The Conditional Cauchy-Schwarz Divergence with Applications to Time-Series Data and Sequential Decision Making

Shujian Yu, Hongming Li, Sigurd Løkse, Robert Jenssen, and José C. Príncipe *Life Fellow, IEEE*

Abstract—The Cauchy-Schwarz (CS) divergence was developed by Príncipe *et al.* in 2000. In this paper, we extend the classic CS divergence to quantify the closeness between two conditional distributions and show that the developed conditional CS divergence can be simply estimated by a kernel density estimator from given samples. We illustrate the advantages (e.g., rigorous faithfulness guarantee, lower computational complexity, higher statistical power, and much more flexibility in a wide range of applications) of our conditional CS divergence over previous proposals, such as the conditional KL divergence and the conditional maximum mean discrepancy. We also demonstrate the compelling performance of conditional CS divergence in two machine learning tasks related to time series data and sequential inference, namely time series clustering and uncertainty-guided exploration for sequential decision making.

Index Terms—Conditional Cauchy-Schwarz divergence, Time series clustering, Sequential decision making

1 Introduction

Quantifying the dissimilarity or distance between two probability distributions (i.e., $p_s(\mathbf{y})$ with respect to $p_t(\mathbf{y})$, where $\mathbf{y}$ is a random variable or vector, s and t refer to domain index) is a fundamental problem in pattern analysis and machine intelligence, receiving significant interests in both machine learning and statistics [1], [2]. It also plays a central role in various problems, such as the generative models [3], the independence or conditional independence tests [4], [5], and the design of robust deep learning loss functions [6].

Different types of divergence or discrepancy measures have been developed over the past few decades [7]. Among them, integral probability metrics (IPMs) [8] and f-divergences [9], [10] are perhaps the two most popular choices. IPMs aim to find a well-behaved function f from a class of function $\mathcal{F}$ such that $|\mathbb{E}_{p_s}f(\mathbf{y}) - \mathbb{E}_{p_t}f(\mathbf{y})|$ is maximized. Notable examples in this category include the Wasserstein distance and maximum mean discrepancy (MMD) [11]. On the other hand, f-divergences, such as the Kullback-Leibler (KL) divergence and the Jensen-Shannon divergence, regards two distributions as identical if they assign the same likelihood to every point [12].

In the early 2000s, Principe *et al.* [13], [14] suggested a way to quantify the distributional dissimilarity simply by measuring the tightness of the famed Cauchy-Schwarz (CS) inequality associated with two distributions[1]:

$$\left|\int p(\mathbf{y})q(\mathbf{y})d\mathbf{y}\right|^2 \leq \int \mid p(\mathbf{y}) \mid^2 d\mathbf{y} \int \mid q(\mathbf{y}) \mid^2 d\mathbf{y}, \tag{1}$$

with equality if and only if $p(\mathbf{y})$ and $q(\mathbf{y})$ are linear independent, a measure of the "distance" between the probability density functions (PDFs) can be defined with:

$$\begin{aligned} D_{\text{CS}}(p;q) &= -\log\left(\frac{\left|\int p(\mathbf{y})q(\mathbf{y})d\mathbf{y}\right|^2}{\int \mid p(\mathbf{y}) \mid^2 d\mathbf{y} \int \mid q(\mathbf{y}) \mid^2 d\mathbf{y}}\right) \\ &= -2\log(\int p(\mathbf{y})q(\mathbf{y})d\mathbf{y}) + \log(\int p(\mathbf{y})^2 d\mathbf{y}) \\ &\quad + \log(\int q(\mathbf{y})^2 d\mathbf{y}), \end{aligned} \tag{2}$$

which was named the CS divergence.

The CS divergence enjoys a few appealing properties [15]. For example, it has closed-form expression for mixture-of-Gaussians (MoG) [16], a property that KL divergence does not hold. Moreover, it can be simply evaluated with the kernel density estimator (KDE). Specifically, given $\{\mathbf{y}_i^s\}_{i=1}^M$ and $\{\mathbf{y}_i^t\}_{i=1}^N$, both in $\mathbb{R}^p$, drawn *i.i.d.* from $p_s(\mathbf{y})$ and $p_t(\mathbf{y})$, respectively. Using the kernel density estimator

This work was funded in part by the Research Council of Norway (RCN) under grant 309439, and the U.S. ONR under grant ONR N00014-21-1-2295.

Shujian Yu is with the Machine Learning Group, UiT - The Arctic University of Norway, Tromsø, Norway, and with the Quantitative Data Analytics Group, Vrije Universiteit Amsterdam (email:yusj9011@gmail.com).
Hongming Li and José C. Príncipe are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (e-mail:hongmingli@ufl.edu; principe@cnel.ufl.edu).
Sigurd Løkse is with the Drones and Autonomous Systems group at NORCE Norwegian Research Centre, Tromsø, Norway (email:sigl@norceresearch.no).
Robert Jenssen is with the Machine Learning Group, UiT - The Arctic University of Norway, Tromsø, Norway, with the Pioneer AI Centre, Copenhagen University, and with the Norwegian Computing Center, Oslo, Norway (email:robert.jenssen@uit.no).

1. In this work, we interchangeably use $p_s(\mathbf{y})$ wrt. $p_t(\mathbf{y})$ or $p(\mathbf{y})$ wrt. $q(\mathbf{y})$ to denote two distributions.

(KDE) [17] with Gaussian kernel $G_\sigma(\cdot) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{\|\cdot\|^2}{2\sigma^2})$, Eq. (2) can be estimated as [15]:

$$\widehat{D}_{\text{CS}}(p_s(\mathbf{y}); p_t(\mathbf{y})) = -2\log\left(\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} G_\sigma(\mathbf{y}_i^s - \mathbf{y}_j^t)\right) + \log\left(\frac{1}{M^2}\sum_{i,j=1}^{M} G_\sigma(\mathbf{y}_i^s - \mathbf{y}_j^s)\right) + \log\left(\frac{1}{N^2}\sum_{i,j=1}^{N} G_\sigma(\mathbf{y}_i^t - \mathbf{y}_j^t)\right). \tag{3}$$

The estimator in Eq. (3) is also closely related to the famed MMD. We refer interested readers to Appendix A.3 for more details.

Due to these properties, the CS divergence has been widely used in a variety of practical machine learning applications. Popular examples include measuring the similarity of two Poisson point processes [18] or data clustering by maximizing the sum of divergences between pairwise clusters [19]. Recently, it has been applied to variational autoencoders (VAE) [20], [21] by imposing a MoG prior, which significantly improves the representation power and the quality of generation. In another application to scene flow estimation for a pair of consecutive point clouds, the CS divergence demonstrated noticeable performance gain over the Chamfer distance and the Earth Mover's distance [22].

All the above-mentioned measures can only be used to evaluate the dissimilarity of two (marginal) distributions, whereas, in practice, one is also frequently interested to quantify the dissimilarity of two conditional distributions, i.e., $p(\mathbf{y}|\mathbf{x})$ with respect to $q(\mathbf{y}|\mathbf{x})$. In fact, the conditional divergence itself has also been used, as a key ingredient, in recent machine learning applications. For example, in domain adaptation and generalization, given input $\mathbf{x}$ and class label y, suppose ϕ is a feature extractor, it is crucially important to align both the (marginal) distribution of latent representations $p(\phi(\mathbf{x}))$ and the conditional distribution $p(y|\phi(\mathbf{x}))$ [23], although in practice people resort to match the class conditional distribution $p(\phi(\mathbf{x})|y)$ for simplicity (because y is discrete) [24], [25]. In self-supervised learning, let $\mathbf{v}_1$ and $\mathbf{v}_2$ denote two different augmented views of the same input entity $\mathbf{x}$, suppose $\mathbf{z}_1$ and $\mathbf{z}_2$ are the latent representations of $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. To ensure minimum sufficient information in $\mathbf{z}_1$ and $\mathbf{z}_2$, it is necessary to minimize the conditional divergence between $p_\theta(\mathbf{z}_1|\mathbf{v}_1)$ and $p_\varphi(\mathbf{z}_2|\mathbf{v}_2)$, in which θ and φ are parameters of view-specific encoders [26].

Despite the growing attention and increasing importance of the conditional divergence, methods on quantifying the dissimilarity or discrepancy of two conditional distributions are less investigated. In this paper, we follow the line of the classic CS divergence in Eq. (2) and extend it to conditional distributions. We illustrate the advantages of the developed conditional CS divergence over the conditional KL divergence and the conditional maximum mean discrepancy (CMMD) [27], [28]. We also demonstrate its versatile applications and consistent performance boost in two tasks related to time series data, namely time series clustering and uncertainty-guided exploration for reinforcement learning without rewards. We finally discuss the potential usages of the new divergence to other fundamental problems and identify its limitations that deserve further research.

2 Background Knowledge

2.1 Problem Formulation

We have two sets of observations $\psi_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^M$ and $\psi_t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^N$ that are assumed to be independently and identically distributed (*i.i.d.*) with density functions $p_s(\mathbf{x}, \mathbf{y})$ and $p_t(\mathbf{x}, \mathbf{y})$, respectively. Here, $\mathbf{x}$ is a random vector which contains input or explanatory variables, whereas $\mathbf{y}$ is the response or dependent variable (or vector).

Typically, the conditional distributions $p(\mathbf{y}|\mathbf{x})$ in ψ_s and ψ_t are unknown and unspecified. The aim of this paper is to develop a computationally efficient measure to quantify the closeness between $p_s(\mathbf{y}|\mathbf{x})$ and $p_t(\mathbf{y}|\mathbf{x})$ based on observations from ψ_s and ψ_t, which we denote as $D(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x}))$.

2.2 Existing Measures of $D(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x}))$

We now briefly review previous efforts to measure $D(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x}))$. The most natural choice is obviously the Kullback-Leibler (KL) divergence. Indeed, by the chain rule for KL divergence, $D_{\text{KL}}(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x}))$ can be decomposed as:

$$D_{\text{KL}}(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x})) = D_{\text{KL}}(p_s(\mathbf{x}, \mathbf{y}); p_t(\mathbf{x}, \mathbf{y})) - D_{\text{KL}}(p_s(\mathbf{x}); p_t(\mathbf{x})). \tag{4}$$

Eq. (4) suggests that one can measure $D_{\text{KL}}(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x}))$ by simply taking the difference between $D_{\text{KL}}(p_s(\mathbf{x}, \mathbf{y}); p_t(\mathbf{x}, \mathbf{y}))$ and $D_{\text{KL}}(p_s(\mathbf{x}); p_t(\mathbf{x}))$, in which both terms can be evaluated by popular plug-in KL divergence estimators, such as the k-nearest neighbors (k-NN) estimator [29] and the KDE estimator. Albeit its simplicity, the decomposition rule requires estimation in the joint space of two variables, which further increase dimensionality and makes estimation more difficult. On the other hand, a common drawback for k-NN estimator and its variants (e.g., [30]) is that they are not differentiable, which limits their practical applications in modern deep learning tasks.

The second approach defines a distance metric through the embedding of probability functions in a reproducing kernel Hilbert space (RKHS) $\mathcal{F}$. The RKHS estimators are much more well-behaved with respect to the dimensionality of the input data, which is also particularly important for our CS divergence. Specifically, let $\{\mathbf{y}_i^s\}_{i=1}^M$ and $\{\mathbf{y}_i^t\}_{i=1}^N$ be the sets of samples from $p_s(\mathbf{y})$ and $p_t(\mathbf{y})$ respectively, the maximum mean discrepancy (MMD) [11] is defined as the distance of feature means μ_s and μ_t in $\mathcal{F}$. That is,

$$\text{MMD}[p_s(\mathbf{y}), p_t(\mathbf{y})] = \|\mu_s - \mu_t\|_{\mathcal{F}}^2 \tag{5}$$

In practice, an estimate of the MMD objective compares the square difference between the empirical kernel mean embeddings:

$$\widehat{\text{MMD}}[p_s(\mathbf{y}), p_t(\mathbf{y})] = \|\frac{1}{M}\sum_{i=1}^{M}\phi(\mathbf{y}_i^s) - \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{y}_i^t)\|_{\mathcal{F}}^2 = \left[\frac{1}{M^2}\sum_{i,j}^{M}\kappa(\mathbf{y}_i^s, \mathbf{y}_j^s) - \frac{2}{MN}\sum_{i,j}^{MN}\kappa(\mathbf{y}_i^s, \mathbf{y}_j^t) + \frac{1}{N^2}\sum_{i,j}^{N}\kappa(\mathbf{y}_i^t, \mathbf{y}_j^t)\right], \tag{6}$$

where $\phi(\mathbf{y}) := \kappa(\mathbf{y}, \cdot)$ refers to a (usually infinite dimension) feature map of $\mathbf{y}$, $\kappa : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a positive definite kernel (usually Gaussian).

An unbiased and more commonly used MMD estimator is given by:

$$\widehat{\mathrm{MMD}}_u[p_s(\mathbf{y}), p_t(\mathbf{y})] = \Bigg[\frac{1}{M(M-1)}\sum_{i}^{M}\sum_{i\neq j}^{M}\kappa(\mathbf{y}_i^s, \mathbf{y}_j^s) - \frac{2}{MN}\sum_{i,j}^{MN}\kappa(\mathbf{y}_i^s, \mathbf{y}_j^t) + \frac{1}{N(N-1)}\sum_{i}^{N}\sum_{i\neq j}^{N}\kappa(\mathbf{y}_i^t, \mathbf{y}_j^t)\Bigg]. \tag{7}$$

The RKHS embedding of conditional distributions was developed by Song *et al.* [28], [31] via a conditional covariance operator $\mathcal{C}_{Y|X} = \mathcal{C}_{YX}\mathcal{C}_{XX}^{-1} = \mathcal{C}_{YX}\left(\mathcal{C}_{XX} + \lambda I\right)^{-1}$, in which $\mathcal{C}_{XX}$ and $\mathcal{C}_{YX}$ are respectively the uncentered covariance and cross-covariance operator. Later, Ren *et al.* [27] measures the difference of two conditional distributions by the square difference between the empirical estimates of the conditional embedding operators. The so-called conditional maximum mean discrepancy (CMMD) is defined as:

$$\begin{aligned}
&\widehat{\mathrm{CMMD}}[p_s(\mathbf{y}|\mathbf{x}), p_t(\mathbf{y}|\mathbf{x})] = \|\hat{\mathcal{C}}^s_{Y|X} - \hat{\mathcal{C}}^t_{Y|X}\|^2_{\mathcal{F}\otimes\mathcal{G}} \\
&= \|\Phi_s(K_s+\lambda I)^{-1}\Upsilon_s^T - \Phi_t(K_t+\lambda I)^{-1}\Upsilon_t^T\|^2_{\mathcal{F}\otimes\mathcal{G}} \\
&= \mathrm{tr}(K_s\tilde{K}_s^{-1}L_s\tilde{K}_s^{-1}) + \mathrm{tr}(K_t\tilde{K}_t^{-1}L_t\tilde{K}_t^{-1}) \\
&\quad - 2\,\mathrm{tr}(K_{st}\tilde{K}_t^{-1}L_{ts}\tilde{K}_s^{-1}),
\end{aligned} \tag{8}$$

in which $\otimes$ denotes tensor product, $\mathcal{G}$ is the RKHS corresponding to $\mathbf{x}$, $\tilde{K} = K + \lambda I$, tr denotes matrix trace. $\Phi_s = [\phi(\mathbf{y}_1^s), \phi(\mathbf{y}_2^s), \cdots, \phi(\mathbf{y}_M^s)]$ and $\Upsilon_s = [\phi(\mathbf{x}_1^s), \phi(\mathbf{x}_2^s), \cdots, \phi(\mathbf{x}_M^s)]$ are implicitly formed feature matrices for dataset ψ_s, Φ_t and Υ_t are defined similarly for dataset ψ_t. $K_s = \Upsilon_s^T\Upsilon_s \in \mathbb{R}^{M\times M}$ and $K_t = \Upsilon_t^T\Upsilon_t \in \mathbb{R}^{N\times N}$ are the Gram matrices for input variables X, $L_s = \Phi_s^T\Phi_s \in \mathbb{R}^{M\times M}$ and $L_t = \Phi_t^T\Phi_t \in \mathbb{R}^{N\times N}$ are the Gram matrices for output variables Y. Finally, $K_{st} = \Upsilon_s^T\Upsilon_t \in \mathbb{R}^{M\times N}$ and $L_{ts} = \Phi_t^T\Phi_s \in \mathbb{R}^{N\times M}$ are the Gram matrices between ψ_s and ψ_t on input and output variables, i.e., $(K_{st})_{ij} = \kappa(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and $(L_{ts})_{ij} = \kappa(\mathbf{y}_i^t, \mathbf{y}_j^s)$.

The recently developed conditional Bregman matrix divergence [32] applies the decomposition rule in Eq. (4) and quantifies the difference of two conditional distributions via:

$$D_{\varphi,B}(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x})) = D_{\varphi,B}(C^s_{\mathbf{xy}}; C^t_{\mathbf{xy}}) - D_{\varphi,B}(C^s_{\mathbf{x}}; C^t_{\mathbf{x}}), \tag{9}$$

in which $D_{\varphi,B}$ refers to the Bregman matrix divergence [33]. $C_{\mathbf{xy}} \in \mathcal{S}_+^{p+q}$ denotes the sample covariance matrix that characterizes the joint distribution $p(\mathbf{x},\mathbf{y})$, whereas $C_{\mathbf{x}} \in \mathcal{S}_+^{p}$ denotes the sample covariance matrix that characterizes the distribution $p(\mathbf{x})$. Formally, given a strictly convex, differentiable function φ, the matrix Bregman divergence from a matrix ρ to a matrix σ is defined as:

$$D_{\varphi,B}(\sigma;\rho) = \varphi(\sigma) - \varphi(\rho) - \mathrm{tr}\left((\nabla\varphi(\rho))^T(\sigma-\rho)\right), \tag{10}$$

where $\mathrm{tr}(\cdot)$ denotes the trace. For example, when $\varphi(\sigma) = \mathrm{tr}(\sigma\log\sigma - \sigma)$, where $\log\sigma$ is the matrix logarithm, the resulting Bregman matrix divergence is:

$$D_{\mathrm{vN}}(\sigma;\rho) = \mathrm{tr}(\sigma\log\sigma - \sigma\log\rho - \sigma + \rho), \tag{11}$$

which is also referred to von Neumann divergence [34].

We summarize in Table 1 different properties (such as computational complexity, differentiability and faithfulness) associated with different existing measures. We also include our new measure for a comparison. Here, we define "faithfulness" of a measure when $D(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x})) \geq 0$ in general and $D(p_s(\mathbf{y}|\mathbf{x}); p_t(\mathbf{y}|\mathbf{x})) = 0$ if and only if $p_s(\mathbf{y}|\mathbf{x}) = p_t(\mathbf{y}|\mathbf{x})$. The computational complexity of conditional KL divergence estimator by k-NN graph can be reduced to $\mathcal{O}(kN\log N)$. The computational complexity of conditional MMD comes from the computation of Gram matrices ($\mathcal{O}(N^2d)$ complexity) and matrix inversion ($\mathcal{O}(N^3)$ complexity). Our conditional CS divergence also requires computation of Gram matrices, but avoids matrix inversion. The computational complexity of conditional Bregman divergence is dominated by the value of d, rather than N. Because it requires evaluating a covariance matrix of size $d\times d$ and its eigenvalues. Note that, the computation of Gram matrices can take only $\mathcal{O}(RN\log d)$ complexity, where R is the number of basis functions for approximating kernels which determines the approximation accuracy [35].

TABLE 1
Properties of different conditional divergences. "Diff." refers to the differentiability; "Faith." refers to the faithfulness.

	Hyperparameter	Complexity[2]	Diff.	Faith.
Cond. KL[1]	k	$\mathcal{O}(kN\log N)$	✗	✓
Cond. MMD	kernel size σ; λ	$\mathcal{O}(N^2d+N^3)$	✓	?[3]
Cond. Bregman	free	$\mathcal{O}(Nd^2+d^3)$	✓	✗
Cond. CS (ours)	kernel size σ	$\mathcal{O}(N^2d)$	✓	✓

[1] We assume the conditional KL divergence is estimated nonparametrically with k-nearest neighbors (k-NN) graph.
[2] N refers to number of samples, d is the dimension of $\mathbf{x}$ or $\mathbf{y}$.
[3] Indeed, there is no universally agreed-upon definition for conditional MMD, and the faithfulness property depends on the specific definition [36]. But this is beyond the scope of our work.

3 The Conditional Cauchy-Schwarz Divergence

3.1 Extending Cauchy-Schwarz divergence for conditional distributions

Following Eq. (2), the CS divergence for two conditional distributions $p(\mathbf{y}|\mathbf{x})$ and $q(\mathbf{y}|\mathbf{x})$ can be expressed naturally as:

$$\begin{aligned}
&D_{\mathrm{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x})) = -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p(\mathbf{y}|\mathbf{x})q(\mathbf{y}|\mathbf{x})d\mathbf{x}d\mathbf{y}\right) \\
&+ \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p^2(\mathbf{y}|\mathbf{x})d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} q^2(\mathbf{y}|\mathbf{x})d\mathbf{x}d\mathbf{y}\right) \\
&= -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}\right) \\
&+ \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}\right),
\end{aligned} \tag{12}$$

which contains two conditional quadratic terms (i.e., $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ and $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$) and a cross term (i.e., $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}$).

