Generative Neural Networks for Sinkhorn Distributionally Robust Hypothesis Testing

Fenglin Zhang, Teyan Liu, Jie Wang*

August 20, 2026

Abstract

This paper studies the Sinkhorn distributionally robust hypothesis testing (SDRHT) problem, seeking a robust detector against least-favorable distributions in Sinkhorn discrepancy-based ambiguity sets centered at the empirical distributions. Existing approaches solve this problem by solving large-scale conic programs, which are not scalable. To overcome this, we propose a generative framework that learns least-favorable distributions and supports efficient training and end-to-end sampling. For the Sinkhorn discrepancy-based ambiguity sets, we first derive an equivalent conditional-KL-divergence representation with respect to kernel-smoothed reference distributions. This property allows us to prove strong duality for both constrained and unconstrained minimax SDRHT formulations. Based on the closed-form optimal detector and Brenier's theorem, we reformulate the max-min dual formulation as a maximization problem over convex potentials whose gradients characterize invertible transport maps between kernel-smoothed distributions and their least-favorable counterparts. We efficiently approximate these potentials using Hyper Input Convex Neural Networks (HyCNNs) equipped with stochastic gradient estimators and prove the representation power of HyCNNs and the distributional universality of their induced transport maps. Numerical results show that the proposed method achieves superior accuracy and robustness across different sample sizes and dimensions, while avoiding the scalability limitations of classical SDRHT methods.

Keywords: Sinkhorn distributionally robust optimization, robust hypothesis testing, generative model, least-favorable distribution generation, hyper input convex neural network

1 Introduction

Hypothesis testing is a fundamental problem in statistics and scientific discovery, aiming to find an optimal detector that, given new data, can discriminate between two competing hypotheses while minimizing decision errors. The Neyman–Pearson lemma [1] yields the optimal likelihood-ratio test when the densities of the underlying distributions are known. In practice, these distributions are typically unknown; instead, the decision-maker has only access to their independent

*Fenglin Zhang is affiliated with School of Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen; Teyan Liu is affiliated with Department of Decision Analytics and Operations, College of Business, City University of Hong Kong; Jie Wang is affiliated with School of Artificial Intelligence and School of Data Science, The Chinese University of Hong Kong, Shenzhen. Corresponding author: Jie Wang, jwang@cuhk.edu.cn.

and identically distributed (i.i.d.) samples. This setting motivates data-driven hypothesis tests that are both statistically reliable and computationally efficient.

However, data-driven tests trained on finite samples are vulnerable to distributional misspecification [2]. On the one hand, for some applications like healthcare [3] and anomaly detection [4], where samples are limited and noisy, reliable estimation of the underlying distribution is challenging due to inevitable measurement and statistical errors. On the other hand, real data may be non-stationary or may undergo distributional shifts, leading to differences between training and testing data. To improve the reliability of hypothesis testing, especially in high-stakes scenarios, robust methods are necessary to circumvent these challenges [5, 6].

Distributionally robust optimization (DRO) [7] offers a powerful paradigm for making robust decisions under distributional uncertainty. When applied to hypothesis testing, DRO optimizes the robust detector against the least-favorable distributions (also known as worst-case distributions) within an ambiguity set consisting of plausible distributions, safeguarding against misspecification and enhancing out-of-sample reliability when applied to new, unseen data [8]. The design of the ambiguity set is crucial for balancing computational tractability and robust out-of-sample performance [9]. Popular choices include Huber's contamination model [10], f-divergence [11, 12], moments [13, 14], kernels [15, 16], or Wasserstein distance [8, 17].

DRO with ambiguity sets constructed using the Sinkhorn discrepancy has recently received increasing attention [18, 19]. Conventional Wasserstein DRO-based hypothesis testing restricts the worst-case distributions to have the same support as that of the empirical distributions [8]. By adding entropic regularization to the Wasserstein distance, the Sinkhorn discrepancy provides a smooth alternative to the Wasserstein distance [20], and the Sinkhorn discrepancy-based ambiguity sets naturally encourage continuous distributional estimates. Thus, unlike the Wasserstein distance-based ambiguity set, if a testing sample is outside the support of the observed data, it is still feasible to leverage the likelihood ratio detector to design a robust test. However, finding the least-favorable distributions in Sinkhorn-based hypothesis testing is an infinite-dimensional optimization problem. Existing approaches approximately solve it using sample average approximation and result in large-scale conic programs [21, 22], whose computational cost grows rapidly with the sample size. It is desirable to develop more tractable approaches for solving the robust hypothesis testing problem with Sinkhorn-based ambiguity sets.

Parallel to these developments in robust statistics, flow-based generative models have shown remarkable success in learning complex distributions as pushforwards of simple reference distributions through invertible and differentiable maps [23, 24, 25]. This framework is not only adept at generating new samples but is also naturally suited for solving optimization problems over probability space, e.g., generating least-favorable distributions in DRO [26, 27, 28]. By parameterizing a distribution as a flow-based transformation starting from a probability density, one can efficiently optimize an objective over the probability space using gradient-based methods [29, 30].

In this work, we leverage flow-based generative models to solve the Sinkhorn distributionally robust hypothesis testing (SDRHT). Based on the theoretical results of optimal transport, we introduce a generative framework to learn continuous least-favorable distributions within the Sinkhorn-based ambiguity sets by optimizing convex potentials. By parameterizing the convex

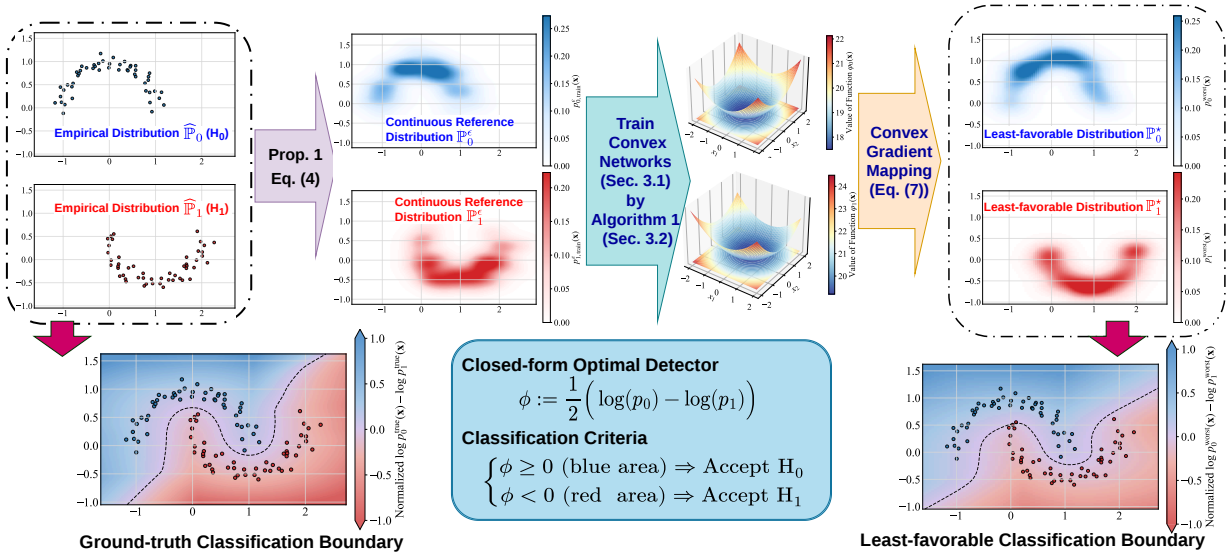


Figure 1: Illustration of the training and testing processes of our least-favorable detector.

potentials, our approach converts the original infinite-dimensional optimization into a finite-dimensional and tractable problem. Our main contributions are summarized as follows:

- **Novel Reformulation:** We establish a strong duality result for SDRHT and thereby reformulate the original minimax optimization as a max-min problem. As the inner minimized detector can be explicitly derived, this problem is equivalent to a maximum problem over distributions. Leveraging Brenier's theorem, we further reformulate it as a maximization problem over convex potentials.

- **Generative Framework:** We parameterize the convex potentials by Hyper Input Convex Neural Networks (HyCNNs) and develop stochastic gradient estimators for the objective. This transforms the infinite-dimensional optimization into a finite-dimensional one, enabling efficient and scalable training and sampling. We also prove the representation power of HyCNNs and the distributional universality of their induced gradient maps.

- **Empirical Validation:** We validate the performance of our method through extensive numerical studies on both synthetic datasets and real-world data with different sample sizes and dimensions, demonstrating its superiority over existing baselines in terms of both accuracy and robustness.

Figure 1 illustrates the training and testing processes of the proposed method using a toy Moons dataset. By exploiting the geometry of the Sinkhorn-based ambiguity set (Proposition 1 and Eq. (4)), the least-favorable distributions $(\mathbb{P}_0^\star, \mathbb{P}_1^\star)$ are generated by first applying kernel smoothing to the empirical distributions $\widehat{\mathbb{P}}_0$ and $\widehat{\mathbb{P}}_1$ to obtain $\mathbb{P}_0^\epsilon$ and $\mathbb{P}_1^\epsilon$, and next robustifying them. In the robustification procedure, we learn convex-gradient maps (see Eq. (7) and Algorithm 1) that generate $\mathbb{P}_0^\star$ and $\mathbb{P}_1^\star$ based on the kernel-smoothed reference distributions $\mathbb{P}_0^\epsilon, \mathbb{P}_1^\epsilon$. The resulting least-favorable distributions induce a nonlinear robust decision boundary with high accuracy.

The remainder of this paper is organized as follows. Section 2 presents the models and properties of SDRHT. Section 3 develops a framework for generating least-favorable distributions. Section 4 shows the representation power of HyCNNs. Section 5 reports the numerical results. Section 6 concludes this paper. Proofs and additional technical notes are provided in the online appendix to this paper.

Notations. For $K \in \mathbb{N}$, let $[K] := \{1, \cdots, K\}$. We write $\mathbb{P} \ll \nu$ if the probability measure $\mathbb{P}$ is absolutely continuous with respect to the measure ν. We denote $\mathbb{P} \otimes \mathbb{Q}$ by the product measure of two probability measures $\mathbb{P}$ and $\mathbb{Q}$. For a transport map $T : \Omega \to \Omega$, the pushforward of a probability measure $\mathbb{P}$ is denoted as $T_{\#}\mathbb{P}$, such that $T_{\#}\mathbb{P}(A) = \mathbb{P}(T^{-1}(A))$ for any measurable set $A \subseteq \Omega$. For a convex function $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$, denote its convex conjugate by $\varphi^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \mathbb{R}^d} \{\boldsymbol{y}^\top \boldsymbol{x} - \varphi(\boldsymbol{x})\}$ for any $\boldsymbol{y} \in \mathbb{R}^d$.

2 Sinkhorn Distributionally Robust Hypothesis Testing

In this section, we introduce the Sinkhorn distributionally robust hypothesis testing (SDRHT), derive its strong duality property, and obtain its equivalent functional optimization reformulation.

Let $\Omega \subseteq \mathbb{R}^d$ be a closed sample space. Denote $\mathscr{P}(\Omega)$ as the set of all probability measures on Ω, and $\mathcal{M}(\Omega)$ as the set of measures on Ω. Denote by $\mathcal{P}_k \subset \mathscr{P}(\Omega)$ the ambiguity set under hypotheses H_k, where $k \in \mathbb{K} = \{0, 1\}$. Given a testing sample $\boldsymbol{\omega}$, the composite hypothesis testing decides which one of the following hypotheses is true:

$$H_0 : \boldsymbol{\omega} \sim \mathbb{P}_0, \mathbb{P}_0 \in \mathcal{P}_0; \qquad H_1 : \boldsymbol{\omega} \sim \mathbb{P}_1, \mathbb{P}_1 \in \mathcal{P}_1.$$

A hypothesis test (detector) $\phi : \Omega \to \mathbb{R}$ is a measurable function. For any testing sample $\boldsymbol{\omega}$, it accepts the null hypothesis H_0 and rejects alternative hypothesis H_1 whenever $\phi(\boldsymbol{\omega}) \geq 0$, otherwise accepts H_1 and rejects H_0. The exact risk of this detector is defined as the sum of type-I and type-II errors. For this non-convex objective, we follow the generating-function approach [8, 31] and optimize the following convex surrogate risk instead:

$$\Phi(\phi; \mathbb{P}_0, \mathbb{P}_1) := \mathbb{E}_{\boldsymbol{\omega} \sim \mathbb{P}_0}[\ell \circ (-\phi)(\boldsymbol{\omega})] + \mathbb{E}_{\boldsymbol{\omega} \sim \mathbb{P}_1}[\ell \circ \phi(\boldsymbol{\omega})],$$

where the generating function ℓ is defined as follows.

Definition 1 (Generating Function)**.** *A generating function $\ell : \mathbb{R} \to \mathbb{R}_+ \cup \{\infty\}$ is a non-negative, non-decreasing, convex function such that $\ell(0) = 1$ and $\lim_{t \to -\infty} \ell(t) = 0$.* ◇

For some common choices of the generating function, Table 1 summarizes the corresponding closed-form optimal detectors ϕ_{opt} given distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ [8, 21], where p_0 and p_1 represent the probability density functions of $\mathbb{P}_0$ and $\mathbb{P}_1$. In this paper, we choose the generating function $\ell(t) := \exp(t)$.

Given sets of training samples $\{\boldsymbol{x}_k^{(1)}, \cdots, \boldsymbol{x}_k^{(n_k)}\}$ for all $k \in \mathbb{K}$, denote the corresponding empirical distributions as $\widehat{\mathbb{P}}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \delta_{x_k^{(i)}}$, where n_k denotes the number of samples from $\mathbb{P}_k \in \mathcal{P}_k$. The following definition introduces the Sinkhorn discrepancy, an entropy-regularized variant of the Wasserstein distance that quantifies the distance between distributions [18, 21]. Throughout the paper, we use the quadratic transport cost $c(x, z) := \frac{1}{2}\|x - z\|_2^2$.

