# A Unified Data Representation Learning for Non-parametric Two-sample Testing

**Xunye Tian**[1] **Liuhua Peng**[1] **Zhijian Zhou**[1] **Mingming Gong**[1] **Arthur Gretton**[2] **Feng Liu**[1]

[1]The University of Melbourne, Australia
[2]University College London, United Kingdoms

## Abstract

Learning effective data representations has been crucial in non-parametric two-sample testing. Common approaches will first split data into training and test sets and then learn data representations *purely on the training set*. However, recent theoretical studies have shown that, as long as the sample indexes are not used during the learning process, the *whole data* can be used to learn data representations, meanwhile ensuring control of Type-I errors. The above fact motivates us to use the test set (but *without* sample indexes) to facilitate the data representation learning in the testing. To this end, we propose a *representation-learning two-sample testing* (RL-TST) framework. RL-TST first performs purely self-supervised representation learning on the entire dataset to capture *inherent representations* (IRs) that reflect the underlying data manifold. A discriminative model is then trained on these IRs to learn *discriminative representations* (DRs), enabling the framework to leverage both the rich structural information from IRs and the discriminative power of DRs. Extensive experiments demonstrate that RL-TST outperforms representative approaches by *simultaneously* using data manifold information in the test set and enhancing test power via finding the DRs with the training set.

## 1 INTRODUCTION

Two-sample tests aim to answer a question: "Are two samples drawn from the same distribution?". Classical two-sample tests, including *t*-tests which test the empirical mean differences between two samples, often need to assume that samples are drawn from specific distributions (e.g., Gaussian distributions with the same variance). To alleviate the assumptions, non-parametric two-sample tests are proposed to solve the problem only based on observed data [Gretton et al., 2012b, Heller and Heller, 2016, Székely and Rizzo, 2013, Jitkrittum et al., 2016, Chen and Friedman, 2017, Ghoshdastidar et al., 2017, Lopez-Paz and Oquab, 2017, Ramdas et al., 2017, Sutherland et al., 2017, Gao et al., 2018, Ghoshdastidar and von Luxburg, 2018, Lerasle et al., 2019, Liu et al., 2020, Kirchler et al., 2020, Kübler et al., 2020, Cheng and Xie, 2021, Kübler et al., 2022, Kübler et al., 2022, Liu et al., 2021, Deka and Sutherland, 2023, Bonnier et al., 2023, Schrab et al., 2023, Biggs et al., 2023].

For example, the *Kolmogorov-Smirnov* (K-S) test is designed to compare the cumulative distribution functions derived from two samples, but generalisation to higher dimension is challenging [Bickel, 1969]. The *maximum mean discrepancy* (MMD) test adopts the kernel mean embedding of distribution and uses it to measure the discrepancy between two distributions [Gretton et al., 2012a] whose dimensions can be relatively higher than classical methods [Liu et al., 2020]. The statistics used in these non-parametric two-sample tests are also widely adopted in many other fields, such as domain adaptation, generative modeling, adversarial learning, membership inference attack, machine-generated text detection and more [Gong et al., 2016, Bińkowski et al., 2018, Stojanov et al., 2019, Cano and Krawczyk, 2020, Gao et al., 2021, Fang et al., 2021b,a, Song et al., 2021, Tahmasbi et al., 2021, Taskesen et al., 2021, Bergamin et al., 2022, Zhang et al., 2024a,b, Sun et al., 2025, Li et al., 2025].

To improve the test power (i.e., control the type II error) of non-parametric two-sample tests in practical applications, recent studies have shown that learning good data representations is crucial before performing two-sample testing [Kirchler et al., 2020, Liu et al., 2020, 2021, Gao et al., 2021, Bergamin et al., 2022]. For example, Kirchler et al. [2020] directly use a pre-trained feature extractor to extract features of two samples and find it is useful to increase the test power during the testing. Meanwhile, Liu et al. [2020] propose a learning paradigm to learn deep-net representations of data via maximizing the test power of MMD and

*Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025)*, PMLR 286:4208–4227.

show that the learned representations can help capture the difference between two complex-structured samples. Even though utilizing a fraction of the samples to train a classifier [Lopez-Paz and Oquab, 2017] or a kernel function [Liu et al., 2020] enables deriving *discriminative representations* (DRs) of the remaining samples, the data splitting process results in a *trade-off* between the extra power provided by the learned functions/kernels and the sacrificed power due to the decreasing testing sample size.

However, Biggs et al. [2023] have pointed out that, after discarding the sample information (namely, we do not know which sample the data belongs to), learning purely *inherent representations* (IRs) from these samples will not influence the type I error of permutation-based testing methods in theory, showing the possibility to learn useful information from the test set instead of only from the training set. Nevertheless, learning purely IRs will miss the discriminative power, making it underperform on complex-structured data.

Motivated by the above theoretical studies and existing challenges, we propose a *representation-learning two-sample testing* (RL-TST) framework that focuses on learning good representations on the samples, from both IRs and DRs. Since two-sample testing data mainly follows a manifold assumption where the (high-dimensional) data lie (roughly) on a low-dimensional manifold, we could firstly learn an encoder from the representation learning that is responsible for extracting the IRs of *entire samples*. Then, train a discriminative model on the learned IRs will enable the model with discriminative ability directly on the inherent manifolds of samples rather than on the complex embedded space of samples. This framework captures the sample structure information discarded in the data splitting process and exhibits a higher discriminative power than purely unsupervised representation learning on the entire dataset.

We conduct extensive experiments to implement RL-TST on different kinds of MMD-based two-sample testing methods, we verify the empirical effectiveness of RL-TST over the *state-of-the-art* (SOTA) two-sample testing methods on synthetic *high-dimensional Gaussian mixture* (HDGM) dataset, MNIST dataset and ImageNet dataset. These are the commonly used benchmarks to detect the performance of two-sample testing methods. Our main contributions are:

- We propose a novel RL-TST, which can address the challenges of two existing frameworks and provide a new research direction of two-sample testing under the control of both type I and type II errors.
- Empirically, various implementations of RL-TST outperform SOTA methods across different benchmarks.
- Comparatively, we provide the discussion and empirical evidence of why alternative potential frameworks, such as semi-supervised learning or purely self-supervised learning are facing challenges in two-sample testing scenarios.

## 2 PRELIMINARIES

**Two-sample Testing.** Two-sample testing is one of the statistical hypothesis tests that aims to assess whether two *independent and identically distributed* samples, denoted by $S_{\mathbb{P}} = \{x_i\}_{i=1}^n \sim \mathbb{P}^n$ and $S_{\mathbb{Q}} = \{y_j\}_{j=1}^m \sim \mathbb{Q}^m$, where $x_i, y_j \in \mathcal{X}$, are drawn from the same distribution [Lehmann and Romano, 2005]. In two-sample testing, the *null hypothesis* $H_0$ refers to two samples being drawn from the same distribution, which corresponds to $\mathbb{P} = \mathbb{Q}$. The *alternative hypothesis* $H_1$ indicates that two samples are drawn from different distributions, meaning $\mathbb{P} \neq \mathbb{Q}$. Whether we should accept or reject $H_0$ depends on the test statistic $\hat{t}$, which represents the differences between two samples.

**Classifier Two-sample Testing (C2ST).** C2ST aims to train a binary classifier: if the classifier obtains a testing accuracy significantly better than random guessing, it suggests that the two samples come from different distributions [Lopez-Paz and Oquab, 2017]. Specifically, given dataset $\mathcal{S} = \{(x_i, 0)|x_i \in S_{\mathbb{P}}\}_{i=1}^n \cup \{(y_j, 1)|y_j \in S_{\mathbb{Q}}\}_{j=1}^m := \{(z_k, l_k)\}_{k=1}^{m+n}$, where $m = n$, and we can shuffle and split $\mathcal{S}$ into training set $\mathcal{S}_{\mathrm{tr}}$ and testing set $\mathcal{S}_{\mathrm{te}}$, let $f^* : \mathcal{X} \to \{0, 1\}$ be a binary classifier that is well-trained on $\mathcal{S}_{\mathrm{tr}}$, then the test statistic or the accuracy of the classifier $f^*$ on $\mathcal{S}_{\mathrm{te}}$ is

$$\hat{t} = \frac{1}{n_{\mathrm{te}}} \sum_{(z_k, l_k) \in \mathcal{S}_{\mathrm{te}}} \mathbb{I}\left[f^*(z_k) = l_k\right], \tag{1}$$

where $n_{\mathrm{te}} = |\mathcal{S}_{\mathrm{te}}|$ and $\mathbb{I}$ is the indicator function. Finally, we compute the $p$-value to see if the test statistic is significantly greater than the random guessing accuracy, utilizing the approximate null distribution [Lopez-Paz and Oquab, 2017, Kim et al., 2021] or the permutation test [Good, 2004].

**C2ST with logits (C2ST-L).** Moreover, we can also consider using the trained classifier $f^*$ in C2ST not directly to compute the accuracy but to extract representations of two samples [Cheng and Cloninger, 2022]. Let $h$ be the feature extractor of $f^*$, then $h(z)$ (model's output should be logits) can be regarded as representations of two samples as the new two samples with the information of prediction confidence. For these new two samples, we can use L2 norm to compute the difference between two samples. Let $S_{\mathbb{P}}^{\mathrm{te}}$ and $S_{\mathbb{Q}}^{\mathrm{te}}$ be the splitting samples of $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$ in the testing set $\mathcal{S}_{\mathrm{te}}$ and $n_x^{\mathrm{te}}$ and $n_y^{\mathrm{te}}$ be the sample size of $S_{\mathbb{P}}^{\mathrm{te}}$ and $S_{\mathbb{Q}}^{\mathrm{te}}$. In general, the statistic used in C2ST-L is

$$\hat{t}_L = \left\| \frac{1}{n_x^{\mathrm{te}}} \sum_{x_i \in S_{\mathbb{P}}^{\mathrm{te}}} h(x_i) - \frac{1}{n_y^{\mathrm{te}}} \sum_{y_i \in S_{\mathbb{Q}}^{\mathrm{te}}} h(y_i) \right\|_2^2, \tag{2}$$

where $\|\cdot\|_2$ is the L2 norm.

**Maximum Mean Discrepancy (MMD) Test with Deep Kernel (MMD-D).** A quick recap on unbiased $U$-statisticd

estimator for $\mathrm{MMD}^2$ when $m = n$:

$$\widehat{\mathrm{MMD}}_u^2(S_{\mathbb{P}}, S_{\mathbb{Q}}; k) := \frac{1}{n(n-1)} \sum_{i \neq j} H_{ij} \tag{3}$$

$$H_{ij} := k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i).$$

Compared to training a classifier, MMD-D focuses on learning a powerful deep kernel function $k_\theta$,

$$k_\phi(x, y) = [(1 - \epsilon)\kappa(\phi(x), \phi(y)) + \epsilon]q(x, y), \tag{4}$$

where $\phi : \mathcal{X} \to \mathbb{R}^k$ is the deep neural network (with parameters $\theta_\phi$) which outputs the DRs of samples, $\epsilon$ is the interpolation weigth that $0 < \epsilon < 1$, and $\kappa$ and $q$ are characteristic kernels with hyperparameters $\theta_\kappa$ and $\theta_q$ respectively. To ensure the deep kernel can directly measure the distance of representations of complex-structured samples, optimizing the kernel with the highest test power will approximately maximize [Sutherland et al., 2017, Liu et al., 2020]

$$\mathcal{J} := \mathrm{MMD}^2(\mathbb{P}, \mathbb{Q}; k_\phi)/\sigma_{\mathcal{H}_1}(\mathbb{P}, \mathbb{Q}; k_\phi), \tag{5}$$

where $\sigma^2_{\mathcal{H}_1}(\mathbb{P}, \mathbb{Q}; k_\phi) := 4(\mathbb{E}[H_{12}H_{13}] - \mathbb{E}[H_{12}]^2])$ is the variance of $\sqrt{n}\mathrm{MMD}_u^2 - \mathrm{MMD}^2$ under the alternative hypothesis $H_1 : \mathbb{P} \neq \mathbb{Q}$ by a standard central limit theorem [Liu et al., 2020], and the $H_{ij}$ follows the definition above.