Proposition 1. *The conditional CS divergence defined in Eq. (12) is a "faithful" measure on the closeness between $p(\mathbf{y}|\mathbf{x})$ and $q(\mathbf{y}|\mathbf{x})$.*

Proof. All proof(s) are demonstrated in Appendix B. □

We demonstrate below that Eq. (12) has closed-form empirical estimator. The estimation technique used in our paper is a bit different to that in [15], but enjoys more advantages. Interested readers can refer to Appendix A.2.

Proposition 2. *Given observations $\psi_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^M$ and $\psi_t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^N$ which are sampled from distributions $p(\mathbf{x}, \mathbf{y})$ and $q(\mathbf{x}, \mathbf{y})$, respectively. Let K^p and L^p denote, respectively, the Gram matrices for the variable $\mathbf{x}$ and the output variable $\mathbf{y}$ in the distribution p. Similarly, let K^q and L^q denote, respectively, the Gram matrices for the variable $\mathbf{x}$ and the output variable $\mathbf{y}$ in the distribution q. Meanwhile, let $K^{pq} \in \mathbb{R}^{M\times N}$ (i.e., $(K^{pq})_{ij} = \kappa(\mathbf{x}_i^s - \mathbf{x}_j^t)$) denote the Gram matrix from distribution p to distribution q for input variable $\mathbf{x}$, and $L^{pq} \in \mathbb{R}^{M\times N}$ the Gram matrix from distribution p to distribution q for output variable $\mathbf{y}$. Similarly, let $K^{qp} \in \mathbb{R}^{N\times M}$ (i.e., $(K^{qp})_{ij} = \kappa(\mathbf{x}_i^t - \mathbf{x}_j^s)$) denote the Gram matrix from distribution q to distribution p for input variable $\mathbf{x}$, and $L^{qp} \in \mathbb{R}^{N\times M}$ the Gram matrix from distribution q to distribution p for output variable $\mathbf{y}$. The empirical estimation of $D_{CS}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x}))$ is given by:*

$$
\begin{aligned}
&\widehat{D}_{CS}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x})) \approx \log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M} K_{ji}^{p} L_{ji}^{p}}{(\sum_{i=1}^{M} K_{ji}^{p})^2}\right)\right) \\
&+ \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{q} L_{ji}^{q}}{(\sum_{i=1}^{N} K_{ji}^{q})^2}\right)\right) \\
&- \log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N} K_{ji}^{pq} L_{ji}^{pq}}{(\sum_{i=1}^{M} K_{ji}^{p})(\sum_{i=1}^{N} K_{ji}^{pq})}\right)\right) \\
&- \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{M} K_{ji}^{qp} L_{ji}^{qp}}{(\sum_{i=1}^{M} K_{ji}^{qp})(\sum_{i=1}^{N} K_{ji}^{q})}\right)\right)
\end{aligned}
\tag{13}
$$

Remark 1 (Difference between CS and conditional CS)**.** *Taking a close look on the expressions of CS divergence and conditional CS divergence, we can find some interesting connections. Suppose $M = N$ for simplicity, the expression of CS divergence according to Eq. (3) can be reformulated as*[2]*:*

$$
\begin{aligned}
D_{CS}(p(\mathbf{y}); q(\mathbf{y})) = &\underbrace{\log\left(\frac{1}{M^2}\operatorname{tr}(L^p \cdot \mathbf{1})\right)}_{\text{within-distrib. similarity}} + \underbrace{\log\left(\frac{1}{N^2}\operatorname{tr}(L^q \cdot \mathbf{1})\right)}_{\text{within-distrib. similarity}} \\
&- \underbrace{\log\left(\frac{1}{MN}\operatorname{tr}(L^{pq} \cdot \mathbf{1})\right)}_{\text{cross-distrib. similarity}} - \underbrace{\log\left(\frac{1}{NM}\operatorname{tr}(L^{qp} \cdot \mathbf{1})\right)}_{\text{cross-distrib. similarity}},
\end{aligned}
\tag{14}
$$

where $\mathbf{1}$ is the all-ones matrix. $\log\left(\frac{1}{M^2}\operatorname{tr}(L^p\cdot\mathbf{1})\right) = \log\left(\frac{1}{M^2}\sum_{i,j=1}^{M}\kappa(\mathbf{y}_i^s - \mathbf{y}_j^s)\right) \approx \log\left(\mathbb{E}_{\mathbf{y}_i\sim p, \mathbf{y}_j\sim p}\kappa(\mathbf{y}_i^s - \mathbf{y}_j^s)\right)$ *measures the logarithm of the expected* within-distribution similarity *for any two samples that are drawn from p (the same interpretation applies for the term $\log\left(\frac{1}{N^2}\operatorname{tr}(L^q\cdot\mathbf{1})\right)$). By contrast,* $\log\left(\frac{1}{MN}\operatorname{tr}(L^{pq}\cdot\mathbf{1})\right) = \log\left(\frac{1}{MN}\sum_{i=}^{M}\sum_{j=1}^{N}\kappa(\mathbf{y}_i^s - \mathbf{y}_j^t)\right) \approx \log\left(\mathbb{E}_{\mathbf{y}_i\sim p, \mathbf{y}_j\sim q}\kappa(\mathbf{y}_i^s - \mathbf{y}_j^t)\right)$ *measures the logarithm of the expected* cross-distribution similarity *for any two samples such that one is drawn from p, whereas another is drawn from q (the same interpretation applies for the term* $\log\left(\frac{1}{NM}\operatorname{tr}(L^{qp}\cdot\mathbf{1})\right)$*).*

2. In fact, the last two terms in Eq. (14) are equal. We reformulate the CS divergence in this form only for the ease of the following analysis.

Similarly, the expression of conditional CS divergence (in Eq. (56)) can be reformulated as:

$$
\begin{aligned}
D_{CS}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x})) = &\log\left(\operatorname{tr}(L^p \cdot C_1)\right) + \log\left(\operatorname{tr}(L^q \cdot C_2)\right) \\
&- \log\left(\operatorname{tr}(L^{pq} \cdot C_3)\right) - \log\left(\operatorname{tr}(L^{qp} \cdot C_4)\right),
\end{aligned}
\tag{15}
$$

where C_1, C_2, C_3 and C_4 are some matrices based on the conditional variables $\mathbf{x}$ in both data sets.

Comparing Eq. (14) with Eq. (15), it is easy to observe that both CS divergence and conditional CS divergence leverage the general idea of summing up within-distribution similarities and then subtracting cross-distribution similarity, and rely on the Gram matrices L^p, L^q, L^{qp} and L^{pq}. However, the CS divergence assigns uniform weights on elements of those Gram matrices, whereas the conditional CS divergence applies non-uniform weights, in which the weights are only determined by the conditional variable $\mathbf{x}$. Specifically, $C_1 \in \mathbb{R}^{M\times M}$, $C_2 \in \mathbb{R}^{N\times N}$, $C_3 \in \mathbb{R}^{N\times M}$ and $C_4 \in \mathbb{R}^{M\times N}$ are given by:

$$
C_1 = \begin{pmatrix}
\frac{K_{11}^p}{(\sum_{i=1}^M K_{1i}^p)^2} & \cdots & \frac{K_{M1}^p}{(\sum_{i=1}^M K_{Mi}^p)^2} \\
\vdots & \ddots & \vdots \\
\frac{K_{1M}^p}{(\sum_{i=1}^M K_{1i}^p)^2} & \cdots & \frac{K_{MM}^p}{(\sum_{i=1}^M K_{Mi}^p)^2}
\end{pmatrix}, \tag{16}
$$

$$
C_2 = \begin{pmatrix}
\frac{K_{11}^q}{(\sum_{i=1}^N K_{1i}^q)^2} & \cdots & \frac{K_{N1}^q}{(\sum_{i=1}^N K_{Ni}^q)^2} \\
\vdots & \ddots & \vdots \\
\frac{K_{1N}^q}{(\sum_{i=1}^N K_{1i}^q)^2} & \cdots & \frac{K_{NN}^q}{(\sum_{i=1}^N K_{Ni}^q)^2},
\end{pmatrix}, \tag{17}
$$

$$
C_3 = \begin{pmatrix}
\frac{K_{11}^{pq}}{(\sum_{i=1}^M K_{1i}^p)(\sum_{i=1}^N K_{1i}^{pq})} & \cdots & \frac{K_{M1}^{pq}}{(\sum_{i=1}^M K_{Mi}^p)(\sum_{i=1}^N K_{Mi}^{pq})} \\
\vdots & \ddots & \vdots \\
\frac{K_{1N}^{pq}}{(\sum_{i=1}^M K_{1i}^p)(\sum_{i=1}^N K_{1i}^{pq})} & \cdots & \frac{K_{MN}^{pq}}{(\sum_{i=1}^M K_{Mi}^p)(\sum_{i=1}^N K_{Mi}^{pq})}
\end{pmatrix}, \tag{18}
$$

$$
C_4 = \begin{pmatrix}
\frac{K_{11}^{qp}}{(\sum_{i=1}^M K_{1i}^{qp})(\sum_{i=1}^N K_{1i}^{q})} & \cdots & \frac{K_{1N}^{qp}}{(\sum_{i=1}^M K_{Mi}^{qp})(\sum_{i=1}^N K_{Mi}^{q})} \\
\vdots & \ddots & \vdots \\
\frac{K_{1M}^{qp}}{(\sum_{i=1}^M K_{1i}^{qp})(\sum_{i=1}^N K_{1i}^{q})} & \cdots & \frac{K_{NM}^{qp}}{(\sum_{i=1}^M K_{Mi}^{qp})(\sum_{i=1}^N K_{Mi}^{q})}
\end{pmatrix}, \tag{19}
$$

We provide a geometric interpretation in Fig. 1 to further clarify the difference and use the cross-distribution similarity term as an example, i.e., $\operatorname{tr}(L^{pq}\cdot\mathbf{1})$ with respect to $\operatorname{tr}(L^{pq}\cdot C_3)$. For both marginal and conditional CS divergences, given a point $\mathbf{y}_i^s$ (i.e., the i-th sample that is drawn from $p(\mathbf{y})$), we need to evaluate all its cross-distribution similarities, i.e., $L_{i1}^{pq} = \kappa(\mathbf{y}_i^s - \mathbf{y}_1^t)$ (solid line), $L_{i2}^{pq} = \kappa(\mathbf{y}_i^s - \mathbf{y}_2^t)$ (dotted line), $\cdots$, and $L_{iN}^{pq} = \kappa(\mathbf{y}_i^s - \mathbf{y}_N^t)$ (dashed line).

Without random variable $\mathbf{x}$, the sum of all similarities for $\mathbf{y}_i^s$ is simply $\sum_{j=1}^N L_{ij}^{pq}$ (i.e., equal weight). Now, if there is a conditional variable $\mathbf{x}$, it can be expected that the sum of cross-distribution similarities will be influenced by this variable. Due to the *i.i.d.* assumption, we find that the weight on L_{ij}^{pq} is only determined by the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (i.e., K_{ij}^{pq}), and is independent to K_{i1}^{pq}, $\cdots$, $K_{i,j-1}^{pq}$, $K_{i,j+1}^{pq}$, $\cdots$, K_{iN}^{pq} (if we ignore the normalization term), which makes sense. Moreover, the reason

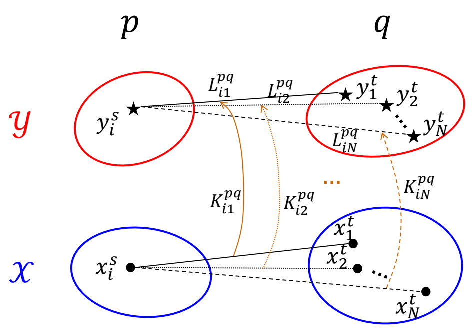


Fig. 1. To evaluate the expected value of cross-distribution similarity for $\mathbf{y}_i^s$, the weight on L_{ij}^{pq} is only determined by K_{ij}^{pq}, and is independent to $K_{i1}^{pq}, \cdots, K_{i,j-1}^{pq}, K_{i,j+1}^{pq}, \cdots, K_{iN}^{pq}$.

why the impact of K_{ij}^{pq} to L_{ij}^{pq} takes the form $L_{ij}^{pq}K_{ij}^{pq}$ (rather than other nonlinear forms) can be attributed to the Gaussian kernel that we used in our paper, which satisfies $\kappa\left(\begin{bmatrix}\mathbf{x}_i\\ \mathbf{y}_i\end{bmatrix} - \begin{bmatrix}\mathbf{x}_j\\ \mathbf{y}_j\end{bmatrix}\right) = \kappa(\mathbf{x}_i - \mathbf{x}_j)\kappa(\mathbf{y}_i - \mathbf{y}_j)$.

Remark 2 (Difference between conditional CS and conditional MMD)**.** *Interestingly, the relationship between CS divergence (Eq. (14)) and conditional CS divergence (Eq. (15)) also holds for MMD and conditional MMD [27]. Specifically, suppose $M = N$, the objective of MMD in Eq. (6) can be rewritten as:*

$$\widehat{MMD} = \frac{1}{M^2}\operatorname{tr}(L^p \cdot \mathbf{1}) + \frac{1}{N^2}\operatorname{tr}(L^q \cdot \mathbf{1}) - \frac{2}{MN}\operatorname{tr}(L^{pq} \cdot \mathbf{1}). \tag{20}$$

The objective of conditional MMD in Eq. (8) can be expressed as:

$$\widehat{CMMD} = \frac{1}{M^2}\operatorname{tr}(L^p \cdot C_1') + \frac{1}{N^2}\operatorname{tr}(L^q \cdot C_2') - \frac{2}{MN}\operatorname{tr}(L^{pq} \cdot C_3'), \tag{21}$$

where C_1', C_2' and C_3' are some matrices based on the conditional variables x in both data sets.

Despite the high similarity, we elaborate two major differences between conditional CS divergence and conditional MMD.

- *($C_1 \neq C_1'$, $C_2 \neq C_2'$ and $C_3 \neq C_3'$). Conditional CS divergence is not simply computed by just putting a "log" on each term of conditional MMD. That is, $C_1 \neq C_1'$, $C_2 \neq C_2'$ and $C_3 \neq C_3'$. For simplicity, we only analysis in detail the difference between C_1 and C_1'.*
  Let $D^p = \operatorname{diag}(K^p \cdot \mathbf{1}) \in \mathbb{R}^{M\times M}$, where $\operatorname{diag}(\cdot)$ refers to a diagonal matrix, i.e.,
  $$D^p = \begin{pmatrix} \sum_{i=1}^M K_{1i}^p & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum_{i=1}^M K_{Mi}^p \end{pmatrix}. \tag{22}$$
  We have $C_1 = K^p(D^p)^{-2}$. By comparing Eq. (8) with Eq. (21), we also have $C_1' = (\tilde{K}^p)^{-1}K^p(\tilde{K}^p)^{-1}$, where $\tilde{K}^p = K^p + \lambda I$. Therefore,
  $$\begin{aligned} C_1' &= (\tilde{K}^p)^{-1}K^p(\tilde{K}^p)^{-1} \\ &= (\tilde{K}^p)^{-1}K^p(D^p)^{-2}(D^p)^2(\tilde{K}^p)^{-1} \\ &= (\tilde{K}^p)^{-1}C_1(D^p)^2(\tilde{K}^p)^{-1} \neq C_1. \end{aligned} \tag{23}$$
  C_1' involves matrix inverse of $\tilde{K}^p$ with computational complexity $\mathcal{O}(M^3)$. By contrast, C_1 only involves matrix inverse of a diagonal matrix D^p with computational complexity $\mathcal{O}(M)$. Additionally, C_1 avoids the introduction of an additional hyperparameter λ, which is actually hard to tune in practice. The difference between C_2 (or C_3) and C_2' (or C_3') can be analyzed similarly.
- *($\operatorname{tr}(L^{pq} \cdot C_3) \neq \operatorname{tr}(L^{qp} \cdot C_4)$). The last two terms in conditional CS divergence are not the same.*

3.2 Two special cases of conditional CS divergence

We then discuss two special cases of the basic conditional CS divergence. Our purpose is to illustrate the flexibility and versatility offered by the definition of the conditional CS divergence, the elegance of its sample estimator (by just relying on the quadratic form and the inner products of samples), and its great potential to different downstream applications.

The last advantage does not hold with other divergence measures that have been discussed in Section 2.2. For example, if we stick to the decomposition rule in Eq. (4) (as in conditional KL divergence and conditional Bregman divergence), we implicitly assume that the dimension of $\mathbf{x}$ remains the same in p_s and p_t, which may not hold true in certain scenarios (e.g., comparing $p(\mathbf{y})$ with respect to $q(\mathbf{y}|\mathbf{x})$). On the other hand, there is still no universal agreement on a rigorous definition of the embedding of conditional distributions in RKHS (due to improper assumptions or difficulty of interpretation) [36], which limits the usages of conditional MMD in a wider setting.

3.2.1 $p(\mathbf{y}_1|\mathbf{x})$ *with respect to* $p(\mathbf{y}_2|\mathbf{x})$

Our first case assumes that the variable $\mathbf{x}$ in both distributions remains the same and aims to quantify the divergence between $p(\mathbf{y}_1|\mathbf{x})$ and $p(\mathbf{y}_2|\mathbf{x})$, in which $\mathbf{y}_1$ and $\mathbf{y}_2$ are dependent. This case is common in supervised learning.

Recall a standard supervised learning paradigm, we have a training set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ of input feature $\mathbf{x}$ and desired response variable y. We assume that $\mathbf{x}_i$ and y_i are sampled *i.i.d.* from a true but unknown data distribution $p(\mathbf{x}, y) = p(y|\mathbf{x})p(\mathbf{x})$. The high-level goal of supervised learning is to use the dataset $\mathcal{D}$ to learn a particular conditional distribution $q_\theta(\hat{y}|\mathbf{x})$ of the task outputs given the input features parameterized by θ, which is a good approximation of $p(y|\mathbf{x})$, in which $\hat{y}$ refers to the predicted output.

If we measure the closeness between $p(y|\mathbf{x})$ and $q_\theta(\hat{y}|\mathbf{x})$ with the KL divergence, the learning objective becomes [37]:

$$\begin{aligned} \min D_{\mathrm{KL}}(p(y|\mathbf{x}); q_\theta(\hat{y}|\mathbf{x})) &= \min \mathbb{E}\left(-\log(q_\theta(\hat{y}|\mathbf{x}))\right) - H(y|\mathbf{x}) \\ &\Leftrightarrow \min \mathbb{E}\left(-\log(q_\theta(\hat{y}|\mathbf{x}))\right), \end{aligned} \tag{24}$$

where $H(y|\mathbf{x})$ only depends on $\mathcal{D}$ that is independent to the optimization over parameters θ.

For regression, suppose $q_\theta(\hat{y}|\mathbf{x})$ is distributed normally $\mathcal{N}(h_\theta(\mathbf{x}), \sigma^2 I)$, and the network $h_\theta(\mathbf{x})$ gives the prediction of the mean of the Gaussian, the objective reduces to $\mathbb{E}\left(\|y - h_\theta(\mathbf{x})\|_2^2\right)$, which amounts to the mean squared error (MSE) loss[3] and is empirically estimated by $\frac{1}{N}\sum_{i=1}^N(y_i - \hat{y}_i)^2$.

3. Note that, $\log(q_\theta(\hat{y}|\mathbf{x})) = \log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|y-h_\theta(\mathbf{x})\|_2^2}{2\sigma^2}\right)\right) = -\log\sigma - \frac{1}{2}\log(2\pi) - \frac{\|y-f_\theta(\mathbf{x})\|_2^2}{2\sigma^2}$.

The CS divergence between $p(y|\mathbf{x})$ and $q_\theta(\hat{y}|\mathbf{x})$ is defined as:

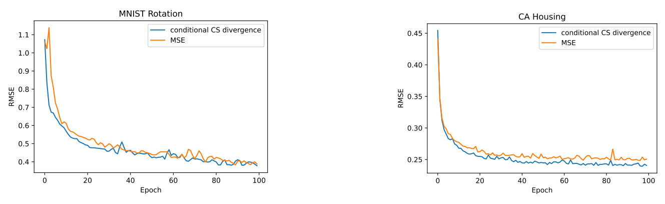
$$
\begin{aligned}
&D_{\mathrm{CS}}(p(y|\mathbf{x}); q_\theta(\hat{y}|\mathbf{x})) = -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p(y|\mathbf{x}) q_\theta(\hat{y}|\mathbf{x}) d\mathbf{x}dy\right) \\
&+ \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p^2(y|\mathbf{x}) d\mathbf{x}dy\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} q_\theta^2(\hat{y}|\mathbf{x}) d\mathbf{x}dy\right) \\
&= -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x},y) q_\theta(\mathbf{x},\hat{y})}{p^2(\mathbf{x})} d\mathbf{x}dy\right) \\
&+ \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},y)}{p^2(\mathbf{x})} d\mathbf{x}dy\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{q_\theta^2(\mathbf{x},\hat{y})}{p^2(\mathbf{x})} d\mathbf{x}dy\right),
\end{aligned}
\tag{25}
$$

which can be elegantly estimated as shown in Proposition 3 in a non-parametric way, without any parametric assumptions (e.g., Gaussian) on the underlying distribution $q_\theta(\hat{y}|\mathbf{x})$ as in the KL divergence case.

