are the Gaussian mixtures $P_s(x_s) = \sum_{k_1=1}^{K_s} \alpha_{sk_1} P_{Sk_1}(x_s)$ and $P_t(x_t) = \sum_{k_2=1}^{K_t} \alpha_{tk_2} P_{Tk_2}(x_t)$, where $K_s = n_s$ and $K_t = n_t$ in our setting. Given this representation, we then solve the entropic GMM-OT problem by solving equation 9 between the source and target mixtures. We then define the *induced continuous coupling* as:

$$\pi_\varepsilon^*(x_s, x_t) \overset{def.}{=} \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x_s)\, P_{Tk_2}(x_t). \tag{11}$$

Lemma 4.1 (Marginals of the induced continuous coupling)**.** *The marginals of $\pi_\varepsilon^*(x_s, x_t)$ defined in equation 11 satisfy*

$$\int_{\mathbb{R}^d} \pi_\varepsilon^*(x_s, x_t)\, dx_t = P_s(x_s), \qquad \int_{\mathbb{R}^d} \pi_\varepsilon^*(x_t, x_s)\, dx_t = P_t(x_s). \tag{12}$$

Proof. See supp. mat. A.1 □

Lemma 4.1 confirms that $\pi_\varepsilon^*(x_s, x_t)$ is a valid continuous joint distribution whose marginals recover exactly P_s and P_t, which guarantees that the lifted GMM representations are preserved through the coupling, a property we now exploit to build a graph encoding the geometry of both domains.

4.2 Graph Construction and Koopman Eigenfunctions

Adapting the graph formulation of (Sad Saoud et al., 2026) to the continuous setting, we use the optimal transport plan π_ε^* to construct a weighted undirected graph $\mathcal{G} = ((\mathbf{X_s}, \mathbf{X_t}), E, \mathcal{A})$, where $E \subseteq \mathbb{R}^d \times \mathbb{R}^d$ is a set of edges and $\mathcal{A} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ is a weight function constructed by symmetrizing the continuous coupling equation 11. The adjacency and degree functions are given by

$$\mathcal{A}(x_s, x_t) = \frac{1}{2}[\pi_\varepsilon^*(x_s, x_t) + \pi_\varepsilon^*(x_t, x_s)], \tag{13a}$$

$$d(x_s) = \int_{\mathbb{R}^d} \mathcal{A}(x_s, x_t)\, dx_t = \frac{1}{2}\Big[\underbrace{\int_{\mathbb{R}^d} \pi_\varepsilon^*(x_s, x_t)\, dx_t}_{P_s(x_s)} + \underbrace{\int_{\mathbb{R}^d} \pi_\varepsilon^*(x_t, x_s)\, dx_t}_{P_t(x_s)}\Big], \tag{13b}$$

where the marginal identities follow from Lemma 4.1. This graph induces a natural Markov chain on $\mathbb{R}^d$, with the probability of transitioning from x_s to x_t in one step is given by the row-stochastic transition kernel

$$p(x_s, x_t) \;=\; \frac{\mathcal{A}(x_s, x_t)}{d(x_s)} \;\overset{(a)}{=}\; \frac{\pi_\varepsilon^*(x_s, x_t) + \pi_\varepsilon^*(x_t, x_s)}{P_s(x_s) + P_t(x_s)}, \tag{14}$$

where equality (a) follows from the substitution of equation 13a and equation 13b, and cancellation of the factor $\frac{1}{2}$. Since every Gaussian component P_{Sk_1} and P_{Tk_2} is strictly positive on $\mathbb{R}^d$, the coupling $\pi_\varepsilon^*(x, y) > 0$ everywhere, making the induced graph connected and implying by Lemma 3.4 that $\lambda = 1$ is a simple eigenvalue of the associated Koopman operator. Since f is defined on the full ambient space, we write the eigenproblem using generic variables $x, y \in \mathbb{R}^d$ in place of x_s, x_t. Substituting equation 14 into equation 10, the eigenproblem $\mathcal{K}f = f$ takes the explicit form

$$\mathcal{K}f(x) = \int_{\mathbb{R}^d} f(y)\, \frac{\mathcal{A}(x, y)}{d(x)}\, dy \tag{15a}$$

$$= \frac{1}{P_s(x) + P_t(x)} \int_{\mathbb{R}^d} f(y) \Big[\pi_\varepsilon^*(x, y) + \pi_\varepsilon^*(y, x)\Big]\, dy \;=\; f(x). \tag{15b}$$

Proposition 4.2 (Expansion using component expectations)**.** *Let $\theta_{Sk_1} := \int_{\mathbb{R}^d} f(y)\, P_{Sk_1}(y)\, dy$ and $\theta_{Tk_2} := \int_{\mathbb{R}^d} f(y)\, P_{Tk_2}(y)\, dy$ be the component expectations of f under the source and target Gaussian components respectively, and $g_{Sk_1}(x) := \frac{P_{Sk_1}(x)}{P_s(x)+P_t(x)}$, $g_{Tk_2}(x) := \frac{P_{Tk_2}(x)}{P_s(x)+P_t(x)}$. Then equation 15b admits the following closed-form fixed-point equation:*

$$f(x) = \sum_{k_1=1}^{K_s} g_{Sk_1}(x) \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, \theta_{Tk_2} + \sum_{k_2=1}^{K_t} g_{Tk_2}(x) \sum_{k_1=1}^{K_s} \pi_{k_1,k_2}\, \theta_{Sk_1}. \tag{16}$$

Proposition 4.2 reveals that the infinite-dimensional eigenproblem in equation 15 admits an exact finite-dimensional reduction: rather than searching for f in all of $L^2(\mathbb{R}^d)$, it suffices to determine the $K_s + K_t$ component expectations $\{\boldsymbol{\theta}_{Sk_1}\}_{k_1=1}^{K_s}$ and $\{\boldsymbol{\theta}_{Tk_2}\}_{k_2=1}^{K_t}$, from which f is reconstructed pointwise via equation 16. This is possible because the GMM structure of the coupling π_ε^* confines the eigenfunction to the span of the normalized component weights $\{g_{Sk_1}, g_{Tk_2}\}$, which form a natural basis adapted to the joint geometry of the source and target distributions.

From fixed-point equation of f to fixed-point equation of $\boldsymbol{\theta}$. While Proposition 4.2 expresses f in closed form, the component expectations θ_S and θ_T are themselves defined as integrals of f. We now show that this circularity resolves into an explicit finite-dimensional fixed-point system. For $i \in \{1, \dots, K_s\}$ (resp. $j \in \{1, \dots, K_t\}$), by substituting equation 16 into $\theta_{Si} = \int_{\mathbb{R}^d} P_{Si}(x) f(x)\, dx$ (resp. $\theta_{Tj} = \int_{\mathbb{R}^d} P_{Tk}(x) f(x)\, dx$) and collecting terms, we introduce the *overlap matrices* between source-source $\mathbf{M_{SS}}$, source-target $\mathbf{M_{ST}}$, target-source $\mathbf{M_{TS}}$ and target-target $\mathbf{M_{TT}}$, respectively (see supp. mat. A.3)

$$M_{SS}[i, k_1] := \int_{\mathbb{R}^d} \frac{P_{Si}(x) P_{Sk_1}(x)}{P_s(x) + P_t(x)}\, dx, \quad M_{ST}[i, k_2] := \int_{\mathbb{R}^d} \frac{P_{Si}(x) P_{Tk_2}(x)}{P_s(x) + P_t(x)}\, dx, \tag{17a}$$

$$M_{TS}[j,k_1] := \int_{\mathbb{R}^d} \frac{P_{Tj}(x)P_{Sk_1}(x)}{P_s(x)+P_t(x)}\,dx, \quad M_{TT}[j,k_2] := \int_{\mathbb{R}^d} \frac{P_{Tj}(x)P_{Tk_2}(x)}{P_s(x)+P_t(x)}\,dx. \tag{17b}$$

In terms of these matrices, the fixed-point equations for θ_{Si} and θ_{Tj} take the compact form (see supp. mat. A.3 for the detailed derivation)

$$\theta_{Si} = \sum_{k_1} M_{SS}[i,k_1] \sum_{k_2} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2} M_{ST}[i,k_2] \sum_{k_1} \pi_{k_1,k_2}\theta_{Sk_1}, \tag{18a}$$

$$\theta_{Tj} = \sum_{k_1} M_{TS}[j,k_1] \sum_{k_2} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2} M_{TT}[j,k_2] \sum_{k_1} \pi_{k_1,k_2}\theta_{Sk_1}. \tag{18b}$$

Finite-dimensional fixed-point system. Collecting Eqs. 18a–18b in vector form gives

$$\begin{aligned} \boldsymbol{\theta}_S &= \mathbf{M}_{SS}\,\boldsymbol{\pi}\,\boldsymbol{\theta}_T + \mathbf{M}_{ST}\,\boldsymbol{\pi}^\top\boldsymbol{\theta}_S, \\ \boldsymbol{\theta}_T &= \mathbf{M}_{TS}\,\boldsymbol{\pi}\,\boldsymbol{\theta}_T + \mathbf{M}_{TT}\,\boldsymbol{\pi}^\top\boldsymbol{\theta}_S, \end{aligned} \quad \boldsymbol{\Theta} = [\boldsymbol{\theta}_S^\top, \boldsymbol{\theta}_T^\top]^\top, \quad \boldsymbol{\Theta} = \begin{bmatrix} \mathbf{M}_{ST}\boldsymbol{\pi}^\top & \mathbf{M}_{SS}\boldsymbol{\pi} \\ \mathbf{M}_{TT}\boldsymbol{\pi}^\top & \mathbf{M}_{TS}\boldsymbol{\pi} \end{bmatrix} \boldsymbol{\Theta}. \tag{19}$$

Thus Θ is an eigenvector of the block operator in equation 19 associated with eigenvalue 1, and its entries encode the domain-invariant spectral coordinates of the source and target distributions jointly. Once θ_S and θ_T are known, the Koopman eigenfunction is reconstructed pointwise as

$$f(x) = \sum_{k_1=1}^{K_s} g_{Sk_1}(x)\,(\pi\theta_T)_{k_1} + \sum_{k_2=1}^{K_t} g_{Tk_2}(x)\,(\pi^\top\theta_S)_{k_2}. \tag{20}$$