**Permutation Testing.** According to the standard central limit theorem [Serfling, 2009], the test statistic $\hat{t}$ in Eq. (1) converges to normal distributions under both the null or alternative hypothesis [Lopez-Paz and Oquab, 2017]. Although it is feasible for us to derive the threshold $t_\alpha$ of the null hypothesis distribution and perform a traditional Z-Test, it is simpler and faster to instead implement a permutation test for all test statistics Eq. (1), Eq. (2) and Eq. (3) [Sutherland et al., 2017]. We will permute and randomly assign samples to new $S_{\mathbb{P}}^{\mathrm{te}'}$ and $S_{\mathbb{Q}}^{\mathrm{te}'}$ for many times. Under $H_0$, samples from $\mathbb{P}$ and $\mathbb{Q}$ should be interchangeable, implying that the test statistic should exhibit minimal variation between its value based on the original sequence of samples and its computation from several randomly permuted sequences. Thus, if the original test statistic is large enough than most of the statistic derived from the randomly permuted sequences, we can reject $H_0$ [Good, 2004].

# 3 REPRESENTATION-LEARNING TWO-SAMPLE TESTING

In this section, we introduce our proposed RL-TST framework and several implementations that could leverage the information from unlabelled data in two-sample testing. Next, we provide an understanding of why learning good representations could enhance the power of two-sample testing. At the end, we discuss the significant challenges if we want to use mainstream semi-supervised learning methods (e.g., methods based on label propagation [Lee et al., 2013]) to address two-sample testing problems, which is another framework to exploit information from unlabelled data.

## 3.1 OUR PROPOSAL: RL-TST

The key to enhance learning representations in two-sample testing is to leverage the information from both the labelled training and unlabelled testing data. Various semi- or self-supervised learning techniques can achieve it, but due to the unique properties of two-sample testing data, two samples often follow two very similar but in fact different distributions under the alternative hypothesis, which makes it difficult to obtain effective information from unlabeled samples through most label propagation [Lee et al., 2013] or augmentation-based [Grill et al., 2020] techniques. From the recent studies, Cheng and Xie [2024] claims that most of the two-sample testing data follow the manifold assumption, where the data are low-dimensional intrinsic manifolds embedded in high-dimensional space. Thus, we propose to use a two-phase pipeline in two-sample testing that leverages the labelled and unlabelled samples to learn IRs and DRs respectively [Dai and Le, 2015].

Generally, since the effectiveness of *auto-encoder-based* (AE-based) representation learning *mainly relies on the manifold assumption* [Vincent et al., 2008], the first phase is an unsupervised AE-based representation learning, which learns a feature extractor that captures the inherent features for both samples. The next phase is to train a *multilayer perceptron* (MLP, used to classify two samples) or a characteristic kernel (with optimized parameters) on those IRs of two-sample testing data, so the final model will exhibit the discriminative ability directly on the intrinsic manifolds of two samples [Belkin et al., 2006]. Finally, we apply the final model to the remaining samples (excluded in the second phase) to obtain their DRs and perform permutation testing on DRs to derive the final testing result. Overall, our framework can be generalised in three main steps: *learn IRs, learn DRs, and then testing*. The general framework can be visualised in Figure 1. In the following, we will introduce our framework in detail.

**Learning Details.** Since RL-TST has two phases, for C2ST, we need to decompose the classifier-based model $f$ into two parts: a feature extractor $\phi \in \mathcal{F} : \mathcal{X} \to \mathbb{R}^k$ that used to learn IRs and followed by a classifier $g \in \mathcal{G} : \mathbb{R}^k \to \{0, 1\}$ that used to learn DRs. We denote by $\phi_f$ and $g_f$ the feature extractor and the classifier of a specified model $f$. For the input samples, removing the label information will leave an unlabeled dataset $\mathcal{S}_{\mathrm{unl}} = \{z_k\}_{k=1}^{m+n}$ that is equal to $S_{\mathbb{P}} \cup S_{\mathbb{Q}}$.

**Learning IRs.** The first step is to train a representation learning encoder on the whole unlabelled dataset $\mathcal{S}_{\mathrm{unl}}$, with the training objective mainly to minimize the differences between input and reconstructed output. Generally, we aim to learn a featurizer $\phi^*$ such that

$$\phi^*, \psi^* = \arg\min_{\phi, \psi} \widehat{\mathcal{R}}_{\mathrm{IR}}(\phi, \psi), \tag{6}$$

where $\psi : \mathbb{R}^k \to \mathcal{X}$ is the decoder. For a specific example

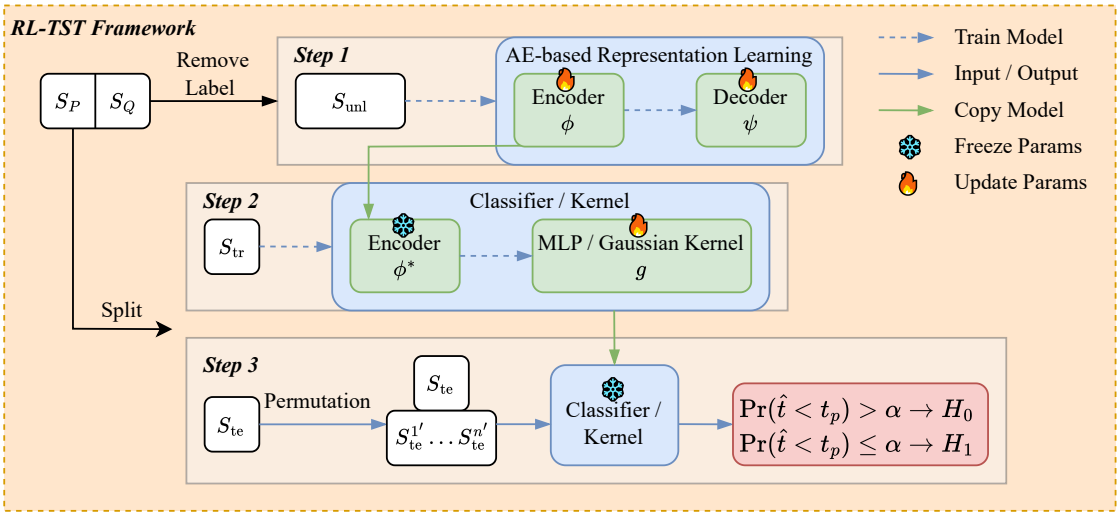



Figure 1: Overview of the RL-TST framework. Firstly, an encoder was learned from any AE-based representation learning algorithms on whole data, which can chosen from standard auto-encoder, wasserstein auto-encoder, etc. Secondly, fine-tune the learned encoder followed by a component that has the discriminative ability. At last, utilizing the final classifier or deep kernel to perform the permutation test based on statistic Eq. (1), Eq. (2) or Eq. (3) to derive the two-sample testing result.

(e.g., *mean squared error* (MSE) in basic auto-encoder)

$$\widehat{\mathcal{R}}_{\text{IR}}(\phi, \psi) = \frac{1}{|\mathcal{S}_{\text{unl}}|} \sum_{z_i \sim \mathcal{S}_{\text{unl}}} \|\psi(\phi(z_i)) - z_i\|_2^2, \quad (7)$$

and the objective will be slightly varied with some penalization terms depending on different AE-based algorithms, such as variational auto-encoder [Kingma and Welling, 2019] or wasserstein auto-encoder [Tolstikhin et al., 2018]. After training, $\phi^*(z_i)$ is called the IR of $z_i$.

**Learning DRs.** Then, utilize the featurizer $\phi^*$ from the representation learning model and concatenate with either an MLP $g$ or a deep kernel $k$ to form a final model $\mathcal{M}$. The combined model is fine-tuned on $\mathcal{S}_{\text{tr}}$, focusing on maximizing the distance of MLP's output of two samples or the test power of MMD regarding the two samples. Formally, for MLP-based $\mathcal{M} := g \circ \phi^*$, we aim to learn a function $g^*$ on $\mathcal{S}_{\text{tr}}$ by minimizing

$$\mathcal{L}_{\text{DR}}(g) = \frac{1}{|\mathcal{S}_{\text{tr}}|} \sum_{(z_i, l_i) \sim \mathcal{S}_{\text{tr}}} \ell_{\text{DR}}(\phi^*(z_i), l_i, g), \quad (8)$$

where $\ell_{\text{DR}}(\phi^*(z_i), l_i, g)$ can be empirically implemented using a loss function such as *binary cross entropy* (BCE) loss, defined for binary classification as:

$$\hat{\ell}_{\text{DR}}(\phi^*(z_i), l_i, g) = -[l_i \log \hat{p}_i + (1 - l_i)(1 - \log \hat{p}_i)], \quad (9)$$

where $\hat{p}_i = (1 + g \circ \phi^*(z_i))^{-1}$ is the estimate of $p$. As $g^*$ is an MLP, so $g^*$ can be expressed by $g^* = h^* \circ h^*_{\text{rep}}$ where $h^*_{\text{rep}} \in \{h_{\text{rep}} : \mathbb{R}^k \to \mathbb{R}^{d_{\text{rep}}}\}$ and $h^* \in \{h : \mathbb{R}^{d_{\text{rep}}} \to \{0, 1\}\}$. Normally, $h^*$ is called a classification head, and $h^*_{\text{rep}}$ is called a representation function. Thus, a DR of $z_i$ is $h^*_{\text{rep}} \circ \phi^*(z_i)$ if we use C2ST-based methods for testing.

For a MMD-based $\mathcal{M} := k_{\phi^*}$, we aim to empirically learn a deep kernel $k^*$ (shown in Eq. (4)) on the $\mathcal{S}_{\text{tr}}$ by maximizing the empirical estimate of $\mathcal{J}$ in Eq. (5)

$$\widehat{\mathcal{J}}_{\text{DR}}(S^{\text{tr}}_{\mathbb{P}}, S^{\text{tr}}_{\mathbb{Q}}; k_{\phi^*}) = \frac{\widehat{\text{MMD}}^2_u(S^{\text{tr}}_{\mathbb{P}}, S^{\text{tr}}_{\mathbb{Q}}; k_{\phi^*})}{\hat{\sigma}_{\mathcal{H}_1, \lambda}(S^{\text{tr}}_{\mathbb{P}}, S^{\text{tr}}_{\mathbb{Q}}; k_{\phi^*})} \quad (10)$$

where $S^{\text{tr}}_{\mathbb{P}}$ and $S^{\text{tr}}_{\mathbb{Q}}$ be the splitting samples of $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$ in the training set $\mathcal{S}_{\text{tr}}$ and $n^{\text{tr}}_x$ and $n^{\text{tr}}_y$ be the sample size of $S^{\text{tr}}_{\mathbb{P}}$ and $S^{\text{tr}}_{\mathbb{Q}}$. $\hat{\sigma}_{\mathcal{H}_1, \lambda}$ represents for the regularized estimator of $\sigma_{\mathcal{H}_1}$ defined in [Liu et al., 2020].

**Testing.** In the end, compute any of the three test statistics in Eq. (1) (by setting $f^*$ as $g^* \circ \phi^*$), in Eq. (2) (by setting $h^*$ as $h^*_{\text{rep}} \circ \phi^*$, or in Eq. (3) (by setting $k^*(\cdot, \cdot) = [(1 - \epsilon)\kappa^*(\phi^*(\cdot), \phi^*(\cdot)) + \epsilon]q^*(\cdot, \cdot)$ based on the original sequence of samples and the $r$ times permuted samples [Doran et al., 2014], reject $H_0$ if original statistic is larger than the threshold derived from permuted statistics. The difference of these three statistics are pure accuracy in Eq. (1), linear kernel that contain confidence information from accuracy in Eq. (2), and higher-order deep kernel that explains complex structure information in Eq. (3).

**Discussion of Alternatives.** In the first phase, if the data sometimes follow a smoothness assumption where *small perturbations will not influence the distribution of data* (i.e., the distance between two distributions will be significantly larger than the distance between the augmentations [Xie et al., 2020]), an augmentation-based self-supervised rep-

**Algorithm 1** Training models in RL-TST framework

**Input:** $S_{\mathbb{P}}, S_{\mathbb{Q}}, \phi, \psi, g, \kappa_\phi$
$\phi^* \leftarrow \arg\min_\phi \widehat{\mathcal{R}}(\phi, \psi, S_{\mathbb{P}} \cup S_{\mathbb{Q}})$ *Eq. (7): learning IRs*
**Split:** $S_{\mathbb{P}}, S_{\mathbb{Q}}$ into $S^{\text{tr}}_{\mathbb{P}}, S^{\text{tr}}_{\mathbb{Q}}, S^{\text{te}}_{\mathbb{P}}, S^{\text{te}}_{\mathbb{Q}}$
$S^{\text{tr}} \leftarrow (S^{\text{tr}}_{\mathbb{P}}, \mathbf{0}) \cup (S^{\text{tr}}_{\mathbb{Q}}, \mathbf{1})$
**if** learn DRs by classifier **then**
  $g^* \leftarrow \arg\min_g \widehat{\mathcal{L}}(g, S^{\text{tr}})$ *Eq. (8)*
  $\mathcal{M} \leftarrow g^* \circ \phi^*$ *learning DRs by classifiers*
**else**
  $\kappa^* \leftarrow \arg\min_{\kappa_{\phi^*}} \widehat{\mathcal{J}}(\kappa_{\phi^*}, S^{\text{tr}}_{\mathbb{P}}, S^{\text{tr}}_{\mathbb{Q}})$ *Eq. (10)*
  $\mathcal{M} \leftarrow \kappa^*_{\phi^*}$ *learning DRs by deep kernels*
**end if**
**Output:** $\mathcal{M}$

resentation learning can also be used [Grill et al., 2020, Li et al., 2021]. Instance-level representation learning through contrastive discrimination, like SimCLR [Chen et al., 2020], is not recommended in two-sample testing scenarios, even when both the smoothness and manifold assumptions are satisfied, since only holistic approaches, such as BYOL [Grill et al., 2020], can effectively capture the IRs of whole data. However, we are proposing a general framework that can be applied to all the two-sample data, so we do not include such representation learning algorithm in the framework.