Proposition 3. *Given observations $\{(\mathbf{x}_i, y_i, \hat{y}_i)\}_{i=1}^N$, where $\mathbf{x} \in \mathbb{R}^p$ denotes a p-dimensional input variable, y is the desired response, and $\hat{y}$ is the predicted output generated by a model h_θ. Let K, L^1 and L^2 denote, respectively, the Gram matrices for the variable $\mathbf{x}$, y, and $\hat{y}$ (i.e., $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $L^1_{ij} = \kappa(y_i, y_j)$ and $L^2_{ij} = \kappa(\hat{y}_i, \hat{y}_j)$). Further, let L^{21} denote the Gram matrix between $\hat{y}$ and y (i.e., $L^{21}_{ij} = \kappa(\hat{y}_i, y_j)$). The prediction term $D_{\mathrm{CS}}(p(y|\mathbf{x}); q_\theta(\hat{y}|\mathbf{x}))$ is given by:*

$$
\begin{aligned}
&\widehat{D}_{CS}(p(y|\mathbf{x}); q_\theta(\hat{y}|\mathbf{x})) = \log\left(\sum_{j=1}^N\left(\frac{\sum_{i=1}^N K_{ji}L^1_{ji}}{(\sum_{i=1}^N K_{ji})^2}\right)\right) \\
&+ \log\left(\sum_{j=1}^N\left(\frac{\sum_{i=1}^N K_{ji}L^2_{ji}}{(\sum_{i=1}^N K_{ji})^2}\right)\right) - 2\log\left(\sum_{j=1}^N\left(\frac{\sum_{i=1}^N K_{ji}L^{21}_{ji}}{(\sum_{i=1}^N K_{ji})^2}\right)\right).
\end{aligned}
\tag{26}
$$

To test the effectiveness of Eq. (71) as a valid loss function, we train neural networks for prediction purpose. The first data is the benchmark California housing[4], which consists of $20,640$ samples and 8 features. We randomly select 70% training samples and 30% test samples. The task is to predict the median house value which has been rescaled between 0.15 and 5. The second data is the rotation MNIST[5], in which the goal is to predict the rotation angles of handwritten digits. Specifically, $10,000$ samples were selected from MNIST dataset. Each sample was randomly rotated with a degree that is uniformly distributed between $-45°$ and $45°$. The training and test sets each contain $5,000$ images. For simplicity, we use fully-connected networks ($8-128-32-128-1$ for California housing and $784-256-196-36-1$ for rotation MNIST) and Sigmoid activation function. We choose SGD optimizer with learning rate $1e-3$ and mini-batch size 128. We observe that the conditional CS divergence loss achieves slightly better prediction accuracy than the MSE loss, as shown in Fig. 2.



(a) Rotation MNIST (b) California Housing

Fig. 2. The root mean square error (RMSE) of the regression network trained with MSE loss and conditional CS divergence loss on test data in each epoch.

In case of multi-output regression which is also named as multi-task regression [38] (i.e., there are more than two targets need to be predicted), some of the tasks are often more closely related and more likely to share common relevant covariates than other tasks. Thus, it is necessary to take into account the complex correlation structure in the outputs for a more effective multi-task learning [38], [32]. It makes sense to measure relatedness between the i-th regression task and the j-th regression task by the conditional CS divergence between $p(y_i|\mathbf{x})$ and $p(y_j|\mathbf{x})$. This proposal is empirically justified in our Section 4.2.

3.2.2 $p(\mathbf{y}|\mathbf{x}_1)$ *with respect to* $p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\})$

Our second case assumes that the variable $\mathbf{y}$ remains the same and aims to quantify the divergence between $p(\mathbf{y}|\mathbf{x}_1)$ and $p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\})$, i.e., there is a third variable $\mathbf{x}_2$ that influences the distribution of $\mathbf{y}$. From a probabilistic perspective, $p(\mathbf{y}|\mathbf{x}_1) = p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\})$ implies that $\mathbf{y}$ is conditionally independent to $\mathbf{x}_2$ given $\mathbf{x}_1$, written symbolically as $\mathbf{y} \perp\!\!\!\perp \mathbf{x}_2|\mathbf{x}_1$. Hence, the divergence between $p(\mathbf{y}|\mathbf{x}_1)$ and $p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\})$ is also a good indicator on the degree of conditional independence, which is considerably difficult to measure.

The CS divergence for $p(\mathbf{y}|\mathbf{x}_1)$ and $p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\})$ can be expressed as (denote $\vec{\mathbf{x}} = [\mathbf{x}_1; \mathbf{x}_2]$, i.e., the concatenation of $\mathbf{x}_1$ and $\mathbf{x}_2$):

$$
\begin{aligned}
&D_{\mathrm{CS}}(p(\mathbf{y}|\mathbf{x}_1); p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\}) = -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p(\mathbf{y}|\mathbf{x}_1)p(\mathbf{y}|\vec{\mathbf{x}}) d\vec{\mathbf{x}}d\mathbf{y}\right) \\
&+ \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p^2(\mathbf{y}|\mathbf{x}_1) d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p^2(\mathbf{y}|\vec{\mathbf{x}}) d\vec{\mathbf{x}}d\mathbf{y}\right) \\
&= -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x}_1,\mathbf{y})p(\vec{\mathbf{x}},\mathbf{y})}{p(\mathbf{x}_1)p(\vec{\mathbf{x}})} d\vec{\mathbf{x}}d\mathbf{y}\right) \\
&+ \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x}_1,\mathbf{y})}{p^2(\mathbf{x}_1)} d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\vec{\mathbf{x}},\mathbf{y})}{p^2(\vec{\mathbf{x}})} d\vec{\mathbf{x}}d\mathbf{y}\right),
\end{aligned}
\tag{27}
$$

which can be efficiently estimated from samples as shown in Proposition 4.

Proposition 4. *Given observations $\psi = \{(\mathbf{x}^1_i, \mathbf{x}^2_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}^1 \in \mathbb{R}^{p_1}$, $\mathbf{x}^2 \in \mathbb{R}^{p_2}$ and $\mathbf{y} \in \mathbb{R}^q$. Let K^1, K^{12} and L denote, respectively, the Gram matrices for the variable $\mathbf{x}^1$, the concatenation of variables $\{\mathbf{x}^1, \mathbf{x}^2\}$, and the variable $\mathbf{y}$. That is, $(K^{12})_{ji} = \kappa\left(\begin{bmatrix}\mathbf{x}^1_j \\ \mathbf{x}^2_j\end{bmatrix} - \begin{bmatrix}\mathbf{x}^1_i \\ \mathbf{x}^2_i\end{bmatrix}\right) = \kappa(\mathbf{x}^1_j - \mathbf{x}^1_i)\kappa(\mathbf{x}^2_j - \mathbf{x}^2_i)$. The*

4. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html.

5. https://de.mathworks.com/help/deeplearning/ug/train-a-convolutional-neural-network-for-regression.html.

empirical estimation of $D_{CS}(p(\mathbf{y}|\mathbf{x}_1); p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\})$ is given by:

$$
\begin{aligned}
D_{CS}(p(\mathbf{y}|\mathbf{x}_1); p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\}) &\approx \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{1} L_{ji}}{(\sum_{i=1}^{N} K_{ji}^{1})^2}\right)\right) \\
&+ \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{12} L_{ji}}{(\sum_{i=1}^{N} K_{ji}^{12})^2}\right)\right) \\
&- 2\log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{1} L_{ji}}{(\sum_{i=1}^{N} K_{ji}^{1})(\sum_{i=1}^{N} K_{ji}^{12})}\right)\right).
\end{aligned}
\tag{28}
$$

There are vast AI applications that may benefit from an efficient sample estimator to the conditional independence [39]. Taking the representation learning as an example, our aim is to learn a representation function φ for the features $\mathbf{x}$, such that our predictions $\hat{y}$ are "invariant" to some metadata $\mathbf{z}$. If $\mathbf{z}$ refers to some protected attribute(s) such as race or gender, the *Equalized Odds* condition [40] requires the conditional independence between prediction and protected attribute(s) given ground truth of the target, i.e., $\hat{y} \perp\!\!\!\perp \mathbf{z}|y$ or equivalently $p(\hat{y}|y) = p(\hat{y}|\mathbf{z}, y)$. On the other hand, if $\mathbf{z}$ is the environment index in which the data was collected, the condition $y \perp\!\!\!\perp \mathbf{z}|\varphi(\mathbf{x})$ or equivalently $p(y|\varphi(\mathbf{x})) = p(y|\mathbf{z}, \varphi(\mathbf{x}))$ is commonly used as a target for invariant learning in domain generalization [41].

Since the attention of this paper is on dynamic data, we demonstrate the implication of Eq. (79) on time series causal discovery [42]: given two time series $\{x_t\}$ and $\{y_t\}$, identify the true causal direction between them, i.e., does $\{x_t\}$ cause $\{y_t\}$, or does $\{y_t\}$ cause $\{x_t\}$, which is also known as bivariate *causal direction identification* [43].

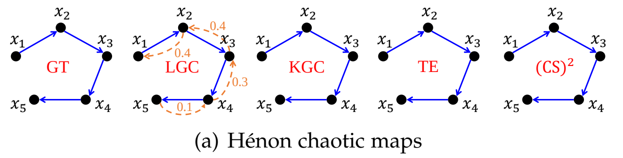
According to Granger [44], [45] (the 2003 Nobel Prize laureate in Economics), a time series (or process) $\{x_t\}$ causes another time series (or process) $\{y_t\}$ if the past of $\{x_t\}$ has unique information about the future of $\{y_t\}$. Essentially, Granger proposed to test the following hypothesis for identification of a causal effect of $\{x_t\}$ on $\{y_t\}$ [46]:

$$
\begin{cases}
\mathcal{H}_0: p(y_{t+1}|\mathbf{y}_t^n) = p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m), \\
\qquad \{\mathbf{x}_t\} \text{ is not the cause of } \{\mathbf{y}_t\} \\
\mathcal{H}_1: p(y_{t+1}|\mathbf{y}_t^n) \neq p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m), \\
\qquad \{\mathbf{x}_t\} \text{ is the cause of } \{\mathbf{y}_t\}
\end{cases}
\tag{29}
$$

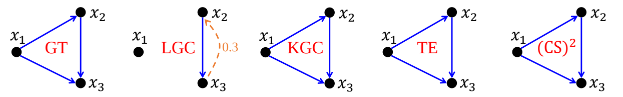
where y_{t+1} refers to the future observation of $\{y_t\}$. $\mathbf{x}_t^m = [x_t, x_{t-\tau}, \cdots, x_{t-(m-1)\tau}]$ denotes the past observation (or reconstructed state-space vector) of $\{x_t\}$, in which τ is the time delay, m is the embedding dimension. $\mathbf{y}_t^n = [y_t, y_{t-\tau}, \cdots, y_{t-(n-1)\tau}]$ denotes the past observation of $\{y_t\}$ with embedding dimension n.

From an information-theoretic perspective, one can directly evaluate the closeness between $p(y_{t+1}|\mathbf{y}_t^n)$ and $p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m)$ to perform the above test. If we use the expected KL divergence, we get:

$$
\begin{aligned}
&\mathbb{E}\left(\log\left(\frac{p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m)}{p(y_{t+1}|\mathbf{y}_t^n)}\right)\right) \\
&= \int\int\int p(y_{t+1}, \mathbf{y}_t^n, \mathbf{x}_t^m)\log\left(\frac{p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m)}{p(y_{t+1}|\mathbf{y}_t^n)}\right) dy_t d\mathbf{y}_t^n d\mathbf{x}_t^m \\
&= -\mathbb{E}\left(\log p(\mathbf{y}_t^n, \mathbf{x}_t^m)\right) + \mathbb{E}\left(\log p(y_{t+1}, \mathbf{y}_t^n, \mathbf{x}_t^m)\right) \\
&\quad - \mathbb{E}\left(\log p(y_{t+1}, \mathbf{y}_t^n)\right) + \mathbb{E}\left(\log p(\mathbf{y}_t^n)\right) \\
&= H(\mathbf{y}_t^n, \mathbf{x}_t^m) - H(y_{t+1}, \mathbf{y}_t^n, \mathbf{x}_t^m) + H(y_{t+1}, \mathbf{y}_t^n) - H(\mathbf{y}_t^n),
\end{aligned}
\tag{30}
$$



(a) Hénon chaotic maps



(b) NLVAR3

Fig. 3. The ground truth (GT) causal graph and that was identified by linear Granger causality (LGC), kernel Granger causality (KGC), transfer entropy (TE) with kNN estimator, and our causal score with CS divergence (CS)^2. The blue solid line represents the detected bivariate causal direction (after significance test). The orange dashed curve represents anti-causal direction that could be incorrectly detected (i.e., a false positive). The ratio behind the curve is the possibility of a false positive over 10 independent trials.

which is also known as the transfer entropy (TE) [47].

An alternative choice is our conditional CS divergence as shown in Eq. (79), i.e., $D_{\text{CS}}(p(y_{t+1}|\mathbf{y}_t^n); p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m))$, by simply taking $\mathbf{y} = y_{t+1}$, $\mathbf{x}^1 = \mathbf{y}_t^n$, and $\mathbf{x}^2 = \mathbf{x}_t^m$. Hence, we can define a causal score for direction $\mathbf{x} \to \mathbf{y}$ by:

$$
\begin{aligned}
C_{\mathbf{x}\to\mathbf{y}} &= D_{\text{CS}}(p(y_{t+1}|\mathbf{y}_t^n); p(y_{t+1}|\mathbf{y}_t^n, \mathbf{x}_t^m)) \\
&\quad - D_{\text{CS}}(p(x_{t+1}|\mathbf{x}_t^m); p(x_{t+1}|\mathbf{x}_t^m, \mathbf{y}_t^n)).
\end{aligned}
\tag{31}
$$

A causal direction $\mathbf{x} \to \mathbf{y}$ is confirmed if $C_{\mathbf{x}\to\mathbf{y}}$ is significantly larger than 0. On the other hand, the inverse direction $\mathbf{y} \to \mathbf{x}$ is confirmed if $C_{\mathbf{x}\to\mathbf{y}}$ is significantly smaller than 0.

To demonstrate the effectiveness of our causal score, we test its performances on two benchmark simulations: the 5 coupled Hénon chaotic maps in [48] with the true causal relation $x_{i-1} \to x_i$, and the NLVAR3 model in [49] which is a nonlinear vector autoregression (VAR) process of order 2 with 3 variables. We also compare our measure of causality with the classic linear Granger causality test [44], the popular kernel Granger causality (KGC) [50] (a generalization of linear Granger causality to nonlinear case by kernel method) and TE with kNN estimator [51]. We generate $1,024$ samples for each model and determine all pairwise causal directions by the corresponding causal score coupled with a significance test. Equations of these two models, details about the implementation regarding different competing methods and the significance test all can be found in Appendix C.1.

As can be seen from Fig. 3, the linear Granger causality fails in nonlinear data, whereas our method (CS)^2, the popular KGC and TE can precisely detect all pairwise causal directions. Compared with KGC, our measure is much more computationally efficient; compared with TE with kNN estimator, our measure is differentiable and easy-to-implement (it only requires evaluation of three Gram matrices), which facilitates more potential usages as demonstrated in the next section.

4 Numerical Simulations on Synthetic Data

We carry out two numerical simulations on synthetic data to demonstrate the behaviors (especially the statistical power) of our conditional CS divergence with respect to previous proposals mentioned in Table 1. Both simulations are designed to examine, qualitatively and quantitatively, how powerful of our divergence is to distinguish two different conditional distributions.

4.1 Simulation I

Motivated by previous literature on two-sample conditional distribution test [52], we generate 5 sets of data that have distinct conditional distributions. Specifically, in set (a), the dependent variable y is generated by $y = 1 + \sum_{i=1}^{p} x_i + \epsilon$, where p refers to the dimension of explanatory variable $\mathbf{x}$, ϵ denotes standard normal distribution. In set (b), $y = 4 + \sum_{i=1}^{p} x_i + \epsilon$, i.e., there is a mean (or 1st order moment) shift with respect to set (a). In set (c), $y = 1 + \sum_{i=1}^{p} x_i + \psi$, where ψ denotes a Logistic distribution with location parameter $\mu = 1$ and the scale parameter $s = 1$. In set (d), $y = 1 + \sum_{i=1}^{p} x_i^2 + \epsilon$. In set (e), $y = 1 + \sum_{i=1}^{p} x_i^2 + \psi$. For each set, the input distribution $p(\mathbf{x})$ is an isotropic Gaussian, but the conditional distribution $p(y|\mathbf{x})$ differs from each other. For example, in set (a), $p(y|\mathbf{x}) \sim \mathcal{N}(y - \sum_{i=1}^{p} x_i - 1, 1)$, whereas in set (e), $p(y|\mathbf{x}) \sim \textit{Logistic}(y - \sum_{i=1}^{p} x_i^2 - 2, 1)$.

To evaluate the statistical power of each conditional divergence measure to distinguish any two sets of data, we randomly simulate 500 samples from each set, each with input dimension $p = 10$. We apply non-parametric permutation test with number of permutations $P = 500$ and significant rate $\eta = 0.05$ to test if one measure can distinguish these two sets[6]. We repeat this procedure 100 independent times and use the percentage of success (a success refers to the tested measure is able to distinguish two sets) as the statistical power. Table 2 summarizes the power test results. Specifically, the (i, j)-th element of each matrix reports the quantitative statistical power of one measure to distinguish set (i) from set (j) via the following hypothesis test:

$$\begin{cases} \mathcal{H}_0 : p_i(y|\mathbf{x}) = p_j(y|\mathbf{x}), \\ \mathcal{H}_1 : p_i(y|\mathbf{x}) \neq p_j(y|\mathbf{x}). \end{cases} \tag{32}$$

An ideal measure is expected to generate a diagonal matrix with zeros on the main diagonal and ones on all off-diagonal elements.

As can be seen, the conditional CS divergence is much more powerful to distinguish two different conditional distributions (especially the ability to distinguish set (c) from set (a), and set (d) from set (e)), although it also has few false alarms in the main diagonal.

If we look deeper, the conditional von Neumann divergence on covariance matrix C has nearly zero power to distinguish set (a) from set (b), in which there is only a mean shift. This is because the covariance matrix only encodes the 2nd moment information such that it is insufficient to distinguish two conditional distributions if the distributional shift comes from 1st or higher-order moments. This also indicates that the conditional von Neumann divergence has no guarantee on the "faithfulness", which is a big limitation. On the other hand, it is surprising to find that the conditional MMD fails to identify the distinctions in these tests. Note that, this result does not mean that the conditional MMD is incapable of quantifying the conditional discrepancy. It simply suggests that a non-parametric permutation test is not a reliable way to characterize the tail probability of the extremal value of conditional MMD statistics. Hence, a more computationally efficient way is required such that one can determine the optimal detection threshold. However, to the best of our knowledge, this is still an open problem.

4.2 Simulation II

Motivated by the multi-task learning literature [53], we construct a synthetic data set with 15 related regression tasks, each with input dimension 20. For each task, the input variable $\mathbf{x}_t$ are generated $i.i.d.$ from the same distribution. The corresponding output is generated as $y_t = \mathbf{w}_t^T \mathbf{x}_t + \epsilon$, where $\mathbf{w}_t \in \mathbb{R}^{20}$ is the regression coefficients or weights of the t-th task, $\epsilon \sim \mathcal{N}(0, 1)$ is the independent noise. Because different tasks have the same input distribution $p(\mathbf{x})$, their relatedness is mainly manifested by the conditional distribution $p(y|\mathbf{x})$, which is also parameterized by $\mathbf{w}$.

In our data, each task is mostly related to its neighboring tasks to manifest strong locality relationships. Specifically, the weight in the 1st task is generated by $\mathbf{w}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{20})$, The weights from the 2nd task to the 15th task (i.e., $\mathbf{w}_2$ to $\mathbf{w}_{15}$) share the same regression coefficients with $\mathbf{w}_1$ on dimensions 3 to 20. However, the first two dimensions of $\mathbf{w}_2$ to $\mathbf{w}_{15}$ are generated by applying a rotation matrix of the form $R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$ to the first two dimensions of $\mathbf{w}_1$, in which θ is evenly spaced between $[0, 2\pi]$. This way, $\mathbf{w}_{15}$ gets back to $\mathbf{w}_1$ and the task i is mostly related to task $(i-1)$ and task $(i+1)$. In other words, the relatedness amongst these 15 tasks forms a circular structure.

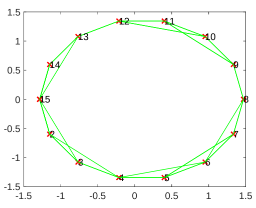
We simulate 200 samples from each task and apply the four different conditional divergence measures to quantify the closeness between pairs of tasks, i.e., the discrepancy from task i to task j is quantified by $D(p_i(y|\mathbf{x}); p_j(y|\mathbf{x}))$. For each measure, we can obtain a 15×15 matrix that encodes all pairwise discrepancies. We then apply the multidimensional scaling (MDS) to project the obtained discrepancy matrix into a 2-dimensional space to visualize the (dis)similarity between individual tasks. We consider two types of input distributions: $\mathbf{x}_t$ follows an isotropic multivariate Gaussian distribution (i.e., $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{20})$) and each element of $\mathbf{x}_t$ is drawn from an uniform distribution $\mathcal{U}(0, 1)$.