Table 1: Common choices of generating function (first column), together with their corresponding optimal detector (second column) and surrogate risk (third column).

$\ell(t)$	**Optimal** ϕ	**Corresponding optimal** $1-\Phi(\mathbb{P}_0,\mathbb{P}_1)/2$
$\exp(t)$	$\log\sqrt{p_0/p_1}$	$H^2(\mathbb{P}_0,\mathbb{P}_1)$
$\log(1+\exp(t))/\log 2$	$\log(p_0/p_1)$	$\mathrm{JS}(\mathbb{P}_0,\mathbb{P}_1)/\log 2$
$(t+1)^2$	$1-2p_0/(p_0+p_1)$	$\chi^2(\mathbb{P}_0,\mathbb{P}_1)$
$(t+1)_+$	$\mathrm{sign}(p_0-p_1)$	$\mathrm{TV}(\mathbb{P}_0,\mathbb{P}_1)$

H^2: Hellinger divergence; JS: Jensen-Shannon divergence; TV: Total variation distance.

Definition 2 (Sinkhorn Discrepancy)**.** *Consider any two distributions $\mathbb{P},\mathbb{Q}\in\mathscr{P}(\Omega)$ and reference measure $\nu\in\mathcal{M}(\Omega)$ such that $\mathbb{Q}\ll\nu$. For regularization parameter $\epsilon>0$, the Sinkhorn discrepancy between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is defined as*

$$\mathcal{S}_\epsilon(\mathbb{P},\mathbb{Q})=\inf_{\gamma\in\Gamma(\mathbb{P},\mathbb{Q})}\left\{\mathbb{E}_{(\boldsymbol{x},\boldsymbol{z})\sim\gamma}[c(\boldsymbol{x},\boldsymbol{z})]+\epsilon\cdot\mathcal{D}_{\mathrm{KL}}(\gamma\|\mathbb{P}\otimes\nu)\right\},$$

where $c(\boldsymbol{x},\boldsymbol{z})$ represents the transport cost function, $\Gamma(\mathbb{P},\mathbb{Q})$ denotes the set of couplings of $\mathbb{P}$ and $\mathbb{Q}$, respectively, and $\mathcal{D}_{\mathrm{KL}}(\gamma\|\mathbb{P}\otimes\nu)$ denotes the relative entropy of distribution γ with respect to measure $\mathbb{P}\otimes\nu$, i.e.,

$$\mathcal{D}_{\mathrm{KL}}(\gamma\|\mathbb{P}\otimes\nu)=\int_{\Omega\times\Omega}\log\frac{\mathrm{d}\gamma(\boldsymbol{x},\boldsymbol{z})}{\mathrm{d}\mathbb{P}(\boldsymbol{x})\mathrm{d}\nu(\boldsymbol{z})}\mathrm{d}\gamma(\boldsymbol{x},\boldsymbol{z}).\qquad\diamond$$

Let the space of detectors be $\mathcal{F}$. We consider the following SDRHT problem, optimizing a detector to minimize the least-favorable risk in ambiguity sets:

$$\inf_{\phi\in\mathcal{F}}\sup_{\mathbb{P}_0\in\mathbb{B}_{\rho_0}(\widehat{\mathbb{P}}_0),\mathbb{P}_1\in\mathbb{B}_{\rho_1}(\widehat{\mathbb{P}}_1)}\Phi(\phi;\mathbb{P}_0,\mathbb{P}_1),\tag{SDRHT}$$

where the ambiguity sets are constructed by the Sinkhorn discrepancy $\mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k,\mathbb{P})$, offering a flexible, non-parametric framework to characterize the least-favorable distributions

$$\mathbb{B}_{\rho_k}(\widehat{\mathbb{P}}_k):=\{\mathbb{P}:\mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k,\mathbb{P})\le\rho_k\},\quad\forall k\in\mathbb{K}.\tag{1}$$

We also consider the following soft-constrained counterpart of problem (SDRHT):

$$\inf_{\phi\in\mathcal{F}}\sup_{\mathbb{P}_0,\mathbb{P}_1\in\mathscr{P}(\Omega)}\Phi(\phi;\mathbb{P}_0,\mathbb{P}_1)-\sum_{k\in\mathbb{K}}\lambda_k\mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k,\mathbb{P}_k),\tag{soft-SDRHT}$$

where $\lambda_k>0$ for all $k\in\mathbb{K}$ are penalty parameters.

2.1 Strong Duality

In this subsection, since the optimal detector admits a closed form for fixed distributions, we prove the strong duality of problems (SDRHT) and (soft-SDRHT) by interchanging the minimization over detectors and the maximization over distributions. Note that the functional space $\mathcal{F}$ is infinite-dimensional and naturally convex, but is generally not compact. Hence, the classical

Sion's minimax theorem cannot be used to verify strong duality. Instead, we prove strong duality primarily using the lopsided minimax theorem [7], which does not require $\mathcal{F}$ to be compact.

For the transport cost and Sinkhorn discrepancy, we introduce the following assumptions.

Assumption 1. (I) *The transport cost function $c(\boldsymbol{x},\boldsymbol{z})$ is $\widehat{\mathbb{P}}\otimes\nu$-measurable and satisfies $\nu\{z : 0 \le c(\boldsymbol{x},\boldsymbol{z}) < \infty\} = 1$ for $\widehat{\mathbb{P}}$-almost every x.*

(II) $\mathbb{E}_{\boldsymbol{z}\sim\nu}[e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon}] < \infty$ *for $\widehat{\mathbb{P}}$-almost every x.*

(III) *The cost function $c(\boldsymbol{x},\boldsymbol{z})$ is lower semicontinuous in $(\boldsymbol{x},\boldsymbol{z})$.*

(IV) *For every joint distribution γ on $\Omega\times\Omega$ with first marginal distribution $\mathbb{P}$, it has a regular conditional distribution $\gamma_{\boldsymbol{x}}$ given the value of the first marginal equals $\boldsymbol{x}$.*

According to [18], Assumptions 1(I) and (II) ensure that both the Sinkhorn discrepancy and the equivalent reformulation in the following Proposition 1 are well-defined. Assumption 1(III) ensures that the Sinkhorn discrepancy is weakly lower semicontinuous. Assumption 1(IV) ensures that the joint distribution γ can be written as $\mathrm{d}\gamma(\boldsymbol{x},\boldsymbol{z}) = \mathrm{d}\widehat{\mathbb{P}}(\boldsymbol{x})\mathrm{d}\gamma_{\boldsymbol{x}}$ and the law of total expectation holds. Our choice of transport cost $c(x,z) = \frac{1}{2}\|x-z\|_2^2$ naturally satisfies Assumption 1(III).

Define the non-negative loss function

$$L_k(\phi,\boldsymbol{\omega}) := \ell\Big((-1)^{k+1}\cdot\phi(\boldsymbol{\omega})\Big), \quad \forall k\in\mathbb{K}. \tag{2}$$

In this paper, we focus on continuous detectors and consider the following assumptions for the loss function.

Assumption 2 (Loss Function)**.** (I) *For any $k\in\mathbb{K}$, $\phi\in\mathcal{F}$, and $\beta_k > 0$, the reference distribution $\mathbb{P}_k^\epsilon$ in Lemma 1 satisfies $\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}[\exp(\beta_k\cdot L_k(\phi,\boldsymbol{\omega}))] < \infty$.*

(II) *There exists $\mathbb{Q}\in\mathscr{P}(\Omega)$ with $\inf_{\phi\in\mathcal{F}}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\Big[L_k(\phi,\boldsymbol{\omega}) - \lambda_k\mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k,\mathbb{Q})\Big] > -\infty$ for any $k\in\mathbb{K}$.*

Assumption 2(I) leads to the following Lemma 2. Assumption 2(II) ensures that the optimal value of the dual of problem (soft-SDRHT) is strictly larger than $-\infty$.

The following proposition provides an equivalent conditional-KL-divergence representation of the Sinkhorn-based ambiguity set.

Proposition 1 (Conditional-KL Representation of the Sinkhorn Ambiguity Set)**.** *The ambiguity sets based on Sinkhorn discrepancy with $\epsilon > 0$ are equivalent to*

$$\mathbb{B}_{\bar\rho_k}(\mathbb{P}_k^\epsilon) := \left\{\mathbb{Q} : \inf_{\gamma\in\Gamma(\widehat{\mathbb{P}}_k,\mathbb{Q})}\mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}\Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}}\|\mathcal{K}_{\boldsymbol{x},\epsilon})\Big] \le \bar\rho_k/\epsilon\right\}, \quad \forall k\in\mathbb{K}, \tag{3}$$

where $\bar\rho_k := \rho_k + \epsilon\mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}\Big[\log\int_{\mathbb{R}^d} e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon}\mathrm{d}\boldsymbol{z}\Big]$, probability density $\mathrm{d}\mathbb{P}_k^\epsilon(\boldsymbol{z}) := \mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}[\mathrm{d}\mathcal{K}_{\boldsymbol{x},\epsilon}(\boldsymbol{z})]$,

$$\mathrm{d}\mathcal{K}_{\boldsymbol{x},\epsilon}(\boldsymbol{z}) := \frac{e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon}}{\int_{\mathbb{R}^d} e^{-c(\boldsymbol{x},\boldsymbol{u})/\epsilon}\cdot\mathrm{d}\boldsymbol{u}}\cdot\mathrm{d}\nu(\boldsymbol{z}).$$

The proof of Proposition 1 is provided in Appendix B.1. Proposition 1 shows the connection between the ambiguity set based on Sinkhorn discrepancy and those based on the KL divergence [12]. Since the reference measure $\nu \in \mathcal{M}(\Omega)$, $\mathcal{K}_{\boldsymbol{x},\epsilon}$ is a probability kernel, and its corresponding continuous reference distributions in Eq. (3) are probability measures, i.e.,

$$\mathbb{P}_k^\epsilon = \frac{1}{n_k}\sum_{i=1}^{n_k} \mathcal{K}_{\boldsymbol{x}_k^{(i)},\epsilon} \in \mathscr{P}(\Omega), \qquad \forall k \in \mathbb{K}. \tag{4}$$

When $\Omega = \mathbb{R}^d$, ν is a Lebesgue measure, and the transport cost is defined as $\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|_2^2$, distribution $\mathbb{P}_k^\epsilon$ is a Gaussian mixture because $\mathcal{K}_{\boldsymbol{x}_k^{(i)},\epsilon} = \mathcal{N}(\boldsymbol{x}_k^{(i)}, \epsilon \cdot \boldsymbol{I}_d)$ for each $k \in \mathbb{K}$ and $i \in [n_k]$.

Based on Assumptions 1 and 2 and Proposition 1, the following strong duality results hold.

Theorem 1 (Strong Duality). *For the convex risk function Φ and Sinkhorn discrepancy $\mathcal{S}_\epsilon$, the following results hold:*

(I) *If Assumptions 1 and 2(I) hold, then*

$$\text{Optimal Value of (SDRHT)} = \max_{\mathbb{P}_0 \in \mathbb{B}_{\rho_0}(\widehat{\mathbb{P}}_0), \mathbb{P}_1 \in \mathbb{B}_{\rho_1}(\widehat{\mathbb{P}}_1)} \inf_{\phi \in \mathcal{F}} \quad \Phi(\phi; \mathbb{P}_0, \mathbb{P}_1);$$

(II) *If Assumptions 1 and 2 hold, then*

$$\text{Optimal Value of (soft-SDRHT)} = \max_{\mathbb{P}_0, \mathbb{P}_1 \in \mathscr{P}(\Omega)} \inf_{\phi \in \mathcal{F}} \quad \Phi(\phi; \mathbb{P}_0, \mathbb{P}_1) - \sum_{k \in \mathbb{K}} \lambda_k \mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k, \mathbb{P}_k),$$

where $\lambda_k > 0$ for any $k \in \mathbb{K}$.

The proof of Theorem 1 is based on the lopsided minimax theorem [7, Theorem 5.15], and is provided in Appendix B.2. We prove this by showing that the Sinkhorn-based ambiguity sets are weakly compact and the expected loss is weakly upper semicontinuous.

Theorem 1 allows us to eliminate the detector minimization before optimizing over the least-favorable distributions. Thus, for generating function $\ell(t) = \exp(t)$, problem (soft-SDRHT) is equivalent to

$$\max_{\mathbb{P}_0, \mathbb{P}_1 \in \mathscr{P}(\Omega)} \quad \Phi(\phi_{\text{opt}}; \mathbb{P}_0, \mathbb{P}_1) - \sum_{k \in \mathbb{K}} \lambda_k \mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k, \mathbb{P}_k). \tag{5}$$

where

$$\Phi(\phi_{\text{opt}}; \mathbb{P}_0, \mathbb{P}_1) = \sum_{k \in \mathbb{K}} \mathbb{E}_{\boldsymbol{\omega} \sim \mathbb{P}_k}\left[\exp\left[(-1)^{k+1} \cdot \frac{1}{2}\Big(\log p_0(\boldsymbol{\omega}) - \log p_1(\boldsymbol{\omega})\Big)\right]\right]. \tag{6}$$

For a testing sample $\boldsymbol{\omega} \in \Omega$, it suffices to evaluate the log-density of least-favorable distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ at $\boldsymbol{\omega}$, and then make a decision by our explicitly optimal detector. Thus, the robust detector retains the likelihood-ratio form, but its densities are those of the least-favorable distributions. Problem (5) nevertheless remains infinite-dimensional because the least-favorable distributions are still optimization variables. In the next subsection, we characterize these distributions using parameterized convex-gradient transport maps.