### 3.2 DISCUSSION OF THE TYPE I ERROR AND TYPE II ERROR CONTROL

**Control of type I error** $(\alpha)$**.** In testing phase, since we are conducting the permutation test on the learned representations of test data, it can guarantee the validity of the type I error rate under exchangeability conditions [Hemerik and Goeman, 2017, Biggs et al., 2023]. In general, RL-TST learned an autoencoder-based feature map in an unsupervised manner, independent of the sample labels. Hence, the unsupervisedly learned nonlinear transformation applies symmetrically to both samples. This symmetry preserves the exchangeability required for a permutation test, controlling of type I error regardless of nonlinear transformations.

For the example to understand the type I error control, given the two samples $S_{\mathbb{P}} \sim \mathbb{P}^n$ and $S_{\mathbb{Q}} \sim \mathbb{Q}^m$, we first treat them as unlabeled data $Z = \{z_1, \ldots, z_{n+m}\}$ and learn an unsupervised map $\phi^*$ from $\mathcal{X}$ to some feature space. Because $\phi^*$ depends only on the unlabeled observations, it remains the same under any permutation of the labels that split $Z$ into two groups $S'_{\mathbb{P}}$ and $S'_{\mathbb{Q}}$ (i.e., $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$ are interchangeable from the perspective of $\phi^*$), so that under the null hypothesis $\mathbb{P} = \mathbb{Q}$, all label assignments are equally plausible and the embedded data preserve exchangeability. When we define a test statistic $T(\phi^*(S_{\mathbb{P}}), \phi^*(S_{\mathbb{Q}}))$, permuting the labels leaves $\phi^*$ unchanged, implying that $T(\phi^*(S_{\mathbb{P}}), \phi^*(S_{\mathbb{Q}}))$ has the same distribution as $T(\phi^*(S'_{\mathbb{P}}), \phi^*(S'_{\mathbb{Q}}))$ for any permutation. Hence, the permutation test on the embedded data remains valid, preserving the type I error control by leveraging the symmetry inherent in unsupervised representation learning [Biggs et al., 2023].

Table 1: Main theoretical results from [Yan and Zhang, 2023], which displays that the relationship between the dimension of the data (denoted by $p$) and the sample size (denoted by N) will affect the $l$-order moment discrepancy the kernel two-sample testing being detected.

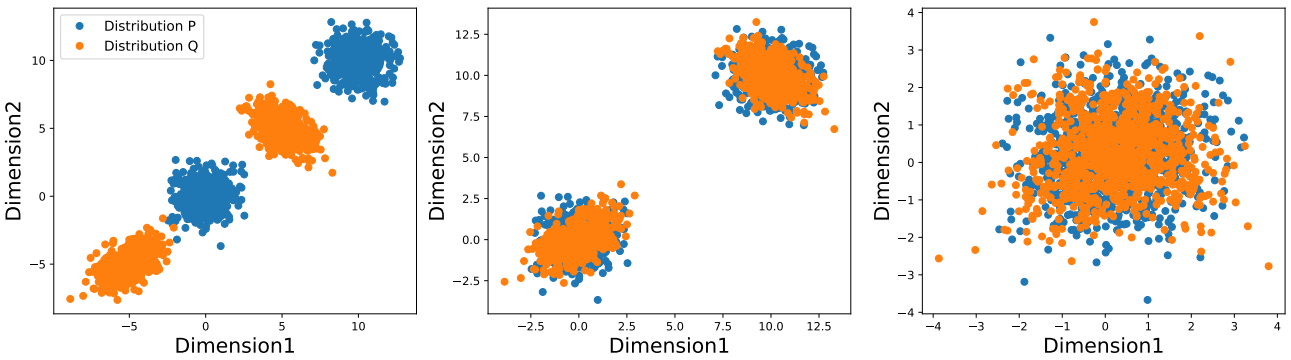
| Dimension and sample size orders | Main features captured |
|---|---|
| $N = o(\sqrt{p})$ | Mean and trace of covariance |
| $N = o(p^{3/2})$ | Mean and covariance |
| $N = o(p^{l-1/2})$ | The first $l$th moments |
| Fixed $p$, growing $N$ | Total homogeneity |

**Control of type II error** $(\beta)$**.** For the test power $(1 - \beta)$ under alternative hypothesis, the analysis of how well the learned representations (both IRs and DRs) preserve or amplify distributional differences remains an active research challenge in the field. However, under the manifold assumption, the learned IRs retain the key compressed information necessary for distinguishing the two samples. In recent theoretical research of two-sample testing, as shown in Table 1, researchers have found that the lower dimension is reduced relative to the sample size, the higher-order moment discrepancy the MMD test is capable to detect [Yan and Zhang, 2023]. Therefore, if we learn better representations of input samples under the manifold assumption, it will effectively extract the lower dimensional features without increasing sample size, making the MMD test more likely to capture the higher-order discrepancy. Since C2ST or C2ST-L are essentially MMD-based two-sample testing methods with sign kernel or linear kernel [Liu et al., 2020], it is compelling that if we can learn better representations on the two-sample testing data, we will derive higher test power from any two-sample testing methods. In Appendix C.5, we also provide the empirical results to show that the learned IRs will not lose information of the original distributional difference.

### 3.3 CAN WE USE SEMI-SUPERVISED LEARNING METHODS FOR TESTING?

After introducing the RL-TST framework, we find a way to make good use of the information from unlabelled data to improve test power. In machine learning, there is another modern technique called *semi-supervised learning* (SSL), which is also designed to utilize the information from unlabelled data to improve classification. In this section, we will discuss why mainstream SSL methods (e.g., label propagation [Lee et al., 2013]) cannot be generally used to address

[1]The result does not include standard deviation, since each trial we are testing whether two groups of drawn samples are from same distribution or not, and the result of each trial is either 0 or 1.




(a) HDGM-Easy (b) HDGM-Medium (c) HDGM-Hard

Figure 2: Visualisation of first two dimensions of samples for different levels of the high-dimensional Gaussian mixture (HDGM) dataset whose dimension is 10. For the HDGM-Easy and HDGM-Medium, the cluster mean difference $\Delta_\mu$ within the same distribution is 10, while for the HDGM-Hard, $\Delta_\mu$ is 0.5. For the HDGM-Easy, the distribution mean difference $\Delta_q$ between $\mathbb{P}$ and $\mathbb{Q}$ is 5, while for HDGM-Medium and HDGM-Hard, $\Delta_q$ is 0. Other setting of how to generate HDGM dataset is described in Appendix C.4.

Table 2: Result of C2ST test power on HDGM-Easy, -Medium and -Hard (d=10), on different total size $N$ of two samples inputed in 100 trials. Compared to other application of mainstream SSL methods on C2ST, where C2ST-CR, C2ST-PL, C2ST-GM, and C2ST-HB represent that we learn the classifier of C2ST using four different mainstream SSL frameworks. [1]

| Method | HDGM-Easy | | | HDGM-Medium | | | HDGM-Hard | | |
|---|---|---|---|---|---|---|---|---|---|
| | N=60 | N=80 | N=100 | N=2000 | N=3000 | N=4000 | N=4000 | N=6000 | N=8000 |
| C2ST | 0.64 | 0.91 | 0.99 | **0.44** | 0.82 | **0.97** | 0.29 | **0.49** | **0.78** |
| C2ST-CR | 0.65 | 0.92 | **1.00** | 0.40 | 0.84 | **0.97** | 0.32 | 0.42 | 0.75 |
| C2ST-PL | 0.72 | 0.96 | 0.99 | 0.40 | 0.76 | 0.93 | **0.36** | 0.45 | 0.77 |
| C2ST-GM | 0.64 | 0.92 | **1.00** | 0.43 | **0.85** | **0.97** | 0.22 | 0.40 | 0.72 |
| C2ST-HB | **0.99** | **1.00** | **1.00** | 0.25 | 0.43 | 0.58 | 0.28 | 0.43 | 0.65 |

two-sample testing problems where $S_{\text{tr}}$ is regarded as the training set and $S_{\text{unl}}$ is regarded as the unlabeled set.

We first recall the basic assumptions required by SSL methods [Chapelle et al., 2006]:

- *Smoothness assumption:* If points $x_1$ and $x_2$ are close, then so should be their labels $y_1$, $y_2$.
- *Cluster assumption:* If points are in the same cluster, they are likely to be of the same class.
- *Manifold assumption:* The (high-dimensional) data lie (roughly) on a low-dimensional manifold.

Based on those assumptions, there are five representative semi-superivsed learning frameworks [Yang et al., 2023]: consistency-regularisation [Xie et al., 2020], pseudo-labelling [Lee et al., 2013], graph-based [Song et al., 2022], generative-models [Kingma and Welling, 2014] and hybrid [Sohn et al., 2020] SSL methods. The consistency regularisation techniques assume that the model can predict the same label between the augmented or permuted samples and original samples; the pseudo-labelling techniques assume that if the samples form a cluster, then all of samples have same label in the same cluster; graph-based techniques assume that the input samples are graph-structured or the input can be represented as graph-structured data; the generative-model techniques assume that the generative samples have the same distribution as input samples. Hybrid techniques can embrace the advantages of the above techniques, however they also require all assumptions to hold. The details of above SSL methods are demonstrated in Appendix B.1.

Even though those methods are comprehensive and advanced in the field of SSL to leverage the information from unlabelled data, they are inherently *incompatible* to the two-sample testing scenarios. In the general case, data in two-sample testing often form a high-degree of overlapping between two samples, which will decrease the useful information content of unlabelled data [Chapelle et al., 2006], and the empirical verification on the challenges of applying semi-supervised techniques is presented below.

**Empirical Results for Validity of Mainstream Semi-supervised Learning Techniques.** In Figure 2, it shows different levels of overlap in the two-sample testing data, and we will conduct experiments on these datasets. Since

SSL methods mainly applied on the classifier-based model, we explore the performance of C2ST-based methods.

In Table 2[2], the empirical results show that even though the application of mainstream SSL methods on C2ST can have better performance on the HDGM-Easy dataset, but it often yields poorer results compared to the original C2ST on HDGM-Medium and HDGM-Hard datasets, which represents the common overlapping distribution data in the context of two-sample testing. This underperformance can be attributed to the fundamental nature of the testing procedure, which is distinct from accuracy evaluation in the classification tasks. In two-sample testing, our aim is to maximize the distance of two whole samples, rather than focusing on correctly classifying all the unseen data points (which is also impossible in two-sample testing). During the training of classifiers, we manually assign labels to facilitate distinction by the classifier, whereas in testing, we consider the two samples holistically rather than focusing on individual instance accuracy.

Furthermore, mainstream SSL methods, which primarily enhance classification through data augmentation based on smoothness assumptions or propagate pseudo labels based on clustering assumptions, aim to generate high-confidence training data. However, in two-sample testing, these approaches are flawed; data augmentation may alter the samples' distributions, and pseudo label propagation often proves inaccurate. These discrepancies lead to the frequent ineffectiveness of these SSL methods in two-sample testing contexts. The details of why testing data does not always satisfy the assumptions made by many SSL methods is analyzed in Appendix B.2. Moreover, two-phase representation learning can also be considered as semi-supervised learning [Dai and Le, 2015], and RL-C2ST is the classifier-based model implemented on the RL-TST framework, so we will also provide more concrete theoretical analysis on how to understand the test power improvement of RL-C2ST in a semi-supervised discriminator's view in Appendix D.