4.3 Learning Procedure

Given source and target samples lifted to Gaussian mixtures as described in Section 4.1, we solve the entropic GMM-OT problem (equation 9) via Sinkhorn to obtain $\boldsymbol{\pi}^*_{\boldsymbol{\varepsilon}}$, and initialize $\boldsymbol{\theta_S} \in \mathbb{R}^{n_s\times k}$ and $\boldsymbol{\theta_T} \in \mathbb{R}^{n_t\times k}$ as learnable parameters, where k is the embeddings dimension. The overlap matrices $\widehat{\mathbf{M}}_{\mathbf{SS}}, \widehat{\mathbf{M}}_{\mathbf{ST}}, \widehat{\mathbf{M}}_{\mathbf{TS}}, \widehat{\mathbf{M}}_{\mathbf{TT}}$ are estimated once via Monte Carlo (see proposition A.1) and kept fixed throughout training. The Koopman loss is derived directly from the fixed-point system (equation 19), penalizing deviations of the learned θ_S, θ_T from the fixed-point equations and driving them toward the Koopman eigenvector Θ:

$$\mathcal{L}_{\text{Koop}} \;=\; \|\boldsymbol{\theta_S} - \widehat{\boldsymbol{M}}_{\boldsymbol{SS}}\boldsymbol{\pi}^*\boldsymbol{\theta_T} - \widehat{\boldsymbol{M}}_{\boldsymbol{ST}}\boldsymbol{\pi}^{*\top}\boldsymbol{\theta_S}\|_F^2 \;+\; \|\boldsymbol{\theta_T} - \widehat{\boldsymbol{M}}_{\boldsymbol{TS}}\boldsymbol{\pi}^*\boldsymbol{\theta_T} - \widehat{\boldsymbol{M}}_{\boldsymbol{TT}}\boldsymbol{\pi}^{*\top}\boldsymbol{\theta_S}\|_F^2. \tag{21}$$

To enforce class-discriminative structure in the learned spectral coordinates, we introduce a classifier $\mathcal{C}_\phi : \mathbb{R}^k \to \mathbb{R}^{N_c}$, parameterized as a multi-layer perceptron $\mathcal{C}_\phi = \mathcal{C}_{\phi_L} \circ \cdots \circ \mathcal{C}_{\phi_1}$, where N_c is the number of classes. At each iteration, the eigenfunction at the each source sample $f(\mathbf{x}_s^i)$ is reconstructed via equation 20, producing spectral embeddings $f(\mathbf{X_s})$ that are fed forward to $\mathcal{C}_\phi$. The full objective function to be minimized is

$$\mathcal{L} \;=\; \mathcal{L}_{\text{Koop}} \;+\; \eta\,\mathcal{L}_{\text{CE}}(\mathcal{C}_\phi(f(\mathbf{X_s})), \mathbf{Y_s}), \tag{22}$$

where $\eta > 0$ balances the two terms. The classification gradient $\frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{\phi}}$ back-propagates through $f(\mathbf{X_s})$ into $\boldsymbol{\theta_S}$ and $\boldsymbol{\theta_T}$, which regularizes the spectral coordinates to prevent trivial or collapsed solutions in practice. It may seem more natural to simply solve the fixed-point system in equation 19 for its eigenfunctions directly. But two things make this the wrong target. First, the system is minimized exactly by the constant function at $\lambda = 1$ (Lemma 3.4), together with the collapsed solution $\boldsymbol{\Theta} = 0$. Both are blind to labels, and the subdominant modes are no

Algorithm 1 DoDyn: Domain Dynamics

Require: X_s, Y_s, X_t, Σ, ε, N, dimension k, η, $\text{It}_{\max}$
Ensure: θ_S, θ_T, $\mathcal{C}_\phi$

```
Lift: P_{Sk_1} = N(x_s^{k_1}, σ²I), P_{Tk_2} = N(x_t^{k_2}, σ²I)
π*_ε ← Solve equation 9 via Sinkhorn
Estimate Ŵ_SS, Ŵ_ST, Ŵ_TS, Ŵ_TT via Monte Carlo
Compute g_{Sk_1}(x), g_{Tk_2}(x) for x ∈ X_s ∪ X_t
Initialize θ_S, θ_T, and C_φ
for it = 1, ..., It_max do
    f ← Eq. 20
    L ← L_Koop + η L_CE(C_φ(f(X_s)), Y_s)
    θ_S, θ_T, φ ← θ_S, θ_T, φ − lr · ∇L_{θ_S,θ_T,φ}
end for
return θ_S, θ_T, C_φ
```

better in that respect, since nothing forces the spectral cuts to align with the classes without supervision. A direct eigensolver would return precisely these uninformative modes, while just minimizing $\mathcal{L}_{\text{Koop}}$ collapses to $\boldsymbol{\Theta} = 0$. Second, the transport plan $\boldsymbol{\pi}_{\boldsymbol{\varepsilon}}^*$ is itself estimated once, so the exact eigenfunctions inherit the estimation error of the plan and are not a target we should follow. Instead, among the functions that nearly satisfy the fixed-point equations, we want to select those that also separate the classes, and this is the role of the classifier. The full parameter set $\{\boldsymbol{\theta_S}, \boldsymbol{\theta_T}, \boldsymbol{\phi}\}$ is optimized jointly via gradient descent (Rumelhart et al., 1986; Kingma & Ba, 2014). At inference, *no fine-tuning is required*. For any new test point $\mathbf{x}$, its eigenfunction $f(\mathbf{x})$ is reconstructed via equation 20 using the learned $\boldsymbol{\theta_S}, \boldsymbol{\theta_T}$, and classification is performed by $\mathcal{C}_\phi(f(\mathbf{x}))$ directly.

5 Experiments

In this section, we present an evaluation of **DoDyn** alongside state-of-the-art domain adaptation methods on domain adaptation tasks.

Benchmarks. We evaluate **DoDyn** on four benchmarks spanning audio, signals and visual modalities. *Music Genre Recognition (MGR)* (Tzanetakis & Cook, 2002) and *Music and Speech Detection (MSD)* (Tzanetakis, 2001) are audio classification datasets where domain shifts arise from five environmental noise conditions. *Compressed Sensing Reflectometry (CS-RT)* (Slimani et al., 2024) is a cable fault diagnosis benchmark based on time domain reflectometry and compressive sensing, where shifts arise from different signal compression factors and cable characteristics. *CMU Pose, Illumination and Expression (CMU-PIE)* (Sim et al., 2002) is a face recognition benchmark with four camera viewpoints as domains. Since we are particularly interested in out-of-sample generalization, a portion of each target domain is withheld entirely from the transport plan computation and used exclusively for OOS evaluation. for CS-RT, an out-of-sample dataset of the same size as the training set is generated separately.

Baselines. We compare **DoDyn** against the following methods: **Source-only** trains a classifier on source data without any adaptation; $\mathbf{OTDA_{Sink}}$ and $\mathbf{OTDA_{Lap}}$ (Courty et al., 2017b) are shallow OT methods; **DeepJDOT** (Damodaran et al., 2018), **DANN** (Ganin et al., 2016), **(Seguy et al., 2018)** for which we implement their Monge map estimation procedure (Algorithm 1+2 with entropic regularization); and **NeuralSPOT**, a baseline we propose that solves a discrete OT problem and approximates a spectral embedding through a neural network (see supp. mat. A.5 for details). This comparison covers shallow OT, deep OT, adversarial, and neural OT-based adaptation methods. Note that since not all methods provide out-of-sample mapping, those that do not are evaluated only on the target domain. On the experimental setup: all experiments were implemented in PyTorch 2.5.1 (CUDA 11.8, cuDNN 9.1.0) and run on a single NVIDIA H200 GPU (141 GB HBM3e) hosted on a server with an AMD EPYC 9555 64-core CPU.

Main Results. Table 1 reports results on the MSD benchmark. All adaptation methods improve over the source-only baseline. Shallow OT methods perform well, while DeepJDOT, (Seguy et al., 2018), and DANN perform comparably, suggesting that deeper feature extraction or adversarial training alone does not bring a consistent advantage on this benchmark. DoDyn and NeuralSPOT achieve the two highest target averages, consistently outperforming all baselines. The real difference becomes apparent at out-of-sample generalization, where DoDyn outperforms NeuralSPOT, This gap is particularly telling that even both methods share the same spectral embedding principle, the continuous Koopman formulation yields more transferable spectral coordinates than the neural approximation. The results on MGR benchmark are reported in Table 2, as discussed in (Montesuma & Mboula, 2021; Sad Saoud et al., 2026), this is a harder setting where transport plans are more sensitive to mismatches, especially when no class-aware regularization is used. Shallow OT methods surprisingly underperform the source-only baseline, confirming that such methods degrades under increased class complexity. DANN performs competitively while DeepJDOT lags behind. The gap between DoDyn and NeuralSPOT is more obvious than on MSD, as with more classes, the discrete spectral approximation of NeuralSPOT is more exposed to transport mismatches, whereas DoDyn's continuous Koopman formulation retains a more robust spectral structure across the increased class complexity. The CS-RT benchmark (Tables 3 and 4) evaluates adaptation on real reflectometric measurements where domain shifts reflect genuine physical differences in compression and cable characteristics. $OTDA_{sink}$ and (Seguy et al., 2018) fall below the source-only baseline, while $OTDA_{Lap}$ and DeepJDOT improves the performance on the

Table 1: Classification results on the MSD benchmark. Methods with OOS evaluation report both accuracy on target domain ($\mathcal{A}_{(T)} \uparrow$) and evaluation on out-of-samples data ($\mathcal{A}_{(OOS)} \uparrow$), with values reported as mean accuracy ± standard deviation, averaged over multiple seeds. Colored values in the average row indicate the performance difference *w.r.t* the source-only baseline.