2.2 Reformulation via Convex-Gradient Transport Maps

By setting the transport cost to the quadratic function $c(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{2}\|x - z\|_2^2$, we can leverage the following fundamental result from optimal transport [32] to simplify Problem (5).

Theorem 2 (Brenier's Theorem). *Let μ, ν be two finite-second-moment probability measures on $\mathbb{R}^d$, and assume that μ is absolutely continuous with respect to the Lebesgue measure. Then there exists a proper convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$ such that $(\nabla\varphi)_{\#}\mu = \nu$. In addition, $\nabla\varphi$ is the unique solution to*

$$\inf_{T:\, T_{\#}\mu=\nu} \mathbb{E}_{\mu}\|\boldsymbol{x} - T(\boldsymbol{x})\|_2^2,$$

and $\gamma_{opt} \sim (\mathrm{Id}, \nabla\varphi)_{\#}\mu$ is the unique solution to

$$\inf_{\gamma\in\Gamma(\mu,\nu)} \mathbb{E}_{\gamma}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2.$$

Theorem 2 shows that the gradient of a convex potential can characterize the transport map between continuous distributions. Proposition 1 and Theorem 2 enable us to reformulate (5) as an equivalent optimization problem over convex gradient transport maps, i.e.,

$$\max_{\varphi_0,\varphi_1\in\mathcal{F}_{\mathrm{CVX}}} \left\{ \Phi(\phi_{\mathrm{opt}}; \mathbb{P}_0, \mathbb{P}_1) - \sum_{k\in\mathbb{K}} \lambda_k \mathcal{S}_{\epsilon}(\widehat{\mathbb{P}}_k, \mathbb{P}_k) :\ \mathbb{P}_k = (\nabla\varphi_k)_{\#}\mathbb{P}_k^{\epsilon} \right\}, \tag{7}$$

where $\mathcal{F}_{\mathrm{CVX}}$ is the set of convex functions $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and the distribution $\mathbb{P}_k^{\epsilon}$ for each $k \in \mathbb{K}$ is defined in Eq. (4). Note that we use continuous distributions $\mathbb{P}_k^{\epsilon}$ as source distributions of the transport maps for all $k \in \mathbb{K}$, thanks to the connection between Sinkhorn discrepancy and KL divergence shown in Proposition 1.

For computation, we restrict the convex function $\varphi_k \in C^2(\mathbb{R}^d)$ to be strongly convex, and $\nabla\varphi_k$ to be a global diffeomorphism for each $k \in \mathbb{K}$. In formulation (7), given strongly convex functions φ_0 and φ_1, the log-density terms in risk function $\Phi(\phi_{\mathrm{opt}}; \mathbb{P}_0, \mathbb{P}_1)$ can be explicitly computed by the log-density form of the change-of-variable formula [23]. Specifically, for any sample $\boldsymbol{\omega}_k \sim \mathbb{P}_k$, we have

$$\log p_k(\boldsymbol{\omega}_k) = \log p_k^{\epsilon}(\boldsymbol{z}_k) - \log \left| \det \nabla^2\varphi_k(\boldsymbol{z}_k) \right|, \quad \forall k \in \mathbb{K}, \tag{8}$$

where $\boldsymbol{z}_k = \nabla\varphi_k^{-1}(\boldsymbol{\omega}_k)$, and p_k^{ϵ} represents the probability density function of $\mathbb{P}_k^{\epsilon}$ for each $k \in \mathbb{K}$. Based on the Fenchel–Young inequality, the gradient of a differentiable strongly convex function φ is the inverse of the gradient of its convex conjugate, i.e., $\nabla\varphi^{-1} = \nabla\varphi^*$. In this case, we have $\boldsymbol{z}_k = \arg\max_{\boldsymbol{z}\in\mathbb{R}^d} \{\boldsymbol{\omega}_k^{\top}\boldsymbol{z} - \varphi_k(\boldsymbol{z})\}$. Similarly, to evaluate the log-density of sample $\boldsymbol{\omega}_k \sim \mathbb{P}_k$ at the other distribution $\mathbb{P}_{|1-k|}^{\epsilon}$, i.e., $\log p_{|1-k|}(\boldsymbol{\omega}_k)$, we trace $\boldsymbol{\omega}_k$ back to its corresponding source sample $\bar{\boldsymbol{z}}_k$ in $\mathbb{P}_{|1-k|}^{\epsilon}$:

$$\bar{\boldsymbol{z}}_k = \nabla\varphi_{|1-k|}^{-1}(\boldsymbol{\omega}_k) = \arg\max_{\boldsymbol{z}\in\mathbb{R}^d} \left\{ \boldsymbol{\omega}_k^{\top}\boldsymbol{z} - \varphi_{|1-k|}(\boldsymbol{z}) \right\}, \quad \forall k \in \mathbb{K}, \tag{9}$$

and then we have

$$\log p_{|1-k|}(\boldsymbol{\omega}_k) = \log p_{|1-k|}^{\epsilon}(\bar{\boldsymbol{z}}_k) - \log \left| \det \nabla^2\varphi_{|1-k|}(\bar{\boldsymbol{z}}_k) \right|, \quad \forall k \in \mathbb{K}. \tag{10}$$

Since the potential φ_k in (7) is assumed to be twice differentiable and strongly convex, its Hessian determinant is strictly positive, i.e., $\left|\det \nabla^2 \varphi_k(\boldsymbol{z})\right| = \det \nabla^2 \varphi_k(\boldsymbol{z})$, and the log-determinant operators in Eqs. (8) and (10) are well defined. In high-dimensional applications, the exact values of the log-determinant-Hessian operators are computationally expensive, but can be estimated efficiently by the stochastic Lanczos quadrature (SLQ) in [33, 34].

Although formulation (7) exposes the transport structure of the least-favorable distributions, it remains an infinite-dimensional optimization. Moreover, the induced densities, inverse maps, and log-determinant terms are generally unavailable in closed form. However, this formulation offers a new perspective on generating least-favorable distributions by characterizing the gradients of convex potentials. In the next section, we propose a generative method to solve (7).

3 A HyCNN-Based Generative Method

Flow-based generative models represent a target distribution as the pushforward of a simple reference distribution through an invertible, differentiable map, allowing efficient density evaluation and sample generation [23]. In flow-based DRO, these models can also generate least-favorable distributions when guided by proper loss functions [26]. For problem (7), a natural approach is to parameterize the convex potentials by the Input Convex Neural Network (ICNN) [34, 35]. In this section, we develop a generative framework that parametrizes convex potentials by a variant of ICNN, i.e., *Hyper Input Convex Neural Network (HyCNN)*, extending the classical ICNN with more flexible architectures [36].

3.1 The Architecture of HyCNN

In this subsection, we first introduce the HyCNN architecture. Let the number of layers of HyCNN be L, and the output of HyCNN given an input $\boldsymbol{x} \in \mathbb{R}^d$ be $\mathsf{HyCNN}(\boldsymbol{x}) \in \mathbb{R}$. Figure 2 illustrates the architecture of HyCNN, including the detailed structure of each block. The forward propagation process of HyCNN can be reformulated as follows

$$\mathsf{HyCNN}(\boldsymbol{x}) := y_{L+1} = \boldsymbol{V}_L \boldsymbol{y}_L + \boldsymbol{W}_L \boldsymbol{x} + b_L, \tag{11a}$$

$$\boldsymbol{y}_{l+1} := \bar{\sigma}\Big(\boldsymbol{V}_l^{(1)} \boldsymbol{y}_l + \boldsymbol{W}_l^{(1)} \boldsymbol{x} + \boldsymbol{b}_l^{(1)}, \boldsymbol{V}_l^{(2)} \boldsymbol{y}_l + \boldsymbol{W}_l^{(2)} \boldsymbol{x} + \boldsymbol{b}_l^{(2)}\Big), \quad \forall l \in \{0, \cdots, L-1\}. \tag{11b}$$

where $\{\boldsymbol{V}, \boldsymbol{W}, \boldsymbol{b}\}$ are trainable parameters, $\boldsymbol{y}_0 := \boldsymbol{0}$, and $\bar{\sigma} : \mathbb{R}^2 \to \mathbb{R}$ is a scalar activation function that is convex and component-wise non-decreasing, applied element-wise to vector inputs.

As shown in Figure 2, in each hidden layer $l \geq 1$ of HyCNN, a "passthrough" layer (red arrow) directly connects the input vector with the hidden units. Since the nonnegative sums of convex functions are also convex and the composition of a convex and convex non-decreasing function is also convex, according to Proposition 2.1 in [36], this construction ensures the convexity of HyCNNs.

Proposition 2 (Convexity of HyCNN [36]). *Suppose that $\bar{\sigma} : \mathbb{R}^2 \to \mathbb{R}$ is convex and component-wise non-decreasing, and weight matrices $\{\boldsymbol{V}_l^{(1)}, \boldsymbol{V}_l^{(2)}\}_{l=0}^{L-1}$ and $\boldsymbol{V}_L$ are component-wise nonnegative. Then, $\mathsf{HyCNN}(\boldsymbol{x})$ in Eq. (11a) is convex in $\boldsymbol{x}$.*

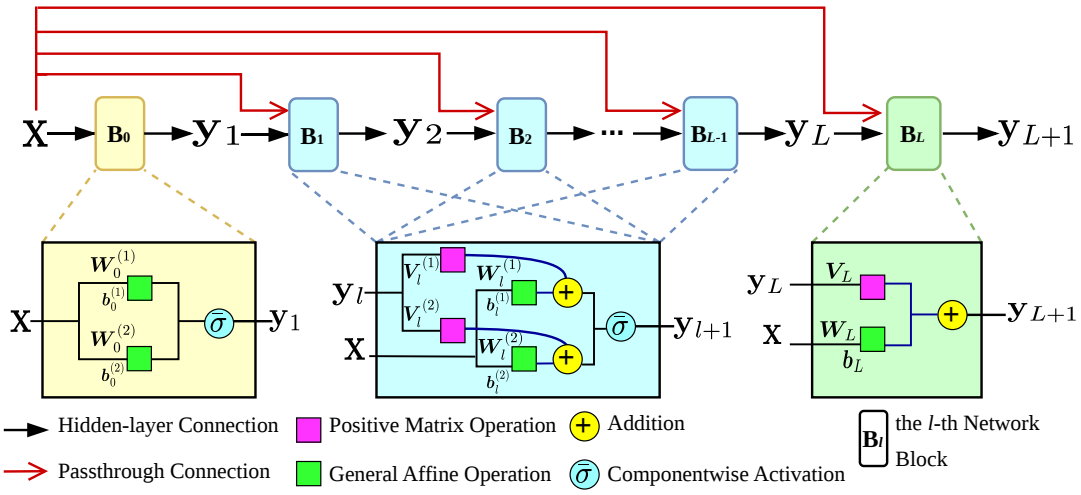


Figure 2: The architecture of HyCNN

In this paper, the non-negativity of weight matrices $\{\boldsymbol{V}_l^{(1)}, \boldsymbol{V}_l^{(2)}\}_{l=0}^{L-1}$ and $\boldsymbol{V}_L$ is enforced by softplus operators. Note that when we choose $\bar{\sigma}(a_1, a_2) = \sigma(a_1)$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function that is convex and non-decreasing (e.g., ReLU), the HyCNN reduces to the standard ICNN architecture [35]. Thus, HyCNNs can be interpreted as a generalization of ICNNs. When we choose a maxout unit [37] as our activation function, i.e., $\bar{\sigma}(a_1, a_2) = \max(a_1, a_2)$, the resulting HyCNN is piecewise-affine but is nondifferentiable at ties. Therefore, to ensure the trained HyCNN is differentiable, we use a smooth log-sum-exp approximation of the maxout activation function with parameter $\tau > 0$:

$$\bar{\sigma}_\tau(a_1, a_2) := \tau \cdot \log\left(e^{a_1/\tau} + e^{a_2/\tau}\right), \tag{12}$$

which is referred to as the *smooth* maxout activation function below.

3.2 Approximation for Convex Potentials

In this subsection, we present the procedure for training HyCNNs. Given sample $\boldsymbol{z} \in \Omega$, denote the two HyCNNs we use for the problem (7) by $\mathsf{HyCNN}_0(\boldsymbol{z})$ and $\mathsf{HyCNN}_1(\boldsymbol{z})$, respectively. Denote the approximated convex function by $\widetilde{\varphi}_k$ for each $k \in \mathbb{K}$. To ensure the gradient of convex potentials is invertible, as the smooth maxout activation makes HyCNN_k twice continuously differentiable, we further add a positive quadratic term to ensure strong convexity, i.e., $\widetilde{\varphi}_k(\boldsymbol{z}) = \mathsf{HyCNN}_k(\boldsymbol{z}) + \frac{\alpha}{2}\|\boldsymbol{z}\|_2^2$ where α is a sufficient small positive number, such that the Hessian matrix $H_{\widetilde{\varphi}_k} \succeq \alpha \boldsymbol{I} > \boldsymbol{0}$. Unlike block-wise progressive methods that train a series of maps in [26], we directly learn one convex-gradient map for each hypothesis, since a composition of convex-gradient maps may not be the gradient of a convex potential.

The training process is given by the following Algorithm 1. For the gradient of the objective function, in Step 5, we propose two N-sample estimators to approximate the gradient of the risk

function and the Sinkhorn discrepancy, respectively; see Algorithms 3 and 4 in the next subsection for details. As shown, to generate the samples for estimators, we first draw N samples from the continuous reference distribution $\mathbb{P}_k^\epsilon$ (Step 2), and then map them to the target distribution space by the gradient of trained convex potentials (Step 3). In Step 6, the stochastic gradient ascent algorithm is used to update the network parameters.