**Insights For Futures.** An interesting view of SSL is that SSL presumes perfectly reliable labels and focuses on propagating that clean supervision to unlabeled data. However, the alternative hypothesis in two-sample testing posits a distributional shift in the feature and label joint distribution, which will directly treat the observed labels as potentially corrupted. This mismatch suggests a new research direction: build noise-aware SSL objectives that estimate a label-corruption channel and couple it with geometry-based regularisation, or design representations optimised to maximise the power of label-shift tests. These approaches would reconcile SSL with the realities captured by distribution testing, turning label noise from an obstacle into a signal.

[2]The experimental details of this table can be found in Appendix B.1, where all detailed description of semi-supervised methods and how to use these methods in testing are introduced.

# 4 EXPERIMENTS

**Datasets.** We conducted experiments on five different datasets to thoroughly evaluate our methods in two different aspects: 1) To assess the performance of alternative SSL learning methods directly applied to two-sample testing methods, we utilized three synthetic datasets: *HDGM-Easy*, *HDGM-Medium*, and *HDGM-Hard*. As we have already mentioned in Appendix B, these datasets represent three different levels of data structure complexity often encountered in the two-sample testing scenarios, which can verify whether the mainstream SSL techniques are robust to various two-sample testing tasks or not; 2) To verify the effectiveness of proposed two-phase RL-TST framework applied on two-sample testing methods (i.e., RL-C2ST, RL-MMD-D) than the other existing work, we conduct the experiments of an implemented RL-TST against other SOTA two-sample testing methods. These experiments were carried out on three representative datasets: *MNIST*, *ImageNet*, and *HDGM-D* (a.k.a. *HDGM-Hard*) to evaluate the enhanced performance of our RL-TST framework. Detailed descriptions of these datasets are provided in Appendix C.1.

**Baselines.** The baselines are the SOTA two-sample testing methods from the existing frameworks. Our main empirical experiments aim to evaluate the performance of two-sample testing methods built on the RL-TST framework (i.e., RL-C2ST and RL-MMD-D) against several SOTA methods in two-sample testing, specifically C2ST, C2ST-L, MMD-D, and MMD-FUSE. These methods serve as competitive references to highlight the improvements achieved by focusing on learning good representations from RL-TST framework. The following are the overall descriptions of each method.

- C2ST: C2ST learns a classifier and uses statistic in Eq. (1) to measure the difference of two samples [Lopez-Paz and Oquab, 2017].
- C2ST-L: same as C2ST, except it uses the statistic in Eq. (2) to measure the absolute mean differences between the probability of the logits of two samples, as we discuss in the Section 2 [Cheng and Cloninger, 2022].
- MMD-D: MMD test trains a neural network to derive a deep kernel [Liu et al., 2020].
- MMD-FUSE: a SOTA testing method that learns IRs from different fixed kernels without data splitting [Biggs et al., 2023].
- RL-C2ST: RL-C2ST is a C2ST improved by our proposed RL-TST framework, as we discussed in the Section 3.1. RL-C2ST-L uses Eq. (2) as test statistic.
- RL-MMD-D: RL-MMD-D is the implementation of RL-TST on the MMD-D as we discussed in the Section 3.1.

For their detailed implementations and parameter settings, please refer to Appendix C.3. Moreover, the methods listed

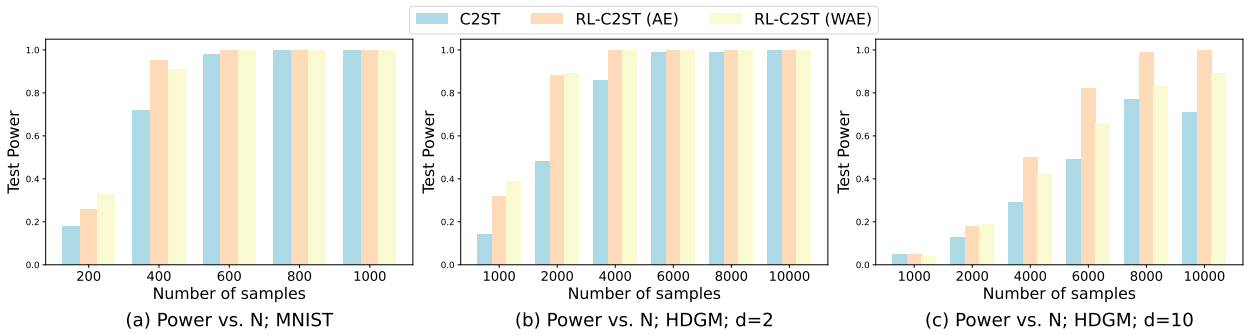



Figure 3: Test power of two different implementations of RL-TST framework on the two-sample testing method C2ST. Barplot to show how standard auto-encoder RL-C2ST and wasserstein auto-encoder RL-C2ST both outperform C2ST in the *MNIST* dataset (a), *HDGM-D* when $d = 2$ (b) and *HDGM-D* when $d = 10$ (c).

in the Table 2 represent our implementations, which serves as an alternative idea's motivation experiment to highlight the challenges of the research gap between the fields of semi-supervised learning and two-sample testing. Thus, they are not intended to be considered as formal baselines.

**Ablation Study.** We conduct the ablation study on RL-TST framework by not solely applying one single representation learning algorithm on the original two-sample testing methods. To ensure the effectiveness of the RL-TST framework can be further investigated with other advanced representation learning algorithms, except for comparing one basic standard AE [Schmidhuber, 2015], we also implement another AE-based representation learning algorithm, *Wasserstein auto-encoder* (WAE) [Tolstikhin et al., 2018], where they both show that various IR learning algorithms can all leverage the information discarded by data splitting to improve the test power of original methods.

For the details of these two AE-based representation learning, the standard auto-encoder has an unrestricted latent space which can more focus on the reconstruction [Schmidhuber, 2015], while the wasserstein auto-encoder can match the latent space with a target prior distribution (i.e., Gaussian) to make sure the generating power [Tolstikhin et al., 2018]. Thus, in application, depending on the characteristics of different AEs, we can choose the suitable one for the downstream task or target data structure.

For the empirical experiments result, the visualized result of how two kinds of RL-C2STs outperform C2ST is displayed in Figure 3. In both dataset *MNIST* and *HDGM-Hard*, we can see that the test powers of RL-C2STs are higher than that of C2ST no matter how many numbers of two samples are drawn from the distribution. Although the differences between two methods are little when $N$ is small, the test powers of RL-C2STs have a huge gap over C2ST when $N$ is large enough and converges to 1 with a relative smaller $N$ compare to C2ST. Since under the alternative hypothesis $H_1 : \mathbb{P} \neq \mathbb{Q}$, if the number of samples goes to infinity, an effective two-sample testing method will always reject the null hypothesis $H_0 : \mathbb{P} = \mathbb{Q}$, thus, the less samples needed to reach the test power of 1, the better performance the method has. Thus, the empirical results can clearly verify that no matter what kinds of representation learning algorithms that can effectively learn IRs from two-sample testing data before learning DRs, we can finally derive a better representation than purely learning DRs.

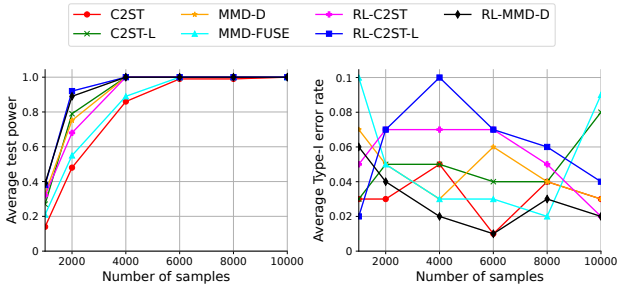
Compared to C2ST, both RL-C2STs learn a compact and potentially more informative representation of the whole data, which makes efficient use of the unlabelled test data. This can not only discover underlying patterns or features that might not be directly related to the labels yet to the data distribution itself, but also provide a regularizing effect to prevent the model being more likely to overfit the training data. Similar to all of the applications of RL-TST, such featurizer in the RL-TST can result in better generalization from the learned representations and improve the classifier's performance on the testing set predictions. The empirical outperformance of different learning algorithms also validates that RL-TST is a compelling framework for two-sample testing methods learning from unlabelled data.

**Result Analysis.** After we validate the effectiveness of RL-TST on C2ST, we will also display how effective the RL-TST applied on two advanced two-sample testing methods, which results in RL-C2ST-L and RL-MMD-D. They both show not only how they improve the original C2ST-L and MMD-D, but also how they outperform the most SOTA testing method MMD-FUSE.

The overall results on the HDGM dataset are shown in Figure 4. We can see that all the RL-TST methods have higher test power than their original methods, no matter how we choose $N$, while all type-I errors are reasonably controlled around $\alpha = 0.05$. We provide an extra type I error checking experiment under different levels of significance level and under $1,000$ trial repetitions in Appendix C.9. For *MNIST* and *ImageNet* datasets, the results of all methods are shown in Table 3, all the RL-TST methods still outperform the original methods. Moreover, RL-C2ST-L and RL-MMD-D,

Table 3: MNIST and ImageNet ($\alpha = 0.05$). Average test power for comparing $M$ real MNIST images to $M$ DCGAN-generated MNIST images, and Average test power for comparing $M$ real ImageNet images to $M$ StyleGAN-XL-generated ImageNet images. The three implementations of RL-TST are all using standard auto-encoder in the learning IRs step, we could replace it into other alternative auto-encoders, such as wasserstein auto-encoder discussed in the Section 4.

| Method | MNIST | | | | | | ImageNet | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M=200 | M=400 | M=600 | M=800 | M=1000 | Avg. | M=200 | M=400 | M=600 | M=800 | M=1000 | Avg. |
| C2ST | $0.180_{\pm.046}$ | $0.720_{\pm.023}$ | $0.980_{\pm.013}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | 0.776 | $0.150_{\pm.022}$ | $0.300_{\pm.029}$ | $0.350_{\pm.026}$ | $0.600_{\pm.036}$ | $0.850_{\pm.016}$ | 0.450 |
| C2ST-L | $0.250_{\pm.047}$ | $0.730_{\pm.053}$ | $0.990_{\pm.009}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | 0.794 | $0.150_{\pm.042}$ | $0.350_{\pm.030}$ | $0.450_{\pm.040}$ | $0.700_{\pm.049}$ | $0.850_{\pm.034}$ | 0.500 |
| MMD-D | $0.290_{\pm.017}$ | $0.996_{\pm.009}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | 0.857 | $0.210_{\pm.031}$ | $0.400_{\pm.039}$ | $0.570_{\pm.033}$ | $0.780_{\pm.041}$ | $\mathbf{1.000}_{\pm.000}$ | 0.592 |
| MMD-FUSE | $0.320_{\pm.032}$ | $0.870_{\pm.033}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | 0.838 | $0.230_{\pm.029}$ | $0.450_{\pm.034}$ | $0.610_{\pm.037}$ | $0.790_{\pm.029}$ | $\mathbf{1.000}_{\pm.000}$ | 0.616 |
| RL-C2ST | $0.260_{\pm.049}$ | $0.950_{\pm.022}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | 0.842 | $0.200_{\pm.036}$ | $0.400_{\pm.049}$ | $0.500_{\pm.061}$ | $0.650_{\pm.050}$ | $0.950_{\pm.022}$ | 0.540 |
| RL-C2ST-L | $\mathbf{0.491}_{\pm.060}$ | $0.985_{\pm.013}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | **0.895** | $\mathbf{0.400}_{\pm.059}$ | $\mathbf{0.500}_{\pm.059}$ | $0.650_{\pm.056}$ | $0.750_{\pm.054}$ | $\mathbf{1.000}_{\pm.000}$ | 0.660 |
| RL-MMD-D | $0.420_{\pm.072}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | $\mathbf{1.000}_{\pm.000}$ | 0.884 | $0.330_{\pm.051}$ | $0.470_{\pm.069}$ | $\mathbf{0.680}_{\pm.055}$ | $\mathbf{0.890}_{\pm.037}$ | $\mathbf{1.000}_{\pm.000}$ | **0.674** |




(a) Power vs. N; d = 2 (b) Level vs. N; d = 2

Figure 4: Results on *HDGM-D* and *HDGM-S* for $\alpha = 0.05$. (a) average test power and (b) average type I error keeping $d = 2$ in 100 trials when increasing $N$ from $N = 1000$ to $N = 10000$. All RL-TST methods use the standard auto-encoder, we could replace it with other alternative auto-encoders, such as Wasserstein auto-encoder (see Section 4).

which are the most two powerful applications, can have the highest test power than other SOTA methods among different sample size $N$ or $M$, which can verify the improvement of our RL-TST on different two-sample testing methods.