Domains	Source-only	OTDA$_{\text{Sink}}$	OTDA$_{\text{Lap}}$	DeepJDOT	DANN		Seguy et al. (2018)		NeuralSPOT		DoDyn	
	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$	$\mathcal{A}_{(T)} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$
$N \to B$	$79.69_{\pm 0.78}$	$91.02_{\pm 0.39}$	$91.41_{\pm 0.00}$	$90.36_{\pm 2.05}$	$86.20_{\pm 4.10}$	$89.33_{\pm 3.09}$	$90.62_{\pm 0.64}$	$94.79_{\pm 2.05}$	96.09	$99.00_{\pm 0.00}$	$95.70_{\pm 0.50}$	$97.50_{\pm 0.50}$
$N \to D$	$76.17_{\pm 0.39}$	$92.97_{\pm 0.78}$	$90.23_{\pm 0.39}$	$82.03_{\pm 1.28}$	$73.44_{\pm 12.30}$	$76.00_{\pm 16.67}$	$90.62_{\pm 1.91}$	$89.32_{\pm 4.70}$	90.62	$84.50_{\pm 1.50}$	$90.23_{\pm 0.39}$	$84.50_{\pm 1.50}$
$N \to F_{16}$	$76.17_{\pm 1.17}$	$88.67_{\pm 0.39}$	$89.06_{\pm 0.78}$	$89.06_{\pm 2.78}$	$85.94_{\pm 4.98}$	$90.33_{\pm 4.92}$	$89.06_{\pm 0.00}$	$91.15_{\pm 5.35}$	90.62	$70.00_{\pm 21.00}$	$91.80_{\pm 1.17}$	$90.50_{\pm 0.50}$
$N \to F$	$86.33_{\pm 1.17}$	$95.70_{\pm 0.39}$	$96.09_{\pm 0.00}$	$92.19_{\pm 1.91}$	$90.10_{\pm 1.95}$	$89.33_{\pm 5.44}$	$95.05_{\pm 0.37}$	$96.09_{\pm 2.30}$	97.66	$94.00_{\pm 1.00}$	$98.44_{\pm 0.78}$	$96.50_{\pm 2.50}$
$B \to N$	$90.62_{\pm 0.00}$	$89.06_{\pm 0.00}$	$93.36_{\pm 0.39}$	$94.53_{\pm 3.98}$	$87.76_{\pm 0.37}$	$83.00_{\pm 6.68}$	$84.38_{\pm 2.21}$	$86.98_{\pm 2.24}$	96.09	$97.00_{\pm 1.00}$	$96.09_{\pm 0.78}$	$96.00_{\pm 2.00}$
$B \to D$	$78.91_{\pm 0.00}$	$91.02_{\pm 0.39}$	$88.28_{\pm 0.00}$	$63.28_{\pm 18.78}$	$85.16_{\pm 4.60}$	$86.67_{\pm 4.64}$	$85.94_{\pm 2.30}$	$80.99_{\pm 7.69}$	97.66	$96.50_{\pm 0.50}$	$93.36_{\pm 1.17}$	$96.00_{\pm 3.00}$
$B \to F_{16}$	$94.14_{\pm 1.95}$	$93.75_{\pm 0.00}$	$93.36_{\pm 0.39}$	$93.75_{\pm 0.00}$	$87.24_{\pm 4.70}$	$88.33_{\pm 4.78}$	$89.32_{\pm 0.37}$	$85.68_{\pm 2.24}$	93.75	$94.50_{\pm 0.50}$	$94.53_{\pm 0.00}$	$94.00_{\pm 1.00}$
$B \to F$	$76.95_{\pm 0.39}$	$94.14_{\pm 0.39}$	$94.53_{\pm 0.78}$	$93.75_{\pm 1.10}$	$86.72_{\pm 0.64}$	$84.33_{\pm 2.36}$	$92.97_{\pm 1.28}$	$89.06_{\pm 7.76}$	96.88	$91.00_{\pm 4.00}$	$96.88_{\pm 0.78}$	$96.00_{\pm 1.00}$
$D \to N$	$90.23_{\pm 0.39}$	$92.19_{\pm 0.00}$	$91.80_{\pm 0.39}$	$94.53_{\pm 1.91}$	$86.98_{\pm 3.90}$	$85.67_{\pm 3.09}$	$90.89_{\pm 1.84}$	$89.58_{\pm 1.95}$	92.19	$87.50_{\pm 1.50}$	$96.09_{\pm 0.78}$	$96.09_{\pm 0.00}$
$D \to B$	$77.73_{\pm 1.17}$	$92.97_{\pm 0.00}$	$94.14_{\pm 0.39}$	$94.79_{\pm 0.97}$	$97.66_{\pm 0.00}$	$98.33_{\pm 1.25}$	$84.64_{\pm 4.53}$	$86.98_{\pm 4.87}$	98.44	$92.50_{\pm 0.50}$	$96.09_{\pm 0.00}$	$92.00_{\pm 0.00}$
$D \to F_{16}$	$95.31_{\pm 0.78}$	$89.45_{\pm 0.39}$	$84.77_{\pm 0.39}$	$96.35_{\pm 0.74}$	$95.05_{\pm 0.37}$	$96.67_{\pm 3.40}$	$87.76_{\pm 0.37}$	$86.72_{\pm 6.15}$	94.53	$96.00_{\pm 0.00}$	$92.58_{\pm 1.17}$	$95.00_{\pm 0.00}$
$D \to F$	$76.56_{\pm 0.78}$	$91.02_{\pm 0.39}$	$88.67_{\pm 0.39}$	$96.88_{\pm 0.64}$	$96.88_{\pm 0.64}$	$98.67_{\pm 1.25}$	$86.20_{\pm 0.37}$	$86.72_{\pm 6.15}$	93.75	$94.50_{\pm 0.50}$	$94.92_{\pm 0.39}$	$95.00_{\pm 1.00}$
$F_{16} \to N$	$91.80_{\pm 0.39}$	$91.80_{\pm 0.39}$	$90.62_{\pm 0.78}$	$97.14_{\pm 0.97}$	$90.89_{\pm 2.58}$	$90.00_{\pm 4.32}$	$92.45_{\pm 1.61}$	$89.84_{\pm 3.38}$	96.09	$87.50_{\pm 1.50}$	$95.31_{\pm 0.78}$	$95.50_{\pm 0.50}$
$F_{16} \to B$	$92.97_{\pm 0.00}$	$93.36_{\pm 0.39}$	$96.09_{\pm 0.00}$	$96.09_{\pm 1.10}$	$96.09_{\pm 1.28}$	$97.67_{\pm 3.30}$	$91.41_{\pm 0.00}$	$93.23_{\pm 2.95}$	93.75	$97.50_{\pm 1.50}$	$94.14_{\pm 1.17}$	$97.00_{\pm 2.00}$
$F_{16} \to D$	$94.53_{\pm 0.00}$	$87.89_{\pm 0.39}$	$87.50_{\pm 0.78}$	$80.21_{\pm 21.37}$	$95.57_{\pm 0.37}$	$94.67_{\pm 4.19}$	$80.73_{\pm 3.51}$	$74.22_{\pm 8.14}$	94.53	$92.00_{\pm 1.00}$	$90.23_{\pm 1.17}$	$93.00_{\pm 4.00}$
$F_{16} \to F$	$78.12_{\pm 0.00}$	$94.92_{\pm 0.39}$	$94.14_{\pm 0.39}$	$96.35_{\pm 0.37}$	$96.88_{\pm 0.00}$	$98.33_{\pm 1.70}$	$93.23_{\pm 2.24}$	$94.27_{\pm 2.66}$	96.09	$95.00_{\pm 0.00}$	$96.48_{\pm 0.39}$	$97.00_{\pm 0.00}$
$F \to N$	$92.19_{\pm 0.00}$	$94.14_{\pm 0.39}$	$94.92_{\pm 0.39}$	$96.35_{\pm 1.95}$	$94.53_{\pm 2.21}$	$93.33_{\pm 2.62}$	$92.45_{\pm 0.97}$	$88.80_{\pm 2.42}$	99.22	$96.00_{\pm 3.00}$	$98.05_{\pm 0.39}$	$97.00_{\pm 1.00}$
$F \to B$	$86.72_{\pm 0.00}$	$89.06_{\pm 0.78}$	$87.11_{\pm 0.39}$	$82.03_{\pm 0.00}$	$90.89_{\pm 1.33}$	$94.33_{\pm 1.70}$	$88.54_{\pm 1.95}$	$92.97_{\pm 3.38}$	97.66	$97.50_{\pm 1.50}$	$94.92_{\pm 0.39}$	$96.50_{\pm 2.50}$
$F \to D$	$83.98_{\pm 0.39}$	$87.89_{\pm 1.17}$	$89.84_{\pm 0.00}$	$77.60_{\pm 3.15}$	$79.95_{\pm 5.12}$	$79.67_{\pm 6.60}$	$88.02_{\pm 0.97}$	$88.28_{\pm 3.38}$	93.75	$93.00_{\pm 0.00}$	$92.58_{\pm 0.39}$	$92.00_{\pm 2.00}$
$F \to F_{16}$	$85.94_{\pm 0.00}$	$91.02_{\pm 1.17}$	$88.67_{\pm 0.39}$	$82.29_{\pm 1.33}$	$89.06_{\pm 1.28}$	$88.67_{\pm 1.70}$	$85.94_{\pm 1.28}$	$87.24_{\pm 3.68}$	96.09	$96.00_{\pm 1.00}$	$96.09_{\pm 0.00}$	$97.00_{\pm 1.00}$
Avg.	85.25	91.60 (6.35 ↑)	91.23 (5.98 ↑)	89.48 (4.23 ↑)	89.65 (4.40 ↑)	90.17 (4.92 ↑)	89.01 (3.76 ↑)	88.68 (3.43 ↑)	**94.96** (9.71 ↑)	<u>92.57</u> (7.32 ↑)	<u>94.73</u> (9.48 ↑)	**94.63** (9.38 ↑)

Table 2: Classification results on the MGR benchmark. Methods with OOS evaluation report both accuracy on target domain ($\mathcal{A}_{(T)} \uparrow$) and evaluation on out-of-sample data ($\mathcal{A}_{(OOS)} \uparrow$). Values are reported as mean accuracy ± standard deviation, averaged over independent seeds.