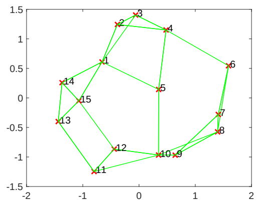
As can be seen from Fig. 4, when input data is Gaussian distributed, our conditional CS divergence (excluding the task 5), the conditional KL divergence, and the conditional von Neumann divergence are capable of identifying a roughly circular structure across all tasks. The result of conditional von Neumann divergence is the closest to a standard circle. This is because the data in each task is Gaussian distributed and does not contain mean shift, such that the 2nd moment information in covariance matrix C is sufficient to distinguish two distributions. The conditional MMD can discover precisely the locality relationships (e.g., tasks $2, 3, 4$ are closely related and tasks $1, 14, 15$ are closely related). However, it is hard to identify the global circular structure. When input data is uniformly distributed, the performance of the von Neumann divergence drops significantly, whereas our conditional CS divergence still identifies

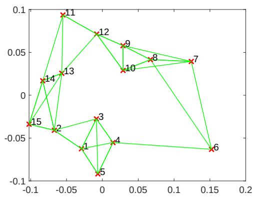
6. Please refer to Appendix C.2 on the details of the permutation test.

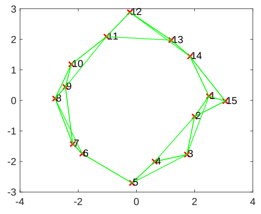
TABLE 2
Power test for conditional CS divergence, conditional KL divergence (with k-NN graph estimator), conditional Bregman divergence (operated on covariance matrix C), and conditional MMD.

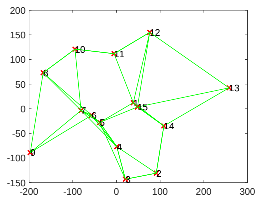
	Conditional CS					Conditional KL					von Neumann (C)					Conditional MMD				
	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
(a)	0.05	1	1	1	1	0.03	1	1	1	1	0.03	0.02	0.86	1	1	0.06	0	0	0	0
(b)	1	0.05	1	1	1	1	0.05	0.98	1	1	0.04	0.09	0.87	1	1	0	0.07	0	0	0
(c)	1	1	0.05	1	1	0.99	0.99	0.06	1	1	0.87	0.88	0.06	1	1	0	0	0.02	0	0
(d)	1	1	1	0.08	0.92	1	1	1	0.03	0.79	1	1	1	0.07	0.11	0	0	0	0.04	0
(e)	1	1	1	0.91	0.10	1	1	1	0.79	0.04	1	1	1	0.14	0.04	0	0	0	0	0.10

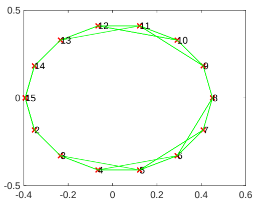

(a) Ground Truth

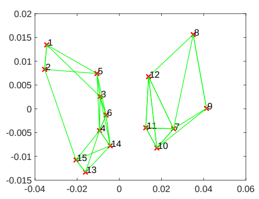

(b) Conditional CS

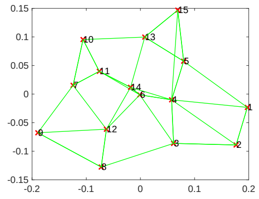

(c) Conditional KL

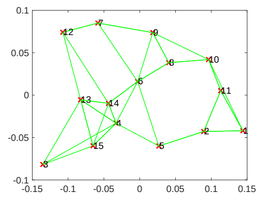

(d) Conditional von Neumann


(e) Conditional MMD


(f) Ground Truth


(g) Conditional CS


(h) Conditional KL


(i) Conditional von Neumann

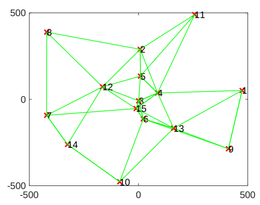

(j) Conditional MMD

Fig. 4. The ground truth task structure (first column) and that is learned by the conditional CS divergence (second column); the conditional KL divergence (third column); the conditional von Neumann divergence (fourth column); and the conditional MMD (fifth column) when the input variable $\mathbf{x}$ is Gaussian distributed (first row) and uniform distributed (second row), respectively. We connect each task with its 3 nearest tasks.

that tasks $7-12$ are closely related, forming a small circle, and tasks $13-15$, $1-6$ are closely related, forming another small circle. It should be noted that our measure only missed capturing the locality relationship between tasks 6 and 7. In contrast, both conditional KL divergence and conditional MMD revealed a few spurious relationships. For example, task 6 with respect to tasks 12 and 14 in Fig. 4(h) and task 6 with respect to tasks $3, 13, 15$ in Fig. 4(j).

The quantitative evaluation in Table 3 is consistent with the visualization results, where we utilize two measures to quantify the closeness between the estimated graph structure and the ground truth. The first measure involves calculating the percentage p based on the difference between the estimated graph adjacency matrix A_S and the true adjacency matrix A_G. The second measure is the geodesic distance $d_{\vec{x}}(L_G, L_S)$ [54]:

$$d_{\vec{x}}(L_G, L_S) = \operatorname{arccosh}\left(1 + \frac{\|(L_G - L_S)\vec{x}\|_2^2\|\vec{x}\|_2^2}{2(\vec{x}^T L_G \vec{x})(\vec{x}^T L_S \vec{x})}\right), \quad (33)$$

in which L_G is the ground truth graph Laplacian, L_S is the estimated graph Laplacian, and we select $\vec{x}$ to be the smallest non-trivial eigenvector of L_G which encodes the global structure of a graph. For both measures, a smaller value indicates better performance.

TABLE 3
Quantitative evaluation on task structure discovery for Gaussian input (left of /) and uniform input (right of /).

Methods	cond. CS	cond. KL	cond. vN	cond. MMD
$p(A_G \neq A_S)$	0.196/**0.187**	0.204/0.293	**0.062**/0.293	0.231/0.418
$d_{\vec{x}}(L_G, L_S)$	1.785/**2.607**	2.246/2.696	**1.494**/2.658	2.872/3.273

5 Applications to Time Series Data and Sequential Decision Making

Our conditional CS divergence can be used in diverse applications associated with time series and sequential data. In the following, we comprehensively evaluate its performances against other SOTA methods in time series clustering and exploration in the absence of explicit rewards.

5.1 Time Series Clustering

Time series clustering is an unsupervised machine learning technique to partition time series data into groups. The similarity based approach is a dominating direction for time series clustering, in which the general idea is to infer the similarity (or distance) between pairwise time series and perform clustering based on the obtained similarities.

Popular time series similarity measures include for example dynamic time warping (DTW) [55], the time warp edit distance (TWED) [56], and the move-split-merge

(MSM) [57]. However, many of these measures cannot be straightforwardly applied to multivariate time series as they did not take relations between different attributes into account [58]. The recently proposed learned pattern similarity (LPS) [59] and time-series cluster kernel (TCK) [60] are two exceptions. Both measures rely on an ensemble strategy to compute the similarity, whereas the latter leverages the Gaussian mixture model (GMM) to fit the data which makes it very effective to deal with missing values.

In contrast to the above-mentioned measures that aim to align two sequences locally or rely on ensemble strategies which is computationally expensive, we suggest a new way to measure time series similarity from a probabilistic perspective. Specifically, given two time series $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$, let K denote a predefined embedding dimension, the conditional distributions $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_{t-2},\cdots,\mathbf{x}_{t-K})$ and $p(\mathbf{y}_t|\mathbf{y}_{t-1},\mathbf{y}_{t-2},\cdots,\mathbf{y}_{t-K})$ characterize the predictive behavior or *dynamics* of $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$, respectively. Hence, our new measure directly evaluates the dissimilarity between $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$ by the conditional CS divergence between $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_{t-2},\cdots,\mathbf{x}_{t-K})$ and $p(\mathbf{y}_t|\mathbf{y}_{t-1},\mathbf{y}_{t-2},\cdots,\mathbf{y}_{t-K})$, i.e.,

$$D_{\text{CS}}(p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_{t-2},\cdots,\mathbf{x}_{t-K});p(\mathbf{y}_t|\mathbf{y}_{t-1},\mathbf{y}_{t-2},\cdots,\mathbf{y}_{t-K})). \tag{34}$$

That is, we expect to quantify the closeness of two (multivariate) time series by measuring the discrepancy of their internal *dynamics*. One can also understand our conditional divergence in Eq. (34) from a kernel adaptive filtering (KAF) [61] perspective. Specifically, given a time series $\{\mathbf{x}_t\}$ (or $\{\mathbf{y}_t\}$), the KAF aims to learn a nonlinear (kernel) regression function $f_{\mathbf{x}}$ (or $f_{\mathbf{y}}$) to predict $\mathbf{x}_t$ (or $\mathbf{y}_t$) using its past K values $\{\mathbf{x}_{t-1},\mathbf{x}_{t-2},\cdots,\mathbf{x}_{t-K}\}$ (or $\{\mathbf{y}_{t-1},\mathbf{y}_{t-2},\cdots,\mathbf{y}_{t-K}\}$) in an online manner, i.e., $\mathbf{x}_t = f_{\mathbf{x}}(\mathbf{x}_{t-1},\mathbf{x}_{t-2},\cdots,\mathbf{x}_{t-K})$ and $\mathbf{y}_t = f_{\mathbf{y}}(\mathbf{y}_{t-1},\mathbf{y}_{t-2},\cdots,\mathbf{y}_{t-K})$. Here, K is also called the filter order. Obviously, $f_{\mathbf{x}}$ is an approximation to $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_{t-2},\cdots,\mathbf{x}_{t-K})$. Hence, our divergence can also be interpreted as the discrepancy between filters $f_{\mathbf{x}}$ and $f_{\mathbf{y}}$. However, we would like to emphasize that, although our divergence has such an interpretation, it does not mean we need to explicitly learn filters $f_{\mathbf{x}}$ and $f_{\mathbf{y}}$ or identify their parameters. That is, our divergence is model-free.

For simplicity, we assume $\{\mathbf{x}_t\}$ and $\{\mathbf{y}_t\}$ have the same length L. For each time series, we can reformulate the sequential observations into a so-called *Hankel matrix* of size $(K+1)\times(L-K)$ as shown in Fig. 5. That is, for sample index i, we have a vector observation $\{\mathbf{x}_i,\mathbf{x}_{i+1},\cdots,\mathbf{x}_{i+K-1}\}$ (or $\{\mathbf{y}_i,\mathbf{y}_{i+1},\cdots,\mathbf{y}_{i+K-1}\}$) and its corresponding desired response $\mathbf{x}_{i+K}$ (or $\mathbf{y}_{i+K}$). Then, the problem reduces to how to estimate Eq. (34) from the $L-K$ pairs of observations $\{[\mathbf{x}_i,\mathbf{x}_{i+1},\cdots,\mathbf{x}_{i+K-1}]^T,\mathbf{x}_{i+K}\}_{i=1}^{L-K}$ that are drawn from $p(\mathrm{X})$ and another $L-K$ observations $\{[\mathbf{y}_i,\mathbf{y}_{i+1},\cdots,\mathbf{y}_{i+K-1}]^T,\mathbf{y}_{i+K}\}_{i=1}^{L-K}$ that are drawn from $q(\mathrm{Y})$, in which $\mathrm{X}\in\mathbb{R}^{K+1}$ and $\mathrm{Y}\in\mathbb{R}^{K+1}$. By Eq. (56), we only need to compute eight Gram matrices of size $(L-K)\times(L-K)$, without any parametric model or parametric assumptions on the underlying distribution. For example, the Gram matrix L^p for the desired response variable is evaluated as $(L^p)_{ij}=\kappa(\mathbf{x}_{i+K}-\mathbf{x}_{j+K})$; whereas the cross Gram matrix L^{pq} is $(L^{pq})_{ij}=\kappa(\mathbf{x}_{i+K}-\mathbf{y}_{j+K})$, in which $1\leq i,j\leq L-K$.

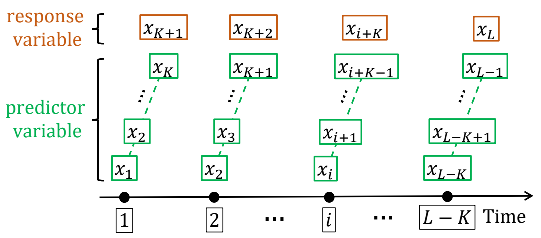
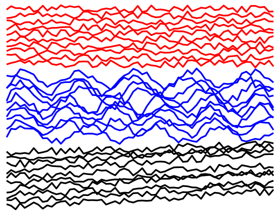
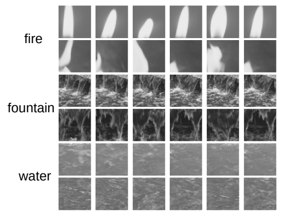


Fig. 5. Reformulating a time series $\{\mathbf{x}_t\}$ into a *Hankel* matrix.



(a) 10 time series from classes "Normal", "Cyclic", and "Increasing trend" in Synthetic Control dataset



(b) Example snapshots of classes "fire", "fountain", "water" in UCLA dataset

Fig. 6. Exemplar time series in (a) Synthetic Control; and (b) UCLA datasets. Each row is a time series.

We test our measure on 12 benchmark time series datasets from the UCI[7] and UCR[8] databases, which cover a wide spectrum of domains, ranging from biomedical data to industrial processes. Additionally, we also test on two challenging dynamic texture (DT) datasets: Highway Traffic [62] and UCLA [63]. The DT is a sequence of images of moving scenes such as flames, smoke, and waves that exihibits certain stationarity in time. The statistics of all datasets are summarized in Appendix C.3.1. Fig. 6 demonstrates exemplar time series from different classes in Synthetic Control and UCLA datasets. For datasets like Traffic and UCLA, the dimension d is substantially larger than the length T, posing significant challenges for clustering tasks.

Our measure is compared to four other similarity measures, namely DTW, TWED, MSM, and TCK. The original DTW can only be applied to univariate time series, which has later been extended for multivariate scenario [64]. In our work, we use a state-of-the-art (SOTA) multivariate implementation in [65] for comparison. Details of all competing measures and their hyperparameters setting are discussed in Appendix C.3.2. For our method, similar to the KAF, the embedding dimension or filter order K is a crucial parameter. Practically, we can rely on Takens' embedding theorem [66] or set it heuristically with some prior knowledge. In our experiment, we observed that the Taken's embedding usually underestimate the value of K that gives the best clustering performance. Therefore, for all the univariate time series, the value of K is selected among 3 values: 10, 15 and 20. For multivariate time series, such as PenDigits and Robot failures, we set $K=1$ due to data prior knowledge. For Traffic and UCLA, we also set $K=1$, because the frame dimension is much larger than time series length.

7. https://archive.ics.uci.edu/ml/datasets.php
8. https://www.cs.ucr.edu/~eamonn/time_series_data/

TABLE 4
Clustering performance comparison (by spectral clustering) in terms of normalized mutual information (NMI). "-" indicates the corresponding measures cannot be extended to multivariate time series or fail to obtain meaningful results. The best performance is in bold; the second best performance is underlined.

Datasets	DTW	MSM	TWED	TCK	**Cond. CS (ours)**
Coffee	0.689	0.592	0.689	0.689	**1**
Diatom	0.788	**0.837**	0.774	0.806	0.743
DistalPhalanxTW	0.570	0.522	0.571	0.491	**0.584**
ECG5000	**0.833**	0.777	0.725	0.695	0.727
FaceAll	0.508	0.796	**0.849**	0.559	0.750
Synthetic control	0.744	0.565	0.565	**0.781**	0.758
PenDigits	**0.725**			0.488	0.655
Libras	**0.652**			0.554	0.606
uWave	**0.834**			0.624	0.759
Robot failure LP1	0.0607			0.176	**0.686**
Robot failure LP2	0.227			0.363	**0.438**
Robot failure LP3	0.170			0.144	**0.328**
Robot failure LP4	0.241			0.080	**0.516**
Robot failure LP5	0.091			0.080	**0.393**
Traffic	0.144			-	**0.145**
UCLA	0.149			-	**0.559**

TABLE 5
Clustering performance comparison (by k-medoids) in terms of normalized mutual information (NMI). "-" indicates the corresponding measures cannot be extended to multivariate time series or fail to obtain meaningful results. The best performance is in bold; the second best performance is underlined.

Datasets	DTW	MSM	TWED	TCK	**Cond. CS (ours)**
Coffee	0.592	0.258	0.689	0.811	**1**
Diatom	0.552	0.768	0.760	**0.821**	0.730
DistalPhalanxTW	0.484	0.456	0.495	0.458	**0.587**
ECG5000	**0.846**	0.756	0.707	0.727	0.595
FaceAll	0.694	0.722	**0.849**	0.584	0.705
Synthetic control	**0.909**	0.899	0.856	0.869	0.745
PenDigits	**0.722**			0.416	0.640
Libras	0.440			**0.596**	0.390
uWave	0.748			0.707	**0.754**
Robot failure LP1	0.132			0.455	**0.658**
Robot failure LP2	0.232			0.315	**0.427**
Robot failure LP3	0.106			0.149	**0.453**
Robot failure LP4	0.084			0.113	**0.365**
Robot failure LP5	0.098			0.234	**0.261**
Traffic	0.139			-	**0.152**
UCLA	0.381			-	**0.416**

For each test dataset, we can obtain a $N \times N$ dissimilarity matrix D that encodes all pairwise dissimilarities using any competing measures (N is the number of time series). To quantitatively evaluate the quality of D, we apply two clustering methods on top of D: k-medoids and spectral clustering [67]. To apply spectral clustering, we first convert the dissimilarity matrix D to a valid adjancency matrix A by a kernel smooth function, i.e., $A_{ij} = \exp\left(-D_{ij}/b\right)$. For each measure, the parameter b is chosen from $\{0.1, 0.2, 1, 2, 10, 20\}$, and the best performance is taken. We use normalized mutual information (NMI) as the clustering evaluation metric. Please refer to [68] for detail definitions of NMI. Table 4 and Table 5 summarize the clustering results using, respectively, spectral clustering and k-medoids. We can summarize a few observations: 1) the clustering performances in terms of two different clustering methods roughly remain consistent; 2) there is no obvious winner for univariate time series, all methods can achieve competitive performance; this makes sense, as DTW, MSM, TWED and TCK are all established methods; 3) our conditional CS divergence has obvious performance gain for multivariate time series; it is also generalizable to Traffic and UCLA, in which the dimension is significantly larger than the length; 4) the performance of our conditional CS divergence is stable in the sense that our measure does not have failing case; by contrast, DTW get very low NMI values in Robert failure LP1-LP5, whereas TCK completely fails in Traffic and UCLA.

Finally, one should note that, different from DTW and other competing measures, our conditional CS divergence is not specifically designed for just measuring similarity between two temporal sequences, it is versatile, flexible, and has wide usages in other machine learning problems.

Apart from the above quantitative analysis, we provide in Appendix C.3.3 how to perform exploratory data analysis for real-world time series data to identify useful patterns. Additionally, we planned to conduct the same experiment using conditional KL divergence and conditional MMD. However, applying conditional KL divergence straightforwardly presents a challenge as the kNN estimator may yield negative values with a noticeable likelihood, hindering the application of spectral clustering or k-medoids. This issue is not encountered with conditional CS. On the other hand, the performance of conditional MMD is always poor, due to the difficulty of tuning the additional hyperparameter λ in Eq. (8). Furthermore, the computational burden of conditional MMD is cubic, stemming from matrix inversion.

5.2 Uncertainty-Guided Exploration for Sequential Decision Making

Our second application deals with sequential decision making under uncertainty, especially in the scenarios where clear rewards are sparse or not available. Let us recall a (discounted) Markov Decision Process (MDP) defined by a 5-tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the set of all possible states (state space), $\mathcal{A}$ is the set of all possible actions (action space). The agent learns a policy function $\pi : \mathcal{S} \mapsto \mathcal{A}$ mapping states to actions at every time step. $P : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$ is the transition function or probability. At each timestep t, upon observing the state s_t, the execution of action a_t triggers an extrinsic reward $r_t = r(s_t, a_t) \in R$ given by the reward function $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, and a transition to a new state $s_{t+1} \sim P(\cdot|s_t, a_t)$. The optimal agent maximizes the discounted cumulative extrinsic reward $Q = \mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_t, a_t\right)$, where $\gamma \in (0, 1)$ is a discounting factor which decays rewards received further in the future.

Even though the reward paradigm is fundamentally flexible in many ways, it is also brittle and limits the agent's ability to learn about its environment, especially when the rewards are sparse or unavailable. To make agents to discover the environment without the requirement of a reward signal, such that the learned policy is more flexible and generalizable, recent studies seek to find an alternative form of reinforcement learning with objective that is reward-independent and favors exploration [69].

Many exploration schemes have been developed over the past years. Here, we consider an uncertainty-guided exploration in the sense that if a state has not been visited sufficiently for the agent to be familiar with it, then that state will have high uncertainty and an agent will be driven to it. This ensures that an agent will thoroughly investigate new areas of the action-state space [70].