**Discussion of Sequential Two-sample Testing.** Sequential two-sample testing methods also utilize information from the test data. Thus, in this part, we will compare ours and sequential two-sample tests. Briefly, we find that sequential two-sample testing has a different problem setting from what we are interested in [Li et al., 2022, Pandeva et al., 2022, Li et al., 2023]. In our problem setting, we assume the total number of samples is *fixed and given*, and we are trying to distinguish whether these two given samples are from the same distribution or not. No more extra data are provided for testing data and the test data is known, so it can be regarded as a *transductive learning problem*, while the sequential two-sample testing assumes the testing data can infinitely arrive as batch.

In sequential two-sample testing, a classifier is trained to determine whether two samples from a single batch originate from the same distribution. Initially, batches are split and fed sequentially into the classifier as testing data. Batches that do not reject the null hypothesis are concatenated with previous batches and used as training data for the classifier, continuing until all batches are exhausted or a single batch rejects the null hypothesis. The sequential nature of the test emerges from the use of e-values, which are updated as more data becomes available, allowing for a dynamic assessment of the testing hypothesis. However, this method should not be directly compared to our method due to different problem settings and designs. Firstly, in sequential two-sample testing, data are split into several batches and tests are conducted on single, small batches. Conversely, in other supervised two-sample testing approaches, data are only split into two halves, creating a trade-off between the number of training and testing samples.

Furthermore, the design of our RL-TST framework is compatible with any other supervised two-sample testing framework, including sequential two-sample testing. As long as a proportion of data is used for testing, we can remove the labels from this testing data and concatenate it into the training data. This allows us to learn IRs through representation learning, followed by the original supervised two-sample testing framework. We also provide experimental results in Appendix C.7 demonstrating that RL-TST can outperform these sequential approaches within the same setting.

## 5 CONCLUSION

This paper presents a unified view, *focusing on learning good representations from both labelled and unlabelled samples*, to both leverage the discarded information in the data splitting process and enhance the discriminative ability, which can address the existing drawbacks of two-sample testing methods. In order to examine the viability of the view, we conduct a thorough survey in the field of two-sample testing and the potential fields that enable to utilize information from unlabelled data, and propose a feasible framework that empirically improve the performance of two-sample testing methods. In the future, many advanced representation learning techniques for two-sample testing can be developed based on the new research direction proposed in this paper.

## 6 ACKNOWLEDGMENT

This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative. LP is supported by ARC (Grant No. LP240100101). FL is supported by the Australian Research Council (ARC) with grant number DE240101089, LP240100101, DP230101540 and the NSF&CSIRO Responsible AI program with grant number 2303037. MG was supported by ARC grants DE210101624 and DP240102088, as well as WIS-MBZUAI grant 142571.

### References

Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *Journal of the ACM*, 57(3):19:1–19:46, 2010.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006.

Federico Bergamin, Pierre-Alexandre Mattei, Jakob Drachmann Havtorn, Hugo Sénétaire, Hugo Schmutz, Lars Maaløe, Soren Hauberg, and Jes Frellsen. Model-agnostic out-of-distribution detection using combined statistical tests. In *AISTATS*, 2022.

Peter J Bickel. A distribution free version of the smirnov two sample test in the p-variate case. *The Annals of Mathematical Statistics*, 40(1):1–23, 1969.

Felix Biggs, Antonin Schrab, and Arthur Gretton. MMD-FUSE: Learning and Combining Kernels for Two-Sample Testing Without Data Splitting. In *NeurIPS*, 2023.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.

Patric Bonnier, Harald Oberhauser, and Zoltán Szabó. Kernelized cumulants: Beyond kernel mean embeddings. In *NeurIPS*, 2023.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. A sharp concentration inequality with application. *Random Struct. Algorithms*, 16(3):277–292, May 2000. ISSN 1042-9832.

Alberto Cano and Bartosz Krawczyk. Kappa updated ensemble for drifting data stream mining. *Machine Learning*, 109(1):175–218, 2020.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006. ISBN 9780262033589.

Hao Chen and Jerome H. Friedman. A new graph-based two-sample test for multivariate and object data. *Journal of the American Statistical Association*, 112(517):397–409, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Xiuyuan Cheng and Alexander Cloninger. Classification logit two-sample testing by neural networks for differentiating near manifold densities. *IEEE Transactions on Information Theory*, 68(10):6631–6662, 2022.

Xiuyuan Cheng and Yao Xie. Neural tangent kernel maximum mean discrepancy. *NeurIPS*, 2021.

Xiuyuan Cheng and Yao Xie. Kernel two-sample tests for manifold data. *Bernoulli*, 30(4):2572–2597, 2024.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *NeurIPS*, 2015.

Namrata Deka and Danica J. Sutherland. Mmd-b-fair: Learning fair representations with statistical testing. In *AISTATS*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Gary Doran, Krikamol Muandet, Kun Zhang, and Bernhard Schölkopf. A permutation-based kernel conditional independence test. In *UAI*, 2014.

Zhen Fang, Jie Lu, Anjin Liu, Feng Liu, and Guangquan Zhang. Learning bounds for open-set learning. In *ICML*, 2021a.

Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4309–4322, 2021b.

Rui Gao, Liyan Xie, Yao Xie, and Huan Xu. Robust hypothesis testing using Wasserstein uncertainty sets. In *NeurIPS*, 2018.

Ruize Gao, Feng Liu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *ICML*, 2021.

Debarghya Ghoshdastidar and Ulrike von Luxburg. Practical methods for graph two-sample testing. In *NeurIPS*, 2018.

Debarghya Ghoshdastidar, Maurilio Gutzeit, Alexandra Carpentier, and Ulrike von Luxburg. Two-sample tests for large random graphs using network statistics. In *COLT*, 2017.

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Behrnhard Schölkopf. Domain adaptation with conditional transferable components. In *ICML*, 2016.

Phillip I. Good. *Permutation, Parametric, and Bootstrap Tests of Hypotheses*. Springer-Verlag, 2004.

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

Arthur Gretton, Bharath Sriperumbudur, Dino Sejdinovic, Heiko Strathmann, and Massimiliano Pontil. Optimal kernel choice for large-scale two-sample tests. In *NeurIPS*, 2012b.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020.

Ruth Heller and Yair Heller. Multivariate tests of association based on univariate tests. In *NeurIPS*, 2016.

Jesse Hemerik and Jelle Goeman. Exact testing with random permutations. *TEST*, 27(4):811–825, 2017.

Wittawat Jitkrittum, Zoltan Szabo, Kacper Chwialkowski, and Arthur Gretton. Interpretable distribution features with maximum testing power. In *NeurIPS*, 2016.

Ilmun Kim, Aaditya Ramdas, Aarti Singh, and Larry Wasserman. Classification accuracy as a proxy for two sample testing. *Annals of Statistics*, 49(1):411 – 434, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Matthias Kirchler, Shahryar Khorasani, Marius Kloft, and Christoph Lippert. Two-sample testing using deep learning. In *AISTATS*, 2020.

Jonas M. Kübler, Wittawat Jitkrittum, Bernhard Schölkopf, and Krikamol Muandet. Learning kernel tests without data splitting. In *NeurIPS*, 2020.

Jonas M. Kübler, Vincent Stimper, Simon Buchholz, Krikamol Muandet, and Bernhard Schölkopf. Automl two-sample test. In *NeurIPS*, 2022.

Jonas M. Kübler, Wittawat Jitkrittum, Bernhard Schölkopf, and Krikamol Muandet. A Witness Two-Sample Test. In *AISTATS*, 2022.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML*, 2013.

E. L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses*. Springer Texts in Statistics. Springer, 2005.

Matthieu Lerasle, Zoltán Szabó, Timothée Mathieu, and Guillaume Lecué. MONK outlier-robust mean embedding estimation by median-of-means. In *ICML*, 2019.

Muxing Li, Zesheng Ye, Yixuan Li, Andy Song, Guangquan Zhang, and Feng Liu. Membership inference attack should move on to distributional statistics for distilled generative models. *arXiv preprint arXiv:2502.02970*, 2025.

Weizhi Li, Gautam Dasarathy, Karthikeyan Natesan Ramamurthy, and Visar Berisha. A label efficient two-sample test. In *UAI*, 2022.

Weizhi Li, Prad Kadambi, Pouria Saidi, Karthikeyan Natesan Ramamurthy, Gautam Dasarathy, and Visar Berisha. Active sequential two-sample testing. *arXiv preprint arXiv:2301.12616*, 2023.

Yazhe Li, Roman Pogodin, Danica J. Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. In *NeurIPS*, 2021.

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *ICML*, 2020.

Feng Liu, Wenkai Xu, Jie Lu, and Danica J. Sutherland. Meta Two-Sample Testing: Learning Kernels for Testing with Limited Data. In *NeurIPS*, 2021.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *ICLR*, 2017.

Teodora Pandeva, Tim Bakker, Christian A. Naesseth, and Patrick Forré. E-valuating classifier two-sample tests, 2022.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, January 2017.

Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH*, 2022.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

Antonin Schrab, Ilmun Kim, Melisande Albert, Beatrice Laurent, Benjamin Guedj, and Arthur Gretton. MMD Aggregated Two-Sample Test. *Journal of Machine Learning Research*, 2023.

Robert J Serfling. *Approximation theorems of mathematical statistics*. John Wiley & Sons, 2009.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

Yiliao Song, Jie Lu, Anjin Liu, Haiyan Lu, and Guangquan Zhang. A segment-based drift adaptation method for data streams. *IEEE Transactions on Neural Networks and Learning Systems*, Early Access, 2021.

Zixing Song, Xiangli Yang, Zenglin Xu, and Irwin King. Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8174–8194, 2022.

Petar Stojanov, Mingming Gong, Jaime G. Carbonell, and Kun Zhang. Data-driven approach to multiple-source domain adaptation. In *AISTATS*, 2019.

Yuhao Sun, Jiacheng Zhang, Zesheng Ye, Chaowei Xiao, and Feng Liu. Sample-specific noise injection for diffusion-based adversarial purification. In *ICML*, 2025.

Danica J. Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*, 2017.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013.

Ashraf Tahmasbi, Ellango Jothimurugesan, Srikanta Tirthapura, and Phillip B Gibbons. Driftsurf: Stable-state/reactive-state learning under concept drift. In *ICML*, 2021.

Bahar Taskesen, Man-Chung Yue, Jose H. Blanchet, Daniel Kuhn, and Viet Anh Nguyen. Sequential domain adaptation by synthesizing distributionally robust experts. In *ICML*, 2021.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *ICLR*, 2018.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *NeurIPS*, 2020.

Jian Yan and Xianyang Zhang. Kernel two-sample tests in high dimensions: interplay between moment discrepancy and dimension-and-sample orders. *Biometrika*, 110(2): 411–430, 2023.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023.

Jiacheng Zhang, Feng Liu, Dawei Zhou, Jingfeng Zhang, and Tongliang Liu. Improving accuracy-robustness trade-off via pixel reweighted adversarial training. In *ICML*, 2024a.

Shuhai Zhang, Yiliao Song, Jiahao Yang, Yuanqing Li, Bo Han, and Mingkui Tan. Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. In *ICLR*, 2024b.

# A ALGORITHM

We present the details of Algorithm 1 about framework for RL-TST in the following Algorithm 2.