Algorithm 1 Convex Potentials Approximation based on HyCNNs.

Require: Number of epochs T, training samples $\{\boldsymbol{x}_k^{(i)}\}_{i=1}^{n_k}$, reference distributions $\mathbb{P}_{k,0} = \mathbb{P}_k^\epsilon$.
Output: Convex potential approximations $\widetilde{\varphi}_{k,T}$ for all $k \in \mathbb{K}$.
1: Initialize strongly convex potential $\widetilde{\varphi}_{k,0}$ with parameters $\boldsymbol{\theta}_{k,0} \in \Theta$ for each $k \in \mathbb{K}$;
2: **for** $t = 0, \cdots, T-1$ **do**
3: Draw N i.i.d. samples $\{\boldsymbol{z}_k^{(i)}\}_{i=1}^N$ from initial distribution $\mathbb{P}_{k,0}$ for each $k \in \mathbb{K}$;
4: Generate samples $\{\boldsymbol{\omega}_{k,t}^{(i)}\}_{i=1}^N \leftarrow \{\nabla\widetilde{\varphi}_{k,t}(\boldsymbol{z}_k^{(i)})\}_{i=1}^N$ in target distribution $\mathbb{P}_{k,t}$ for each $k \in \mathbb{K}$;
5: Estimate the gradient of objective function of (7) by Algorithms 3 and 4:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}_{k,t}} F(\boldsymbol{\theta}_{0,t}, \boldsymbol{\theta}_{1,t}) \leftarrow\ & \mathsf{Risk\_Gradient\_Estimator}(\{\boldsymbol{\omega}_0^{(i)}\}_{i=1}^N, \{\boldsymbol{\omega}_1^{(i)}\}_{i=1}^N, \widetilde{\varphi}_{0,t}, \widetilde{\varphi}_{1,t}, \boldsymbol{\theta}_{k,t}) \\ & - \nabla_{\boldsymbol{\theta}_{k,t}} \sum_{k\in\mathbb{K}} \lambda_k \cdot \mathsf{Sinkhorn\_Estimator}(\{\boldsymbol{x}_k^{(i)}\}_{i=1}^{n_k}, \{\boldsymbol{\omega}_{k,t}^{(i)}\}_{i=1}^N), \quad \forall k \in \mathbb{K};\end{aligned}$$

6: Update $\boldsymbol{\theta}_{k,t+1} \leftarrow \boldsymbol{\theta}_{k,t} + \eta_t \cdot \nabla_{\boldsymbol{\theta}_{k,t}} F(\boldsymbol{\theta}_{0,t}, \boldsymbol{\theta}_{1,t})$, where η_t is the learning rate;
7: **end for**

3.3 Stochastic Gradient Estimation

Next, we develop stochastic estimators required by Algorithm 1 to solve Problem (7). Let $\{\boldsymbol{z}_0^{(i)}\}_{i=1}^N$ and $\{\boldsymbol{z}_1^{(i)}\}_{i=1}^N$ be N independent and identically distributed (i.i.d.) samples from reference distributions $\mathbb{P}_0^\epsilon$ and $\mathbb{P}_1^\epsilon$ (see Eq. 4), respectively. These points are subsequently transformed into samples from the target distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ via transport maps, i.e., $\boldsymbol{\omega}_0 = \nabla\varphi_0(\boldsymbol{z}_0)$ and $\boldsymbol{\omega}_1 = \nabla\varphi_1(\boldsymbol{z}_1)$, where the underlying distributions satisfy $\mathbb{P}_0 = [\nabla\varphi_0]_\# \mathbb{P}_0^\epsilon$ and $\mathbb{P}_1 = [\nabla\varphi_1]_\# \mathbb{P}_1^\epsilon$.

3.3.1 Risk Gradient Estimator

In this part, we present the gradient estimator of the risk function with respect to the parameters of HyCNNs in Algorithm 1. For every $k \in \mathbb{K}$ and $\boldsymbol{\omega}_k \sim \mathbb{P}_k$, we define the basic element of the risk function as

$$h_k(\boldsymbol{\omega}_k, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1) := \exp\left[\frac{1}{2}(\log p_{|1-k|}(\boldsymbol{\omega}_k) - \log p_k(\boldsymbol{\omega}_k))\right], \quad \forall k \in \mathbb{K}.$$

As an example, given convex functions $\widetilde{\varphi}_0$ and $\widetilde{\varphi}_1$ with parameters $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$, respectively, the gradients of $h_0(\boldsymbol{\omega}_0, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ with respect to $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are given by

$$\frac{\partial h_0(\boldsymbol{\omega}_0, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_0} = \frac{h_0(\boldsymbol{\omega}_0, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)}{2} \cdot \left[\left(\frac{\partial \boldsymbol{\omega}_0}{\partial \boldsymbol{\theta}_0}\right)^\top [\nabla^2\widetilde{\varphi}_1(\tilde{\boldsymbol{z}}_0)]^{-1}\boldsymbol{g} + \frac{\partial}{\partial \boldsymbol{\theta}_0}\log\det\nabla^2\widetilde{\varphi}_0(\boldsymbol{z}_0)\right], \tag{13a}$$

$$\frac{\partial h_0(\boldsymbol{\omega}_0, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} = -\frac{h_0(\boldsymbol{\omega}_0, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)}{2} \cdot \left[\left(\frac{\partial \nabla\widetilde{\varphi}_1(\tilde{\boldsymbol{z}}_0)}{\partial \boldsymbol{\theta}_1}\right)^\top [\nabla^2\widetilde{\varphi}_1(\tilde{\boldsymbol{z}}_0)]^{-1}\boldsymbol{g} + \frac{\partial}{\partial \boldsymbol{\theta}_1}\log\det\nabla^2\widetilde{\varphi}_1(\tilde{\boldsymbol{z}}_0)\right], \tag{13b}$$

where $\boldsymbol{g} := \frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \bar{\boldsymbol{z}}_0} = \frac{\partial[\log p_1^{\epsilon}(\bar{\boldsymbol{z}}_0) - \log\det \nabla^2 \widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)]}{\partial \bar{\boldsymbol{z}}_0}$, and $\bar{\boldsymbol{z}}_k$ is defined in Eq. (9) for each $k \in \mathbb{K}$. For brevity, the detailed gradient derivation can be found in Appendix C.1. Eqs. (13a) and (13a) indicate that directly differentiating the risk requires backpropagation through inverse maps and the log-determinant-Hessian (LDH), which is computationally expensive. Therefore, in the following, we develop a matrix-free stochastic gradient estimator for the LDH in Algorithm 2, and then present the estimator for the risk function in Algorithm 3.

For the gradient of LDH, we first introduce an $\mathcal{O}(1)$-memory estimator for the gradient of the log-determinant [34]. Specifically, for any invertible matrix H with parameters $\boldsymbol{\theta}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log\det H = \operatorname{tr}(H^{-1}\frac{\partial H}{\partial \boldsymbol{\theta}}) = \mathbb{E}_{\boldsymbol{v}}[\boldsymbol{v}^\top H^{-1}\frac{\partial H}{\partial \boldsymbol{\theta}}\boldsymbol{v}], \tag{14}$$

where the last equality uses the Hutchinson trace estimator [38] with an isotropic random vector (e.g., Rademacher or standard Gaussian) $\boldsymbol{v} \in \mathbb{R}^d$ satisfying $\mathbb{E}[\boldsymbol{v}] = 0$ and $\mathbb{E}[\boldsymbol{v}\boldsymbol{v}^\top] = \boldsymbol{I}$. With exact linear solves, the Hutchinson construction offers an unbiased estimator of the gradient of the log-determinant. In this problem, H is a symmetric positive definite Hessian matrix. Thus, $\boldsymbol{v}^\top H^{-1}$ in Eq. (14) can be efficiently solved by the following quadratic optimization problem

$$\min_{\boldsymbol{u}} \quad \frac{1}{2}\boldsymbol{u}^\top H\boldsymbol{u} - \boldsymbol{v}^\top \boldsymbol{u}, \tag{15}$$

which has the unique minimizer $\boldsymbol{u}_{\text{opt}} = H^{-1}\boldsymbol{v} = (\boldsymbol{v}^\top H^{-1})^\top$. Then, the gradient of the LDH in our problem can be estimated as $\mathbb{E}_{\boldsymbol{v}}[\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{u}_{\text{opt}}^\top H\boldsymbol{v})]$. In this paper, we summarize this procedure as Algorithm 2.

Algorithm 2 GLDH_Estimator: Estimator for the Gradient of LDH.

Require: Symmetric positive definite matrix H, parameter $\boldsymbol{\theta}$, and sample size M.

1: Initialize estimator $\boldsymbol{E} = \boldsymbol{0}$ with the same dimension as $\boldsymbol{\theta}$;
2: **for** $m = 1, \cdots, M$ **do**
3: Sample Rademacher random vector $\boldsymbol{v}^{(m)}$ with d dimensions;
4: $\boldsymbol{u}^{(m)} \leftarrow \mathsf{ConjugateGradient}(H, \boldsymbol{v}^{(m)})$;
5: $\overline{\boldsymbol{u}}^{(m)} \leftarrow \mathsf{StopGradient}\left(\boldsymbol{u}^{(m)}\right)$;
6: $\boldsymbol{E} \leftarrow \boldsymbol{E} + \nabla_{\boldsymbol{\theta}}\left[\left(\overline{\boldsymbol{u}}^{(m)}\right)^\top H\boldsymbol{v}^{(m)}\right]$;
7: **end for**
8: **Return** $\boldsymbol{E}/M$.

In Step 4 of Algorithm 2, the function $\mathsf{ConjugateGradient}(H, \boldsymbol{v}) := \arg\min_{\boldsymbol{u}} \frac{1}{2}\boldsymbol{u}^\top H\boldsymbol{u} - \boldsymbol{v}^\top \boldsymbol{u}$ represents that we use the conjugate gradient method to solve the unconstrained optimization problem (15). In Step 5, $\mathsf{StopGradient}(\cdot)$ treats its argument as a constant during backpropagation. Thus, in Step 6, only $H(\boldsymbol{\theta})$ is differentiated, and the term $\nabla_{\boldsymbol{\theta}}[\left(\overline{\boldsymbol{u}}^{(m)}\right)^\top H\boldsymbol{v}^{(m)}]$ can be efficiently computed by Hessian-vector and vector-Jacobian products.

Then, the gradient of the risk function (6) can be estimated by Algorithm 3. Both functions ConjugateGradient in Step 5 and GLDH_Estimator have been defined in Algorithm 2. In Step 6, we compute the value of the log-determinant in $h_k(\boldsymbol{\omega}_k, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ exactly when $d \le 32$ and estimate it by the stochastic Lanczos quadrature (SLQ) [33, 34] when $d > 32$.

Algorithm 3 Risk_Gradient_Estimator: Gradient Estimator for the Risk Function.

Require: Generated samples $\{\boldsymbol{\omega}_k^{(i)}\}_{i=1}^N$ and convex potentials $\widetilde{\varphi}_k$ for all $k \in \mathbb{K}$, parameter $\boldsymbol{\theta}$.

1: Initialize estimator $\boldsymbol{E} = \mathbf{0}$ with the same dimension as $\boldsymbol{\theta}$;

2: **for** $k = 0, 1$ **do**

3: **for** $i = 1, \cdots, N$ **do**

4: Compute $\boldsymbol{z}_k^{(i)} = \nabla \widetilde{\varphi}_k^{-1}(\boldsymbol{\omega}_k^{(i)})$ and $\bar{\boldsymbol{z}}_k^{(i)} = \nabla \widetilde{\varphi}_{|1-k|}^{-1}(\boldsymbol{\omega}_k^{(i)})$ by Eq. (9);

5: Compute $\boldsymbol{u}_k^{(i)} \leftarrow \mathsf{ConjugateGradient}\left(\nabla^2 \widetilde{\varphi}_{|1-k|}(\bar{\boldsymbol{z}}_k^{(i)}), \frac{\partial \log p_{|1-k|}(\boldsymbol{\omega}_k^{(i)})}{\partial \bar{\boldsymbol{z}}_k^{(i)}}\right)$;

6: Update the gradient estimator

$$\boldsymbol{E} \leftarrow \boldsymbol{E} + \frac{h_k(\boldsymbol{\omega}_k^{(i)}, \boldsymbol{\theta}_0, \boldsymbol{\theta}_1)}{2} \cdot \left\{ \left[\frac{\partial \left(\boldsymbol{\omega}_k^{(i)} - \nabla \widetilde{\varphi}_{|1-k|}(\bar{\boldsymbol{z}}_k^{(i)}) \right)}{\partial \boldsymbol{\theta}} \right]^\top \boldsymbol{u}_k^{(i)} \right.$$
$$\left. + \mathsf{GLDH\_Estimator}\left(\nabla^2 \widetilde{\varphi}_k(\boldsymbol{z}_k^{(i)}), \boldsymbol{\theta} \right) - \mathsf{GLDH\_Estimator}\left(\nabla^2 \widetilde{\varphi}_{|1-k|}(\bar{\boldsymbol{z}}_k^{(i)}), \boldsymbol{\theta} \right) \right\};$$

7: **end for**

8: **end for**

9: **Return** $\boldsymbol{E}/N$.

3.3.2 Sinkhorn Discrepancy Estimator

Computing the Sinkhorn discrepancy term $\mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k, \mathbb{P}_k)$ exactly (see Definition 2) in the objective function is challenging. In this part, we present a differentiable finite-sample estimator for the Sinkhorn discrepancy. The gradient of this estimator can be computed efficiently by backpropagation and is used in Algorithm 1.

We first introduce the following proposition regarding the dual formulation of Sinkhorn discrepancy [32, 39].