At each timestep t, we observe a transition of state $s_t \rightarrow s_{t+1}$, we can update our belief distribution $p(s_{t+1}|s_t, a_t)$. Following the observation of a new state transition, we can compute the divergence $D(p_{\text{new}}(s_{t+1}|s_t, a_t); p_{\text{old}}(s_{t+1}|s_t, a_t))$ between the two belief functions, before and after incorporating the new knowledge. In practice, we introduce a replay buffer to record 2τ steps of $\{s_{t+1}, s_t, a_t\}$ trios. Here we define p_{old} to be the conditional distribution of trios in the old half of the buffer, while p_{new} corresponds the new half. When a new state transition is observed that greatly changes the model, this means that the agent was quite uncertain about the outcome of the action taken in that particular state. In order to encourage exploration of the agent to states that have not been visited sufficiently, the action selection module should be trained by maximizing $D(p_{\text{new}}(s_{t+1}|s_t, a_t); p_{\text{old}}(s_{t+1}|s_t, a_t))$. We term such exploration strategy the "divergence-to-go (DTG)", and the optimal policy π_{dtg} aims to maximize the divergence between old and new experience at each visited state:

$$\pi_{\text{dtg}} = \underset{\pi}{\text{argmax}}\mathbb{E}\left(\sum_{t=0}^{\infty} D(p_{\text{new}}(s_{t+1}|s_t, a_t); p_{\text{old}}(s_{t+1}|s_t, a_t))\right). \tag{35}$$

The DTG exploration was initially proposed in [71], but the authors estimate $D(p_{\text{new}}(s_{t+1}|s_t, a_t); p_{\text{old}}(s_{t+1}|s_t, a_t))$ by the Euclidean distance and make a few additional assumptions on the underlying distribution of $p(s_{t+1}|s_t, a_t)$ to make the estimation tractable. In this section, we straightforwardly estimate the conditional CS divergence between $p_{\text{new}}(s_{t+1}|s_t, a_t)$ and $p_{\text{old}}(s_{t+1}|s_t, a_t)$ using Eq. (56) in which we treat s_{t+1} as variable $\mathbf{y}$ and the concatenation of $[s_t, a_t]$ as variable $\mathbf{x}$, without any distributional assumptions. We denote our improved exploration method the DTG-CS. Identical to the original DTG, we also incorporate a kernelized Q-learning backbone [72] into our methodology. The detailed algorithm of DTG-CS and its major differences to standard reward-based Q-learning are provided in Appendix C.4.

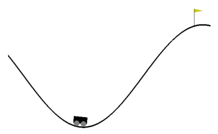
We apply our method to a mountain car, pendulum and maze task, as shown in Fig. 7. The mountain car is a benchmark reinforcement learning (RL) test bed in which the agent attempts to drive an underpowered car up a 2-dimensional hill. Each time step, the agent selects an action from $\mathcal{A} = \{-1, 0, 1\}$ based on the 2-dimensional state space $\mathcal{S}$ consisting of the cart's position and velocity. Zero represents no action. Negative actions drive the cart to the left while positive actions drive it to the right. To reach the goal, the agent must climb the hill to the left and then allow the combination of gravity and positive actions to drive the cart to the top of the hill on the right. The pendulum is another classic control problem in reinforcement learning, which consists of a pendulum attached at one end to a fixed point, and the other end being free. We set it to start at downward position and the goal is to apply torque on the free end to swing it into an upright position. Each step, the agent chooses a action (torque) from the range $[-1, +1]$. In maze game, an agent is placed from a start (red point). The goal of the agent is to reach the exit (blue point) as quickly as possible. For every step, the agent must decide which action to take among four options: left, right, up or down.

We compare our DTG-CS with basic Q-learning, the random exploration, the baseline DTG [71] and the recently proposed maximum entropy (MaxEnt) exploration [69]. To demonstrate the superiority of conditional CS divergence over its KL divergence and MMD counterparts, we also implement DTG-KL and DTG-MMD. Basic Q-learning usually requires pre-collected state-reward buffer with goal rewards or random policy to avoid converging to local optimum (or even non-optimum), if rewards are sparse. Hence, we introduce random actions using ϵ-greedy ($\epsilon_0 = 0.5$) in the Q-learning for the mountain car and the Maze. MaxEnt encourages efficient discovery of an unknown state space by maximizing the entropy of state distribution $-\mathbb{E}_{s\in\mathcal{S}}[\log_2 p(s)]$, in which $\mathcal{S}$ is the set of all visited states. We repeat the training process for 100 independent runs with different global seeds and record the mean value of the number of steps used to achieve the goal for the first time. Note that, for maze game, the agent may not reach the exit given sufficient number of training steps (because it has a chance to converge to a dead end). Therefore, we additionally report the success rate over 100 independent runs. If the method cannot achieve the goal within $50,000$ training steps, we regard it as a failure.

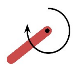
Table 6 shows the quantitative results. For mountain car, our method outperforms all competing methods, although the improvement over DTG is marginal. This is not surprising since mountain car is a simple task. In short, exploration-based methods (MaxEnt, DTG, and its variants including our DTG-CS) can achieve the goal faster than reward-driven Q-learning and random policy. For pendulum, contrast to the other two environments where the agent can only obtain rewards by achieving the goals, the agent of the pendulum receives rewards at arbitrary positions. Closer to the upright position, higher the rewards[9]. Hence, it is not supervising that the reward-based Q-learning outperforms others. Our DTG-CS achieves best result among remaining methods, without explicitly utilizing rewards. For maze, our method outperforms all other methods by a substantial margin. We obtain similar results to basic DTG but improve significantly the success rate. MaxEnt obtains higher success rate than Q-learning and random policy but takes longer time to achieve the goal.

Fig. 8 visualizes results using heatmaps of training steps. For mountain car, we plot the log-probability of occupancy of the 2D state space with location on the X-axis and speed on the Y-axis. In summary, states of Q-learning and random policy cluster near the initial state while exploration-based methods (especially DTG and DTG-CS) evenly distribute in the entire state space. For pendulum, we plot the 2D positions of the free end to show the traces of the bar. A full circle indicates the agent explores the state space

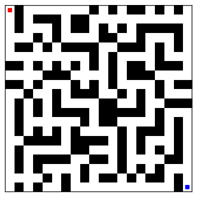
9. https://www.gymlibrary.dev/environments/classic_control/pendulum/



(a) Mountain Car



(b) Pendulum



(c) Maze

Fig. 7. (a) The goal of mountain car is to touch the flag; (b) The goal of Pendulum is to make the bar upright. (c) A 20×20 maze. The goal is to reach the exit (blue point) as quickly as possible from the start (red point).

TABLE 6
For mountain car and Pendulum, we show the number of steps used to achieve the goal for the first time. For Maze, we also show the success rate after "/" given $50,000$ training steps.

Methods	Mountain car (step)	Pendulum (step)	Maze (step/success rate)
Random	42085	5884	15666/0.15
Q-learning	53204	3276	12906/0.18
MaxEnt	13761	5213	31039/0.4
DTG	979	4562	2618/0.58
DTG-KL	1659	4780	4181/0.53
DTG-MMD	1853	4282	4383/0.61
DTG-CS	**946**	**4105**	**2581/0.75**

completely. Random policy stays in the lower semicircle due to gravity. Q-learning can explore the full space with the help of dense rewards. MaxEnt explores more than random but cannot distribute evenly before reaching $50,000$ steps training. DTG-based methods produce similar results to Q-learning but employ no rewards. For maze, we plot 2D state spaces with (x, y) coordinates. States of Q-learning and random policy still cluster near the start point, and the density decreases monotonically with respect to the distance to the exit. MaxEnt produces two local maximums apart from the initial state (see the middle and the upper right corner) which makes it has a larger probability to stuck in dead end. DTG-based approach avoids both limitations.

6 Conclusions and Implications for Future Work

We developed the conditional Cauchy-Schwarz (CS) divergence to quantify the closeness between two conditional distributions from samples, which can be elegantly evaluated with kernel density estimator (KDE). Our conditional CS divergence enjoys simultaneously relatively lower computational complexity, differentiability, and faithfulness guarantee. The new divergence can be applied to a variety of time series and sequential decision making applications in a versatile way. With regard to time series clustering, it demonstrated obvious performance gain for multivariate or high-dimensional time series. With regard to reinforcement learning without explicit rewards, it outperforms the popular maximum entropy strategy and encourages significantly exploration to states that have not been visited sufficiently for the agent to be familiar with it. We additionally analyzed two special cases of conditional CS divergence and illustrated their implications in other challenging areas such as time series causal discovery and the loss function design of deep regression models.

Our research opens up exciting avenues for future work, especially from an application perspective. Given the promising performance of causal score with CS divergence $(\text{CS})^2$ on benchmark simulated data in Fig. 3, it would be interesting to systematically investigate its ability on causal discovery for realistic data in other domains, such as neuroscience, econometrics, and climate science. Meanwhile, it is compelling to apply our conditional CS divergence estimator in Eq. (56) and a by-product on sample efficient measure on conditional independence in Eq. (79) to different downstream tasks that aim to learn *invariant* and *fair* latent representations (see also discussions in Sec. 3.2.2).

We conclude this paper with an initial investigation to unsupervised domain adaptation (UDA) on EEG data. Our motivation is to inspire interested readers to jointly investigate more wider applications of the conditional CS divergence and unveil its undiscovered theoretical properties.

In UDA, we have M labeled samples $\mathcal{D}_s = \{(\mathbf{x}_s^i, y_s^i)\}_{i=1}^M$ from a source domain with distribution p_s and N unlabeled samples $\mathcal{D}_t = \{\mathbf{x}_t^j\}_{j=1}^N$ from a target domain with distribution p_t ($p_t \neq p_s$). The primary goal of UDA is to learn a neural network $h_\theta = f \circ g$ such that the risk on the target domain is minimized, where $f : \mathbf{x} \mapsto \mathbf{z}$ is a feature extractor and $g : \mathbf{z} \mapsto y$ is a classifier. We denote the prediction by $\hat{y} = g(\mathbf{z})$. According to [73], the loss l_{test} in the test distribution (a.k.a., target domain) is upper bounded by:

$$l_{\text{test}} \leq l_{\text{train}} + \frac{M}{\sqrt{2}}\sqrt{D_{\text{KL}}(p_t(\mathbf{z}); p_s(\mathbf{z})) + D_{\text{KL}}(p_t(y|\mathbf{z}); p_s(y|\mathbf{z}))}, \tag{36}$$

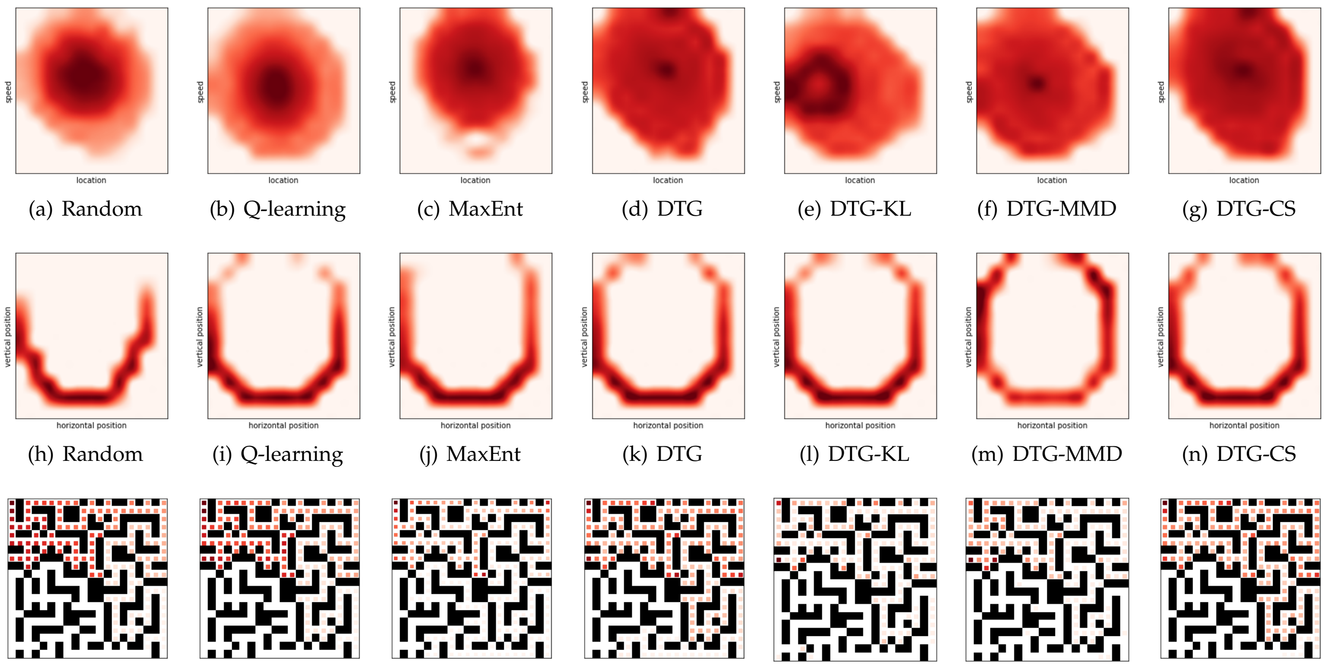
in which l_{train} is the loss in source domain, and it is assumed that $-\log \hat{p}(y|\mathbf{z})$ is upper bounded by a constant M.

Eq. (36) implies that achieving small test error necessitates matching both the marginal distribution $p(\mathbf{z})$ and the conditional distribution $p(y|\mathbf{z})$, which aligns with [23]. Motivated by this theoretical basis, it makes sense to replace KL divergence (due to difficulty of estimation) with our CS divergence, and optimize the following objective:

$$L_{\text{CE}} + \alpha \left[D_{\text{CS}}(p_t(\mathbf{z}); p_s(\mathbf{z})) + D_{\text{CS}}(p_t(\hat{y}|\mathbf{z}); p_s(y|\mathbf{z})) \right], \tag{37}$$

where $L_{\text{CE}} = -\frac{1}{M}\sum_{i=1}^M y_s^i \log \hat{y}_s^i$ is the cross-entropy loss on source domain and $\alpha > 0$ controls the strength of regularization. Note that, in target domain, there is no ground truth y_t. Hence, we approximate y_t with its prediction $\hat{y}_t = h_\theta(x_t)$, which is a common trick in UDA literature.

We apply CS divergence-based domain adaptation to EEG classification on two benchmark real-world datasets, namely BCI Competition IIIb [74] and SEED [75]. We also compare our method with representative MMD-based approaches, which include DDC [76], DAN [77], JAN [24], DJP-MMD [78]. The baseline model is EEGNet [79], and $\mathbf{z}$ is selected to be the latent representation before the final linear classification layer. An detailed discussion on experimental setup and the difference amongst all competing approaches are provided in Appendix C.5. According to Table 7, our CS divergence always outperforms the best MMD-based approaches.



(a) Random (b) Q-learning (c) MaxEnt (d) DTG (e) DTG-KL (f) DTG-MMD (g) DTG-CS

(h) Random (i) Q-learning (j) MaxEnt (k) DTG (l) DTG-KL (m) DTG-MMD (n) DTG-CS

(o) Random (p) Q-learning (q) MaxEnt (r) DTG (s) DTG-KL (t) DTG-MMD (u) DTG-CS

Fig. 8. The log-probability of occupancy of the two-dimensional state space in mountain car (first row), the log-probability of occupancy of x-y coordinates in Pendulum (second row) and the heatmap of traces for maze within $50,000$ training steps (third row). We repeat the experiment 100 times with different random seeds and show the average results.

TABLE 7
Classification accuracy for different approaches.

Methods	S4 $\rightarrow$ X11	X11 $\rightarrow$ S4	SEED
EEGNet (no adaptation)	0.608	0.718	0.515
DDC	0.610	0.766	0.564
DAN	0.615	0.768	0.547
JAN	0.609	0.771	0.543
DJP-MMD	0.628	0.768	0.536
CS (ours)	**0.631**	**0.801**	**0.578**

REFERENCES

[1] R. Flamary, N. Courty, D. Tuia, A. Rakotomamonjy, Optimal transport for domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (9) (2017) 1853–1865.
[2] A. Cherian, S. Sra, A. Banerjee, N. Papanikolopoulos, Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (9) (2012) 2161–2174.
[3] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International conference on machine learning, PMLR, 2017, pp. 214–223.
[4] A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, A. Smola, A kernel statistical test of independence, Advances in neural information processing systems 20.
[5] X. Wang, W. Pan, W. Hu, Y. Tian, H. Zhang, Conditional distance correlation, Journal of the American Statistical Association 110 (512) (2015) 1726–1734.
[6] E. Amid, M. K. Warmuth, R. Anil, T. Koren, Robust bi-tempered logistic loss based on bregman divergences, Advances in Neural Information Processing Systems 32.
[7] A. Basu, H. Shioya, C. Park, Statistical inference: the minimum distance approach, CRC press, 2011.
[8] A. Müller, Integral probability metrics and their generating classes of functions, Advances in Applied Probability 29 (2) (1997) 429–443.
[9] A. Rényi, et al., On measures of entropy and information, in: Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, Vol. 1, Berkeley, California, USA, 1961.
[10] I. Csiszár, Information-type measures of difference of probability distributions and indirect observation, studia scientiarum Mathematicarum Hungarica 2 (1967) 229–318.
[11] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, The Journal of Machine Learning Research 13 (1) (2012) 723–773.
[12] S. Zhao, A. Sinha, Y. He, A. Perreault, J. Song, S. Ermon, Comparing distributions by measuring differences that affect decision making, in: International Conference on Learning Representations, 2021.
[13] J. C. Principe, D. Xu, J. Fisher, S. Haykin, Information theoretic learning, Unsupervised adaptive filtering 1 (2000) 265–319.
[14] J. C. Principe, D. Xu, Q. Zhao, J. W. Fisher, Learning from examples with information theoretic criteria, Journal of VLSI signal processing systems for signal, image and video technology 26 (1) (2000) 61–77.
[15] R. Jenssen, J. C. Principe, D. Erdogmus, T. Eltoft, The cauchy–schwarz divergence and parzen windowing: Connections to graph theory and mercer kernels, Journal of the Franklin Institute 343 (6) (2006) 614–629.
[16] K. Kampa, E. Hasanbelliu, J. C. Principe, Closed-form cauchy-schwarz pdf divergence for mixture of gaussians, in: The 2011 International Joint Conference on Neural Networks, IEEE, 2011, pp. 2578–2585.
[17] E. Parzen, On estimation of a probability density function and mode, The annals of mathematical statistics 33 (3) (1962) 1065–1076.
[18] H. G. Hoang, B.-N. Vo, B.-T. Vo, R. Mahler, The cauchy–schwarz divergence for poisson point processes, IEEE Transactions on Information Theory 61 (8) (2015) 4475–4485.
[19] M. Kampffmeyer, S. Løkse, F. M. Bianchi, L. Livi, A.-B. Salberg, R. Jenssen, Deep divergence-based approach to clustering, Neural Networks 113 (2019) 91–101.
[20] D. P. Kingma, M. Welling, Auto-encoding variational {Bayes}, in: International Conference on Learning Representations, 2014.
[21] L. Tran, M. Pantic, M. P. Deisenroth, Cauchy-schwarz regularized autoencoder, Journal of Machine Learning Research 23.
[22] P. He, P. Emami, S. Ranka, A. Rangarajan, Self-supervised robust scene flow estimation via the alignment of probability density

functions, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 861–869.

[23] H. Zhao, R. T. Des Combes, K. Zhang, G. Gordon, On learning invariant representations for domain adaptation, in: International Conference on Machine Learning, PMLR, 2019, pp. 7523–7532.

[24] M. Long, H. Zhu, J. Wang, M. I. Jordan, Deep transfer learning with joint adaptation networks, in: International conference on machine learning, PMLR, 2017, pp. 2208–2217.

[25] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2272–2281.

[26] M. Federici, A. Dutta, P. Forré, N. Kushman, Z. Akata, Learning robust representations via multi-view information bottleneck, in: International Conference on Learning Representations, 2020.

[27] Y. Ren, J. Zhu, J. Li, Y. Luo, Conditional generative moment-matching networks, Advances in Neural Information Processing Systems 29.

[28] L. Song, J. Huang, A. Smola, K. Fukumizu, Hilbert space embeddings of conditional distributions with applications to dynamical systems, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 961–968.

[29] Q. Wang, S. R. Kulkarni, S. Verdú, Divergence estimation for multidimensional densities via k-nearest-neighbor distances, IEEE Transactions on Information Theory 55 (5) (2009) 2392–2405.

[30] M. Noshad, K. R. Moon, S. Y. Sekeh, A. O. Hero, Direct estimation of information divergence using nearest neighbor ratios, in: 2017 IEEE International Symposium on Information Theory (ISIT), IEEE, 2017, pp. 903–907.

[31] L. Song, K. Fukumizu, A. Gretton, Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models, IEEE Signal Processing Magazine 30 (4) (2013) 98–111.

[32] S. Yu, A. Shaker, F. Alesiani, J. Principe, Measuring the discrepancy between conditional distributions: methods, properties and applications, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 2777–2784.

[33] B. Kulis, M. A. Sustik, I. S. Dhillon, Low-rank kernel learning with bregman matrix divergences., Journal of Machine Learning Research 10 (2).

[34] M. A. Nielsen, I. L. Chuang, Quantum computation and quantum information.

[35] J. Zhao, D. Meng, Fastmmd: Ensemble of circular discrepancy for efficient two-sample test, Neural computation 27 (6) (2015) 1345–1372.

[36] J. Park, K. Muandet, A measure-theoretic approach to kernel conditional mean embeddings, Advances in neural information processing systems 33 (2020) 21247–21259.

[37] B. Rodriguez Galvez, The information bottleneck: Connections to other problems, learning and exploration of the ib curve, https://www.diva-portal.org/smash/get/diva2:1332068/FULLTEXT01.pdf.

[38] X. Chen, S. Kim, Q. Lin, J. G. Carbonell, E. P. Xing, Graph-structured multi-task regression and an efficient optimization method for general fused lasso, arXiv preprint arXiv:1005.3579.

[39] R. Pogodin, N. Deka, Y. Li, D. J. Sutherland, V. Veitch, A. Gretton, Efficient conditionally invariant representation learning, in: International Conference on Learning Representations, 2023.

[40] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Advances in neural information processing systems 29.

[41] B. Li, Y. Shen, Y. Wang, W. Zhu, D. Li, K. Keutzer, H. Zhao, Invariant information bottleneck for domain generalization, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 7399–7407.