Domains	Source-only	OTDA$_{\text{Sink}}$	OTDA$_{\text{Lap}}$	DeepJDOT	DANN		NeuralSPOT		DoDyn	
	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$	$\mathcal{A}_{(\text{T})} \uparrow$	$\mathcal{A}_{(\text{OOS})} \uparrow$
$N \to B$	$32.11_{\pm 1.15}$	$40.18_{\pm 0.52}$	$25.32_{\pm 0.57}$	$33.71_{\pm 1.56}$	$40.29_{\pm 1.37}$	$38.33_{\pm 1.18}$	$32.06_{\pm 0.31}$	$20.25_{\pm 2.25}$	$45.12_{\pm 1.06}$	$33.17_{\pm 4.13}$
$N \to D$	$31.05_{\pm 0.63}$	$47.49_{\pm 0.15}$	$27.88_{\pm 0.55}$	$40.37_{\pm 1.34}$	$51.00_{\pm 1.60}$	$54.83_{\pm 2.05}$	$45.56_{\pm 4.19}$	$27.75_{\pm 0.25}$	$51.29_{\pm 0.95}$	$41.50_{\pm 1.47}$
$N \to F_{16}$	$33.62_{\pm 0.39}$	$47.54_{\pm 1.06}$	$27.81_{\pm 0.38}$	$39.87_{\pm 1.50}$	$49.42_{\pm 0.82}$	$47.50_{\pm 0.41}$	$38.50_{\pm 5.75}$	$30.50_{\pm 5.00}$	$54.12_{\pm 0.41}$	$42.17_{\pm 3.01}$
$N \to F$	$49.83_{\pm 0.86}$	$62.49_{\pm 1.36}$	$32.40_{\pm 0.79}$	$52.54_{\pm 0.31}$	$60.75_{\pm 2.81}$	$58.17_{\pm 6.41}$	$63.19_{\pm 4.44}$	$35.75_{\pm 0.25}$	$67.87_{\pm 0.31}$	$51.00_{\pm 1.22}$
$B \to N$	$35.39_{\pm 0.34}$	$43.68_{\pm 0.91}$	$22.48_{\pm 0.45}$	$36.29_{\pm 1.50}$	$47.17_{\pm 3.65}$	$49.50_{\pm 4.71}$	$38.56_{\pm 3.31}$	$24.75_{\pm 2.25}$	$45.96_{\pm 0.31}$	$42.17_{\pm 1.43}$
$B \to D$	$39.34_{\pm 1.15}$	$39.09_{\pm 0.37}$	$31.78_{\pm 0.43}$	$36.50_{\pm 2.37}$	$58.25_{\pm 3.23}$	$59.50_{\pm 0.82}$	$52.06_{\pm 4.31}$	$31.75_{\pm 0.25}$	$81.25_{\pm 1.33}$	$68.67_{\pm 1.84}$
$B \to F_{16}$	$52.11_{\pm 0.16}$	$45.83_{\pm 0.22}$	$33.88_{\pm 0.59}$	$43.62_{\pm 3.22}$	$68.92_{\pm 0.46}$	$67.67_{\pm 3.12}$	$66.37_{\pm 1.63}$	$41.50_{\pm 0.50}$	$91.92_{\pm 0.72}$	$82.50_{\pm 2.55}$
$B \to F$	$33.96_{\pm 0.11}$	$37.38_{\pm 1.81}$	$20.96_{\pm 0.37}$	$38.21_{\pm 3.49}$	$61.00_{\pm 3.33}$	$58.50_{\pm 2.86}$	$52.31_{\pm 0.94}$	$26.50_{\pm 0.00}$	$72.00_{\pm 0.35}$	$60.00_{\pm 1.47}$
$D \to N$	$46.33_{\pm 0.43}$	$46.37_{\pm 2.13}$	$32.07_{\pm 1.72}$	$42.67_{\pm 1.93}$	$53.75_{\pm 2.91}$	$57.33_{\pm 2.72}$	$42.56_{\pm 0.94}$	$23.50_{\pm 1.50}$	$54.88_{\pm 0.27}$	$52.50_{\pm 4.30}$
$D \to B$	$41.88_{\pm 1.77}$	$35.36_{\pm 0.96}$	$23.75_{\pm 0.76}$	$32.46_{\pm 1.47}$	$45.04_{\pm 1.49}$	$47.17_{\pm 0.85}$	$49.44_{\pm 2.56}$	$34.50_{\pm 0.50}$	$81.87_{\pm 0.47}$	$62.83_{\pm 3.30}$
$D \to F_{16}$	$55.08_{\pm 0.27}$	$39.98_{\pm 1.65}$	$37.45_{\pm 0.68}$	$41.00_{\pm 4.32}$	$66.21_{\pm 2.36}$	$67.67_{\pm 4.70}$	$69.56_{\pm 1.69}$	$48.25_{\pm 1.75}$	$92.25_{\pm 0.82}$	$83.00_{\pm 1.12}$
$D \to F$	$56.73_{\pm 1.27}$	$47.19_{\pm 0.70}$	$35.48_{\pm 2.09}$	$43.21_{\pm 1.08}$	$70.04_{\pm 0.66}$	$67.67_{\pm 1.70}$	$66.81_{\pm 1.69}$	$37.75_{\pm 2.25}$	$87.04_{\pm 0.58}$	$77.67_{\pm 2.05}$
$F_{16} \to N$	$51.35_{\pm 1.53}$	$50.16_{\pm 0.80}$	$32.16_{\pm 1.51}$	$41.54_{\pm 2.10}$	$56.71_{\pm 1.74}$	$57.67_{\pm 7.03}$	$39.19_{\pm 6.19}$	$27.75_{\pm 0.25}$	$57.79_{\pm 1.20}$	$53.33_{\pm 3.06}$
$F_{16} \to B$	$57.59_{\pm 0.45}$	$48.97_{\pm 0.26}$	$34.81_{\pm 1.07}$	$39.79_{\pm 1.90}$	$69.29_{\pm 1.09}$	$67.33_{\pm 0.94}$	$57.81_{\pm 0.31}$	$40.00_{\pm 1.00}$	$93.00_{\pm 0.31}$	$74.67_{\pm 2.72}$
$F_{16} \to D$	$61.20_{\pm 0.08}$	$43.79_{\pm 1.18}$	$34.00_{\pm 0.98}$	$38.25_{\pm 1.50}$	$65.62_{\pm 1.59}$	$67.50_{\pm 4.71}$	$61.37_{\pm 3.62}$	$46.75_{\pm 1.75}$	$92.71_{\pm 0.72}$	$83.17_{\pm 1.55}$
$F_{16} \to F$	$62.43_{\pm 0.33}$	$44.95_{\pm 1.26}$	$32.49_{\pm 1.00}$	$47.54_{\pm 3.84}$	$71.08_{\pm 1.60}$	$70.17_{\pm 1.89}$	$44.94_{\pm 14.31}$	$32.75_{\pm 5.75}$	$89.21_{\pm 0.12}$	$83.83_{\pm 3.09}$
$F \to N$	$56.13_{\pm 1.85}$	$65.23_{\pm 0.40}$	$36.04_{\pm 1.31}$	$50.42_{\pm 0.78}$	$66.88_{\pm 1.63}$	$65.50_{\pm 3.89}$	$57.31_{\pm 0.81}$	$28.50_{\pm 2.50}$	$68.75_{\pm 0.89}$	$62.17_{\pm 5.54}$
$F \to B$	$42.75_{\pm 0.42}$	$35.08_{\pm 0.44}$	$25.74_{\pm 0.65}$	$36.67_{\pm 2.01}$	$57.92_{\pm 2.86}$	$58.33_{\pm 2.36}$	$48.50_{\pm 6.62}$	$29.50_{\pm 3.50}$	$73.50_{\pm 0.27}$	$53.33_{\pm 2.72}$
$F \to D$	$49.08_{\pm 0.13}$	$47.32_{\pm 0.51}$	$34.52_{\pm 0.91}$	$40.29_{\pm 0.78}$	$62.96_{\pm 0.85}$	$65.17_{\pm 5.66}$	$68.06_{\pm 3.19}$	$43.75_{\pm 1.25}$	$89.67_{\pm 0.48}$	$72.33_{\pm 2.25}$
$F \to F_{16}$	$56.77_{\pm 1.39}$	$50.97_{\pm 1.79}$	$36.86_{\pm 1.03}$	$46.42_{\pm 2.00}$	$71.46_{\pm 1.83}$	$70.83_{\pm 4.37}$	$64.87_{\pm 1.50}$	$40.75_{\pm 0.25}$	$90.58_{\pm 0.77}$	$77.83_{\pm 4.87}$
Avg.	47.24	45.95 (1.29 ↓)	30.89 (16.35 ↓)	41.07 (6.17 ↓)	<u>59.69</u> (12.45 ↑)	<u>59.82</u> (12.58 ↑)	52.95 (5.71 ↑)	33.62 (13.62 ↓)	**74.04** (26.80 ↑)	**62.89** (15.675 ↑)

target. DANN provides consistent gains on this benchmark and NeuralSPOT performs competitively on target data but degrades sharply on OOS generalization, while DoDyn achieves the best performances on both target domain and Out-of-sample generalization. Table 5 reports results on the CMU-PIE benchmark. OTDA$_{\text{Sink}}$ underperforms the source-only baseline on several tasks and DANN provides modest gain. DoDyn achieves the best performance on both target and OOS data by a considerable margin, which confirms that

Table 3: Classification results on the CS-RT benchmark. Each row shows results for a "source → target" case. Colored values in the average row indicate the performance improvement/drop *w.r.t* source-only.

Domains	**Source-only**	**OTDA$_{\text{Sink}}$**	**OTDA$_{\text{Lap}}$**	**DeepJDOT**	**DANN**	**Seguy et al. (2018)**	**NeuralSPOT**	**DoDyn**
	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$
$F_{16} \to F_4$	$43.60_{\pm 0.42}$	$42.41_{\pm 0.00}$	$56.10_{\pm 0.84}$	$43.67_{\pm 2.05}$	$54.67_{\pm 2.62}$	$28.17_{\pm 1.70}$	$72.00_{\pm 0.71}$	$75.50_{\pm 0.41}$
$F_{16} \to F_2$	$43.75_{\pm 0.73}$	$38.54_{\pm 0.56}$	$54.02_{\pm 0.36}$	$37.17_{\pm 7.96}$	$61.33_{\pm 0.85}$	$26.50_{\pm 3.89}$	$57.83_{\pm 1.31}$	$65.67_{\pm 0.85}$
$F_{16} \to Ph$	$48.51_{\pm 2.20}$	$27.83_{\pm 2.10}$	$29.46_{\pm 0.36}$	$39.83_{\pm 3.42}$	$46.67_{\pm 4.92}$	$25.33_{\pm 4.84}$	$24.50_{\pm 0.00}$	$42.00_{\pm 3.08}$
$F_4 \to F_{16}$	$25.89_{\pm 0.00}$	$25.45_{\pm 0.00}$	$68.60_{\pm 0.21}$	$39.83_{\pm 5.54}$	$56.83_{\pm 7.19}$	$25.00_{\pm 8.78}$	$73.17_{\pm 1.55}$	$73.50_{\pm 0.41}$
$F_4 \to F_2$	$29.76_{\pm 0.21}$	$28.72_{\pm 0.76}$	$30.80_{\pm 6.65}$	$39.17_{\pm 4.87}$	$56.33_{\pm 3.79}$	$17.67_{\pm 2.09}$	$52.00_{\pm 4.08}$	$59.50_{\pm 0.71}$
$F_4 \to Ph$	$36.61_{\pm 0.73}$	$18.75_{\pm 0.00}$	$30.36_{\pm 0.36}$	$32.50_{\pm 2.68}$	$33.17_{\pm 3.27}$	$27.00_{\pm 8.20}$	$31.67_{\pm 6.33}$	$38.83_{\pm 5.72}$
$F_2 \to F_{16}$	$26.79_{\pm 0.00}$	$40.18_{\pm 0.00}$	$27.98_{\pm 0.76}$	$37.00_{\pm 8.15}$	$54.50_{\pm 7.76}$	$23.00_{\pm 6.48}$	$58.67_{\pm 0.62}$	$63.17_{\pm 0.94}$
$F_2 \to F_4$	$27.38_{\pm 0.21}$	$25.45_{\pm 0.00}$	$37.05_{\pm 0.00}$	$36.00_{\pm 7.22}$	$52.17_{\pm 6.51}$	$30.00_{\pm 4.71}$	$54.17_{\pm 1.31}$	$56.33_{\pm 0.94}$
$F_2 \to Ph$	$66.52_{\pm 1.89}$	$38.10_{\pm 0.84}$	$72.02_{\pm 0.42}$	$57.50_{\pm 9.90}$	$76.33_{\pm 5.39}$	$62.67_{\pm 3.57}$	$60.33_{\pm 3.70}$	$61.83_{\pm 6.25}$
$Ph \to F_{16}$	$26.34_{\pm 0.00}$	$21.13_{\pm 0.42}$	$30.06_{\pm 0.21}$	$36.67_{\pm 7.77}$	$46.50_{\pm 5.10}$	$29.17_{\pm 2.39}$	$34.50_{\pm 4.26}$	$55.83_{\pm 1.55}$
$Ph \to F_4$	$26.34_{\pm 0.00}$	$20.98_{\pm 0.34}$	$43.30_{\pm 0.96}$	$37.17_{\pm 1.65}$	$40.17_{\pm 3.06}$	$23.00_{\pm 3.34}$	$26.83_{\pm 9.57}$	$43.00_{\pm 0.71}$
$Ph \to F_2$	$46.73_{\pm 0.76}$	$50.74_{\pm 0.42}$	$61.31_{\pm 0.92}$	$48.50_{\pm 7.78}$	$63.17_{\pm 2.01}$	$50.67_{\pm 11.83}$	$61.50_{\pm 2.12}$	$66.67_{\pm 2.09}$
Avg.	37.35	33.11 (4.24 ↓)	45.09 (7.74 ↑)	40.42 (3.07 ↑)	<u>53.49</u> (16.14 ↑)	30.68 (6.67 ↓)	50.60 (12.26 ↑)	**58.49** (21.14 ↑)