Proposition 3. *Let $\mathbb{P}$ and $\mathbb{Q}$ be probability measures with finite second moments. The strong dual formulation of Sinkhorn discrepancy $\mathcal{S}_\epsilon(\mathbb{P}, \mathbb{Q})$ is given by*

$$\sup_{f \in L^1(\mathbb{P}), g \in L^1(\mathbb{Q})} \left\{ \int f(\boldsymbol{x})\, \mathrm{d}\mathbb{P}(\boldsymbol{x}) + \int g(\boldsymbol{z})\, \mathrm{d}\mathbb{Q}(\boldsymbol{z}) - \epsilon \iint \left[\exp\left(\frac{f(\boldsymbol{x}) + g(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z})}{\epsilon} \right) - 1 \right] \mathrm{d}\mathbb{P}(\boldsymbol{x}) \mathrm{d}\mathbb{Q}(\boldsymbol{z}) \right\},$$

Let f_ϵ and g_ϵ be the solutions of the dual formulation, which is also known as optimal entropic potentials. Then the transport plan

$$\gamma_\epsilon(\mathrm{d}\boldsymbol{x}, \mathrm{d}\boldsymbol{z}) = \exp\left(\frac{f_\epsilon(\boldsymbol{x}) + g_\epsilon(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z})}{\epsilon} \right) \mathrm{d}\mathbb{P}(\boldsymbol{x}) \mathrm{d}\mathbb{Q}(\boldsymbol{z})$$

is the unique solution to the Sinkhorn discrepancy, and the optimal entropic potentials satisfy

$$\int \exp\left(\frac{f_\epsilon(\boldsymbol{x}) + g_\epsilon(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z})}{\epsilon} \right) \mathrm{d}\mathbb{P}(\boldsymbol{x}) = 1, \quad \forall \boldsymbol{z} \in \Omega, \tag{16a}$$

$$\int \exp\left(\frac{f_\epsilon(\boldsymbol{x}) + g_\epsilon(\boldsymbol{z}) - c(\boldsymbol{x}, \boldsymbol{z})}{\epsilon}\right) \mathrm{d}\mathbb{Q}(\boldsymbol{z}) = 1, \quad \forall \boldsymbol{x} \in \Omega, \tag{16b}$$

for $\mathbb{Q}$-almost every $\boldsymbol{z}$ and $\mathbb{P}$-almost every $\boldsymbol{x}$, respectively.

Since the target continuous distribution $\mathbb{P}_k$ is computationally intractable, we approximate it as empirical distributions formed by the generated samples, denoted by $\widetilde{\mathbb{P}}_0 = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\omega}_0^{(i)}}$ and $\widetilde{\mathbb{P}}_1 = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\omega}_1^{(i)}}$. Therefore, it suffices to compute the value $\mathcal{S}_{\epsilon,(n_k,N)}(\widehat{\mathbb{P}}_k, \widetilde{\mathbb{P}}_k)$. The optimal entropic potentials between discrete distributions $\widehat{\mathbb{P}}_k$ and $\widetilde{\mathbb{P}}_k$, denoted by $f_{\epsilon,k,(n_k,N)}$ and $g_{\epsilon,k,(n_k,N)}$, can be obtained by Sinkhorn's algorithm [20, 39], leading to the following estimator for the barycentric projection of the optimal entropic coupling:

$$T_{\epsilon,k,(n_k,N)}(\boldsymbol{x}) = \frac{\frac{1}{N}\sum_{i=1}^N \boldsymbol{\omega}_k^{(i)} e^{\left(g_{\epsilon,k,(n_k,N)}(\boldsymbol{\omega}_k^{(i)}) - c(\boldsymbol{x},\boldsymbol{\omega}_k^{(i)})\right)/\epsilon}}{\frac{1}{N}\sum_{i=1}^N e^{\left(g_{\epsilon,k,(n_k,N)}(\boldsymbol{\omega}_k^{(i)}) - c(\boldsymbol{x},\boldsymbol{\omega}_k^{(i)})\right)/\epsilon}}, \quad \forall k \in \mathbb{K}.$$

However, the discrete dual potential $g_{\epsilon,k,(n_k,N)}$ is initially defined only on the generated support points and is thus not differentiable for all samples in Ω. To make the optimized empirical Sinkhorn discrepancy differentiable with respect to the generated sample locations, we extend g to the general continuous sample space for fixed f:

$$g^c_{\epsilon,k}(\boldsymbol{\omega}) = -\epsilon \log\left(\frac{1}{n_k}\sum_{j=1}^{n_k} \exp\left(\frac{f_k(\boldsymbol{x}_k^{(j)}) - c(\boldsymbol{x}_k^{(j)}, \boldsymbol{\omega})}{\epsilon}\right)\right), \quad \forall \boldsymbol{\omega} \in \Omega. \tag{17}$$

This extension agrees with the original potential g on the empirical support at convergence while making it differentiable. Therefore, our differentiable estimator for the Sinkhorn discrepancy is given by

$$\widehat{\mathcal{S}}_{\epsilon,(n_k,N)}(\widehat{\mathbb{P}}_k, \mathbb{P}_k) := \mathcal{S}_{\epsilon,(n_k,N)}(\widehat{\mathbb{P}}_k, \widetilde{\mathbb{P}}_k) = \frac{1}{n_k}\sum_{j=1}^{n_k} f_{\epsilon,k,(n_k,N)}(\boldsymbol{x}_k^{(j)}) + \frac{1}{N}\sum_{i=1}^{N} g^c_{\epsilon,k}(\boldsymbol{\omega}_k^{(i)}), \tag{18}$$

and its gradient can be computed by backpropagation (see Appendix C.2).

Therefore, the pseudo-code for estimating the differentiable Sinkhorn discrepancy is shown in Algorithm 4. Steps 1-6 in Algorithm 4 are essentially Sinkhorn's algorithm [39], and thus the computational complexity of each iteration is $\mathcal{O}(n_k \cdot N)$. This alternating algorithm computes the entropic potentials by iteratively updating f and g while satisfying one of the two optimal marginal conditions in Eq. (16).

4 Representation Power of HyCNNs

This section provides theoretical analyses of the representation capacity of HyCNNs. Specifically, we focus on their representation power for both convex functions and their gradients, as well as the distributional universality of their induced transport maps.

Algorithm 4 Sinkhorn_Estimator: Estimator for the Sinkhorn Discrepancy.

Require: Training samples $\{\boldsymbol{x}_k^{(i)}\}_{i=1}^{n_k}$ from $\widehat{\mathbb{P}}_k$ and generated samples $\{\boldsymbol{\omega}_k^{(i)}\}_{i=1}^{N}$.

1: Initialize $i = 0$ and $f_{\epsilon,k}^{(0)} = 0$ for each k;

2: **while** not converged **do**

3: update $g_{\epsilon,k}^{(i+1)}(\boldsymbol{\omega}_k) \leftarrow -\epsilon \log \frac{1}{n_k} \sum_{j=1}^{n_k} \exp\left(\frac{f_{\epsilon,k}^{(i)}(\boldsymbol{x}_k^{(j)}) - c(\boldsymbol{x}_k^{(j)}, \boldsymbol{\omega}_k)}{\epsilon}\right)$ for each $k \in \mathbb{K}$;

4: update $f_{\epsilon,k}^{(i+1)}(\boldsymbol{x}_k) \leftarrow -\epsilon \log \frac{1}{N} \sum_{j=1}^{N} \exp\left(\frac{g_{\epsilon,k}^{(i+1)}(\boldsymbol{\omega}_k^{(j)}) - c(\boldsymbol{x}_k, \boldsymbol{\omega}_k^{(j)})}{\epsilon}\right)$ for each $k \in \mathbb{K}$;

5: $i \leftarrow i + 1$;

6: **end while**

7: Extend the entropic potential $g_{\epsilon,k}^{(i+1)}(\boldsymbol{\omega}_k)$ to continuous space by Eq. (17);

8: **Return** $\widetilde{\mathcal{S}}_{\epsilon,(n_k,N)}(\widehat{\mathbb{P}}_k, \mathbb{P}_{k,t})$ computed by Eq. (18).

We refer to HyCNNs equipped with *smooth* maxout activation functions $\bar{\sigma}_\tau$ (see Eq. (12)) as *smooth HyCNNs*. We first show the representation power of both standard and smooth HyCNNs. Let L and h denote the number of layers and the width at each hidden layer of HyCNNs, respectively. Since we take the strongly convex coefficient $\alpha \to 0$, these results hold in our strongly convex setting.

In the following results, we take the strongly convex coefficient $\alpha \to 0$.

Theorem 3 (Representation Power of (Smooth) HyCNNs)**.** *For any L_f-Lipschitz convex function $f : \mathbb{R}^d \to \mathbb{R}$, compact convex set $\mathbb{D} \subset \mathbb{R}^d$, and $\varepsilon > 0$, assume that $\mathbb{D}$ has a diameter $D \in (0, \infty)$ such that $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \le D$ for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{D}$, and then:*

(I) *There exists a hyper input convex neural network $\widehat{\mathsf{HyCNN}}$ with maxout activation functions satisfying $\sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathsf{HyCNN}}(\boldsymbol{x}) - f(\boldsymbol{x})| \le \varepsilon$ and $L \cdot h = \mathcal{O}((\frac{DL_f}{\varepsilon})^d)$.*

(II) *There exists a smooth HyCNN with a sufficiently small parameter $\tau > 0$, denoted by $\widehat{\mathsf{HyCNN}}^\tau$, satisfying $\sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathsf{HyCNN}}^\tau(\boldsymbol{x}) - f(\boldsymbol{x})| \le \varepsilon$ and $L \cdot h = \mathcal{O}((\frac{DL_f}{\varepsilon})^d)$.*

(III) *There exists a sequence of smooth HyCNNs, denoted by $\{\widehat{\mathsf{HyCNN}}_n^\tau\}_{n=1}^\infty$, satisfying $\widehat{\mathsf{HyCNN}}_n^\tau \to f$ uniformly on $\mathbb{D}$, when τ is sufficiently small and $L \cdot h = \mathcal{O}((\frac{DL_f}{\varepsilon})^d)$.*

The proof of Theorem 3 is provided in Appendix B.3. Compared to existing qualitative representation power theorems of ICNNs [34, 40], Theorem 3 extends this result to HyCNNs and provides quantitative guidelines for hyperparameter selection.

Although these properties of (smooth) HyCNNs show that they can uniformly approximate convex potentials on a compact set, convergence of potentials does not generally imply convergence of the gradient fields [34]. Theorem 4 shows that the smooth HyCNNs approximate the gradient of convex functions well.

Theorem 4 (Gradient Convergence of Smooth HyCNNs)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a Lipschitz continuous convex function, and $\{\widehat{\mathsf{HyCNN}}_n^\tau\}_{n=1}^\infty$ be a sequence of smooth HyCNNs with hyperparameters $L_n \cdot h_n = \mathcal{O}({\varepsilon_n}^{-d})$ for any $\varepsilon_n > 0$, where $\widehat{\mathsf{HyCNN}}_n^\tau : \mathbb{R}^d \to \mathbb{R}$. Assume that $\widehat{\mathsf{HyCNN}}_n^\tau \to f$. Then*

(I) *For almost every* $\boldsymbol{x} \in \mathbb{R}^d$, $\nabla \widehat{\mathsf{HyCNN}}_n^{\tau}(\boldsymbol{x}) \to \nabla f(\boldsymbol{x})$.

(II) *If* f *is Lipschitz smooth, the gradient sequence* $\nabla \widehat{\mathsf{HyCNN}}_n$ *converges to* ∇f *uniformly at a rate of* $\mathcal{O}\left(\sqrt{\varepsilon_n}\right)$.

The proof of Theorem 4 is provided in Appendix B.4. Generally, Theorem 4 holds for any differentiable convex function sequence. Note that the uniform convergence assumption in Theorem 4 can be verified using the representation power of smooth HyCNNs on compact sets (see Theorem 3(III)).

Based on the results above and Brenier's theorem, the following Theorem 5 shows that smooth HyCNNs are distributionally universal in our problem.