**Algorithm 2** Paradigm of testing with RL-TST

**Input:** $S_{\mathbb{P}}$, $S_{\mathbb{Q}}$, significance level $\alpha$, an auto-encoder $f_a$ consist of a featurizer $\phi$ and a decoder $\phi^{-1}$, a final classifier $\mathcal{M} := g \circ \phi$ or a deep kernel $\mathcal{M} := k_\phi$ included with the featurizer, total epochs for learning IRs $T_{\mathrm{IR}}$, total epochs for learning DRs $T_{\mathrm{DR}}$.
**1:** Derive the unlabelled data $\mathcal{S}_{\mathrm{unl}} = \mathrm{shuffle}(S_{\mathbb{P}} \cup S_{\mathbb{Q}})$
*# Phase 1: derive Featurizer $\phi$ from learning IRs*
**for** $t = 1, 2, \ldots, T_{\mathrm{IR}}$ **do**
  **2:** $X_t \leftarrow$ minibatch from $\mathcal{S}_{\mathrm{unl}}$;
  **3:** $\phi^* \leftarrow \arg\min_\phi \mathcal{R}(f_a, X)$ based on Eq. (6);
**end for**
*# Phase 2: train a classifier or kernel $\mathcal{M}$ to learn DRs on $S^{\mathrm{tr}} = (S^{\mathrm{tr}}_{\mathbb{P}}, \mathbf{0}) \cup (S^{\mathrm{tr}}_{\mathbb{Q}}, \mathbf{1})$*
**for** $t = 1, 2, \ldots, T_{\mathrm{DR}}$ **do**
  **5:** $(X_t, l_t) \leftarrow$ minibatch from $\mathcal{S}^{\mathrm{tr}}$;
  **6:** $g^* \leftarrow \arg\min_g \mathcal{L}_{\mathrm{DR}}(\phi^*(X_t), l_t, g)$ based on Eq. (8), if learning classifier;
**end for**
or
**7:** $k^* \leftarrow \arg\max_{k_{\phi^*}} \widehat{\mathcal{J}}_{\mathrm{DR}}(S^{\mathrm{tr}}_{\mathbb{P}}, S^{\mathrm{tr}}_{\mathbb{Q}}; k_{\phi^*})$ based on 10, if learning deep kernel;
*# Phase 3: permutation test with $\hat{f}$ on $S^{\mathrm{te}} = S^{\mathrm{te}}_{\mathbb{P}} \cup S^{\mathrm{te}}_{\mathbb{Q}}$*
**8:** $est \leftarrow \hat{t}(S^{\mathrm{te}}_{\mathbb{P}}, S^{\mathrm{te}}_{\mathbb{Q}}; \mathcal{M})$ based on Eq. (1), Eq. (2), or Eq. (3);
**for** $i = 1, 2, \ldots, n_{\mathrm{perm}}$ **do**
  **9:** Shuffle $S^{\mathrm{te}}$ into $X$ and $Y$;
  **10:** $perm_i \leftarrow \hat{t}(X, Y; \mathcal{M})$
**end for**
**Output:** $\mathbb{I}\left[\frac{1}{n_{perm}} \sum_{i=1}^{n_{perm}} \mathbb{I}(est < perm_i) \leq \alpha\right]$

# B DISCUSSION OF SEMI-SUPERVISED LEARNING

## B.1 OVERVIEW OF MAIN CATEGORIES OF SEMI-SUPERVISED LEARNING METHODS

Building on the semi-supervised learning (SSL) assumptions, we will recap how contemporary SOTA SSL methods incorporate these principles and assumptions, setting the stage for an analysis of their applicability to the specific challenges presented by our problem setting.

**Transductive vs Inductive learning.** Classification tasks within machine learning can typically be categorized within two distinct problem settings: transductive and inductive learning [Chapelle et al., 2006]. *Transductive learning* is concerned with predicting the labels of the specific unlabelled data that was present during the training process, emphasizing a tailored fit to this data. *Inductive learning*, on the other hand, focuses on the generalization of the learned classifier to new, unseen data. In learnable two-sample testing, the goal is to test whether the given two samples are drawn from same distributions. To make it, we firstly split samples into labelled set and unlabelled set, then find out that whether it is possible to learn a classifier that can distinguish two samples from the mixed unlabelled set. It becomes apparent that applying SSL methodologies to the two-sample testing problem inherently requires a transductive learning approach. This conceptual groundwork necessitates a detailed examination of current SSL methods to identify their foundational assumptions and evaluate their performance in two-sample test scenarios.

**Major categories.** Currently, we identify that there are five main categories of SOTA SSL methods: consistency regularisation, pseudo-labelling, graph-based, generative models and hybrid (often a combination of consistency regularisation and pseudo-labelling) [Yang et al., 2023]. We will succinctly explicate how they work, and how they are applied for our downstream two-sample testing tasks in the experiments of various levels of HDGM.

- *Consistency Regularisation:* Based on the manifold assumption or the smoothness assumption, the consistency

regularisation methods apply consistency constraints to the final loss function, where the intuition is that if the data follows the smoothness assumption or manifold assumption, even though we construct some perturbations in the inputs, it will not influence the output of classification [Xie et al., 2020].

- *Pseudo-Labelling:* Pseudo-labelling uses its own predictions to generate labels for unlabelled data, which are then used to further train the model. It relies on the assumptions that model's high-confidence predictions are accurate. This assumption is based on the cluster assumption for the validity and efficacy of propagating labels to unlabelled data based on model predictions [Lee et al., 2013].
- *Graph-Based:* Graph-based methods will construct a similarity graph based on the raw dataset, where each node represents a data instance, and weighted-edge represents the similarity between two data instances. Based on the smoothness assumption, the label information can be propagated from labelled nodes to unlabelled nodes, if two nodes are closely connected in the constructed graph [Song et al., 2022].
- *Generative Models:* Generative methods learn to model the underlying distribution of both labelled data and unlabelled data, using this learned representation to generate new data points and infer missing labels. Based on the manifold assumption, the generative models aim to learn the underlying low-dimensional manifold and generate data points that adhere to the same manifold, used for further model training [Kingma and Welling, 2014].
- *Hybrid:* Hybrid methods are just combination of multiple methods, such as consistency regularisation, pseudo-labelling, and sometimes generative approaches. These models typically rely on the smoothness assumption and cluster assumption, in order to infer the labels of unlabelled data [Sohn et al., 2020].

### B.2 ANALYSIS OF WHY TWO-SAMPLE TESTING DATA ARE NOT SATISFIED FOR SSL

In the traditional two-sample testing problem settings, there is often overlap between the two samples. As we can see in Figure 2b and Figure 2c, for the HDGM-Medium and HDGM-Hard datasets, there are high-overlapping areas between two distributions. This will *highly violate the first two assumptions* of SSL mentioned previously. For the smoothness assumption, our dataset will have large amounts of nearly the same data points in two samples, but allocated different labels; this will notably influence the SSL methods that based on such assumption. For the cluster assumptions, we can see in HDGM-Medium that although there are two obvious clusters, they do not have the same labels within the same cluster in a holistic view. The SOTA SSL techniques will rely on at least one of the smoothness assumption or cluster assumption to ensure that the unlabeled samples' label information can be inferred, or extra training samples can be created. However, our samples will face a challenge that they may only *follow the manifold assumption*: to ensure the robustness of the methods, we have to make sure that they can be applied on all the possible scenarios.

## C EXPERIMENTAL DETAILS

### C.1 OVERVIEW OF DATASETS

**High-Dimensional Gaussian mixtures.** The *high dimensional Gaussian mixtures* (HDGM) benchmark is a synthetic dataset that is composed of multiple Gaussian distributions, each representing a cluster, which is proposed by Liu et al. [2020]. In our experiments, we are considering bimodal Gaussian mixtures, which means the number of clusters remains 2 irrelevant to the dimension of the multivariate Gaussian distributions. In Appendix B, we consider there are three levels of *HDGM*, which are *HDGM-Easy*, *HDGM-Medium* and *HDGM-Hard* in order to specify different levels of data distribution existing in the two-sample testing problems. In other places of this paper, rather than Appendix B, we regard *HDGM* as *HDGM-Hard*. Under $H_0$, $\mathbb{P}$ and $\mathbb{Q}$ are the same, which denoted as *HDGM-S* to verify the type I error under control; and under $H_1$, we slightly modify a mild covariance $\pm 0.5$ between first two dimensions in the covariance matrix of $\mathbb{Q}$ and other setups are the same as *HDGM-S*, which is referred to as *HDGM-D*. Thus, *HDGM-S* and *HDGM-D* are both noted by hard-level HDGM. The details of how to synthesize $\mathbb{P}$ and $\mathbb{Q}$ to derive *HDGM-Easy*, *HDGM-Medium*, *HDGM-Hard*, *HDGM-S* and *HDGM-D* are described in Appendix C.4. We regard $n_c$ as the number of samples drawn from each cluster in each distribution and $N$ as the number of total samples drawn from both $\mathbb{P}$ and $\mathbb{Q}$, where $N = n \times c \times 2$. We conduct two experiments on *HDGM-D*, increasing the $N$ from $N = 1000$ to $N = 10000$ when keeping the dimension $d$ remain the same. One experiment is a low-dimensional *HDGM-D* with $d = 2$ and another is a high-dimensional *HDGM-D* with $d = 10$. Moreover, we conduct both low-dimensional and high-dimensional *HDGM-S* to show that the type-I error is controlled. The result is shown in Figures 3 and 4, which will be analyzed in the below subsection.

***MNIST* vs *MNIST*-Fake.** The *MNIST* datasets is a collection of 70,000 grayscale images of handwritten digits, ranging from 0 to 9, divided into a training set of 60,000 images and a test of 10,000 images [LeCun et al., 1998]. The *MNIST*-Fake is the a set of 10,000 images generated by a pretrained *deep convolutional generative adversarial network* (DCGAN) [Radford et al., 2016]. The MNIST benchmark (*MNIST* vs *MNIST*-Fake) is also proposed by Liu et al. [2020], aiming to test the performance of testing methods in the image space. Under $H_0$, we draw samples both from the *MNIST*-Fake. Under $H_1$, we compare the samples from real *MNIST*, $\mathbb{P}$, and samples from *MNIST*-Fake, $\mathbb{Q}$. We regard $N$ as the number of samples each drawn from $\mathbb{P}$ and $\mathbb{Q}$, where we increase $N$ from $N = 200$ to $N = 1000$. The result of the average test power of all methods is displayed in the Table 3. All methods are tested with a reasonable type-I error rate.

**ImageNet vs ImageNet-Fake.** The *ImageNet* dataset is a comprehensive collection of over 14 million labelled high-resolution images belonging to roughly 22,000 categories [Deng et al., 2009]. The *ImageNet*-Fake dataset comprises 10,000 high-quality images generated using the advanced *StyleGAN-XL* model, a state-of-the-art generative adversarial network designed for large and diverse datasets [Sauer et al., 2022]. This benchmark (*ImageNet* vs *ImageNet*-Fake) extends the framework established by Liu et al. [2020] to a more complex and diverse image domain, testing the robustness of two-sample testing methods at a larger scale. Under the null hypothesis $H_0$, samples are drawn from *ImageNet*-Fake, while under the alternative hypothesis $H_1$, we compare samples from the real *ImageNet* dataset, $\mathbb{P}$, with those from *ImageNet*-Fake, $\mathbb{Q}$. We vary the number of samples drawn from each, $\mathbb{P}$ and $\mathbb{Q}$, from $N = 200$ to $N = 1000$ to examine the scalability of the test methods. The outcomes in terms of average test power across various methodologies are summarized in Table 3, with all tests maintaining a reasonable type-I error rate.

### C.2 IMPLEMENTATION DETAILS OF C2ST AND RL-C2ST

- C2ST: C2ST has the ability of learn DRs from two samples, by learning a well-trained classifier to only get prediction accuracy information. Implementation of C2ST paradigm is to only take *Phase 2* and *Phase 3* from Algorithm 2. Most of the implementation details are referenced from Lopez-Paz and Oquab [2017] and Liu et al. [2020]. The splitting portion of training and testing is always half to half, and the model architecture is the same for C2ST and RL-C2ST, where first few layers are feature extractor and followed by a classification layer. Moreover, in the first step of *Phase 3*, we do not utilize the the softmax probability of the first value of the logits returned by the classifier to calculate the statistic of two samples, we apply Eq. (1) which directly derive the mean of the classification prediction accuracy of two samples.
- RL-C2ST: a unified representation learning version of C2ST. We firstly learn an encoder that can extract IRs from whole samples, and boost the discriminative ability by minizing the prediction error of classifier. Replacing the test statistics of RL-C2ST from Eq. (1) into Eq. (2) will result in RL-C2ST-L. Most of the implementation details are described in the Algorithm 2.

In C2ST, we have a classifier $f$ consisting of a randomly initialized feature extractor $\phi_\theta(x)$ followed by a logistic regression layer with parameters $\boldsymbol{w}$ and $\boldsymbol{b}$, where

$$f(x) = \phi_\theta(x) \times \boldsymbol{w} + \boldsymbol{b}.$$

As the $f$ is a binary classifier, $f(x) = [z_0, z_1]$ and $\text{softmax}(f(x)) = [p_0, p_1]$, where $p_0 + p_1 = 1$. All parameters $\theta, \boldsymbol{w}$ and $\boldsymbol{b}$ are updated through the supervised learning on the training set, which aims to minimize the occurrence of incorrect predictions. Then, use the empirical probability of the correct predictions on an unseen testing set to measure the difference between two samples.