Table 4: Classification results on out-of-sample data from the CS-RT benchmark. Values are reported as mean accuracy ± standard deviation.

Domains	**Source-only**	**DANN**	**Seguy et al. (2018)**	**NeuralSPOT**	**DoDyn**
	$\mathcal{A}_{(oos)}\uparrow$	$\mathcal{A}_{(oos)}\uparrow$	$\mathcal{A}_{(oos)}\uparrow$	$\mathcal{A}_{(oos)}\uparrow$	$\mathcal{A}_{(oos)}\uparrow$
$F_{16} \to F_4$	$45.24_{\pm 0.76}$	$56.33_{\pm 3.27}$	$26.17_{\pm 5.44}$	$61.50_{\pm 2.27}$	$79.00_{\pm 0.82}$
$F_{16} \to F_2$	$45.68_{\pm 0.42}$	$62.00_{\pm 1.78}$	$20.67_{\pm 4.09}$	$50.67_{\pm 3.66}$	$75.83_{\pm 0.85}$
$F_{16} \to Ph$	$44.05_{\pm 0.76}$	$47.00_{\pm 3.63}$	$24.50_{\pm 4.02}$	$23.00_{\pm 1.78}$	$57.33_{\pm 5.57}$
$F_4 \to F_{16}$	$26.34_{\pm 0.00}$	$58.33_{\pm 3.79}$	$22.00_{\pm 3.94}$	$69.17_{\pm 6.64}$	$76.00_{\pm 1.63}$
$F_4 \to F_2$	$32.44_{\pm 0.21}$	$52.67_{\pm 3.12}$	$24.83_{\pm 10.07}$	$55.33_{\pm 6.33}$	$63.17_{\pm 1.31}$
$F_4 \to Ph$	$37.35_{\pm 0.56}$	$34.67_{\pm 0.47}$	$25.33_{\pm 8.59}$	$32.83_{\pm 7.15}$	$54.50_{\pm 3.34}$
$F_2 \to F_{16}$	$26.34_{\pm 0.00}$	$54.33_{\pm 6.79}$	$30.17_{\pm 3.52}$	$61.17_{\pm 7.00}$	$67.00_{\pm 0.00}$
$F_2 \to F_4$	$25.15_{\pm 0.21}$	$48.67_{\pm 6.51}$	$31.50_{\pm 4.71}$	$60.33_{\pm 1.84}$	$64.50_{\pm 1.08}$
$F_2 \to Ph$	$70.68_{\pm 0.21}$	$82.83_{\pm 1.93}$	$54.33_{\pm 5.91}$	$55.33_{\pm 6.33}$	$77.33_{\pm 2.90}$
$Ph \to F_{16}$	$26.34_{\pm 0.00}$	$45.00_{\pm 3.19}$	$27.33_{\pm 10.27}$	$29.17_{\pm 7.58}$	$48.67_{\pm 3.06}$
$Ph \to F_4$	$26.34_{\pm 0.00}$	$38.33_{\pm 2.46}$	$22.17_{\pm 8.25}$	$30.67_{\pm 10.07}$	$46.67_{\pm 2.25}$
$Ph \to F_2$	$45.98_{\pm 0.00}$	$58.50_{\pm 0.41}$	$39.83_{\pm 13.12}$	$51.17_{\pm 3.70}$	$66.50_{\pm 1.87}$
Avg.	37.66	<u>53.22</u> (15.56 ↑)	29.07 (8.59 ↓)	48.36 (10.70 ↑)	**64.71** (27.05 ↑)

Table 5: Classification results on the CMU-PIE benchmark. Values are reported as mean accuracy ± standard deviation, averaged over independent runs.

Domains	**Source-only**	**OTDA$_{\text{Sink}}$**	**DANN**		**DoDyn**	
	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(oos)}\uparrow$	$\mathcal{A}_{(T)}\uparrow$	$\mathcal{A}_{(oos)}\uparrow$
c07 → c29	25.83	$36.40_{\pm 0.79}$	$26.32_{\pm 3.23}$	$24.20_{\pm 2.12}$	$57.26_{\pm 0.87}$	$58.96_{\pm 0.92}$
c07 → c05	34.10	$34.41_{\pm 1.93}$	$25.45_{\pm 2.21}$	$26.47_{\pm 2.19}$	$44.00_{\pm 1.29}$	$39.31_{\pm 1.91}$
c07 → c09	37.41	$40.75_{\pm 1.17}$	$34.56_{\pm 1.28}$	$32.17_{\pm 0.79}$	$67.86_{\pm 0.97}$	$70.24_{\pm 0.78}$
c29 → c07	22.62	$14.24_{\pm 3.21}$	$36.37_{\pm 1.24}$	$36.96_{\pm 1.94}$	$58.50_{\pm 0.64}$	$58.87_{\pm 1.11}$
c29 → c05	27.02	$27.85_{\pm 1.01}$	$28.84_{\pm 0.12}$	$27.97_{\pm 0.70}$	$31.25_{\pm 1.02}$	$27.88_{\pm 2.15}$
c29 → c09	29.60	$38.20_{\pm 0.87}$	$30.82_{\pm 2.95}$	$30.09_{\pm 2.00}$	$59.74_{\pm 0.26}$	$59.74_{\pm 1.43}$
c05 → c07	24.82	$14.78_{\pm 4.51}$	$37.08_{\pm 2.05}$	$35.73_{\pm 0.91}$	$43.12_{\pm 0.77}$	$44.63_{\pm 1.74}$
c05 → c29	19.30	$10.88_{\pm 2.45}$	$29.69_{\pm 0.20}$	$28.49_{\pm 2.33}$	$31.83_{\pm 0.34}$	$33.70_{\pm 1.74}$
c05 → c09	20.68	$9.38_{\pm 2.48}$	$27.94_{\pm 4.37}$	$30.09_{\pm 3.69}$	$44.49_{\pm 0.78}$	$46.69_{\pm 0.78}$
c09 → c07	28.23	$18.58_{\pm 0.69}$	$34.81_{\pm 0.92}$	$34.25_{\pm 0.60}$	$67.03_{\pm 0.99}$	$67.40_{\pm 1.30}$
c09 → c29	33.09	$35.72_{\pm 1.06}$	$30.12_{\pm 1.81}$	$30.76_{\pm 3.70}$	$59.50_{\pm 0.49}$	$60.42_{\pm 0.96}$
c09 → c05	32.26	$30.06_{\pm 0.57}$	$27.96_{\pm 1.74}$	$29.05_{\pm 1.39}$	$42.13_{\pm 0.53}$	$37.56_{\pm 0.09}$
Avg.	27.91	25.94 (1.97 ↓)	<u>30.83</u> (2.92 ↑)	<u>30.52</u> (2.61 ↑)	**50.56** (22.65 ↑)	**50.45** (22.54 ↑)

the continuous Koopman formulation generalizes well even under challenging visual domain shifts driven by pose variation. Overall, DoDyn leads across all tasks with the largest OOS margin observed across all benchmarks.

6 Ablation Study

The following section examines the sensitivity of DoDyn to its hyperparameters and design choices.

Entropic regularization parameter ε. We study the sensitivity of DoDyn to the entropic regularization parameter ε on the MSD benchmark, using the adaptation task Noiseless → Buccaneer as a representative pair. we vary ε over the range $\{\mathbf{10^{-5}}, \mathbf{10^{-4}}, \mathbf{10^{-3}}, \mathbf{10^{-2}}, \mathbf{10^{-1}}, \mathbf{1}, \mathbf{10}\}$, With σ fixed to a default value, and report both in-distribution target accuracy and out-of-sample performance averaged over different seeds. As illustrated in Fig. 2 (left), performance is stable and near-optimal for $\varepsilon \in [10^{-4}, 10^{-2}]$ on both target and OOS evaluation. As $\varepsilon \to 0$, the entropic OT converges to exact OT, where each source point is mapped to a single target point with no mass spreading. In graph terms, this produces isolated point pairs as connected components, and degrading performance significantly. At the other extreme, large ε enforces a near-uniform

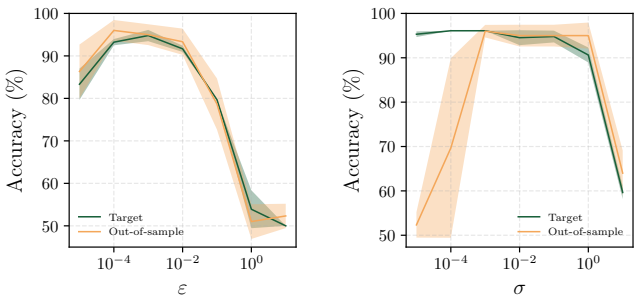

Figure 2: Sensitivity of **DoDyn** to the entropic regularization parameter ε and the standard deviation σ, evaluated on the MSD benchmark (N → B).

mass distribution across all points, collapsing the entire source-target graph into a single component, the resulting embedding carries no discriminative structure, and performance drops sharply.