[42] C. K. Assaad, E. Devijver, E. Gaussier, Survey and evaluation of causal discovery methods for time series, Journal of Artificial Intelligence Research 73 (2022) 767–819.

[43] J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, B. Schölkopf, Distinguishing cause from effect using observational data: methods and benchmarks, The Journal of Machine Learning Research 17 (1) (2016) 1103–1204.

[44] C. W. Granger, Investigating causal relations by econometric models and cross-spectral methods, Econometrica: journal of the Econometric Society (1969) 424–438.

[45] C. W. Granger, Testing for causality: A personal viewpoint, Journal of Economic Dynamics and control 2 (1980) 329–352.

[46] L. Su, H. White, A nonparametric hellinger metric test for conditional independence, Econometric Theory 24 (4) (2008) 829–864.

[47] T. Schreiber, Measuring information transfer, Physical review letters 85 (2) (2000) 461.

[48] D. Kugiumtzis, Direct-coupling information measure from nonuniform embedding, Physical Review E 87 (6) (2013) 062918.

[49] B. Gourévitch, R. L. Bouquin-Jeannès, G. Faucon, Linear and nonlinear causality between signals: methods, examples and neurophysiological applications, Biological cybernetics 95 (4) (2006) 349–369.

[50] D. Marinazzo, M. Pellicoro, S. Stramaglia, Kernel method for nonlinear granger causality, Physical review letters 100 (14) (2008) 144103.

[51] J. Zhu, J.-J. Bellanger, H. Shu, R. Le Bouquin Jeannès, Contribution to transfer entropy estimation via the k-nearest-neighbors approach, Entropy 17 (6) (2015) 4173–4201.

[52] J. X. Zheng, A consistent test of conditional parametric distributions, Econometric Theory 16 (5) (2000) 667–691.

[53] R. Flamary, A. Rakotomamonjy, G. Gasso, Learning constrained task similarities in graphregularized multi-task learning, Regularization, Optimization, Kernels, and Support Vector Machines 103.

[54] G. Bravo Hermsdorff, L. Gunderson, A unifying framework for spectrum-preserving graph sparsification and coarsening, Advances in Neural Information Processing Systems 32.

[55] D. J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: Proceedings of AAAI Workshop on Knowledge Discovery in Databases, 1994, pp. 359–370.

[56] P.-F. Marteau, Time warp edit distance with stiffness adjustment for time series matching, IEEE transactions on pattern analysis and machine intelligence 31 (2) (2008) 306–318.

[57] A. Stefan, V. Athitsos, G. Das, The move-split-merge metric for time series, IEEE transactions on Knowledge and Data Engineering 25 (6) (2012) 1425–1438.

[58] Z. Bankó, J. Abonyi, Correlation based dynamic time warping of multivariate time series, Expert Systems with Applications 39 (17) (2012) 12814–12823.

[59] M. G. Baydogan, G. Runger, Time series representation and similarity based on local autopatterns, Data Mining and Knowledge Discovery 30 (2) (2016) 476–509.

[60] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, R. Jenssen, Time series cluster kernel for learning similarities between multivariate time series with missing data, Pattern Recognition 76 (2018) 569–581.

[61] W. Liu, P. P. Pokharel, J. C. Principe, The kernel least-mean-square algorithm, IEEE Transactions on Signal Processing 56 (2) (2008) 543–554.

[62] A. B. Chan, N. Vasconcelos, Classification and retrieval of traffic video using auto-regressive stochastic processes, in: IEEE Proceedings. Intelligent Vehicles Symposium, 2005., IEEE, 2005, pp. 771–776.

[63] P. Saisan, G. Doretto, Y. N. Wu, S. Soatto, Dynamic texture recognition, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Vol. 2, IEEE, 2001, pp. II–II.

[64] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, E. Keogh, Generalizing dtw to the multi-dimensional case requires an adaptive approach, Data mining and knowledge discovery 31 (1) (2017) 1–31.

[65] D. Schultz, B. Jain, Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces, Pattern Recognition 74 (2018) 340–358.

[66] F. Takens, Detecting strange attractors in turbulence, in: Dynamical systems and turbulence, Warwick 1980, Springer, 1981, pp. 366–381.

[67] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, Advances in neural information processing systems 14.

[68] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 267–273.

[69] E. Hazan, S. Kakade, K. Singh, A. Van Soest, Provably efficient maximum entropy exploration, in: International Conference on Machine Learning, PMLR, 2019, pp. 2681–2691.

[70] I. J. Sledge, M. S. Emigh, J. C. Príncipe, Guided policy exploration for markov decision processes using an uncertainty-based value-of-information criterion, IEEE transactions on neural networks and learning systems 29 (6) (2018) 2080–2098.

[71] M. Emigh, E. Kriminger, J. C. Principe, A model based approach to exploration of continuous-state mdps using divergence-to-go, in: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2015, pp. 1–6.

[72] J. Bae, P. Chhatbar, J. T. Francis, J. C. Sanchez, J. C. Principe, Reinforcement learning via kernel temporal difference, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2011, pp. 5662–5665.

[73] A. T. Nguyen, T. Tran, Y. Gal, P. Torr, A. G. Baydin, KL guided domain adaptation, in: International Conference on Learning Representations, 2022.

[74] B. Blankertz, K.-R. Muller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. R. Millan, M. Schroder, N. Birbaumer, The bci competition iii: Validating alternative approaches to actual bci problems, IEEE transactions on neural systems and rehabilitation engineering 14 (2) (2006) 153–159.

[75] W.-L. Zheng, B.-L. Lu, Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks, IEEE Transactions on autonomous mental development 7 (3) (2015) 162–175.

[76] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance, arXiv preprint arXiv:1412.3474.

[77] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: International conference on machine learning, PMLR, 2015, pp. 97–105.

[78] W. Zhang, D. Wu, Discriminative joint probability maximum mean discrepancy (djp-mmd) for domain adaptation, in: 2020 international joint conference on neural networks (IJCNN), IEEE, 2020, pp. 1–8.

[79] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, B. J. Lance, Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces, Journal of neural engineering 15 (5) (2018) 056013.

[80] E. García-Portugués, Notes for Nonparametric Statistics, 2024, version 6.9.1. ISBN 978-84-09-29537-1.
URL https://bookdown.org/egarpor/NP-UC3M/

[81] Q. Li, J. S. Racine, Nonparametric econometrics: theory and practice, Princeton University Press, 2023.

[82] J. Beirlant, E. J. Dudewicz, L. Györfi, E. C. Van der Meulen, et al., Nonparametric entropy estimation: An overview, International Journal of Mathematical and Statistical Sciences 6 (1) (1997) 17–39.

[83] P. Duan, F. Yang, S. L. Shah, T. Chen, Transfer zero-entropy and its application for capturing cause and effect relationship between variables, IEEE Transactions on Control Systems Technology 23 (3) (2014) 855–867.

[84] G. Doretto, A. Chiuso, Y. N. Wu, S. Soatto, Dynamic textures, International Journal of Computer Vision 51 (2) (2003) 91–109.

[85] X. You, W. Guo, S. Yu, K. Li, J. C. Príncipe, D. Tao, Kernel learning for dynamic texture synthesis, IEEE Transactions on Image Processing 25 (10) (2016) 4782–4795.

[86] Y. Xu, Y. Quan, H. Ling, H. Ji, Dynamic texture classification using dynamic fractal analysis, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 1219–1226.

[87] L. N. Ferreira, From time series to networks in r with the ts2net package, arXiv preprint arXiv:2208.09660.

[88] J. Bae, L. S. Giraldo, P. Chhatbar, J. Francis, J. Sanchez, J. Principe, Stochastic kernel temporal difference for reinforcement learning, in: 2011 IEEE International Workshop on Machine Learning for Signal Processing, IEEE, 2011, pp. 1–6.

[89] Z. Lan, O. Sourina, L. Wang, R. Scherer, G. R. Müller-Putz, Domain adaptation techniques for eeg-based emotion recognition: a comparative study on two public datasets, IEEE Transactions on Cognitive and Developmental Systems 11 (1) (2018) 85–94.

[90] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32.

Appendix A Background Knowledge

A.1 On Kernel Functions in Multivariate Kernel Density Estimation

Kernel density estimation (KDE) can be extended to estimate multivariate densities f in $\mathbb{R}^d$ based on the same principle: perform an average of densities "centered" at the data points [80]. Specifically, given M samples $X_1, X_2, \cdots, X_M$ in $\mathbb{R}^d$, the KDE of f evaluated on an arbitrary point $\mathbf{x}$ is given by:

$$\hat{f}(\mathbf{x}, \Sigma) := \frac{1}{M|\Sigma|^{1/2}} \sum_{i=1}^{M} K(\Sigma^{-1/2}(\mathbf{x} - X_i)), \tag{38}$$

where K is a d-variate symmetric kernel that is unimodal at $\mathbf{0}$ and that depends on the bandwidth matrix Σ, a symmetric positive definite (SPD) matrix. Then, the bandwidth matrix Σ can be thought of as the variance-covariance matrix of a multivariate normal density with mean X_i and the KDE in Eq. (38) can be regarded as a data-driven mixture of those densities [80].

A common notation is $K_\Sigma(\mathbf{z}) := |\Sigma|^{-1/2} K(\Sigma^{-1/2}\mathbf{z})$, so the KDE can be compactly written as:

$$\hat{f}(\mathbf{x}, \Sigma) := \frac{1}{M} \sum_{i=1}^{M} K_\Sigma(\mathbf{x} - X_i). \tag{39}$$

Considering a full bandwidth matrix Σ gives more flexibility to the KDE, but also quadratically increases the amount of bandwidth parameters ($\frac{d(d+1)}{2}$ in total) that need to be chosen precisely, which notably complicates bandwidth selection as the dimension d grows, and increases the variance of the KDE. A common simplification is to consider a diagonal bandwidth matrix $\Sigma = \mathrm{diag}(\sigma_1^2, \cdots, \sigma_d^2)$, which yields the KDE employing product kernels [80], [81]:

$$\hat{f}(\mathbf{x}, \Sigma) := \frac{1}{M} \sum_{i=1}^{M} \kappa_{\sigma_1}(x_1 - X_{i,1}) \times \kappa_{\sigma_2}(x_2 - X_{i,2}) \times \cdots \times \kappa_{\sigma_d}(x_d - X_{i,d}), \tag{40}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_d]^T$, $X_i = [X_{i,1}, X_{i,2}, \cdots, X_{i,d}]^T$, $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_d]^T$ is the vector of bandwidths, $\kappa_\sigma(\cdot)$ is a univariate kernel function such as Gaussian $\kappa_\sigma(\cdot) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\|\cdot\|^2}{2\sigma^2})$.

Note that, the only assumption in Eq. (40) was that the samples are independent. That is, no restrictions were placed on the s index for each dimension $X_{i,s} (s = 1, 2, \cdots, d)$. The product kernel is used simply for convenience, and it certainly *does not* require that the $X_{i,s}$'s are independent across the s index. In other words, the multivariate KDE in Eq. (40) is capable of capturing general dependence among the different dimension of X_i [81, Chapter 1.6].

In practice, a much simpler and common choice is to consider $\sigma = \sigma_1 = \sigma_2 = \cdots = \sigma_d$.

A.2 *Resubstitution Estimator* and *Plug-in Estimator*

The CS divergence involves the estimation of inner product of two density functions. In fact, quantities like $\int p^2 d\mu$ and $\int pq d\mu$ can be estimated in a couple of ways [82]. In this paper, we use a so-called *resubstitution estimator*; whereas the authors in [15] use the *plug-in estimator* that simply inserting KDE of the density into the formula, i.e.,

$$\begin{aligned} \int p^2 d\mu \approx \int \widehat{p}^2(x) dx &= \int \left(\frac{1}{N} \sum_{i=1}^{N} \kappa_\sigma(x_i - x) \right)^2 dx \\ &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \int \kappa_\sigma(x_i - x) \times \kappa_\sigma(x_j - x) dx. \end{aligned} \tag{41}$$

Authors of [15] then assume a Gaussian kernel and rely on the property that the integral of the product of two Gaussians is exactly evaluated as the value of the Gaussian computed at the difference of the arguments and whose variance is the sum of the variances of the two original Gaussian functions. Hence,

$$\int p^2 d\mu \approx \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \int \kappa_\sigma(x_i - x) \times \kappa_\sigma(x_j - x) dx = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \kappa_{\sqrt{2}\sigma}(x_i - x_j). \tag{42}$$

To our knowledge, other kernel functions, however, do not result in such convenient evaluation of the integral because the Gaussian maintains the functional form under convolution.

By contrast, we estimate $\int p^2 d\mu$ as:

$$\int p^2 d\mu = \mathbb{E}_p(p) = \frac{1}{N} \sum_{i=1}^{N} p(x_i) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{N} \sum_{j=1}^{N} \kappa_\sigma(x_i - x_j) \right) dx = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \kappa_\sigma(x_i - x_j). \tag{43}$$

Although Eq. (43) only differs from Eq. (42) by replacing $\sqrt{2}\sigma$ with σ, our estimator offers two immediately advantages over that in [15]: 1) our estimator is generalizable to all valid kernel functions; and 2) the empirical estimator of conditional CS divergence can be achieved much more easily using only the *resubstitution estimator*.

A.3 CS Divergence and its Connection to MMD

Given M samples $\{\mathbf{y}_i^s\}_{i=1}^M$ drawn from distribution p_s and N samples $\{\mathbf{y}_i^t\}_{i=1}^N$ drawn from distribution p_t. By the kernel density estimation (KDE) with a kernel function κ_σ of width σ[10], we have:

$$\hat{p}_s(\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M} \kappa_\sigma(\mathbf{y}-\mathbf{y}_i^s), \tag{44}$$

and

$$\hat{p}_t(\mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N} \kappa_\sigma(\mathbf{y}-\mathbf{y}_i^t). \tag{45}$$

To evaluate the dissimilarity between p_s and p_t, a natural choice is the Euclidean distance:

$$\begin{aligned} D_{\text{ED}}(p_s;p_t) &= \int (p_s(\mathbf{y})-p_t(\mathbf{y}))^2 dy \\ &= \int p_s^2(\mathbf{y})d\mathbf{y} + \int p_t^2(\mathbf{y})d\mathbf{y} - \int p_s(\mathbf{y})p_t(\mathbf{y})d\mathbf{y} \end{aligned} \tag{46}$$

By Eq. (43), we have:

$$\int p_s^2(\mathbf{y})d\mathbf{y} = \mathbb{E}_{p_s}(p_s) = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \kappa_\sigma(\mathbf{y}_j^s-\mathbf{y}_i^s). \tag{47}$$

Similarly,

$$\int p_t^2(\mathbf{y})d\mathbf{y} = \mathbb{E}_{p_t}(p_t) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa_\sigma(\mathbf{y}_j^t-\mathbf{y}_i^t), \tag{48}$$

and

$$\int p_s(\mathbf{y})p_t(\mathbf{y})d\mathbf{y} = \mathbb{E}_{p_s}(p_t) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} \kappa_\sigma(\mathbf{y}_j^t-\mathbf{y}_i^s). \tag{49}$$

Combining Eqs. (47)-(49), we have:

$$D_{\text{ED}}(p_s;p_t) = \underbrace{\frac{1}{M^2}\sum_{i,j=1}^{M} \kappa_\sigma(\mathbf{y}_j^s-\mathbf{y}_i^s)}_{\text{Ⓐ}} + \underbrace{\frac{1}{N^2}\sum_{i,j=1}^{N} \kappa_\sigma(\mathbf{y}_j^t-\mathbf{y}_i^t)}_{\text{Ⓑ}} - \underbrace{\frac{2}{MN}\sum_{i,j=1}^{M,N} \kappa_\sigma(\mathbf{y}_j^t-\mathbf{y}_i^s)}_{\text{Ⓒ}}. \tag{50}$$

Note that Eq. (50) is exactly the same (in terms of mathematical expression) to the square of MMD using V-statistic estimator [11]:

$$\widehat{\text{MMD}}_v[p_s(\mathbf{x}),p_t(\mathbf{x})] = \left[\frac{1}{M^2}\sum_{i,j=1}^{M} \kappa(\mathbf{y}_i^s,\mathbf{y}_j^s) + \frac{1}{N^2}\sum_{i,j=1}^{N} \kappa(\mathbf{y}_i^t,\mathbf{y}_j^t) - \frac{2}{MN}\sum_{i,j=1}^{M,N} \kappa(\mathbf{y}_j^t,\mathbf{y}_i^s)\right]^{\frac{1}{2}}. \tag{51}$$

Although, in practice, the U-statistic estimator of MMD is more widely used:

$$\widehat{\text{MMD}}_u[p_s(\mathbf{x}),p_t(\mathbf{x})] = \left[\frac{1}{M(M-1)}\sum_{i=1}^{M}\sum_{j\neq i}^{M} \kappa(\mathbf{y}_i^s,\mathbf{y}_j^s) + \frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j\neq i}^{N} \kappa(\mathbf{y}_i^t,\mathbf{y}_j^t) - \frac{2}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} \kappa(\mathbf{y}_j^t,\mathbf{y}_i^s)\right]^{\frac{1}{2}}, \tag{52}$$

On the other hand, by substituting Eqs. (47)-(49) into the definition of CS divergence ($D_{\text{CS}}(p\|q) = -2\log(\int p(\mathbf{y})q(\mathbf{y})d\mathbf{y}) + \log(\int p(\mathbf{y})^2 d\mathbf{y}) + \log(\int q(\mathbf{y})^2 d\mathbf{y})$), we obtain:

$$\widehat{D}_{\text{CS}}(p\|q) = \log\left(\frac{1}{M^2}\sum_{i,j=1}^{M} \kappa_\sigma(\mathbf{y}_j^s-\mathbf{y}_i^s)\right) + \log\left(\frac{1}{N^2}\sum_{i,j=1}^{N} \kappa_\sigma(\mathbf{y}_j^t-\mathbf{y}_i^t)\right) - 2\log\left(\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} \kappa_\sigma(\mathbf{y}_j^t-\mathbf{y}_i^s)\right). \tag{53}$$

To summarize, the mathematical expression of MMD can be derived by either taking the distance of kernel mean embedding in a reproducing kernel Hilbert space (RKHS) or taking the Euclidean distance of two distributions which are estimated by KDE. More interestingly, one can estimate MMD by $\left(\text{Ⓐ} + \text{Ⓑ} - 2\text{Ⓒ}\right)^{\frac{1}{2}}$, and CS divergence by $\log\left(\text{Ⓐ}\right) + \log\left(\text{Ⓑ}\right) - 2\log\left(\text{Ⓒ}\right)$. Note, however, that this observation does not apply to conditional MMD and conditional CS divergence.

10. For example, the most popular Gaussian kernel function is expressed as $\kappa_\sigma(\cdot) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{\|\cdot\|^2}{2\sigma^2})$.

Appendix B
Proofs

B.1 Proof to Proposition 1

Proof. Denote $\tilde{p}(\mathbf{y}) = p(\mathbf{y}|\mathbf{x})$ and $\tilde{q}(\mathbf{y}) = q(\mathbf{y}|\mathbf{x})$. By the CS inequality, we have:

$$\int \tilde{p}(\mathbf{y})\tilde{q}(\mathbf{y})d\mathbf{y} \leq \sqrt{\left(\int \tilde{p}(\mathbf{y})^2 d\mathbf{y}\right)\left(\int \tilde{q}(\mathbf{y})^2 d\mathbf{y}\right)}. \tag{54}$$

Therefore, $D_{\text{CS}}(\tilde{p}(\mathbf{y}); \tilde{p}(\mathbf{y}))$ is non-negative and reduces to zero if and only if $\tilde{p}(\mathbf{y})$ and $\tilde{q}(\mathbf{y})$ are linearly dependent, i.e., $\tilde{p}(\mathbf{y}) = \lambda\tilde{q}(\mathbf{y})$.

On the other hand, $\tilde{p}(\mathbf{y})$ and $\tilde{q}(\mathbf{y})$ are valid probability distributions in the sense that $\int \tilde{p}(\mathbf{y})d\mathbf{y} = \lambda \int \tilde{q}(\mathbf{y})d\mathbf{y} = \int \tilde{q}(\mathbf{y})d\mathbf{y} = 1$, which implies $\lambda = 1$. □

B.2 Proof to Proposition 2

The conditional CS divergence for $p(\mathbf{y}|\mathbf{x})$ and $q(\mathbf{y}|\mathbf{x})$ is expressed as:

$$D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x})) = -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}\right), \tag{55}$$

which contains two conditional quadratic terms (i.e., $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ and $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$) and a cross term (i.e., $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}$).