Theorem 5 (Distributional Universality of Smooth HyCNNs)**.** *Let* $\mu, \nu \in \mathscr{P}(\Omega)$ *be regarded as probability measures on* $\mathbb{R}^d$ *supported on* Ω*. Assume that* μ *is absolutely continuous with respect to the* d*-dimensional Lebesgue measure on* $\mathbb{R}^d$*, and that* μ *and* ν *have finite second moments. Then there exists a sequence of smooth HyCNNs, denoted by* $\{\widehat{\mathsf{HyCNN}}_n^{\tau}\}_{n=1}^{\infty}$*, such that* $(\nabla \widehat{\mathsf{HyCNN}}_n^{\tau})_{\#}\mu \Rightarrow \nu$*, where* $\Rightarrow$ *denotes weak convergence of probability measures on* $\mathbb{R}^d$*.*

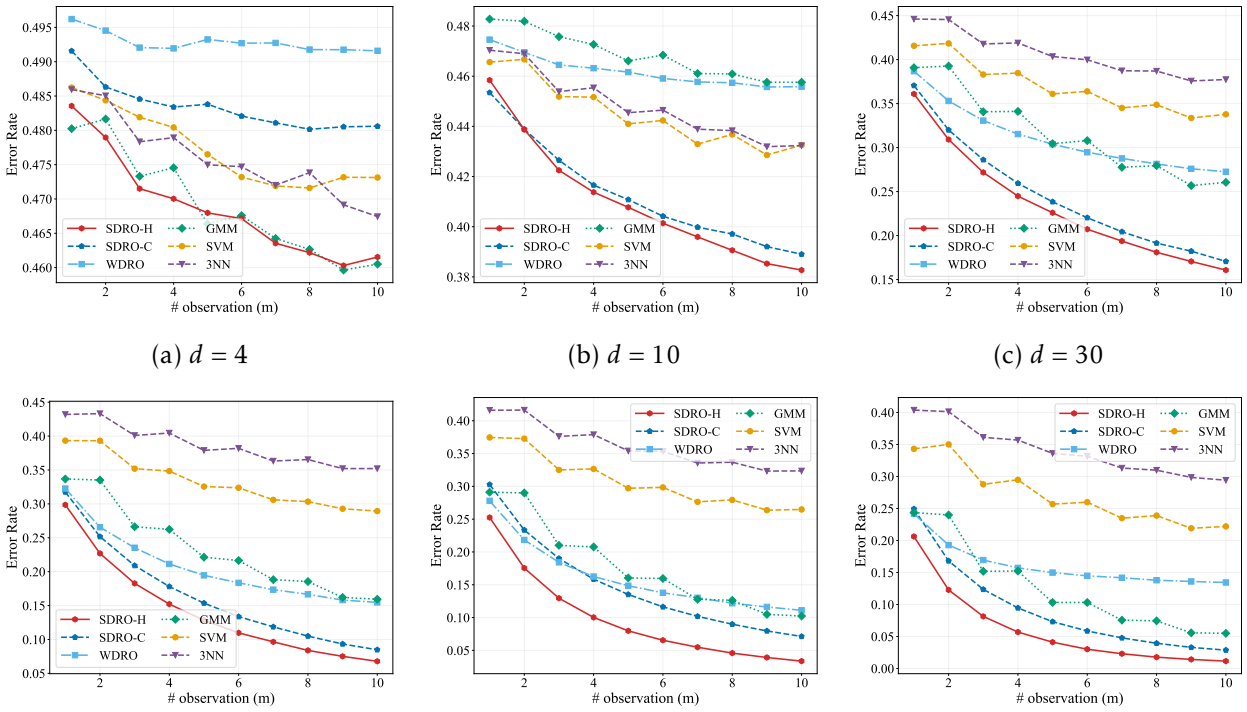
The proof of Theorem 5 is provided in Appendix B.5. It demonstrates that smooth HyCNN gradient maps are distributionally universal for approximating target distributions. For a reference distribution $\mathbb{P}^{\epsilon} \in \mathscr{P}(\Omega)$ and any candidate least-favorable distribution $\mathbb{P}^{\star} \in \mathscr{P}(\Omega)$ in our problem, assume that both $\mathbb{P}^{\epsilon}$ and $\mathbb{P}^{\star}$ have finite second moments, and $\mathbb{P}^{\epsilon}$ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$. Then, the gradient map of smooth HyCNNs is distributionally universal for approximating any candidate least-favorable distribution.

5 Experiments

In this section, we conduct experiments to assess the effectiveness of the proposed method on datasets of varying training sample sizes and dimensions. Specifically, we evaluate its performance on small sample sizes using a synthetic Gaussian mixture dataset (Section 5.1) and the MNIST dataset (Section 5.2), and on large sample sizes using the Higgs dataset (Section 5.3). We also provide a qualitative illustration of our method on a high-dimensional human face generation task (Section 5.4). All experiments were conducted on a laptop with an NVIDIA GeForce RTX 4060 GPU using Python. Compared to the existing methods, our approach can handle hypothesis testing problems across different sample sizes and dimensions, achieving higher accuracy and robustness. Codes and experimental results are available at https://github.com/Arthas819/SDRHT.

5.1 Synthetic Gaussian Mixture Dataset

In this subsection, we consider a synthetic Gaussian mixture dataset based on [17]. Let the samples under the two hypotheses be generated from distributions $0.5\mathcal{N}(0.4\boldsymbol{e}, I_d) + 0.5\mathcal{N}(-0.4\boldsymbol{e}, I_d)$ and $0.5\mathcal{N}(0.4\boldsymbol{f}, I_d) + 0.5\mathcal{N}(-0.4\boldsymbol{f}, I_d)$, respectively. Here $\boldsymbol{e} \in \mathbb{R}^d$ is a vector with all entries equal to 1, and $\boldsymbol{f} \in \mathbb{R}^d$ is a vector with the first $\lceil d/2 \rceil$ entries equal to 1 and the remaining entries equal to -1. Consider a setting with a small number of training samples $n_0 = n_1 = 10$, and then test



(a) $d = 4$ (b) $d = 10$ (c) $d = 30$

(d) $d = 50$ (e) $d = 70$ (f) $d = 100$

Figure 3: Error rate comparison on GMM dataset, averaged over 100 trials. **SDRO-H**: SDRHT method with HyCNN; **SDRO-C**: SDRHT method with standard ICNN.

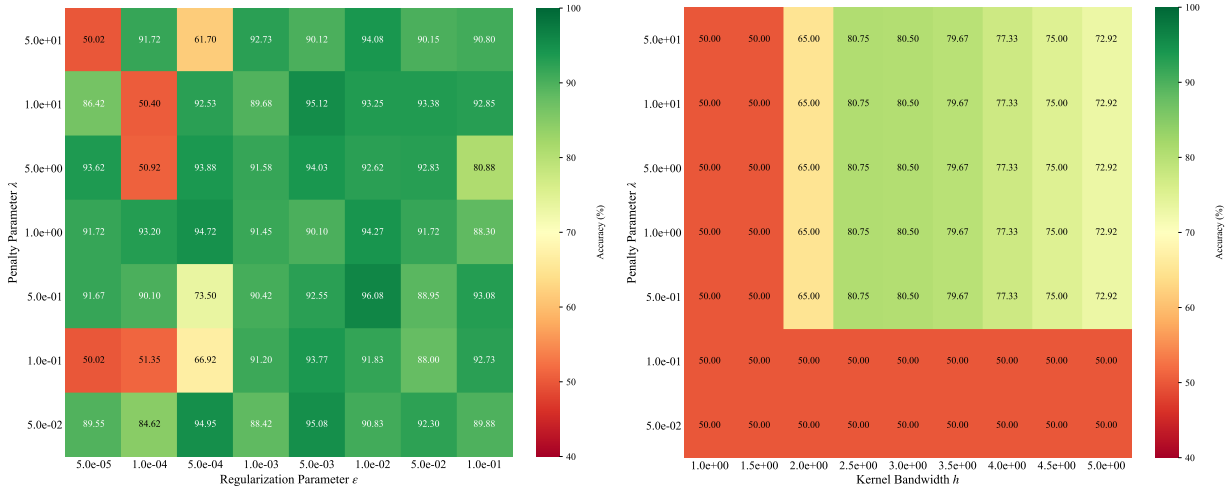
on 1000 new samples from each mixture model. We compare our method against Wasserstein DRO-based hypothesis testing (WDRO) [17], Gaussian Mixture Model (GMM), kernel support vector machine (SVM) with radial basis function (RBF) kernel, and a three-layer neural network (3NN) with hidden layer dimensions $(32, 32, 32)$. The details of the WDRO method are provided in Appendix D. All hyperparameters, including λ and ϵ in SDRO-H (SDRHT with HyCNN) and SDRO-C (SDRHT with standard ICNN), ambiguity set radii, and kernel bandwidth in WDRO, are determined by cross-validation.

Table 2: Computation time (s) of methods on Gaussian mixture dataset, averaged over 100 trials.

d	**Computation Time (s)**					
	SDRO-H	**SDRO-C**	**WDRO**	**GMM**	**SVM**	**3NN**
4	1.906	1.749	0.283	0.075	<0.000	0.052
10	1.507	1.802	0.275	0.002	0.001	0.040
30	1.292	1.900	0.271	0.002	<0.000	0.034
50	1.753	1.897	0.285	0.002	0.001	0.033
70	1.614	1.907	0.290	0.002	<0.000	0.026
100	1.725	1.844	0.327	0.002	<0.000	0.028

Figure 3 compares the classification error rate of selected methods over varying dimensions $d \in \{4, 10, 30, 50, 70, 100\}$. The horizontal axis in Figure 3 represents that there are m independent observations in a testing sample, and the decision is made by majority rule [17]. As illustrated, Hy-CNN is more suitable for approximating convex potentials in SDRO than standard ICNN (though the performance of SDRO-C remains competitive in Figures 3b-3f), and the SDRO-H method achieves the lowest classification error rate in almost all dimensions. These results demonstrate the superior performance of our method on hypothesis-testing tasks with small sample sizes compared to selected baselines. Table 2 shows the average computation time of all methods. Although the training time of SDRO is higher than that of other baselines, it remains under 2s and remains stable as d increases.

Figure 4 reports the accuracy of SDRO-H and WDRO across different hyperparameter combinations when $n_0 = n_1 = 10$ and $d = 100$. As illustrated, the accuracy under the best parameter combination of WDRO is inferior to that of SDRO-H. The result of SDRO-H indicates that one may improve accuracy by moderately increasing the penalty parameter λ and decreasing the entropic regularization parameter ϵ. This is because a small λ leads to excessive conservatism, and a large ϵ induces stronger smoothing and may dilute the correlation between samples and classification labels. The result of WDRO demonstrates that under an appropriate bandwidth selection, the classification accuracy is not sensitive to λ.



(a) SDRO-H (b) WDRO

Figure 4: Average accuracy (%) of SDRO-H and WDRO on the GMM dataset with different hyperparameters ($n_0 = n_1 = 10, d = 100$).

To evaluate the adversarial robustness of our model on adversarial examples, we apply adversarial perturbations to the test data using the projected gradient descent (PGD) attack [26, 41]. Specifically, for a testing sample $\boldsymbol{\omega} \sim \mathbb{P}_k$, the attacked sample for fixed detector ϕ_{opt}, defined by a point-wise perturbation problem, is given by

$$\boldsymbol{\omega}_{\text{attacked}} := \boldsymbol{\omega} + \boldsymbol{\delta}^*_{\boldsymbol{\omega}}, \quad \boldsymbol{\delta}^*_{\boldsymbol{\omega}} := \arg \max_{\|\boldsymbol{\delta}_{\boldsymbol{\omega}}\|_p \le \Delta} \quad L_k(\phi_{\text{opt}}, \boldsymbol{\omega} + \boldsymbol{\delta}_{\boldsymbol{\omega}}). \tag{19}$$

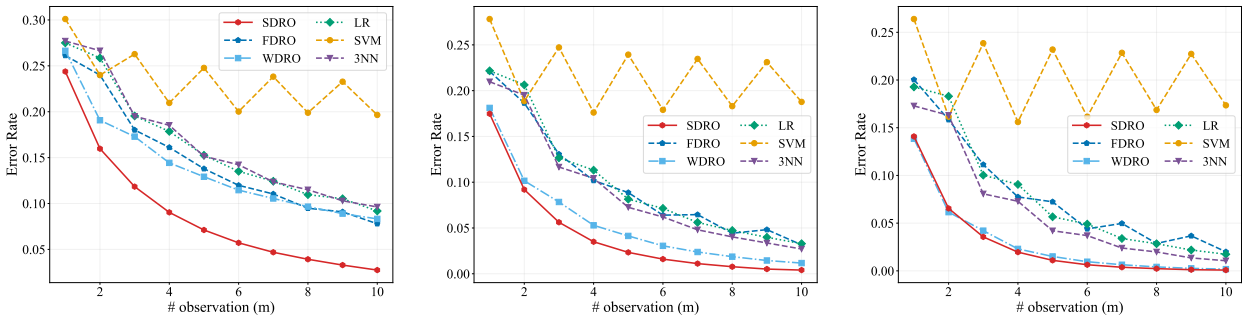
Table 3: Robustness comparison on the GMM dataset after PGD attack, averaged over $m \in [10]$ and 5 trials (accuracy %, $n_0 = n_1 = 10, d = 50$). Bold represents the highest accuracy in each column.

Method	PGD-L_2 Attack				PGD-L_∞ Attack			
	$\Delta = 0.025$	$\Delta = 0.05$	$\Delta = 0.075$	$\Delta = 0.1$	$\Delta = 0.05$	$\Delta = 0.1$	$\Delta = 0.15$	$\Delta = 0.2$
SDRO-H	**85.801**	**85.769**	**85.736**	**85.737**	**85.467**	**85.165**	**84.906**	**84.811**
SDRO-C	82.333	82.347	82.332	82.309	82.175	82.018	81.867	81.774
WDRO	69.693	68.902	68.108	67.924	62.440	54.711	48.096	44.256
GMM	73.401	72.392	71.367	71.121	63.100	51.579	41.804	36.414
SVM	64.477	63.628	62.774	62.561	55.957	47.171	40.166	36.166
3NN	58.187	57.021	55.906	55.638	47.635	37.184	29.483	25.509

In the following, we assess robustness using L_2 and L_∞ norm constraints, i.e., $p = 2$ and $p \to \infty$ in Eq. (19). In Table 3, we compare the classification accuracy of all methods across different levels of Δ. As shown in Table 3, the accuracies of the proposed methods remain higher and more stable than those of the baselines as the attack intensity Δ increases under both L_2 and L_∞ constraints, and SDRO-H consistently yields the best test accuracy across all values of Δ.

5.2 MNIST Handwritten Digits Classification

This subsection considers the MNIST handwritten digits dataset (d=784). In each trial, we randomly select 5 digits as class 0, and the remaining as class 1. We select n' samples for each digit of each class to form the two training sets, and therefore $n_0 = n_1 = 5n'$. The testing data includes 1000 images from both classes. For this dataset, we further include logistic regression (LR) and Flow-based DRO (FDRO) [26] as baselines. The FDRO method, which characterizes least-favorable distributions from a generative perspective, is suitable for high-dimensional robust hypothesis testing and requires training convolutional autoencoders. In the following, the label 'SDRO' represents our SDRHT method with HyCNN. All hyperparameters are determined by cross-validation.