Sensitivity to the standard deviation σ. The sensitivity of DoDyn to the standard deviation σ of the Gaussian components is presented in Fig. 2 (right). We vary σ over $\{\mathbf{10^{-5}}, \mathbf{10^{-4}}, \mathbf{10^{-3}}, \mathbf{10^{-2}}, \mathbf{10^{-1}}, \mathbf{1}, \mathbf{10}\}$. As shown in Fig. 2 (right), performance is stable for $\sigma \in [10^{-4}, 1]$ and degrades at both ends of the range. As $\sigma \to 0$, each Gaussian component collapses toward a Dirac centered at its sample, and the resulting mixtures degenerate into discrete measures. Target accuracy remains high because training points still lie exactly at component centers, but out-of-sample accuracy drops sharply: for a new point $x \notin X_s \cup X_t$, the densities $P_s(x)$ and $P_t(x)$ vanish almost everywhere, and the normalized weights $g_{Sk_1}(x)$ and $g_{Tk_2}(x)$ that govern the eigenfunction reconstruction in equation 20 become ill-defined. As σ controls how each new point is softly assigned to source and target components, small values gives a sharply peaked partition that does not extrapolate beyond training points. On the other hand, large σ causes excessive overlap between Gaussian components, making P_s and P_t nearly indistinguishable and washing out the normalized weights g_{Sk_1} and g_{Tk_2} that form the basis of the eigenfunction reconstruction in equation 20.

Isotropic lifting vs. fitted GMM. A natural question is whether one should rely on a parametric mixture fitted to each domain, in the spirit of (Delon & Desolneux, 2020; Montesuma et al., 2025), or on the parameter-free one-Gaussian-per-sample lifting used in DoDyn. To answer this, we compare the two strategies: in the first, source and target GMMs are fitted with N_{GMM} components via Expectation Maximization (EM), and our procedure is then run on the fitted components; in the second, each empirical sample is lifted to its own isotropic Gaussian with shared covariance $\Sigma = \sigma^2 I$. We vary N_{GMM} over $\{\mathbf{2}, \mathbf{5}, \mathbf{10}, \mathbf{20}, \mathbf{30}, \mathbf{50}, \mathbf{70}, \mathbf{80}, \mathbf{90}, \mathbf{100}, \mathbf{110}, \mathbf{120}\}$, while the isotropic variant serves as a reference independent of N_{GMM}. As shown in Fig. 3, the fitted GMM variant performs poorly at small numbers of components, and its accuracy improves progressively as N_{GMM} grows. However, the fitted GMM remains sensitive to the choice of the number of Gaussian components, as observed from the unstable behavior in the range $N_{GMM} \in [100, 110]$, where Out-of-sample accuracy drops sharply before recovering. As the number of fitted components approaches the number of samples n, the fitted mixture converges to our isotropic lifting, which achieves a comparable performance to DoDyn. Beyond performance, the fitted GMM introduces additional complexity, since it requires running EM on each domain, estimating per-component means, covariances, and weights, and tuning N_{GMM} as an extra hyperparameter. While our isotropic representation avoids all of these steps by construction, which makes it both simpler and more reliable. Additional ablation experiments on the number of eigenfunctions, the Monte Carlo sample budget, and the computational cost are provided in the supplementary material B.

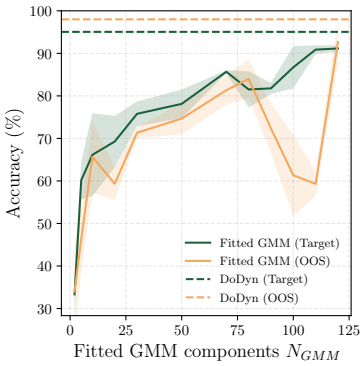

Figure 3: Comparison between DoDyn's empirical-to-Gaussian representation and parametric GMM fitted on the data.

7 Conclusion

In this paper, we presented DoDyn, a continuous domain adaptation method that learns domain-invariant representations as closed-form eigenfunctions of a Koopman operator defined on a random walk induced by an optimal transport coupled graph, and enables accurate out-of-sample embedding without recomputing the transport plan at inference. Empirical validation across benchmarks covering acoustic, real-world signal processing, and visual adaptation tasks confirms that DoDyn consistently outperforms shallow OT, deep domain adaptation, and neural OT baselines, with a decisive advantage on out-of-sample generalization. Ablation studies further confirm the robustness of the method to its key hyperparameters. The current approach operates on the full source and target sets jointly, from the OT plan to the overlap matrices and the eigenfunction reconstruction, which limits scalability to larger datasets due to its memory cost. Extending the method to a minibatch regime, where the Koopman fixed-point system and the eigenfuncions are approximated over subsampled batches at each iteration, would open the door to larger-scale tasks while preserving the continuous formulation that underpins the generalization advantage of DoDyn.

References

Charu C Aggarwal and Chandan K Reddy. Data clustering. *Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra*, 2014.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017a.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017b. doi: 10.1109/TPAMI.2016.2615921.

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

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 447–463, 2018.

George B Dantzig. Linear programming and extensions. 2016.

Julie Delon and Agnes Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.

Mourad El Hamri, Younès Bennani, and Issam Falih. Label propagation through optimal transport. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Rémi Flamary, Karim Lounici, and André Ferrari. Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation. *arXiv preprint arXiv:1905.10155*, 2019.

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, 2016.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.

Leonid V Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pp. 199–201, 1942.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Stefan Klus and Maia Trower. Transfer operators on graphs: spectral clustering and beyond. *Journal of Physics: Complexity*, 5(1):015014, 2024.

Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=bEoxzW_EXsa`.

Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=d8CBRlWNkqH`.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681. PMLR, 2020.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.

Eduardo Fernandes Montesuma and Fred Maurice Ngole Mboula. Wasserstein barycenter for multi-source domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16785–16793, 2021.

Eduardo Fernandes Montesuma, Fred Maurice Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(2):1161–1180, 2024.

Eduardo Fernandes Montesuma, Fred Maurice NGOLE MBOULA, and Antoine Souloumiac. Optimal transport for domain adaptation through gaussian mixture models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=DCAeXwLenB`.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation for discrete optimal transport. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper_files/paper/2016/file/26f5bd4aa64fdadf96152ca6e6408068-Paper.pdf`.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 737–753. Springer, 2017.

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Abdel Djalil Sad Saoud, Fred Maurice Ngolè Mboula, and Hanane Slimani. Beyond mapping: Domain-invariant representations via spectral embedding of optimal transport plans. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4556–4560, 2026. doi: 10.1109/ICASSP55912.2026.11462413.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=B1zlp1bRW`.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000. ISSN 0378-3758. doi: https://doi.org/10.1016/S0378-3758(00)00115-4. URL https://www.sciencedirect.com/science/article/pii/S0378375800001154.

T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pp. 53–58, 2002. doi: 10.1109/AFGR.2002.1004130.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Hanane Slimani, Yosra Gargouri, Fred Maurice Ngole Mboula, and Nicolas Ravot. Detection, localization and characterization of fault in cable via machine learning approach based on compressed sensing time-domain reflectometry. In *2024 IEEE AUTOTESTCON*, pp. 1–9, 2024. doi: 10.1109/AUTOTESTCON47465.2024.10697515.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Amirhossein Taghvaei and Amin Jalali. 2-wasserstein approximation via restricted convex potentials with application to improved training for gans. *arXiv preprint arXiv:1902.07197*, 2019.

Asuka Takatsu. Wasserstein geometry of gaussian measures. 2011.

George Tzanetakis. Automatic musical genre classification of audio signals. In *International Society for Music Information Retrieval Conference*, 2001. URL https://api.semanticscholar.org/CorpusID:14714051.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

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

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Zhiqing Xiao, Haobo Wang, Ying Jin, Lei Feng, Gang Chen, Fei Huang, and Junbo Zhao. Spa: A graph spectral alignment perspective for domain adaptation. *Advances in Neural Information Processing Systems*, 36:37252–37272, 2023.

Appendix

A Proofs and Derivations

In this section, we present the detailed derivations and proofs that underlie our approach.

A.1 Proof of Lemma 4.1

The marginals of $\pi_\varepsilon(x_s, x_t)$ are obtained by integrating Eq. 11 with respect to x_t, and using that each P_{Tk_2} integrates to one:

$$\int_{\mathbb{R}^d} \pi_\varepsilon(x_s, x_t)\, dx_t = \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x_s) \int_{\mathbb{R}^d} P_{Tk_2}(x_t)\, dx_t \tag{23a}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x_s) \tag{23b}$$

$$= \underbrace{\sum_{k_1=1}^{K_s} \alpha_{sk_1}\, P_{Sk_1}(x_s)}_{P_s(x_s)}, \tag{23c}$$

where the last equality uses the marginal constraint $\sum_{k_2=1}^{K_t} \pi_{k_1,k_2} = \alpha_{sk_1}$. The analogous computation for $\int_{\mathbb{R}^d} \pi_\varepsilon(x_t, x_s)\, dx_t$ yields $P_t(x_s)$. □

A.2 Proof of Proposition 4.2

Proof. Let the component expectations be defined as

$$\theta_{Sk_1} := \int_{\mathbb{R}^d} f(y)\, P_{Sk_1}(y)\, dy, \qquad \theta_{Tk_2} := \int_{\mathbb{R}^d} f(y)\, P_{Tk_2}(y)\, dy. \tag{24}$$

Replacing the first term integral in Eq.15b using Eqs.11 and 24 gives

$$\int_{\mathbb{R}^d} \pi(x,y) f(y)\, dy = \int \left[\sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x)\, P_{Tk_2}(y) \right] f(y)\, dy \tag{25a}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x) \int_{\mathbb{R}^d} P_{Tk_2}(y) f(y)\, dy \tag{25b}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x)\, \theta_{Tk_2}. \tag{25c}$$

Similarly for the second integral term,

$$\int_{\mathbb{R}^d} \pi(y,x) f(y)\, dy = \int \left[\sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(y)\, P_{Tk_2}(x) \right] f(y)\, dy \tag{26a}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Tk_2}(x) \int_{\mathbb{R}^d} P_{Sk_1}(y) f(y)\, dy \tag{26b}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Tk_2}(x)\, \theta_{Sk_1}. \tag{26c}$$

Plugging both expansions equation 25c, equation 26c into Eq.15b, we obtain

$$f(x) = \frac{1}{P_s(x) + P_t(x)} \left[\sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Sk_1}(x)\, \theta_{Tk_2} + \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\, P_{Tk_2}(x)\, \theta_{Sk_1} \right]. \tag{27}$$

Moreover, it is convenient to define

$$g_{Sk_1}(x) := \frac{P_{Sk_1}(x)}{P_s(x) + P_t(x)}, \qquad g_{Tk_2}(x) := \frac{P_{Tk_2}(x)}{P_s(x) + P_t(x)}. \tag{28}$$

Then Eq.27 rewrites as

$$f(x) = \sum_{k_1=1}^{K_s} g_{Sk_1}(x) \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} g_{Tk_2}(x) \sum_{k_1=1}^{K_s} \pi_{k_1,k_2}\theta_{Sk_1}. \tag{29}$$