Given observations $\psi_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^M$ and $\psi_t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^N$ which are sampled from distributions $p(\mathbf{x},\mathbf{y})$ and $q(\mathbf{x},\mathbf{y})$, respectively. Let K^p and L^p denote, respectively, the Gram matrices for the variable $\mathbf{x}$ and the output variable $\mathbf{y}$ in the distribution p. Similarly, let K^q and L^q denote, respectively, the Gram matrices for the variable $\mathbf{x}$ and the output variable $\mathbf{y}$ in the distribution q. Meanwhile, let $K^{pq} \in \mathbb{R}^{M\times N}$ (i.e., $(K^{pq})_{ij} = \kappa(\mathbf{x}_i^s - \mathbf{x}_j^t)$) denote the Gram matrix from distribution p to distribution q for input variable $\mathbf{x}$, and $L^{pq} \in \mathbb{R}^{M\times N}$ the Gram matrix from distribution p to distribution q for output variable $\mathbf{y}$. Similarly, let $K^{qp} \in \mathbb{R}^{N\times M}$ (i.e., $(K^{qp})_{ij} = \kappa(\mathbf{x}_i^t - \mathbf{x}_j^s)$) denote the Gram matrix from distribution q to distribution p for input variable $\mathbf{x}$, and $L^{qp} \in \mathbb{R}^{N\times M}$ the Gram matrix from distribution q to distribution p for output variable $\mathbf{y}$. The empirical estimation of $D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x}))$ is given by:

$$\begin{aligned}\widehat{D}_{\text{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x})) \approx{} & \log\left(\sum_{j=1}^M\left(\frac{\sum_{i=1}^M K_{ji}^p L_{ji}^p}{(\sum_{i=1}^M K_{ji}^p)^2}\right)\right) + \log\left(\sum_{j=1}^N\left(\frac{\sum_{i=1}^N K_{ji}^q L_{ji}^q}{(\sum_{i=1}^N K_{ji}^q)^2}\right)\right) \\ & - \log\left(\sum_{j=1}^M\left(\frac{\sum_{i=1}^N K_{ji}^{pq} L_{ji}^{pq}}{(\sum_{i=1}^M K_{ji}^p)(\sum_{i=1}^N K_{ji}^{pq})}\right)\right) - \log\left(\sum_{j=1}^N\left(\frac{\sum_{i=1}^M K_{ji}^{qp} L_{ji}^{qp}}{(\sum_{i=1}^M K_{ji}^{qp})(\sum_{i=1}^N K_{ji}^q)}\right)\right)\end{aligned} \tag{56}$$

In the following, we first demonstrate how to estimate the two conditional quadratic terms (i.e., $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ and $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$) from samples. We then demonstrate how to estimate the cross term (i.e., $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}$). We finally explain the empirical estimation of $D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x}))$.

Proof. [The conditional quadratic term]

The empirical estimation of $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ can be expressed as:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y} = \mathbb{E}_{p(X,Y)}\left[\frac{p(X,Y)}{p^2(X)}\right] \approx \frac{1}{M}\sum_{j=1}^M \frac{p(\mathbf{x}_j,\mathbf{y}_j)}{p^2(\mathbf{x}_j)}. \tag{57}$$

By kernel density estimator (KDE), we have:

$$\frac{p(\mathbf{x}_j,\mathbf{y}_j)}{p^2(\mathbf{x}_j)} \approx M\frac{\sum_{i=1}^M \kappa_\sigma(\mathbf{x}_j^p - \mathbf{x}_i^p)\kappa_\sigma(\mathbf{y}_j^p - \mathbf{y}_i^p)}{\left(\sum_{i=1}^M \kappa_\sigma(\mathbf{x}_j^p - \mathbf{x}_i^p)\right)^2}. \tag{58}$$

Therefore,

$$\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^M\left(\frac{\sum_{i=1}^M \kappa_\sigma(\mathbf{x}_j^p - \mathbf{x}_i^p)\kappa_\sigma(\mathbf{y}_j^p - \mathbf{y}_i^p)}{\left(\sum_{i=1}^M \kappa_\sigma(\mathbf{x}_j^p - \mathbf{x}_i^p)\right)^2}\right). \tag{59}$$

Similarly, the empirical estimation of $\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ is given by:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^q-\mathbf{x}_i^q)\kappa_\sigma(\mathbf{y}_j^q-\mathbf{y}_i^q)}{\left(\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^q-\mathbf{x}_i^q)\right)^2}\right). \tag{60}$$

[The cross term]

Again, the empirical estimation of $\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ can be expressed as:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y} = \mathbb{E}_{p(X,Y)}\left[\frac{q(X,Y)}{p(X)q(X)}\right] \approx \frac{1}{M}\sum_{j=1}^{M}\frac{q(\mathbf{x}_j,\mathbf{y}_j)}{p(\mathbf{x}_j)q(\mathbf{x}_j)}. \tag{61}$$

By KDE, we further have:

$$\frac{q(\mathbf{x}_j,\mathbf{y}_j)}{p(\mathbf{x}_j)q(\mathbf{x}_j)} \approx M\frac{\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^p-\mathbf{x}_i^q)\kappa_\sigma(\mathbf{y}_j^p-\mathbf{y}_i^q)}{\sum_{i=1}^{M}\kappa_\sigma(\mathbf{x}_j^p-\mathbf{x}_i^p)\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^p-\mathbf{x}_i^q)}. \tag{62}$$

Therefore,

$$\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^p-\mathbf{x}_i^q)\kappa_\sigma(\mathbf{y}_j^p-\mathbf{y}_i^q)}{\sum_{i=1}^{M}\kappa_\sigma(\mathbf{x}_j^p-\mathbf{x}_i^p)\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^p-\mathbf{x}_i^q)}\right). \tag{63}$$

Note that, one can also empirically estimate $\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y}$ over $q(\mathbf{x},\mathbf{y})$, which can be expressed as:

$$\begin{aligned}\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y} &= \mathbb{E}_{q(X,Y)}\left[\frac{p(X,Y)}{p(X)q(X)}\right] \approx \frac{1}{N}\sum_{j=1}^{N}\frac{p(\mathbf{x}_j,\mathbf{y}_j)}{p(\mathbf{x}_j)q(\mathbf{x}_j)} \\ &\approx \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{M}\kappa_\sigma(\mathbf{x}_j^q-\mathbf{x}_i^p)\kappa_\sigma(\mathbf{y}_j^q-\mathbf{y}_i^p)}{\sum_{i=1}^{M}\kappa_\sigma(\mathbf{x}_j^q-\mathbf{x}_i^p)\sum_{i=1}^{N}\kappa_\sigma(\mathbf{x}_j^q-\mathbf{x}_i^q)}\right).\end{aligned} \tag{64}$$

[Empirical Estimation]

Let K^p and L^p denote, respectively, the Gram matrices for the input variable $\mathbf{x}$ and output variable $\mathbf{y}$ in the distribution p. Further, let $(K)_{ji}$ denotes the (j,i)-th element of a matrix K (i.e., the j-th row and i-th column of K). We have:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p^2(\mathbf{x},\mathbf{y})}{p^2(\mathbf{x})}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M}K_{ji}^pL_{ji}^p}{(\sum_{i=1}^{M}K_{ji}^p)^2}\right). \tag{65}$$

Similarly, let K^q and L^q denote, respectively, the Gram matrices for input variable $\mathbf{x}$ and output variable $\mathbf{y}$ in the distribution q. We have:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{q^2(\mathbf{x},\mathbf{y})}{q^2(\mathbf{x})}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N}K_{ji}^qL_{ji}^q}{(\sum_{i=1}^{N}K_{ji}^q)^2}\right). \tag{66}$$

Further, let $K^{pq}\in\mathbb{R}^{M\times N}$ denote the Gram matrix between distributions p and q for input variable $\mathbf{x}$, and L^{pq} the Gram matrix between distributions p and q for output variable $\mathbf{y}$. According to Eq. (63), we have:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}}\frac{p(\mathbf{x},\mathbf{y})q(\mathbf{x},\mathbf{y})}{p(\mathbf{x})q(\mathbf{x})}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N}K_{ji}^{pq}L_{ji}^{pq}}{(\sum_{i=1}^{M}K_{ji}^p)(\sum_{i=1}^{N}K_{ji}^{pq})}\right). \tag{67}$$

Therefore, according to Eqs. (65)-(67), an empirical estimate of $D_{\text{CS}}(p(\mathbf{y}|\mathbf{x});q(\mathbf{y}|\mathbf{x}))$ is given by:

$$\begin{aligned}D_{\text{CS}}(p(\mathbf{y}|\mathbf{x});q(\mathbf{y}|\mathbf{x})) \approx{}& \log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M}K_{ji}^pL_{ji}^p}{(\sum_{i=1}^{M}K_{ji}^p)^2}\right)\right) + \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N}K_{ji}^qL_{ji}^q}{(\sum_{i=1}^{N}K_{ji}^q)^2}\right)\right) \\ &- 2\log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N}K_{ji}^{pq}L_{ji}^{pq}}{(\sum_{i=1}^{M}K_{ji}^p)(\sum_{i=1}^{N}K_{ji}^{pq})}\right)\right).\end{aligned} \tag{68}$$

Note that, according to Eq. (64), $D_{\text{CS}}(p(\mathbf{y}|\mathbf{x});q(\mathbf{y}|\mathbf{x}))$ can also be expressed as:

$$\begin{aligned}D_{\text{CS}}(p(\mathbf{y}|\mathbf{x});q(\mathbf{y}|\mathbf{x})) \approx{}& \log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M}K_{ji}^pL_{ji}^p}{(\sum_{i=1}^{M}K_{ji}^p)^2}\right)\right) + \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N}K_{ji}^qL_{ji}^q}{(\sum_{i=1}^{N}K_{ji}^q)^2}\right)\right) \\ &- 2\log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{M}K_{ji}^{qp}L_{ji}^{qp}}{(\sum_{i=1}^{M}K_{ji}^{qp})(\sum_{i=1}^{N}K_{ji}^{q})}\right)\right).\end{aligned} \tag{69}$$

Therefore, to obtain a consistent and symmetric expression, we estimate $D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x}))$ by:

$$\begin{aligned} D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}); q(\mathbf{y}|\mathbf{x})) \approx \log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M} K_{ji}^{p} L_{ji}^{p}}{(\sum_{i=1}^{M} K_{ji}^{p})^2}\right)\right) + \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{q} L_{ji}^{q}}{(\sum_{i=1}^{N} K_{ji}^{q})^2}\right)\right) \\ - \log\left(\sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{N} K_{ji}^{pq} L_{ji}^{pq}}{(\sum_{i=1}^{M} K_{ji}^{p})(\sum_{i=1}^{N} K_{ji}^{pq})}\right)\right) - \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{M} K_{ji}^{qp} L_{ji}^{qp}}{(\sum_{i=1}^{M} K_{ji}^{qp})(\sum_{i=1}^{N} K_{ji}^{q})}\right)\right). \end{aligned} \tag{70}$$

□

B.3 Proof to Proposition 3

Given observations $\psi = \{(\mathbf{x}_i, \mathbf{y}_i^1, \mathbf{y}_i^2)\}_{i=1}^N$, where $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{y}^1 \in \mathbb{R}^q$ and $\mathbf{y}^2 \in \mathbb{R}^q$. Let K, L^1 and L^2 denote, respectively, the Gram matrices for the variable $\mathbf{x}$, $\mathbf{y}^1$, and $\mathbf{y}^2$. Further, let L^{21} denote the Gram matrix between $\mathbf{y}^2$ and $\mathbf{y}^1$. The empirical estimation of $D_{\text{CS}}(p(\mathbf{y}_1|\mathbf{x}); p(\mathbf{y}_2|\mathbf{x}))$ is given by:

$$\begin{aligned} D_{\text{CS}}(p(\mathbf{y}_1|\mathbf{x}); p(\mathbf{y}_2|\mathbf{x})) \approx \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji} L_{ji}^{1}}{(\sum_{i=1}^{N} K_{ji})^2}\right)\right) \\ + \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji} L_{ji}^{2}}{(\sum_{i=1}^{N} K_{ji})^2}\right)\right) - 2\log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji} L_{ji}^{21}}{(\sum_{i=1}^{N} K_{ji})^2}\right)\right) \end{aligned} \tag{71}$$

Proof. Eq. (71) can be obtained by setting $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}$. In this sense, we have:

$$K = K^p = K^q = K^{pq} = K^{qp} \in \mathbb{R}^{N\times N}. \tag{72}$$

Plugging in Eq. (72) into Eq. (56), we obtain Eq. (71). □

B.4 Proof to Proposition 4

Proof. The CS divergence for conditional distribution $p(\mathbf{y}|\mathbf{x}_1)$ and conditional distribution $p(\mathbf{y}|\mathbf{x}_1, \mathbf{x}_2)$ can be expressed as (denote $\vec{\mathbf{x}} = [\mathbf{x}_1; \mathbf{x}_2]$):

$$\begin{aligned} & D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}_1); p(\mathbf{y}|\{\mathbf{x}_1, \mathbf{x}_2\}) \\ &= -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p(\mathbf{y}|\mathbf{x}_1)p(\mathbf{y}|\vec{\mathbf{x}})d\vec{\mathbf{x}}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p^2(\mathbf{y}|\mathbf{x}_1)d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} p^2(\mathbf{y}|\vec{\mathbf{x}})d\vec{\mathbf{x}}d\mathbf{y}\right) \\ &= -2\log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x}_1,\mathbf{y})p(\vec{\mathbf{x}},\mathbf{y})}{p(\mathbf{x}_1)p(\vec{\mathbf{x}})}d\vec{\mathbf{x}}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x}_1,\mathbf{y})}{p^2(\mathbf{x}_1)}d\mathbf{x}d\mathbf{y}\right) + \log\left(\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\vec{\mathbf{x}},\mathbf{y})}{p^2(\vec{\mathbf{x}})}d\vec{\mathbf{x}}d\mathbf{y}\right), \end{aligned} \tag{73}$$

Following the proof to Proposition 2, the two conditional quadratic terms can be estimated empirically by:

$$\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\mathbf{x}_1,\mathbf{y})}{p^2(\mathbf{x}_1)}d\mathbf{x}d\mathbf{y} \approx \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{1} L_{ji}}{(\sum_{i=1}^{N} K_{ji}^{1})^2}\right), \tag{74}$$

and

$$\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p^2(\vec{\mathbf{x}},\mathbf{y})}{p^2(\vec{\mathbf{x}})}d\vec{\mathbf{x}}d\mathbf{y} \approx \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{12} L_{ji}}{(\sum_{i=1}^{N} K_{ji}^{12})^2}\right). \tag{75}$$

We only discuss below the empirical estimation to the term $\int_{\mathcal{X}}\int_{\mathcal{Y}} \frac{p(\mathbf{x}_1,\mathbf{y})p(\vec{\mathbf{x}},\mathbf{y})}{p(\mathbf{x}_1)p(\vec{\mathbf{x}})}d\vec{\mathbf{x}}d\mathbf{y}$.
We have:

$$\int \frac{p(\mathbf{x}_1,\mathbf{y})p(\vec{\mathbf{x}},\mathbf{y})}{p(\mathbf{x}_1)p(\vec{\mathbf{x}})} = \mathbb{E}_{p(\vec{\mathbf{x}},\mathbf{y})}\left[\frac{p(\mathbf{x}_1,\mathbf{y})}{p(\mathbf{x}_1)p(\vec{\mathbf{x}})}\right] \approx \frac{1}{N}\sum_{j=1}^{N}\frac{p((\mathbf{x}_1)_j,\mathbf{y}_j)}{p((\mathbf{x}_1)_j)p(\vec{\mathbf{x}}_j)}. \tag{76}$$

By KDE, we have:

$$\frac{p((\mathbf{x}_1)_j,\mathbf{y}_j)}{p((\mathbf{x}_1)_j)p(\vec{\mathbf{x}}_j)} \approx N\frac{\sum_{i=1}^{N}\kappa_\sigma(\mathbf{y}_j-\mathbf{y}_i)\kappa_\sigma((\mathbf{x}_1)_j-(\mathbf{x}_1)_i)}{\sum_{i=1}^{N}\kappa_\sigma((\mathbf{x}_1)_j-(\mathbf{x}_1)_i)\times\sum_{i=1}^{N}\kappa_\sigma((\mathbf{x}_1)_j-(\mathbf{x}_1)_i)\kappa_\sigma((\mathbf{x}_2)_j-(\mathbf{x}_2)_i)}. \tag{77}$$

Therefore,

$$\begin{aligned} \int \frac{p(\mathbf{x}_1,\mathbf{y})p(\vec{\mathbf{x}},\mathbf{y})}{p(\mathbf{x}_1)p(\vec{\mathbf{x}})} &\approx \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N}\kappa_\sigma(\mathbf{y}_j-\mathbf{y}_i)\kappa_\sigma((\mathbf{x}_1)_j-(\mathbf{x}_1)_i)}{\sum_{i=1}^{N}\kappa_\sigma((\mathbf{x}_1)_j-(\mathbf{x}_1)_i)\times\sum_{i=1}^{N}\kappa_\sigma((\mathbf{x}_1)_j-(\mathbf{x}_1)_i)\kappa_\sigma((\mathbf{x}_2)_j-(\mathbf{x}_2)_i)}\right) \\ &= \sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K_{ji}^{1} L_{ji}}{(\sum_{i=1}^{N} K_{ji}^{1})(\sum_{i=1}^{N} K_{ji}^{12})}\right). \end{aligned} \tag{78}$$

Combining Eq. (74), Eq. (75) and Eq. (78), we obtain Eq. (79).

$$\begin{aligned} D_{\text{CS}}(p(\mathbf{y}|\mathbf{x}_1); p(\mathbf{y}|\{\mathbf{x}_1,\mathbf{x}_2\})) \approx \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K^{1}_{ji}L_{ji}}{(\sum_{i=1}^{N} K^{1}_{ji})^2}\right)\right) \\ + \log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K^{12}_{ji}L_{ji}}{(\sum_{i=1}^{N} K^{12}_{ji})^2}\right)\right) - 2\log\left(\sum_{j=1}^{N}\left(\frac{\sum_{i=1}^{N} K^{1}_{ji}L_{ji}}{(\sum_{i=1}^{N} K^{1}_{ji})(\sum_{i=1}^{N} K^{12}_{ji})}\right)\right) \end{aligned} \tag{79}$$

□

Appendix C
Experimental Details and Additional Results

C.1 Detailed experiment settings of Causal analysis in Section 3.2

We first consider 5 coupled Hénon chaotic maps [48], whose ground truth causal relationship is $x_{i-1} \to x_i$. The system of K coupled Hénon chaotic maps is defined as:

$$\begin{cases} x_{1,t} = 1.4 - x_{1,t-1}^2 + 0.3x_{1,t-2} \\ x_{i,t} = 1.4 - (Cx_{i-1,t-1} + (1-C)x_{i,t-1})^2 + 0.3x_{i,t-2} \quad \text{for } i = 2, 3, \cdots, K. \end{cases} \tag{80}$$

In our simulation, we set the coupling strength $C = 0.3$ and generate $1,024$ samples in 10 independent realizations respectively.

Next, we consider a nonlinear VAR process of order 1 with 3 variables (NLVAR3) [49]:

$$\begin{cases} x_{1,t} = 3.4x_{1,t-1}(1 - x_{1,t-1}^2)\exp(-x_{1,t-1}^2) + 0.01w_{1,t} \\ x_{2,t} = 3.4x_{2,t-1}(1 - x_{2,t-1}^2)\exp(-x_{2,t-1}^2) + 0.5x_{1,t-1}x_{2,t-1} + 0.01w_{2,t} \\ x_{3,t} = 3.4x_{3,t-1}(1 - x_{3,t-1}^2)\exp(-x_{3,t-1}^2) + 0.3x_{2,t-1} + 0.5x_{1,t-1}^2 + 0.01w_{3,t} \end{cases} \tag{81}$$

where $w_{1,t}$, $w_{2,t}$ and $w_{3,t}$ denotes independent standard Gaussian noise. The true causal directions in NLVAR3 are $x_1 \to x_2$, $x_1 \to x_3$, $x_2 \to x_3$. Again, we generate $1,024$ samples in 10 independent realizations respectively.

The delay τ and the embedding dimension d are vital parameters for all Wiener and Granger causality methods. In our simulations, we simply use the ground truth from the synthetic models, that is $\{\tau = 1, d = 1\}$ for NLVAR3 and $\{\tau = 1, d = 2\}$ for Hénon maps, since selecting the embedding automatically is not the topic of this paper.

For kernel Granger causality (KGC), we use the official MATLAB code from authors[11] and select kernel size σ by the median rule. For transfer entropy, we use the kNN entropy estimator from the Information Theoretical Estimators Toolbox[12] and set $k = 3$ as default.

Once we obtain the causal score $C_{\mathbf{x}\to\mathbf{y}}$ with one of the measures, we need to test its significance. To carry out this hypothesis test, we may use the Monte Carlo method by constructing a surrogate time series. The constructed surrogate time series must satisfy the null hypothesis that the causal influence from $\mathbf{x}$ to $\mathbf{y}$ is completely destroyed; at the same time, the statistical properties of $\mathbf{x}$ and $\mathbf{y}$ should remain the same. To construct the surrogate time series that satisfies these two conditions, we apply the surrogate time series construction method in [83].

Specifically, given N samples, let T denotes the length of training set (in our case $T = 1,024$), a pair of surrogate time series for $\mathbf{x}$ and $\mathbf{y}$ is constructed as:

$$\begin{cases} x^{\text{surr}} = [x_i, x_{i+1}, \cdots, x_{i+T-1}] \\ y^{\text{surr}} = [y_j, y_{j+1}, \cdots, y_{j+T-1}], \end{cases} \tag{82}$$

where i and j are randomly chosen from $1, 2, \cdots, N-T+1$ and $|j - i| \geq e$ and e is a sufficient large integer such that there is almost no correlation between x^{surr} and y^{surr}. In our simulation, we set $e = 512$. On these surrogate data we use a permutation test (100 permutations) to obtain a p-value. P-values below 0.05 were considered as significant.