(a) $n' = 2, n_0 = n_1 = 10$ (b) $n' = 5, n_0 = n_1 = 25$ (c) $n' = 10, n_0 = n_1 = 50$

Figure 5: Error rate comparison on the MNIST dataset, averaged over 100 trials.

Figure 5 illustrates that the proposed method achieves a lower error rate than existing well-tuned DRO methods and common classifiers across different training set sizes. Table 4 shows the average computation time of all methods, illustrating that the training time of SDRO is acceptable and stable as n' increases. To evaluate robustness, we compare all methods under both L_2 and L_∞ PGD attacks (see Eq. (19)) in Table 5. As shown, SDRO achieves the highest accuracy across all non-zero attack intensities under both norms, indicating the superior robustness of our method. Under the PGD-L_∞ attack, the three DRO-based models demonstrate better robustness than other baselines.

Table 4: Computation time (s) of methods on the MNIST dataset, averaged over 100 trials.

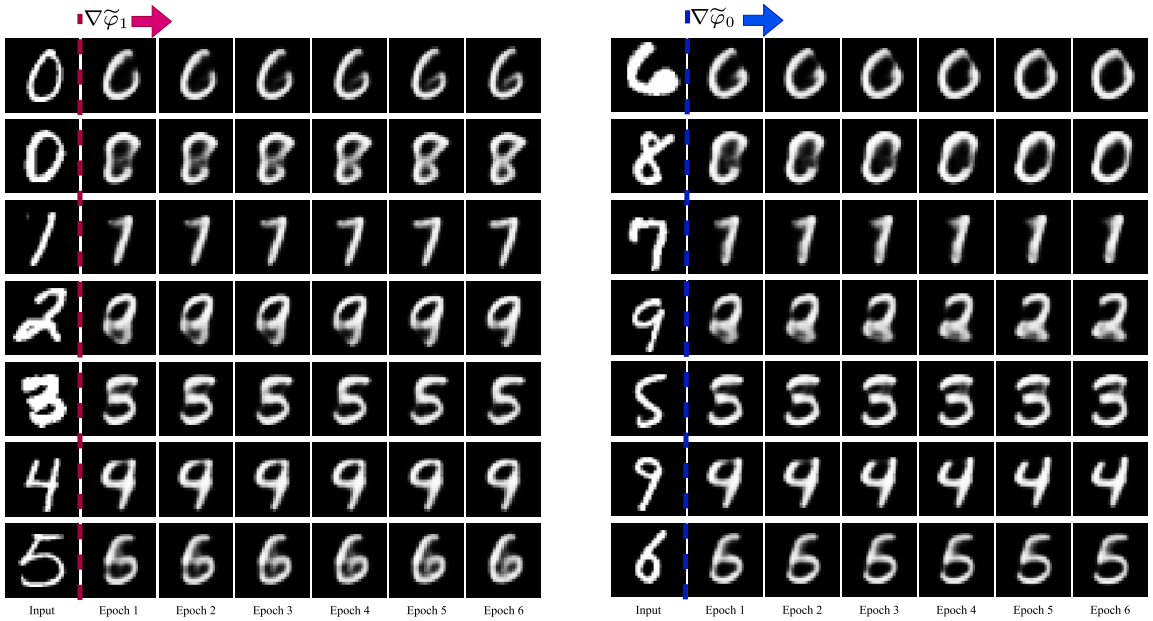
n'	Computation Time (s)					
	SDRO	**FDRO**	**WDRO**	**LR**	**SVM**	**3NN**
2	1.943	4.821	0.022	0.008	<0.001	0.167
5	1.477	8.999	0.045	0.011	0.001	0.231
10	1.465	17.153	0.135	0.018	0.001	0.269

Table 5: Robustness comparison on MNIST dataset after PGD attack, averaged over $m \in [10]$ and 5 trials (accuracy %, $n_0 = n_1 = 25$).

Method	**PGD-L_2 Attack**				**PGD-L_∞ Attack**			
	$\Delta = 0.025$	$\Delta = 0.05$	$\Delta = 0.075$	$\Delta = 0.1$	$\Delta = 0.05$	$\Delta = 0.1$	$\Delta = 0.15$	$\Delta = 0.2$
SDRO	**95.75**	**95.70**	**95.67**	**95.67**	**93.53**	**90.60**	**87.46**	**85.47**
FDRO	91.04	90.93	90.87	90.87	86.51	80.92	75.28	71.91
WDRO	94.54	94.47	94.42	94.42	91.18	85.63	79.42	75.56
LR	91.15	90.98	90.89	90.89	84.79	75.80	67.27	62.34
SVM	76.67	76.50	76.44	76.44	72.65	69.02	66.21	64.73
3NN	91.39	91.21	91.11	91.11	82.10	68.34	55.00	48.09

We next visualize how the trained convex functions induce adversarial distributional attacks on their opposite classes. In each trial, we select two digits that are easily confused in handwritten form and place them in separate classes. To generate a more intuitive and clear trajectory, in the training phase, we first train a variational autoencoder, including two parts, Encoder and Decoder, to embed all images into a 32-dimensional latent space. Then, the convex potentials are trained on this latent space with a small training sample size. In the generation phase, we randomly select an image z_k from each class $k \in \mathbb{K}$, encode it into the latent space, denoted by $\widetilde{z}_k := \mathsf{Encoder}(z_k)$, and then map $\widetilde{z}_k$ by the gradient of the convex potential of the other class, i.e., $\nabla\widetilde{\varphi}_{|1-k|}(\widetilde{z}_k)$, and finally draw the attacked image $\mathsf{Decoder}\Big(\nabla\widetilde{\varphi}_{|1-k|}(\widetilde{z}_k)\Big)$. Figure 6 illustrates the trajectory of injecting adversarial attacks on digits in both classes in 7 trials. As illustrated, our adversarial attack naturally achieves conversions of different classes of numbers through convex

gradient mapping in a few steps.



(a) Adversarial attack for Class 0 by $\nabla\widetilde{\varphi}_1$

(b) Adversarial attack for Class 1 by $\nabla\widetilde{\varphi}_0$

Figure 6: Adversarial attacks on digits by trained convex gradients. The first column displays the original images, while the following six columns show the attacked images from Epochs 1 to 6.

5.3 High-energy Physics Higgs Dataset

In this subsection, we evaluate our method on the Higgs dataset, which distinguishes between a signal process where new theoretical Higgs bosons (HIGGS) are produced, and a background process with the identical decay products but distinct kinematics [42]. We consider 21 original features and large training sample sizes.

Table 6 reports the accuracy and computation time of our method against other baselines over varying training sample sizes. As shown, the SDRO method achieves the highest accuracy in a relatively low computation time. Note that Table 6 does not include the result of WDRO, since when the sample size is large, WDRO did not produce an acceptable solution within the 1,800-second time limit. This is because WDRO requires solving a linear program with $\mathcal{O}(n_0 n_1)$ decision variables (see Appendix D) and becomes computationally prohibitive for large sample sizes.

In Table 7, we show that SDRO remains the highest accuracy over 70% across all non-zero attack intensities under both norms. This result demonstrates that our method can still maintain its accuracy even under significant adversarial perturbations compared to other baselines.

Table 6: Accuracy (%) and computation time (s) of methods on the Higgs dataset, averaged over observations $m \in [10]$ and 10 trials. Bold represents the highest accuracy in each row.

$n_0 = n_1$	Accuracy (%)					Computation Time (s)				
$(\times 10^3)$	**SDRO**	**FDRO**	**LR**	**SVM**	**3NN**	**SDRO**	**FDRO**	**LR**	**SVM**	**3NN**
1	**68.09**	62.07	59.73	64.13	57.92	1.11	2.09	0.02	0.18	5.09
2	**71.32**	65.56	60.45	66.67	58.70	1.06	2.06	0.01	0.58	11.06
3	**72.83**	67.20	60.68	67.87	58.97	1.13	2.69	0.01	0.74	9.86
4	**73.88**	68.03	60.95	68.88	60.17	1.64	3.65	0.02	1.51	11.21
5	**74.62**	68.77	61.01	69.56	61.00	1.79	3.56	0.02	2.20	11.16
6	**75.37**	69.15	61.07	70.27	61.54	1.87	3.30	0.02	3.01	13.74
7	**75.71**	69.21	61.09	70.67	62.86	2.37	3.35	0.03	4.46	15.90
8	**76.25**	69.60	61.22	71.13	63.66	2.47	3.30	0.03	5.66	16.71
9	**76.26**	69.71	61.18	71.14	63.85	2.54	3.73	0.02	7.33	18.53
10	**76.66**	69.91	61.29	71.59	64.89	2.96	3.52	0.03	8.65	21.41

Note: The WDRO method did not produce an acceptable solution within the 1,800-second time limit.

Table 7: Robustness comparison on Higgs dataset after PGD attack, averaged over $m \in [10]$ and 5 trials (accuracy %, $n_0 = n_1 = 2,000$).

Method	**PGD-L_2 Attack**				**PGD-L_∞ Attack**			
	$\Delta = 0.025$	$\Delta = 0.05$	$\Delta = 0.075$	$\Delta = 0.1$	$\Delta = 0.025$	$\Delta = 0.05$	$\Delta = 0.075$	$\Delta = 0.1$
SDRO	**72.342**	**71.998**	**71.887**	**71.891**	**73.352**	**72.758**	**72.436**	**72.284**
FDRO	66.999	64.888	62.736	60.835	62.780	55.396	48.914	44.461
LR	62.945	60.913	59.193	57.582	59.708	54.046	48.528	44.412
SVM	67.669	65.849	64.191	62.716	64.524	58.350	53.250	49.614
3NN	58.204	54.669	50.782	48.015	50.264	38.658	30.418	26.266

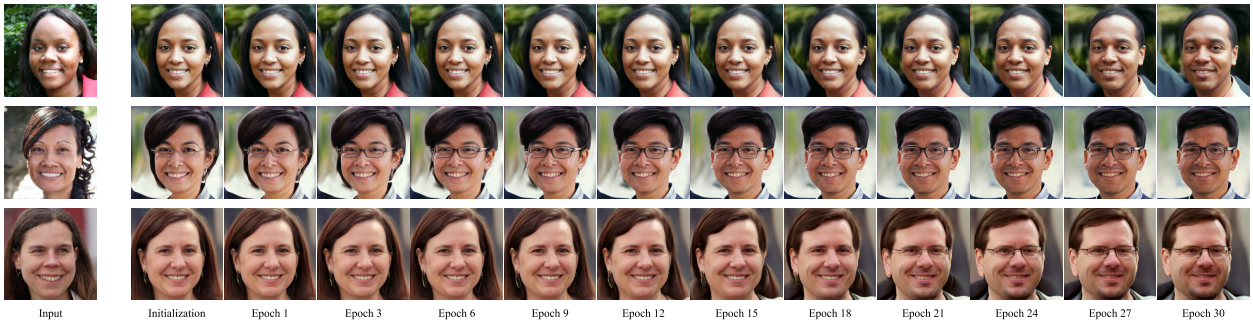
5.4 Human Face Generation

In this subsection, similar to [28], we present a qualitative experiment on the human face dataset FFHQ [43] that illustrates how our method generates adversarial-attacked high-dimensional images.

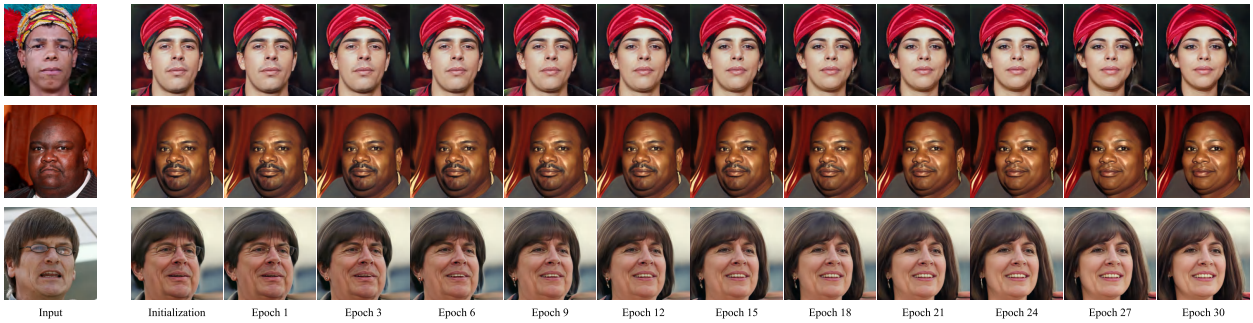
We take the gender attribute of the FFHQ dataset as the binary label in the following image generation experiment. We labeled Class 0 as *Female* and Class 1 as *Male*. In the training phase, we first map the original 1024×1024 images to a 512-dimensional latent space by a pre-trained adversarial latent autoencoder [44]. Then, we train two HyCNNs by Algorithm 1 on the latent space. In the generation phase, we randomly select three images in each class. These samples are first encoded into a latent space, then mapped by our trained convex gradient, and finally recovered in the original space by the pre-trained decoder.

For all experiments, we set $n_0 = n_1 = 500, \lambda = 1, \epsilon = 0.1, T = 30$, and each HyCNN has a single

hidden layer with 128 units. Figure 7 shows the trajectories of generating the attacked images of all selected samples in the gender classification task. In each subfigure, the first column shows the original image, and the other 12 columns represent the images generated using the convex gradient maps trained for 30 epochs, respectively. As illustrated, the samples we generated underwent meaningful modifications to the original images, such that the generated images gradually acquired visual characteristics associated with the opposite dataset label after a small number of training epochs.



(a) *Female* to *Male*



(b) *Male* to *Female*

Figure 7: The trajectory of image generation between *Female* (Class 0) and *Male* (Class 1). The first column displays the original images, while the following columns show the images generated during 30 epochs.