However, in RL-C2ST, we have $g$ consisting of a feature extractor $\phi_a(x)$ trained on $S_P^{\text{tr}} \cup S_P^{\text{te}} \cup S_Q^{\text{tr}} \cup S_Q^{\text{te}}$ without labels via unsupervised learning and a logistic regression layer for subsequent supervised training purpose. In the unsupervised learning step, we use $\phi_a(x)$ to extract a latent feature vector $z$ from the input $x$, and then use a decoder $\phi_a^{-1}(x)$ to reconstruct $z$ to a reconstructed $x'$. We update the parameters of $\phi_a$ by minimizing the difference between the reconstructed input $x'$ and the original input $x$. After the unsupervised training procedure, we add a classification layer after $\phi_a$ to form a classifier $g$, and train the classification layer in the same way as the C2ST.

### C.3 DETAILS OF RL-C2ST-L AND OTHER MMD BASED METHODS

We first introduce RL-C2ST-L and compare the following state-of-the-art testing methods on two benchmark datasets:

- C2ST-L: The name of C2ST-L is originated from Liu et al. [2020], where L refers to logit. It can capture more

discriminative information from the confidence of predictions. The implementation detail is the same as C2ST, except not computing the prediction probability, but using the logit output directly to measure the distance.

- RL-C2ST-L: A RL-TST implemented on C2ST-L. Rather than using the prediction labels (0 or 1) to measure the test accuracy, we utilize MMD to calculate the differences between output features extracted from the RL-C2ST. The output features could be the output of the hidden layer or the logits output of the classifier trained by the RL-C2ST, as we discuss in Section 3.1.
- MMD-D: a SOTA testing method to learn a deep kernel with a neural network that can extract DRs. MMD-D has the training objectives of directly maximizing the test power of MMD, leading to an increase in test power on the testing set. The implementation is strictly aligned with the code provided in Liu et al. [2020], where we simultaneously train a deep neural network and deep kernel Gaussian bandwidths by maximizing the training objectives $\widehat{\mathcal{J}} = \widehat{\mathrm{MMD}}_u / \hat{\sigma}_{\mathcal{H}_1,\lambda}$.
- RL-MMD-D: A RL-TST implemented on MMD-D to alleviate the drawbacks of decreasing the testing samples size from data splitting process, as we discussed in the Section 3.1.
- MMD-FUSE: MMD-FUSE fuses several MMD statistics based on the simple kernel of different combinations of hyperparameters into a new powerful statistic, then conducts a permutation test based on the fused statistic. The implementation is strictly aligned with the code provided in Biggs et al. [2023], where we compute and fuse the test statistics based on different kernel functions and hyperparameters in order to capture complex data structure in an unsupervised way.

### C.4 DETAILS OF HDGM DATASETS

Table 4 displays the details of how HDGM datasets are generated [Liu et al., 2020]. Different levels of HDGM datasets are first proposed in this paper, in order to show why SOTA SSL methods cannot be directly applied in the two-sample testing problem. The level of HDGM is differed from whether the data points are highly overlapping or whether the clusters within the same distribution are isolated. For the *HDGM-Easy*, $\Delta_\mu = 10$ and $\Delta_q = 5$. For the *HDGM-Medium*, $\Delta_\mu = 10$ and $\Delta_q = 0$. For the *HDGM-Hard*, $\Delta_\mu = 0.5$ and $\Delta_q = 0$.

Table 4: Details of how to synthesize $\mathbb{P}$ and $\mathbb{Q}$ in the experiments. Let $c = 2$ be the number of the clusters in each distribution, $d > 2$ be the dimension of multivariate normal distribution of each cluster. $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_c)$ is a set of d-dimensional mean vector $\boldsymbol{\mu}_i$ that specifies that mean of each dimension in the distribution, where $\boldsymbol{\mu}_1 = \mathbf{0}_d, \boldsymbol{\mu}_i = \boldsymbol{\mu}_{i-1} + \Delta_\mu \times \mathbf{1}_d$. $I_d$ is the $d \times d$ identity matrix, $\Delta_\mu$ is the cluster mean difference within the same distribution, and $\Delta_q$ is the mean difference between $\mathbb{P}$ and $\mathbb{Q}$. $\Delta_1 = 0.5$, $\Delta_2 = -0.5$, and $\boldsymbol{\Sigma}_i = \begin{pmatrix} 1 & \Delta_i & \mathbf{0}_{d-2} \\ \Delta_i & 1 & \mathbf{0}_{d-2} \\ \mathbf{0}_{d-2}^T & \mathbf{0}_{d-2}^T & I_{d-2} \end{pmatrix}$.

| Datasets | $\mathbb{P}$ | $\mathbb{Q}$ |
|---|---|---|
| *HDGM-S* | $\sum_{i=1}^{c} \mathcal{N}(\boldsymbol{\mu}_i, I_d)$ | $\sum_{i=1}^{c} \mathcal{N}(\boldsymbol{\mu}_i, I_d)$ |
| *HDGM-D* | $\sum_{i=1}^{c} \mathcal{N}(\boldsymbol{\mu}_i, I_d)$ | $\sum_{i=1}^{c} \mathcal{N}(\boldsymbol{\mu}_i + \Delta_q, \boldsymbol{\Sigma}_i)$ |

### C.5 EMPIRICAL RESULTS OF TYPE II ERROR CONTROL

In this subsection, we empirically show that the learned IRs will not lose the information of original distribution differences. Inspired by Biggs et al. [2023], we implement the IR learning step before the state-of-the-art baselines MMD-FUSE to clearly show that compared to distinguish between the original features, the IRs could at least preserve the original distributions differences between two samples, but also often compress key information to enhance the test power. In Table 5, we can find that the IRs will never underperform than the original features. Moreover, before the test power coverges, the compressed low-dimensional key information can boost the performance.

### C.6 DETAILS OF COMPUTING RESOURCES

The experiments of the work are conducted on three platforms. One platform is a Nvidia-4090 GPU PC with Pytorch framework. The second platform is a High-performance Computer cluster with lots of Nvidia-A100 GPU with Pytorch

Table 5: Experimental results of test power across sample sizes in the HDGM-D (d=10) in 100 trials. IR-MMD-FUSE means we use the IRs learned from Eq. (7) to input the MMD-FUSE testing, compared to the original features.

| | $N = 1000$ | $N = 2000$ | $N = 4000$ | $N = 6000$ | $N = 8000$ | Avg. |
|---|---|---|---|---|---|---|
| MMD-FUSE | 0.06 | 0.14 | 0.36 | 0.70 | 0.95 | 0.442 |
| IR-MMD-FUSE | **0.07** | **0.16** | **0.43** | **0.82** | **0.95** | **0.486** |

framework. The last platform is a Nvidia-4090 GPU Window Subsystem for Linux with Jax framework. The memory of three platforms are all over 16 GB. The storage of disk of three platforms are all over 512 GB.

### C.7 EXPERIMENT RESULT OF SEQUENTIAL TWO-SAMPLE TESTING

In this part, we will display the result of supervised sequential two-sample test that proposed by Pandeva et al. [2022] on the HDGM-Hard dataset, and compared the result with original C2ST and RL-C2ST in our problem setting. We can find that even though this method can have a small increase on the test power over the original C2ST method, but have a large decrease to our method. The number of batches we choose is five, if we choose the number of batches to two, it is exactly similar as C2ST; if we choose the number of batches to a large number like ten, the test power will drop down, since the test data size will be too small. Thus, we decide five as the number of batches, and C2ST-Sequential(5) in the Table 6 represent the supervised sequential two-sample testing with the number of batches equal to five.

Table 6: Experiment results of test power of sequential two-sample testing with Batch5 over original C2ST and our propose RL-C2ST on HDGM-hard dataset. $N$ is the total size of two samples inputed in 100 trials.

| Method | N=4000 | N=6000 | N=8000 | Avg. |
|---|---|---|---|---|
| C2ST-Sequential (5) | 0.32 | 0.57 | 0.79 | 0.56 |
| C2ST | 0.29 | 0.49 | 0.78 | 0.52 |
| RL-C2ST | 0.50 | 0.81 | 0.99 | 0.77 |

### C.8 REPRODUCIBILITY

All the reproducible code can be found in the anonymous link, and some of the two-sample testing methods are used in the package AdapTesting.

### C.9 EXTRA TYPE I ERROR CHECKING EXPERIMENTS RESULTS

In Table 7, we conduct these additional type I error checks when setting the significance level $\alpha$ at different values, which have helped strengthen our analysis of type I error control.

## D THEORETICAL DISCUSSION AND ANALYSIS

The following theoretical discussions are based on the assumption that the input two-sample testing data can follow the assumptions of the applied semi-supervised methods. Moreover, those theorems are only applied to MLP-based two-sample testing methods, such as RL-C2ST or RL-C2ST-L.

### D.1 THEORETICAL DECLARATION AND INTERPRETATION

**Test Power.** Test power is the probability that a test will correctly reject $H_0$, when $H_1$ holds. It represents the ability of the test to detect the difference between $\mathbb{P}$ and $\mathbb{Q}$, so analyzing this power is essential for evaluating the performance of one two-sample testing method.

**Definition D.1.** Let $f' \in \mathcal{C}_\phi : \mathcal{X} \to \{0, 1\}$ denotes the RL-C2ST classifier model with specific feature extractor $\phi$, where $\mathcal{C}_\phi = \{f' | f' = g \circ \phi, g \in \mathcal{G}\} \subseteq \mathcal{C}$ and $\mathcal{C} = \bigcup_{\phi \in \mathcal{F}} \mathcal{C}_\phi$.

Table 7: Type I error (mean $\pm$ standard error) for various sample sizes $N$ and levels $\alpha$ under $1,000$ repetitions on HDGM-S dataset.

| | $N=1000$ | $N=2000$ | $N=4000$ | $N=6000$ | $N=8000$ | Avg |
|---|---|---|---|---|---|---|
| $\alpha = 0.05$ | | | | | | |
| RL-C2ST | $0.0100_{\pm 0.009}$ | $0.0300_{\pm 0.015}$ | $0.0300_{\pm 0.030}$ | $0.0400_{\pm 0.031}$ | $0.0400_{\pm 0.016}$ | 0.0300 |
| RL-C2ST-L | $0.0400_{\pm 0.016}$ | $0.0400_{\pm 0.021}$ | $0.0400_{\pm 0.016}$ | $0.0800_{\pm 0.024}$ | $0.0300_{\pm 0.015}$ | 0.0460 |
| RL-MMD-D | $0.0400_{\pm 0.016}$ | $0.0600_{\pm 0.016}$ | $0.0300_{\pm 0.015}$ | $0.0300_{\pm 0.015}$ | $0.0300_{\pm 0.021}$ | 0.0380 |
| $\alpha = 0.03$ | | | | | | |
| RL-C2ST | $0.0100_{\pm 0.006}$ | $0.0200_{\pm 0.011}$ | $0.0300_{\pm 0.011}$ | $0.0300_{\pm 0.013}$ | $0.0300_{\pm 0.015}$ | 0.0240 |
| RL-C2ST-L | $0.0350_{\pm 0.010}$ | $0.0300_{\pm 0.011}$ | $0.0400_{\pm 0.012}$ | $0.0200_{\pm 0.008}$ | $0.0200_{\pm 0.008}$ | 0.0290 |
| RL-MMD-D | $0.0350_{\pm 0.013}$ | $0.0350_{\pm 0.010}$ | $0.0200_{\pm 0.011}$ | $0.0200_{\pm 0.008}$ | $0.0350_{\pm 0.010}$ | 0.0290 |
| $\alpha = 0.01$ | | | | | | |
| RL-C2ST | $0.0067_{\pm 0.002}$ | $0.0133_{\pm 0.003}$ | $0.0000_{\pm 0.000}$ | $0.0100_{\pm 0.002}$ | $0.0100_{\pm 0.003}$ | 0.0080 |
| RL-C2ST-L | $0.0100_{\pm 0.003}$ | $0.0000_{\pm 0.000}$ | $0.0100_{\pm 0.003}$ | $0.0166_{\pm 0.004}$ | $0.0000_{\pm 0.000}$ | 0.0073 |
| RL-MMD-D | $0.0120_{\pm 0.002}$ | $0.0100_{\pm 0.002}$ | $0.0000_{\pm 0.000}$ | $0.0133_{\pm 0.015}$ | $0.0133_{\pm 0.004}$ | 0.0097 |