C.2 Details of Permutation Test in Section 4.1

11. https://github.com/danielemarinazzo/KernelGrangerCausality
12. https://bitbucket.org/szzoli/ite/src/master/

Algorithm 1: Test the significance of conditional divergence

```
Input: Two groups of observations $\psi_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^M$ and $\psi_t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^N$; Permutation number $P$; Significant rate $\eta$.
Output: Test decision (Is $\mathcal{H}_0 : p_s(\mathbf{y}|\mathbf{x}) = p_t(\mathbf{y}|\mathbf{x})$ $True$ or $False$?).
1: Compute conditional divergence value $d_0$ on $\psi_s$ and $\psi_t$ with one of the conditional divergence measures (e.g., conditional KL, or conditional Bregman divergence, or conditional MMD, or conditional CS divergence).
2: for $m = 1$ to $P$ do
3:     $(\psi_s^m, \psi_t^m) \leftarrow$ random split of $\psi_s \bigcup \psi_t$.
4:     Compute conditional divergence value $d_m$ on $\psi_s^m$ and $\psi_t^m$ with the selected conditional divergence measure.
5: end for
6: if $\frac{1+\sum_{m=1}^P \mathbf{1}[d_0 \leq d_t]}{1+P} \leq \eta$ then
7:     decision$\leftarrow False$
8: else
9:     decision$\leftarrow True$
10: end if
11: return decision
```

C.3 Detailed Experiment Settings and Additional Results of Time Series Clustering in Section 5.1

C.3.1 More on Dynamic Texture Datasets

The dynamic texture (DT) is a sequence of images of moving scenes such as flames, smoke, and waves, which exhibits certain stationarity in time and can be modeled using a linear dynamic system (LDS) [84]:

$$\begin{cases}\mathbf{h}_t = A\mathbf{h}_{t-1} + v_t, v_t \sim N(0,Q)\\ \mathbf{y}_t = C\mathbf{h}_t + w_t, w_t \sim N(0,R)\end{cases} \tag{83}$$

where $\mathbf{y}_t \in \mathbb{R}^m$ is the observational frame at time t, $\mathbf{h}_t \in \mathbb{R}^n$ ($n \ll m$) is the hidden state vector at time t, $A \in \mathbb{R}^{n\times n}$ is the state transition matrix, $C \in \mathbb{R}^{m\times n}$ is the output matrix that maps hidden states to observations. Finally, v_t and w_t are Gaussian noises with covariance matrices Q and R, respectively. Thus, a single dynamic texture can be described by a transition function f from $\mathbf{y}_{t-1}$ to $\mathbf{y}_t$, i.e., $\mathbf{y}_t = f(\mathbf{y}_{t-1})$ [85]. In this sense, it is natural to expect to distinguish different DTs by the conditional distribution $p(\mathbf{y}_t|\mathbf{y}_{t-1})$.

The UCLA database[13] originally contains 200 DT sequences from 50 categories, and each category contains 4 video sequences captured from different viewpoints. All the video sequences are of the size $48 \times 48 \times 75$, where 75 is the number of frames. By combining sequences from different viewpoints, the original 50 categories are merged to 9 categories: boiling water (8), fire (8), flowers (12), fountains (20), plants (108), sea (12), smoke (4), water (12) and waterfall (16), where the numbers in parentheses denote the number of the sequences in each category. This dataset is however very challenging and imbalanced, because the category "plants" contains too many videos. Therefore, we follow [86] and remove the category of "plants". Finally, there are a total of 92 video sequences from 8 categories.

The traffic database[14] [62], consists of 253 videos divided into three classes: light, medium, and heavy traffic. Videos have 42 to 52 frames with a resolution of 48×48 pixels.

The statistics of all datasets are summarized in Table 8.

C.3.2 Competing Baselines and Their Hyperparameters Setting

We include the following representative baselines in the literature of time series clustering for evaluations:

- DTW is a well-known approach to measure the similarity between two temporal time series sequences. It uses the dynamic programming technique to find the optimal temporal matching between elements of two-time series. In our experiments, we use the implementation of DTW and its multivariate extension provided by Schultz and Jain[15] [65]. We apply the default warping window constraint, i.e., the length of the longer sequence.
- MSM distance is a distance metric for time series. It is conceptually similar to other edit distance approaches. MSM metric uses as building blocks three fundamental operations: Move, Split, and Merge, which can be applied in sequence to transform any time series into any other time series. In our experiments, we use the official code provided by authors[16]. There is no hyperparameters need to be tuned for MSM.
- TWED is a distance measure for discrete time series matching with time "elasticity". It has two vital parameters: the penalty for deletion operation λ and the elasticity parameter ν. In our experiments, we use the official code provided by authors[17] and set $\lambda = 1$ and $\nu = 0.001$. A big limitation of TWED is that its computational complexity is usually $\mathcal{O}(n^2)$, in which n is the length of time series.

13. https://drive.google.com/file/d/0BxMIVlhgRmcbN3pRa0dyaHpHV1E/view?resourcekey=0-_OdWQfyRKH_FHh84bxZLcg
14. http://www.svcl.ucsd.edu/projects/dytex/
15. https://www.ai4europe.eu/research/ai-catalog/dtw-mean
16. https://athitsos.utasites.cloud/projects/msm/
17. http://people.irisa.fr/Pierre-Francois.Marteau/

TABLE 8
Properties of the datasets (in time series clustering).

Datasets	Time series	Length	Dimension	Clusters
Coffee	28	286	1	2
Diatom	306	345	1	4
DistalPhalanxTW	400	80	1	6
ECG5000	469	140	1	2
FaceAll	560	131	1	14
Synthetic control	600	60	1	6
PenDigits	3498	8	2	10
Libras	180	45	2	15
uWave	200	315	3	8
Robot failure LP1	88	15	6	4
Robot failure LP2	47	15	6	5
Robot failure LP3	47	15	6	4
Robot failure LP4	117	15	6	3
Robot failure LP5	164	15	6	5
Traffic	253	42	2304	3
UCLA	92	75	2304	8

- TCK is able to learn a valid kernel function to describe the similarity between pairwise time series. Distinct to above mentioned measures, TCK is an ensemble approach that combines multiple GMM models, whose diversity is ensured by training the models on subsamples of data, attributes, and time segment, to capture different levels of granularity in the data. It is robust to missing values and applies to multivariate time series. There are two optional parameters for practitioners: the max number of mixture components for each GMM (C) and the number of randomizations for each number of components (G). In our experiment, we use the default values of C and G provided by authors[18].

C.3.3 A Case Study of Exploratory Data Analysis

This section presents a network modeling example using real data. We use temperature data over 109 cities at US in the year of 2010, which can be obtained from the US National Oceanic and Atmospheric Administration (NOAA)[19]. For each city, we obtain temperature records in each hour and take the daily average over 24 hours, thus forming a univariate time series of length 364.

We first compute a 109×109 matrix which encodes conditional CS divergence measures over all pairs of time series. We then obtain a nearest neighbor network over different cities using 5% of the smallest conditional CS divergence measures (see Fig. 9(a)) and find communities using spectral clustering with the number of clusters set to 2. The result is shown in Fig. 9(b). We can make two observations: 1) closer cities are more likely to have similar temperature patterns; and 2) the north-south difference is much more obvious than the east-west difference. Both observations make sense and match well with recent exploratory data analysis (EDA) package [87].

C.4 Details of divergence to go in Section 5.2

Analogous to Q-learning and its extensions, we employ the divergence-to-go (dtg) metric in a dynamic programming framework. In short, we replace the reward in Q-learning with the divergence between old states and new counterparts $D(p_{\text{new}}(s_{t+1}|s_t,a_t);p_{\text{old}}(s_{t+1}|s_t,a_t))$. To obtain samples for divergence estimation, we introduce a replay buffer B (of size 2τ) to record 2τ steps of $\{s_{t+1},s_t,a_t\}$ trios. We use trios in the first half of buffer to estimate p_{old} and trios in the second half of buffer to estimate p_{new}, such that we can evaluate the corresponding conditional CS divergence using Eq. (56) (by treating s_{t+1} as variable $\mathbf{y}$ and the concatenation of $[s_t,a_t]$ as variable $\mathbf{x}$). Same to the initial dtg paper [71], we use a kernelized version of Q-learning, i.e., Kernel Temporal Difference (KTD) algorithm [88] as the backbone of dtg. More formally, we have:

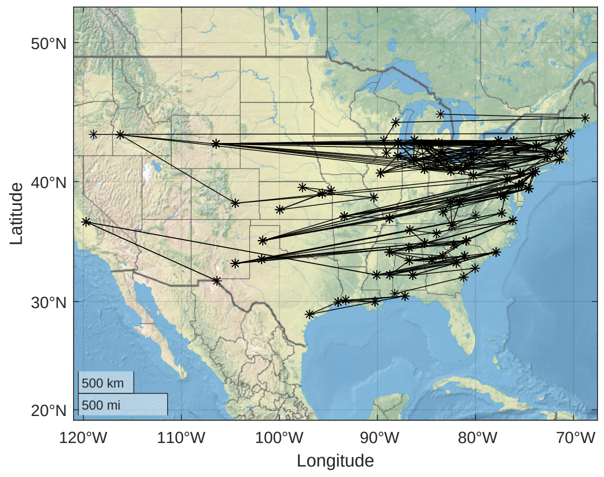
$$\delta_t = D + \gamma \max_{a'}[\text{dtg}_{t+1}(s',a')] - \text{dtg}_t(s,a), \tag{84}$$

and

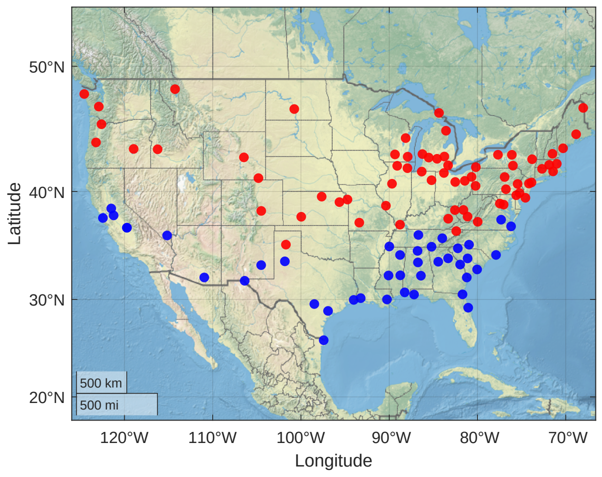
$$\text{dtg}_t(s,a) = \alpha \sum_{j=1}^{t} \delta_j \kappa(z;z_j), \tag{85}$$

where D is the divergence $D(p_{\text{new}}(s_{t+1}|s_t,a_t);p_{\text{old}}(s_{t+1}|s_t,a_t))$; κ denotes Gaussian kernel; $z := \{s,t\}$ denotes the vector of a state-action pair. Steps in detail are shown in Algorithm 2. In our experiment, the kernel size is fixed to be 0.1. The length of the buffer B is $2,000$.

18. https://github.com/kmi010/Time-series-cluster-kernel-TCK-
19. https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/city/time-series



(a)



(b)

Fig. 9. (a) The nearest neighbor network constructed using 5% of the smallest conditional CS divergences; (b) The detected communities by conditional CS divergence. Node colors represent communities.

Algorithm 2: The divergence-to-go (DTG) algorithm using conditional Cauchy-Schwarz divergence.

```
 1: Initialize $\text{dtg}_0(s,a)=0$, set learning rate $\alpha$, discount factor $\gamma$, empty first-in-first-out (FIFO) replay buffer $B$.
 2: while not achieve the goal do
 3:     $s \leftarrow$ current state.
 4:     $a \leftarrow \underset{a'}{\textbf{argmax}}[\text{dtg}(s,a')]$.
 5:     Take action $a$, observe state transition.
 6:     $s' \leftarrow$ new state.
 7:     Record trio $\{s',s,a\}$ to the buffer $B$
 8:     Compute conditional divergence $D(p_{\text{new}}(s_{t+1}|s_t,a_t);p_{\text{old}}(s_{t+1}|s_t,a_t))$ with Eq. (56) from trios in the buffer $B$.
 9:     $\delta = D + \gamma\underset{a'}{\textbf{max}}[\text{dtg}(x',a')] - \text{dtg}(x,a)$
10:     $\text{dtg}(s,a) \leftarrow \text{dtg}(s,a) + \alpha\cdot\delta$
11: end while
```

In the following, we also show detailed steps of the standard reward-based Q-learning in Algorithm 3. The major differences between dtg and Q-learning are highlighted by blue texts. In summary, our dtg algorithm is not reward driven. Instead of obtaining rewards from environments, our dtg estimates conditional CS divergence using state-action pairs only (see line 8 in Algorithm 2 and line 7 in Algorithm 3). In order to estimate divergence, we have to introduce a replay buffer for collecting samples in dtg algorithm. The buffer is not required in original Q-learning though this memory is a popular trick to conduct mini-batch learning and offline learning.

Algorithm 3: The standard reward-based Q-learning. We highlight the main differences between dtg and Q-learning with blue texts.

```
 1: Initialize $Q_0(s,a)=0$, set learning rate $\alpha$, discount factor $\gamma$.
 2: while not achieve the goal do
 3:     $s \leftarrow$ current state.
 4:     $a \leftarrow \underset{a'}{\textbf{argmax}}[Q(s,a')]$.
 5:     Take action $a$, observe state transition.
 6:     $s' \leftarrow$ new state.
 7:     $r \leftarrow$ reward given by the environment.
 8:     $\delta = r + \gamma\underset{a'}{\textbf{max}}[Q(s',a')] - Q(s,a)$
 9:     $Q(s,a) \leftarrow Q(s,a) + \alpha\cdot\delta$
10: end while
```

Another exploration-based reinforcement learning algorithm, which shares the same motivation to us, is the maximum entropy exploration (MaxEnt) [69]. Instead of estimating divergence in time series, the goal of MaxEnt is to obtain the policy which maximizes the entropy of states. In our experiments, we compare to MaxEnt with code in https://github.

TABLE 9
Hyperparameters in DTG-CS and standard Q-learning in Algorithms 2 and 3.

Hyper-parameter	value
discount factor γ	0.99
learning rate α	1e-3
ϵ-greedy ratio	0.1
number of steps to roll out a policy	1000

com/abbyvansoest/maxent_base. We keep all hyper-parameters of DTG and MaxEnt identical, as shown in Table 9.

C.5 Detailed Experiment Settings of Unsupervised Domain Adaptation in Section 6

C.5.1 About the Datasets

We apply CS divergence-based domain adaptation to EEG classification on two benchmark real-world datasets, namely BCI Competition IIIb [74] and SEED [75].

The BCI Competition IIIb focuses on cued motor imagery tasks with online feedback, employing a non-stationary classifier with two classes (left hand and right hand). It comprises EEG recordings from three subjects (S4, X11, and O3VR) with 1080, 1080, and 640 trials. The length of each trial is 7s. Due to a mistake, trials 1-160 and 161-320 in O3VR are repeated. Hence, we only use data from S4 and X11 in the present experiment. EEG data were recorded at a 125 Hz sampling rate, band-pass filtered between 0.5 and 30 Hz, and converted into the GDF format for analysis, using a bipolar EEG amplifier from g.tec with additional Notch filtering.

The SEED consists of data from 15 subjects. Curated movie excerpts were specifically selected to elicit three distinct emotions: positive, neutral, and negative. Each emotion category contains five movie excerpts. All subjects underwent three EEG recording sessions, with a two-week interval between each session. During each recording session, subjects were exposed to fifteen four-minute long movie excerpts, each intended to induce one of the specified emotions. Importantly, the same fifteen movie excerpts were used across all three recording sessions. Thus, the SEED dataset comprises 15 EEG trials recorded for each subject in each session, with each emotion category consisting of 5 trials. EEG signals were acquired using a 62-channel ESI NeuroScan device, sampled at a rate of 1000 Hz, and subsequently downsampled to 200 Hz for analysis [89]. In our experiment, we randomly select 10 pairs of subjects as the source and target domains, and report the average domain adaptation result.

C.5.2 Competing Baselines and Their Hyperparameters Setting

We compare our method with four representative MMD-based approaches, which include DDC [76], DAN [77], JAN [24], DJP-MMD [78]. Given a baseline network as shown in Fig. 10, comprising a shared feature extractor, fully connected layers of depth $|L|$ and a final linear classification layer, all competing methods optimize the following objective:

$$L_{\text{CE}} + \alpha D(p_s, p_t), \tag{86}$$

where L_{CE} is the cross-entropy loss on source domain labeled data, $D(p_s, p_t)$ represents the distributional discrepancy between the source domain and target domain. However, they vary in their approach to evaluate $D(p_s, p_t)$.

Specifically, DDC just minimizes the discrepancy in the last layer (i.e., $\text{MMD}^2(\mathbf{z}_s^{|L|}, \mathbf{z}_t^{|L|})$), DAN improves upon DDC by minimizing the discrepancies in all fully connected layers and replacing the basic MMD with multi-kernel MMD (MK-MMD):

$$L_{\text{CE}} + \alpha \sum_{l=1}^{|L|} \text{MK-MMD}^2(\mathbf{z}_s^l, \mathbf{z}_t^l), \tag{87}$$

in which the kernel function in MK-MMD is expressed as:

$$\kappa(\mathbf{z}_s, \mathbf{z}_t) = \sum_{i=1}^{m} \beta_i \kappa_i(\mathbf{z}_s, \mathbf{z}_t), \quad \text{s.t.} \quad \beta_i > 0, \sum_{i=1}^{m} \beta_i = 1. \tag{88}$$

JAN argues that the $|L|$ constraints in Eq. (87) is independent and does not incorporate the dependence between different layers. Hence, the authors of JAN minimizes the joint MMD for all fully connected layers:

$$L_{\text{CE}} + \alpha \|\mathcal{C}_{\mathbf{z}_s^{1:|L|}} - \mathcal{C}_{\mathbf{z}_t^{1:|L|}}\|_{\otimes_{l=1}^{|L|}\mathcal{H}^l}, \tag{89}$$

in which $\mathcal{C}_{\mathbf{z}^{1:|L|}}$ refers to kernel mean embedding for $|L|$ variables using tensor product feature space:

$$\mathcal{C}_{\mathbf{z}_s^{1:|L|}} = \mathbb{E}_{\mathbf{z}_s^{1:|L|}} \left[\otimes_{l=1}^{|L|} \phi^l(\mathbf{z}^l) \right]. \tag{90}$$

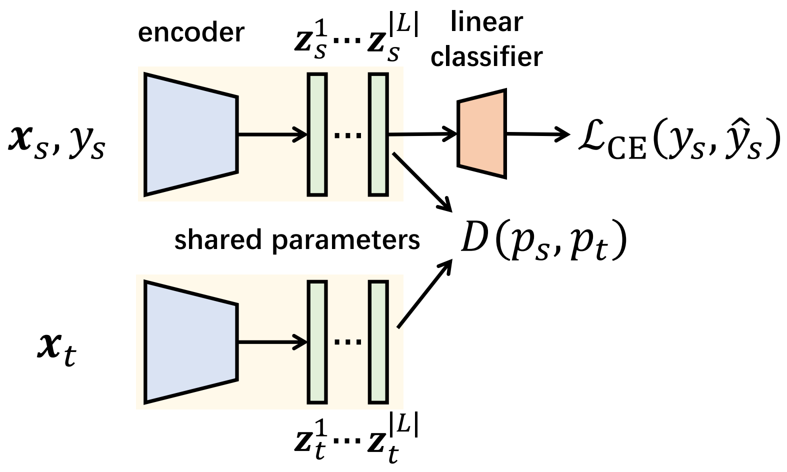


Fig. 10. UDA framework.

DJP-MMD aims to minimize the discrepancy on the joint distribution $p(\mathbf{z}, y)$, but approximates the joint alignment in the following way:

$$D(p_s(\mathbf{z}, y), p_t(\mathbf{z}, y)) = D(p_s(y|\mathbf{z})p_s(\mathbf{z}), p_t(y|\mathbf{z})p_t(\mathbf{z})) \approx \mu_1 D(p_s(\mathbf{z}), p_t(\mathbf{z})) + \mu_2 D(p_s(\mathbf{z}|y), p_t(\mathbf{z}|y)). \tag{91}$$

Obviously, such approximation simplifies the estimation, but does not obey the chain rule that $p(\mathbf{z}, y) = p(y|\mathbf{z})p(\mathbf{z})$ (rather than $p(\mathbf{z}|y)$).

By contrast, our approach directly optimizes the following objective:

$$L_{\text{CE}} + \alpha \left[D_{\text{CS}}(p_t(\mathbf{z}); p_s(\mathbf{z})) + D_{\text{CS}}(p_t(\hat{y}|\mathbf{z}); p_s(y|\mathbf{z})) \right], \tag{92}$$

and approximates y_t (the true label in target domain) with its prediction $\hat{y}_t$, which is a common trick in UDA literature.

Our network for domain adaptation is built upon the EEGNet [79]. Additionally, as a common practice, for all methods, we use a bottleneck subnetwork, comprising two fully connected layers (i.e., $|L| = 2$) with ReLU activation functions before the final linear classification layer. Except for JAN which aligns the joint distribution $p(\mathbf{z}^1, \mathbf{z}^2)$, all other methods only takes $\mathbf{z}^2$ as the learned feature. The training is performed with PyTorch [90] on a NVIDIA GeForce RTX 3090 GPU. Adam optimizer is used with learning rate of $1e - 2$ and batch size 64. In order to stabilize distribution alignment in the training phase, we employ a warm-up strategy, initially training without adaptation for the first 10 epochs. For our method, we normalize the feature and adopt kernel size $\sigma = 1$. For other methods, we choose either adaptive kernel size or fixed size 1, depending on their performances. As for the weight α of the discrepancy term in each method, we perform a grid search from 1 to 100 with an interval of 10.