□

A.3 Detailed Derivation of Fixed-Point Equations 18a and 18b

Source component expectations. For a fixed $i \in \{1, \dots, K_s\}$, by definition

$$\theta_{Si} = \int_{\mathbb{R}^d} P_{Si}(x)\, f(x)\, dx. \tag{30}$$

Substituting the fixed-point expansion of Eq. 16 into the above gives

$$\theta_{Si} = \int_{\mathbb{R}^d} P_{Si}(x) \left[\sum_{k_1=1}^{K_s} g_{Sk_1}(x) \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} g_{Tk_2}(x) \sum_{k_1=1}^{K_s} \pi_{k_1,k_2}\theta_{Sk_1} \right] dx \tag{31a}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \left(\int_{\mathbb{R}^d} P_{Si}(x)\, g_{Sk_1}(x)\, dx \right) \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} \sum_{k_1=1}^{K_s} \left(\int_{\mathbb{R}^d} P_{Si}(x)\, g_{Tk_2}(x)\, dx \right) \pi_{k_1,k_2}\theta_{Sk_1}, \tag{31b}$$

where we exchanged the integral and the sums by linearity of integration. Expanding $g_{Sk_1}(x)$ and $g_{Tk_2}(x)$ using their definitions equation 28, the integrals inside the parentheses become

$$\int_{\mathbb{R}^d} P_{Si}(x)\, g_{Sk_1}(x)\, dx = \int_{\mathbb{R}^d} \frac{P_{Si}(x)\, P_{Sk_1}(x)}{P_s(x) + P_t(x)}\, dx =: M_{SS}[i, k_1], \tag{32}$$

$$\int_{\mathbb{R}^d} P_{Si}(x)\, g_{Tk_2}(x)\, dx = \int_{\mathbb{R}^d} \frac{P_{Si}(x)\, P_{Tk_2}(x)}{P_s(x) + P_t(x)}\, dx =: M_{ST}[i, k_2], \tag{33}$$

which are exactly the overlap matrices defined in equation 17a. Substituting back yields Eq. 18a:

$$\theta_{Si} = \sum_{k_1=1}^{K_s} M_{SS}[i, k_1] \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} M_{ST}[i, k_2] \sum_{k_1=1}^{K_s} \pi_{k_1,k_2}\theta_{Sk_1}. \tag{34}$$

Target component expectations. For a fixed $j \in \{1, \dots, K_t\}$, by definition

$$\theta_{Tj} = \int_{\mathbb{R}^d} P_{Tj}(x)\, f(x)\, dx. \tag{35}$$

Substituting Eq. 16 and exchanging integral and sums by linearity gives

$$\theta_{Tj} = \int_{\mathbb{R}^d} P_{Tj}(x) \left[\sum_{k_1=1}^{K_s} g_{Sk_1}(x) \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} g_{Tk_2}(x) \sum_{k_1=1}^{K_s} \pi_{k_1,k_2}\theta_{Sk_1} \right] dx \tag{36a}$$

$$= \sum_{k_1=1}^{K_s} \sum_{k_2=1}^{K_t} \left(\int_{\mathbb{R}^d} P_{Tj}(x)\, g_{Sk_1}(x)\, dx \right) \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} \sum_{k_1=1}^{K_s} \left(\int_{\mathbb{R}^d} P_{Tj}(x)\, g_{Tk_2}(x)\, dx \right) \pi_{k_1,k_2}\theta_{Sk_1}. \tag{36b}$$

Expanding $g_{Sk_1}(x)$ and $g_{Tk_2}(x)$ using equation 28, the integrals become

$$\int_{\mathbb{R}^d} P_{Tj}(x)\, g_{Sk_1}(x)\, dx = \int_{\mathbb{R}^d} \frac{P_{Tj}(x)\, P_{Sk_1}(x)}{P_s(x) + P_t(x)}\, dx =: M_{TS}[j, k_1], \tag{37}$$

$$\int_{\mathbb{R}^d} P_{Tj}(x)\, g_{Tk_2}(x)\, dx = \int_{\mathbb{R}^d} \frac{P_{Tj}(x)\, P_{Tk_2}(x)}{P_s(x) + P_t(x)}\, dx =: M_{TT}[j, k_2], \tag{38}$$

which are exactly the overlap matrices defined in equation 17b. Substituting back yields Eq. 18b:

$$\theta_{Tj} = \sum_{k_1=1}^{K_s} M_{TS}[j, k_1] \sum_{k_2=1}^{K_t} \pi_{k_1,k_2}\theta_{Tk_2} + \sum_{k_2=1}^{K_t} M_{TT}[j, k_2] \sum_{k_1=1}^{K_s} \pi_{k_1,k_2}\theta_{Sk_1}. \tag{39}$$

A.4 On the Estimation of The Overlap matrices

Proposition A.1 (Monte Carlo estimation of overlap matrices)**.** *Each entry of the overlap matrices admits an unbiased Monte Carlo estimator. For $x_s^{(n)} \sim P_{Si}$ and $x_t^{(n)} \sim P_{Tj}$, drawn independently for $n = 1, \dots, N$,*

$$\widehat{M}_{SS}[i,k_1] = \frac{1}{N}\sum_{n=1}^{N} \frac{P_{Sk_1}(x_s^{(n)})}{P_s(x_s^{(n)}) + P_t(x_s^{(n)})}, \quad \widehat{M}_{ST}[i,k_2] = \frac{1}{N}\sum_{n=1}^{N} \frac{P_{Tk_2}(x_s^{(n)})}{P_s(x_s^{(n)}) + P_t(x_s^{(n)})}, \tag{40}$$

$$\widehat{M}_{TS}[j,k_1] = \frac{1}{N}\sum_{n=1}^{N} \frac{P_{Sk_1}(x_t^{(n)})}{P_s(x_t^{(n)}) + P_t(x_t^{(n)})}, \quad \widehat{M}_{TT}[j,k_2] = \frac{1}{N}\sum_{n=1}^{N} \frac{P_{Tk_2}(x_t^{(n)})}{P_s(x_t^{(n)}) + P_t(x_t^{(n)})}, \tag{41}$$

all converging almost surely to their population counterparts as $N \to \infty$.

In practice, since each Gaussian component $P_{Si} = \mathcal{N}(\mathbf{x}_s^i, \sigma^2 I)$ and $P_{Tj} = \mathcal{N}(\mathbf{x}_t^j, \sigma^2 I)$ is isotropic with known parameters, sampling from them reduces to the reparameterization $x_s^{(n)} = \mathbf{x}_s^i + \sigma\epsilon^{(n)}$ and $x_t^{(n)} = \mathbf{x}_t^j + \sigma\epsilon^{(n)}$ with $\epsilon^{(n)} \sim \mathcal{N}(0, I_d)$. The normalized weights $g_{Sk_1}(x_s^{(n)})$ and $g_{Tk_2}(x_s^{(n)})$ appearing in the estimators are then evaluated in log-space for numerical stability. All four matrices are estimated once before training and kept fixed throughout, since they depend only on the lifted distributions P_s and P_t and not on the learnable parameters θ_S and θ_T.

A.5 NeuralSPOT Baseline

Let $\boldsymbol{X_s} = (\mathbf{x_1^s}, \mathbf{x_2^s}, \dots, \mathbf{x_{n_s}^s}) \in \mathbb{R}^{n_s \times d}$ and $\boldsymbol{X_t} = (\mathbf{x_1^t}, \mathbf{x_2^t}, \dots, \mathbf{x_{n_t}^t}) \in \mathbb{R}^{n_t \times d}$ be two empirical distributions supported on source and target probability measures μ_s and μ_t, respectively, with $n_s \neq n_t$. We consider the discrete entropic OT problem between these empirical measures, whose optimal transport plan $\boldsymbol{\pi_\varepsilon^*}$ is defined as

$$\boldsymbol{\pi_\varepsilon^*} = \underset{\boldsymbol{\pi} \in \Pi(\mu_s, \mu_t)}{\operatorname{argmin}} \sum_{i,j} \pi_{ij} C_{ij} + \varepsilon \sum_{ij} \pi_{ij}(\log \pi_{ij} - 1), \tag{42}$$

where C is the cost matrix and $\Pi = \{\boldsymbol{\pi} \in \mathbb{R}^{n_s \times n_t} \mid \boldsymbol{\pi}\mathbf{1}_{n_t} = \mu_s,\ \boldsymbol{\pi}^\top \mathbf{1}_{n_s} = \mu_t\}$ is the set of admissible couplings. Following (Sad Saoud et al., 2026), the source and target samples are represented as nodes of a bipartite graph whose adjacency matrix is

$$\mathcal{A} = \begin{pmatrix} 0 & \boldsymbol{\pi_\varepsilon^*} \\ \boldsymbol{\pi_\varepsilon^*}^\top & 0 \end{pmatrix} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}, \tag{43}$$

and the corresponding degree matrix is

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D_s} & 0 \\ 0 & \boldsymbol{D_t} \end{pmatrix}, \quad (D_s)_{ii} = \sum_{j=1}^{n_t} \pi^*_{\varepsilon_{ij}}, \quad (D_t)_{jj} = \sum_{i=1}^{n_s} \pi^*_{\varepsilon_{ij}}. \tag{44}$$

The random-walk Laplacian is $\boldsymbol{L}_{\mathrm{rw}} = \boldsymbol{D^{-1}L} = \boldsymbol{I_{n_s+n_t}} - \boldsymbol{D^{-1}}\mathcal{A}$.