**Theorem D.2.** *[Lopez-Paz and Oquab, 2017] Let $H_0 : t = \frac{1}{2}$ and $H_1 : t = 1 - \epsilon(\mathbb{P}, \mathbb{Q}; f')$, where t is the test accuracy and $\epsilon(\mathbb{P}, \mathbb{Q}; f') = \Pr_{(z_i,l_i)\sim\mathcal{D}}\left[f'(z_i) \neq l_i\right]/2 \in \left(0, \frac{1}{2}\right)$ represents the inability of $f'$ to distinguish between $\mathbb{P}$ and $\mathbb{Q}$. The test power of $\hat{t}$ is:*

$$\Pr_{H_1}\left(\hat{t}_{H_0} > t_\alpha\right) = \Phi\left(\frac{\left(\frac{1}{2} - \epsilon(\mathbb{P}, \mathbb{Q}; f')\right)\sqrt{n_{\text{te}}} - \Phi^{-1}(1-\alpha)/2}{\sqrt{\epsilon(\mathbb{P}, \mathbb{Q}; f') - \epsilon(\mathbb{P}, \mathbb{Q}; f')^2}}\right), \tag{11}$$

*where $\alpha \in (0, 1)$ is the significance level, $t_\alpha$ is the $(1-\alpha)$ quantile and $\Phi$ is the CDF of standard normal distribution. The Type-I error of $\hat{t}$ is also controlled no more than $\alpha$, which ensures that the test will not always reject $H_0$, when $H_0$ is true.*

**Understand RL-C2ST via Theorem D.2.** In hypothesis testing, our primary aim is to maximize test power while maintaining control over the Type-I error rate. While we know that via Theorem D.2, $\Phi^{-1}(1-\alpha)/2$ is a constant, for a reasonably fixed large $n_{\text{te}}$, the first term $(\frac{1}{2} - \epsilon(\mathbb{P}, \mathbb{Q}; f'))$ in the numerator dominates the test power. In fact, to ensure that the model can achieve the optimal test power on a fixed test dataset, it is equivalent to minimize

$$\mathcal{J}\left(\mathbb{P}, \mathbb{Q}; f'\right) := \epsilon(\mathbb{P}, \mathbb{Q}; f') \Big/ (1 - \epsilon(\mathbb{P}, \mathbb{Q}; f')), \tag{12}$$

where we estimate it with

$$\hat{\mathcal{J}}\left(S_P, S_Q; f'\right) := \frac{\hat{\epsilon}(S_P, S_Q; f')}{(1 - \hat{\epsilon}(S_P, S_Q; f'))}, \text{ and } \hat{\epsilon}(S_P, S_Q; f') \in \left(0, \frac{1}{2}\right), \tag{13}$$

where

$$\hat{\epsilon}(S_P, S_Q; f') = \frac{1}{2}\widehat{err}(f') = \frac{1}{2|\mathcal{S}|}\sum_{(x_i,l_i)\sim\mathcal{S}} \mathbb{I}[f'(x_i) \neq l_i].$$

From Eq. (13), we can find that if we can learn a classifier $f'$ from Eq. (8) that has a smaller $\hat{\epsilon}(S_P, S_Q; f')$, we can minimize the $\hat{\mathcal{J}}$, leading to maxmizing the test power. Thus, we will analyze how the use of unlabelled data and the size of unlabelled data $m_{\text{u}}$ helps to learn a classifier model $f'$ that have a smaller $\hat{\epsilon}(S_P, S_Q; f')$ in the semi-supervised learning.

We first give a definition of compatibility, an important measurement when analyzing SSL methods.

**Definition D.3** (Compatibility)**.** The compatibility of classifier model $f$ is defined as $\chi : \mathcal{C} \times \mathcal{X} \to [0, 1]$, and $\chi(f, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}[\chi(f, x)]$ estimates how "compatible" the $f$ is with $\mathcal{D}$. Thus, for a given sample $\mathcal{S}$, the incompatibility of $f$ with $\mathcal{S}$ is $1 - \chi(f, \mathcal{S})$. We can also call it unlabelled error rate $err_{\text{unl}}(f)$, where $\widehat{err}_{\text{unl}}(f) = 1 - \chi(f, \mathcal{S})$, e.g., for the consistency regularization technique, $1 - \chi(f, x) = (f(x) - f(\mathcal{A}(x))^2$, where $\mathcal{A}$ is the data augmentation function. Moreover, given value $\xi$, we define $\mathcal{C}_{\mathcal{S},\chi}(\xi) = \{f \in \mathcal{C} : \widehat{err}_{\text{unl}}(f) \leq \xi\}$.

Then, the following theorems show our main theoretical result, based on the compatibility.

**Theorem D.4.** *Balcan and Blum [2010] Let* $f^* = \arg\min_{f \in \mathcal{C}_\phi} [\epsilon(\mathbb{P}, \mathbb{Q}; f) | err_{\text{unl}}(f) \leq \xi]$*. Then, the following holds, with probability at least* $1 - \delta$*, and for any arbitrarily small* $\Delta_{m_{\text{u}}, m_{\text{l}}} > 0$*,*

$$\hat{\epsilon}(S_P, S_Q; f) \leq \epsilon(\mathbb{P}, \mathbb{Q}; f^*) + \frac{\Delta_{m_{\text{u}}, m_{\text{l}}}}{2} + \sqrt{\frac{\ln\left(\frac{4}{\delta}\right)}{8m_{\text{u}}}}, \tag{14}$$

*with the unlabelled sample size*

$$m_{\text{u}} = \mathcal{O}\left(\Delta^{-2} \log \Delta^{-1} \mathcal{V}(\mathcal{C}) + \Delta^{-2} \log(2/\delta)\right),$$

*where* $\mathcal{V}(\mathcal{C}) = \max\left[VCdim\left(\mathcal{C}\right), VCdim\left(\chi(\mathcal{C})\right)\right]$*, and the labelled sample size*

$$m_{\text{l}} = \frac{8}{\Delta^2}\left[\log\left(2\mathcal{C}_{\mathcal{S},\chi}(\xi + 2\Delta)\left[2m_{\text{l}}, \mathcal{S}\right]\right) + \log(4/\delta)\right].$$

*Here,* $\chi(\mathcal{C}) = \{\chi_f : f \in \mathcal{C}\}$ *is assumed to have a finite VC dimension,* $\chi_f(\cdot) = \chi(f, \cdot)$*, and* $\mathcal{C}_{\mathcal{S},\chi}(\xi + 2\Delta)\left[2m_{\text{l}}, \mathcal{S}\right]$ *is the expected split number for* $2m_{\text{l}}$ *points drawn from* $\mathcal{S}$ *using functions in* $\mathcal{C}_{\mathcal{S},\chi}(\xi + 2\Delta)$*.*

Theorem D.4 indicates that when the best model $f^*$ has an unlabelled error rate of at most $\xi$, the empirical inability of $f$ will be at most $\Delta$ larger than that of $f^*$, with given labelled sample size $m_l$ and unlabeled sample size $m_u$.

**Theorem D.5.** *Let* $\mathcal{C} = \{g \circ \phi | \phi \in \mathcal{F}, g \in \mathcal{G}\}$*, and suppose* $\phi' \in \mathcal{F}$ *is fixed (e.g., via pretraining). Then, the following restricted subclass*

$$\mathcal{C}_{\phi'} \;=\; \left\{g \circ \phi' \;\middle|\; g \in \mathcal{G}\right\}, \quad \chi(\mathcal{C}_{\phi'}) \;=\; \left\{\chi_{g \circ \phi'} \;\middle|\; g \in \mathcal{G}\right\}.$$

*satisfy*

- $\mathcal{C}_{\phi'} \subseteq \mathcal{C}$ *and* $\chi(\mathcal{C}_{\phi'}) \subseteq \chi(\mathcal{C})$*;*
- $\text{VCdim}\big(\mathcal{C}_{\phi'}\big) \;\leq\; \text{VCdim}\big(\mathcal{C}\big)$ *and* $\text{VCdim}\big(\chi(\mathcal{C}_{\phi'})\big) \;\leq\; \text{VCdim}\big(\chi(\mathcal{C})\big)$*;*
- $\mathcal{V}(\mathcal{C}_{\phi'}) \leq \mathcal{V}(\mathcal{C})$*.*

**Interpretations.** Combined with Theorem D.4 and Theorem D.5, we can find that compared to letting $\phi$ ba learned from scratch, if we learn a fixed $\phi'$ in the representation learning step, we now need fewer unlabeled samples to achieve the same error $\Delta$; or equivalently, given the same unlabeled sample size, we can push $\Delta$ smaller.

### D.2 PROOF OF THEOREM D.4

**Definition D.6.** Let $\epsilon(\mathbb{P}, \mathbb{Q}; f) \in \left(0, \frac{1}{2}\right)$ be the inability of $f$ to distinguish between distribution $\mathbb{P}$ and $\mathbb{Q}$. Then we define the $err_{\text{te}}(f) = 2\epsilon(\mathbb{P}, \mathbb{Q}; f) \in (0, 1)$ to be the error rate of $f$ on distribution $\mathbb{P}$ and $\mathbb{Q}$.

**Theorem D.7.** *[Boucheron et al., 2000] Suppose function space* $\mathcal{C} : \{f | f : \mathcal{X} \to \{0, 1\}\}$ *has finite VC-dimension for* $V \geq 1$*. For any sample* $\mathcal{S}$*, any function* $f$*, we have*

$$\Pr\left[\sup_{f \in \mathcal{C}} |err_{\text{te}}(f) - \widehat{err}_{\text{te}}(f)| \geq \Delta\right] \leq 8\mathcal{C}[2m_{\text{l}}, \mathcal{S}]e^{-m\Delta^2/8}.$$

*So for any* $\Delta, \delta > 0$*, if we draw from* $\mathcal{S}$ *a sample satisfying*

$$m_{\text{l}} \geq \frac{8}{\Delta}\left(\ln(\mathcal{C}[m_{\text{l}}, \mathcal{S}]) + \ln\left(\frac{8}{\delta}\right)\right),$$

*then, with probability at least* $1 - \delta$*, all functions* $f$ *satify* $|err_{\text{te}}(f) - \widehat{err}_{\text{te}}(f)| \leq \Delta$*.*

*Proof.* The given unlabelled sample size implies that with probability $1 - \delta/2$, all $f \in \mathcal{C}$ have

$$|\widehat{err}_{\text{unl}}(f) - err_{\text{unl}}(f)| \leq \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{2m_{\text{u}}}} \leq \Delta,$$

which also implies that

$$\widehat{err}_{\mathrm{unl}}(f) \leq err_{\mathrm{unl}}(f) + \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{2m_{\mathrm{u}}}} \leq \xi + \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{2m_{\mathrm{u}}}} \leq \xi + \Delta.$$

Using the standard VC bounds (e.g., Theorem D.7), the labelled sample size $m_{\mathrm{l}}$ implies that with probability at least $1-\delta/4$, all $f \in \mathcal{C}_{\mathcal{S},\chi}(\xi + 2\Delta)$ have $|err_{\mathrm{te}}(f) - \widehat{err}_{\mathrm{te}}(f)| \leq \Delta$. Then, by Hoeffding bounds, with probability at least $1-\delta/4$ we have

$$\widehat{err}_{\mathrm{te}}(f^*) \leq err_{\mathrm{te}}(f^*) + \sqrt{\log(4/\delta)/2m_{\mathrm{l}}} \leq err_{\mathrm{te}}(f^*) + \Delta.$$

Therefore, with probability at least $1-\delta$, the $f \in \mathcal{C}$ that optimizes $\widehat{err}_{\mathrm{te}}(f)$ subject to $\widehat{err}_{\mathrm{unl}}(f) \leq \xi + \Delta$ has

$$\widehat{err}_{\mathrm{te}}(f) \leq err_{\mathrm{te}}(f^*) + \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{2m_{\mathrm{u}}}} + \sqrt{\log(4/\delta)/2m_{\mathrm{l}}} \leq err_{\mathrm{te}}(f^*) + \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{2m_{\mathrm{u}}}} + \Delta.$$

Moreover, since we have $\widehat{err}_{\mathrm{te}}(f) = \Pr_{(z_i,l_i)\sim\mathcal{S}}\left[f(z_i) \neq l_i\right] \in (0,1)$ which is proportional to the empirical inability $\hat{\epsilon}(S_P, S_Q; f) \in \left(0, \frac{1}{2}\right)$. Thus, we can conclude the following inequality

$$2\hat{\epsilon}(S_P, S_Q; f) \leq err_{\mathrm{te}}(f^*) + \Delta + \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{2m_{\mathrm{u}}}},$$

since $err_{\mathrm{te}}(f^*) = 2\epsilon(\mathbb{P}, \mathbb{Q}; f^*)$,

$$\hat{\epsilon}(S_P, S_Q; f) \leq \epsilon(\mathbb{P}, \mathbb{Q}; f^*) + \frac{\Delta}{2} + \sqrt{\frac{\ln\left(\frac{4s}{\delta}\right)}{8m_{\mathrm{u}}}},$$

which concludes the proof. □