Lemma A.2. *Let $\mathcal{A}$ and $\boldsymbol{D}$ be defined as above. The matrix $\boldsymbol{P} = \boldsymbol{D^{-1}}\mathcal{A}$ is row-stochastic and defines the transition operator of a discrete-time Markov chain on the bipartite graph with adjacency $\mathcal{A}$:*

$$\boldsymbol{P} = \boldsymbol{D^{-1}}\mathcal{A} = \begin{pmatrix} 0 & \boldsymbol{D_s^{-1}\pi_\varepsilon^*} \\ \boldsymbol{D_t^{-1}\pi_\varepsilon^*}^\top & 0 \end{pmatrix}, \quad \boldsymbol{L_{\mathrm{rw}}} = \boldsymbol{I_{n_s+n_t}} - \boldsymbol{P}. \tag{45}$$

The spectral embedding $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{(n_s+n_t)\times k}$ of source and target samples is given by the null-space of $\boldsymbol{L_{\mathrm{rw}}}$, i.e., $\boldsymbol{L_{\mathrm{rw}}}\boldsymbol{\mathcal{Z}} = 0$. For each column $\boldsymbol{z}$ of $\boldsymbol{\mathcal{Z}}$,

$$\boldsymbol{L_{\mathrm{rw}}}\boldsymbol{z} = 0 \iff \boldsymbol{P}\boldsymbol{z} = \boldsymbol{z}, \tag{46}$$

so $\boldsymbol{z}$ is invariant under one transition step of $\boldsymbol{P}$, equivalently it lies in the stationary subspace of the Markov chain. Decomposing $\boldsymbol{\mathcal{Z}}$ into source and target blocks $\boldsymbol{\mathcal{Z}_s}$ and $\boldsymbol{\mathcal{Z}_t}$,

$$\begin{pmatrix} 0 & \boldsymbol{D_s^{-1}\pi_\varepsilon^*} \\ \boldsymbol{D_t^{-1}\pi_\varepsilon^*}^\top & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\mathcal{Z}_s} \\ \boldsymbol{\mathcal{Z}_t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\mathcal{Z}_s} \\ \boldsymbol{\mathcal{Z}_t} \end{pmatrix}, \tag{47}$$

which yields the coupled eigen-problem

$$\begin{cases} \boldsymbol{D_s^{-1}\pi_\varepsilon^* \mathcal{Z}_t = \mathcal{Z}_s}, \\ \boldsymbol{D_t^{-1}\pi_\varepsilon^{*\top} \mathcal{Z}_s = \mathcal{Z}_t}. \end{cases} \tag{48}$$

Proposition A.3. *Let $\Phi : \mathbb{R}^d \to \mathbb{R}^k$, $f_\Phi = f_{\Phi_l} \circ \cdots \circ f_{\Phi_1}$, be a neural network with learnable parameters $\Phi = \{\Phi_1, \ldots, \Phi_l\}$, where k is the number of eigenvectors. The coupled eigen-problem equation 48 can be solved by learning a shared embedding $\mathcal{Z}_s[i] = f_\Phi(\mathbf{x_i^s})$ and $\mathcal{Z}_t[j] = f_\Phi(\mathbf{x_j^t})$ that minimizes*

$$\mathcal{L}_{\text{spec}}(\Phi) = \sum_{i,j} \|D_s^{-1}\pi^*_{\varepsilon_{ij}} f_\Phi(\mathbf{x_j^t}) - f_\Phi(\mathbf{x_i^s})\|_2^2 + \|D_t^{-1}\pi^{*\top}_{\varepsilon_{ij}} f_\Phi(\mathbf{x_i^s}) - f_\Phi(\mathbf{x_j^t})\|_2^2. \tag{49}$$

To enforce class-discriminative structure, we attach a classifier $\mathcal{C}_\phi : \mathbb{R}^k \to \mathbb{R}^{N_c}$ on top of the learned embedding and train it jointly with f_Φ using the total objective

$$\min_{\Phi,\phi} \underbrace{\sum_{i,j} \left\| n\pi^*_\varepsilon f_\Phi(\mathbf{x_t^j}) - f_\Phi(\mathbf{x_s^i}) \right\|_2^2 + \left\| n\pi^{*\top}_\varepsilon f_\Phi(\mathbf{x_s^i}) - f_\Phi(\mathbf{x_t^j}) \right\|_2^2}_{\text{spectral loss}} + \eta \underbrace{\frac{1}{n_s} \sum_i \mathcal{L}_{CE}\big(\phi(f_\Phi(\mathbf{x_s^i})), y_{si}\big)}_{\text{classification loss}} \tag{50}$$

where Φ and ϕ are optimized jointly via gradient descent, under the same setup as DoDyn.

B Additional Experiments

In this section, we present an additional study on the choice of different components of our method.

B.1 Ablation on the number of eigenfunctions k

The number of eigenfunctions k controls the dimensionality of the learned domain-invariant representation. We study its effect on the MGR benchmark, since it contains more classes, we used the adaptation task F16 → D as a representative pair, with all other hyperparameters fixed to their default values ($\sigma = 0.1$, $\varepsilon = 0.001$, $N = 200$). We vary k over $\{\mathbf{1}, \mathbf{2}, \mathbf{5}, \mathbf{10}, \mathbf{20}, \mathbf{50}, \mathbf{100}\}$ and report both target and out-of-sample accuracies. Ideally, one would set $k = N_c$ where N_c is the number of classes, as the Koopman Eigenfunctions should ideally recover N_c connected components in the random walk graph, each concentrating mass from samples belonging to the same class. The results are illustrated in Fig. 4. As shown, for $k < N_c = 10$, performance is poor since the embedding lacks sufficient expressivity to separate all classes. As k grows, performance starts improving, reaching its best at $k = N_c$. Beyond this point, performance plateaus and remains stable regardless of how many additional eigenfunctions are estimated, which indicates that DoDyn is consistent with the following principle: the relevant spectral information is captured by the first N_c eigenfunctions, and adding more does not degrade nor improve the representation. This also gives a simple and intuitive rule where k is simply set to the number of classes without any tuning.

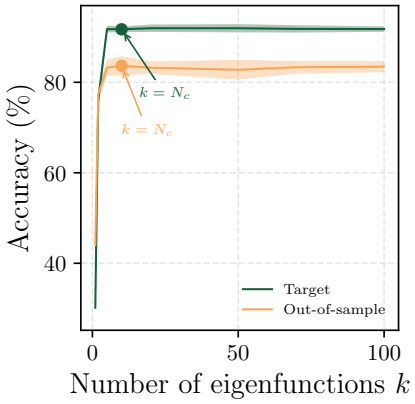

Figure 4: The effect of the number of Eigenfunctions k on the adaptation.

B.2 Monte Carlo sample budget and computational cost

We examine the effect of the number of Monte Carlo samples N on the adaptation performance and the associated computational cost of the overlap matrices. In Fig. 5.(a), we vary N over $\{\mathbf{25}, \mathbf{50}, \mathbf{100}, \mathbf{200}, \mathbf{800}, \mathbf{1600}\}$ on the MSD benchmark (N → B) and report target and out-of-sample accuracies, with all other hyperparameters fixed to their default values. In Fig. 5.(b), we report the wall-clock time required to compute the four overlap matrices on synthetic data of increasing size $n \in \{\mathbf{100}, \mathbf{200}, \mathbf{500}, \mathbf{1000}, \mathbf{3000}, \mathbf{5000}, \mathbf{10000}\}$, where source and target samples are drawn from Gaussian distributions in $\mathbb{R}^d$ with $d = 256$, for five values of

$N \in \{\mathbf{25}, \mathbf{100}, \mathbf{200}, \mathbf{300}, \mathbf{500}, \mathbf{700}\}$. As shown in Fig. 5(a), both target and out-of-sample accuracy remain almost stable across the full range of N, which indicates that the Monte Carlo estimators of the overlap matrices converge rapidly and that a modest N suffices in practice. Concerning the runtime in Fig. 5(b), it grows with n for all values of N, consistent with the $\mathcal{O}(n^2 N d)$ complexity of the estimation step, which motivates the minibatch extension discussed in Section 7 as the natural direction toward larger-scale adaptation.

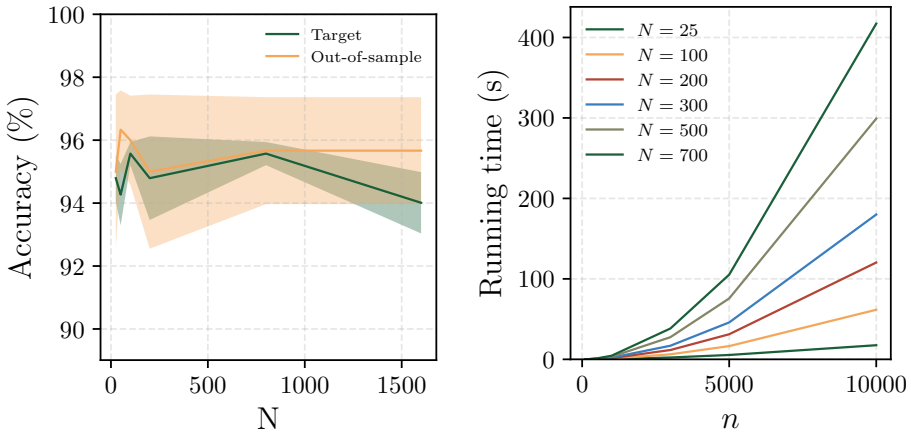

Figure 5: (a) Sensitivity of DoDyn to the Monte Carlo sample budget N used to estimate the overlap matrices, evaluated on the MSD benchmark (Noiseless $\rightarrow$ Buccaneer). (b) Running time for computing the four overlap matrices as a function of the number of samples n, for varying N, on synthetic data with $d = 256$.

B.3 Components Contribution

To study the contribution of each component of DoDyn, we compare three configurations on the MSD benchmark using the adaptation task Noiseless $\rightarrow$ buccaneer2, with all hyperparameters fixed to their default values ($\sigma = 0.1$, $\varepsilon = 0.001$, $N = 200$, $k = 2$), and report both target and out-of-sample accuracies. The configurations are: *(i) Clf.*, a classifier trained on source samples and evaluated at target domain with no adaptation. *(ii) Oper.*, a learned Θ via $\mathcal{L}_{\text{koop}}$ only without the classifier regularizer, once Θ is obtained, a post-hoc classifier is trained on the source eigenfunctions and evaluated on target/OOS embeddings. *(iii) Oper. + Clf.*, which implements the proposed DoDyn objective. Fig. 6 shows that the proposed DoDyn objective (*Oper. + Clf.*) achieves the best performance, with *Oper.* and *Clf.* performing almost comparably, confirming the importance of the combination in equation 22 with the classifier acting as a regularizer. The gradient analysis in Fig. 6 (right) further supports the design of equation 22: $\|\nabla_\Theta \mathcal{L}_{\text{Koop}}\|$ dominates $\|\nabla_\Theta \mathcal{L}_{\text{CE}}\|$ by one to two orders of magnitude throughout training, meaning the classifier acts as a mild regularizer on the spectral embedding rather than distorting it. This also provides a practical guideline to select η in a way that offers a good balance, as larger values risk pushing the CE gradient too close to the Koopman gradient and perturbing the solution.

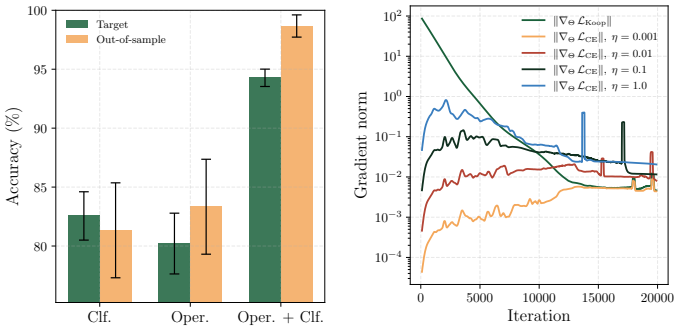

Figure 6: Ablation on DoDyn's structural components. *(Left)* Accuracy of five configurations. *(Right)* Gradient norms throughout training for varying η.