6 Conclusions and Discussions

As a fundamental method in statistics, robust hypothesis testing addresses noisy data, distributional shifts, and measurement errors in real-world, uncertain environments. This paper develops a generative framework for Sinkhorn distributionally robust hypothesis testing (SDRHT), overcoming the non-continuity of the least-favorable distributions of WDRO-based testing and the computational challenges of existing SDRHT methods. By exploiting a conditional-KL representation of Sinkhorn ambiguity sets and strong duality, this framework reformulates the infinite-dimensional problem as an optimization over parametrized convex potentials. We further establish approximation guarantees for the potentials and their gradients, as well as the distributional

universality of the induced transport maps. Numerical results demonstrate the superior performance and robustness of our method compared to existing methods on tasks with different sample sizes and dimensions.

There are several topics worth investigating for future work. First, it is interesting to extend our method to multi-class hypothesis testing, exploring its closed-form optimal detector and efficient algorithms. Then, since the transport map between distributions is the gradient of a convex potential, directly parameterizing the gradient instead of the potential may yield a better approximation of the transport map. Finally, for distributions with extremely high dimensions, e.g., high-resolution images, it is important to develop methods that efficiently solve transport maps while maintaining the convexity of potentials.

References

[1] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, no. 694-706, pp. 289–337, 1933.

[2] B. C. Levy, *Principles of signal detection and parameter estimation*. Springer Science & Business Media, 2008.

[3] P. Schober and T. R. Vetter, "Two-sample unpaired t-tests in medical research," *Anesthesia & Analgesia*, vol. 129, no. 4, p. 911, 2019.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[5] P. J. Huber, "A robust version of the probability ratio test," *The Annals of Mathematical Statistics*, pp. 1753–1758, 1965.

[6] G. Gül and A. M. Zoubir, "Minimax robust hypothesis testing," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5572–5587, 2017.

[7] D. Kuhn, S. Shafiee, and W. Wiesemann, "Distributionally robust optimization," *Acta Numerica*, vol. 34, pp. 579–804, 2025.

[8] R. Gao, L. Xie, Y. Xie, and H. Xu, "Robust hypothesis testing using Wasserstein uncertainty sets," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[9] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1, pp. 115–166, 2018.

[10] P. J. Huber, *Robust statistical procedures*. SIAM, 1996.

[11] B. C. Levy, "Robust hypothesis testing with a relative entropy tolerance," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 413–421, 2008.

[12] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.

[13] E. Delage and Y. Ye, "Distributionally robust optimization under moment uncertainty with application to data-driven problems," *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010.

[14] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014.

[15] Z. Sun and S. Zou, "A data-driven approach to robust hypothesis testing using kernel MMD uncertainty sets," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 3056–3061.

[16] J.-J. Zhu, W. Jitkrittum, M. Diehl, and B. Schölkopf, "Kernel distributionally robust optimization: Generalized duality theorem and stochastic approximation," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 280–288.

[17] L. Xie, R. Gao, and Y. Xie, "Robust hypothesis testing with Wasserstein uncertainty sets," *arXiv preprint arXiv:2105.14348*, 2021.

[18] J. Wang, R. Gao, and Y. Xie, "Sinkhorn distributionally robust optimization," *Operations Research*, 2025.

[19] F. Zhang and J. Wang, "Contextual distributionally robust optimization with causal and continuous structure: An interpretable and tractable approach," *arXiv preprint arXiv:2601.11016*, 2026.

[20] G. Peyré and M. Cuturi, *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

[21] J. Wang and Y. Xie, "A data-driven approach to robust hypothesis testing using Sinkhorn uncertainty sets," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 3315–3320.

[22] J. Wang, R. Gao, and Y. Xie, "Non-convex robust hypothesis testing using Sinkhorn uncertainty sets," in *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2024, pp. 843–848.

[23] C.-H. Lai, Y. Song, D. Kim, Y. Mitsufuji, and S. Ermon, "The principles of diffusion models," *arXiv preprint arXiv:2510.21890*, 2025.

[24] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv preprint arXiv:2210.02747*, 2022.

[25] T. Dai, D. Simchi-Levi, M. X. Wu, and Y. Xie, "Assured autonomy: How operations research powers and orchestrates generative AI systems," *arXiv preprint arXiv:2512.23978*, 2025.

[26] C. Xu, J. Lee, X. Cheng, and Y. Xie, "Flow-based distributionally robust optimization," *IEEE Journal on Selected Areas in Information Theory*, 2024.

[27] J. Wang, "Iterative sampling methods for sinkhorn distributionally robust optimization," *arXiv preprint arXiv:2512.12550*, 2025.

[28] Z. Xu and J.-J. Zhu, "Gradient flow sampler-based distributionally robust optimization," *arXiv preprint arXiv:2510.25956*, 2025.

[29] L. Zhu and Y. Xie, "Distributionally robust optimization via iterative algorithms in continuous probability spaces," *arXiv preprint arXiv:2412.20556*, 2024.

[30] X. Cheng, Y. Xie, L. Zhu, and Y. Zhu, "Worst-case generation via minimax optimization in Wasserstein space," *arXiv preprint arXiv:2512.08176*, 2025.

[31] A. Nemirovski and A. Shapiro, "Convex approximations of chance-constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007.

[32] S. Chewi, J. Niles-Weed, and P. Rigollet, *Statistical optimal transport*. Springer, 2025.

[33] S. Ubaru, J. Chen, and Y. Saad, "Fast estimation of tr(f(a)) via stochastic lanczos quadrature," *SIAM Journal on Matrix Analysis and Applications*, vol. 38, no. 4, pp. 1075–1099, 2017.

[34] C.-W. Huang, R. T. Chen, C. Tsirigotis, and A. Courville, "Convex potential flows: Universal probability distributions with optimal transport and convex optimization," *arXiv preprint arXiv:2012.05942*, 2020.

[35] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 146–155.

[36] S. Hundrieser, I. Kong, and J. Schmidt-Hieber, "Hyper input convex neural networks for shape constrained learning and optimal transport," *arXiv preprint arXiv:2604.26942*, 2026.

[37] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *International Conference on Machine Learning*. PMLR, 2013, pp. 1319–1327.

[38] M. F. Hutchinson, "A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines," *Communications in Statistics-Simulation and Computation*, vol. 18, no. 3, pp. 1059–1076, 1989.

[39] A.-A. Pooladian and J. Niles-Weed, "Entropic estimation of optimal transport maps," *arXiv preprint arXiv:2109.12004*, 2021.

[40] Y. Chen, Y. Shi, and B. Zhang, "Optimal control via neural networks: A convex approach," *arXiv preprint arXiv:1805.11835*, 2018.

[41] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[42] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for exotic particles in high-energy physics with deep learning," *Nature communications*, vol. 5, no. 1, p. 4308, 2014.

[43] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[44] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto, "Adversarial latent autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 104–14 113.

[45] R. Cescon, A. Martin, and G. Ferrari-Trecate, "Sinkhorn ambiguity sets for distributionally robust control: Convexity, weak compactness, and tractability," *arXiv preprint arXiv:2605.03845*, 2026.

[46] S. Shafiee, L. Aolaritei, F. Dörfler, and D. Kuhn, "Nash equilibria, regularization, and computation in optimal transport-based distributionally robust optimization," *Operations Research*, 2025.

[47] M. D. Donsker and S. S. Varadhan, "Asymptotic evaluation of certain markov process expectations for large time. iv," *Communications on pure and applied mathematics*, vol. 36, no. 2, pp. 183–212, 1983.

[48] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on stochastic programming: modeling and theory*. SIAM, 2021.

Appendix

A Summary of Notations

For integer $K \in \mathbb{Z}_+$, define $[K] := \{1, \cdots, K\}$, which is the set of positive running indices to K. The list of important sets, spaces, and notations used in this paper is shown in Table 8. The list of important functions is shown in Table 9.

Table 8: Notations List

Spaces	**Description**
Ω	Closed sample space, $\Omega \subseteq \mathbb{R}^d$.
$\mathcal{F}$	Functional space, which is not necessarily compact.
Basic Sets	
$\mathbb{K}$	Indices set of hypotheses, $\mathbb{K} = \{0, 1\}$.
$\mathscr{P}(\Omega)$	The set of all probability distributions on Ω.
$\mathcal{M}(\Omega)$	The set of measures on Ω.
$\Gamma(\mathbb{P}, \mathbb{Q})$	Joint distribution set of marginal distribution $\mathbb{P}$ and $\mathbb{Q}$.
$\mathcal{P}_k$	The distribution set under hypotheses H_k, $\mathcal{P}_k \subset \mathscr{P}(\Omega)$ for each $k \in \mathbb{K}$.
$\mathcal{F}_{\text{CVX}}$	Set of convex functions $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$.
Parameters	
ϵ	Entropic regularization parameter in Sinkhorn discrepancy, $\epsilon \geq 0$.
ρ_k	Radius for Sinkhorn-based ambiguity set, for each $k \in \mathbb{K}$.
λ_k	Penalty parameter in soft-constraint problem (soft-SDRHT), $\lambda_k \geq 0$ for each $k \in \mathbb{K}$.
Distributions	
$\widehat{\mathbb{P}}_k$	Empirical distribution with n_k samples for each $k \in \mathbb{K}$.
$\mathcal{K}_{\boldsymbol{x}_k^{(i)}, \epsilon}$	Kernel extension for any sample point $i \in [n_k]$ and each $k \in \mathbb{K}$.
$\mathbb{P}_k^\epsilon$	Mixture of continuous distributions $\mathcal{K}_{\boldsymbol{x}_k^{(i)}, \epsilon}$ for all $i \in [n_k]$, see Eq. (4).
$\widetilde{\mathbb{P}}_k$	Discrete approximation of distribution $\mathbb{P}_k^\epsilon$ with N samples, for each $k \in \mathbb{K}$.
Discrepancies	
$\mathcal{S}_\epsilon(\mathbb{P}, \mathbb{Q})$	Sinkhorn discrepancy from distribution $\mathbb{P}$ to $\mathbb{Q}$, see Definition 2.
$\mathcal{D}_{\text{KL}}(\mathbb{P}, \mathbb{Q})$	Kullback–Leibler divergence (KL divergence) from distribution $\mathbb{P}$ to $\mathbb{Q}$.
Ambiguity Sets	
$\mathbb{B}_{\rho_k}(\widehat{\mathbb{P}}_k)$	Sinkhorn discrepancy-based ambiguity set centered at $\widehat{\mathbb{P}}_k$ for each $k \in \mathbb{K}$, see Eq. (1).
$\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$	Equivalent KL-based ambiguity set of $\mathbb{B}_{\rho_k}(\widehat{\mathbb{P}}_k)$ centered at $\mathbb{P}_k^\epsilon$, see Proposition 1.

Table 9: Functions List

Functions	Description
$c : \Omega \times \Omega \to \mathbb{R}$	Transport cost function satisfying Assumption 1, we choose $c(\boldsymbol{x}, \boldsymbol{z}) := \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2$.
$\ell : \mathbb{R} \to \mathbb{R}_+ \cup \{\infty\}$	Generating function (see Definition 1), we choose $\ell(t) = \exp(t)$.
$\phi : \Omega \to \mathbb{R}$	Detector for hypothesis testing, which is measurable and upper semicontinuous.
$\phi_{\text{opt}} : \Omega \to \mathbb{R}$	Optimal detector for generating function $\ell(t) = \exp(t)$, $\phi_{\text{opt}} = \frac{1}{2}\log(p_0/p_1)$.
$\varphi_1, \varphi_2 : \mathbb{R}^d \to \mathbb{R}$	Convex functions, decision variables in the problem (7).
$\varphi^* : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$	Convex conjugate of a convex function $\varphi : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$.
$\mathsf{HyCNN}_k : \mathbb{R}^d \to \mathbb{R}$	Convex function formed by the HyCNN for each $k \in \mathbb{K}$.
$\widetilde{\varphi}_k : \mathbb{R}^d \to \mathbb{R}$	Strongly convex function, $\widetilde{\varphi}_k(\boldsymbol{z}) = \mathsf{HyCNN}_k(\boldsymbol{z}) + \frac{\alpha}{2}\|\boldsymbol{z}\|_2^2$ for each $k \in \mathbb{K}$ and $\alpha > 0$.
$L_k : \Omega \to \mathbb{R}$	Loss function of a given detector ϕ for sample $\boldsymbol{\omega}$, satisfying Assumption 2.
$\Phi : \mathscr{P}(\Omega)^2 \to \mathbb{R}$	Convex surrogate risk of the detector ϕ.
$g^c_{\epsilon,k}$	Continuous extension for the optimal entropic potential $g_{\epsilon,k,(n_k,N)}$ in Algorithm 4.
$\widehat{\mathcal{S}}_{\epsilon,(n_k,N)}$	Finite sample estimator for the Sinkhorn discrepancy.

B Technical Analyses and Proofs

B.1 Proof of Proposition 1

Proof of Proposition 1. By the definition of Sinkhorn discrepancy, for distributions $\mathbb{P}, \mathbb{Q} \in \mathscr{P}(\Omega)$, it is equivalent to

$$\mathcal{S}_\epsilon(\mathbb{P}, \mathbb{Q}) = \inf_{\gamma \in \Gamma(\mathbb{P}, \mathbb{Q})} \int_{\Omega \times \Omega} \Big(c(\boldsymbol{x}, \boldsymbol{z}) + \epsilon \cdot \log \frac{\mathrm{d}\gamma(\boldsymbol{x}, \boldsymbol{z})}{\mathrm{d}\mathbb{P}(\boldsymbol{x})\mathrm{d}\nu(\boldsymbol{z})} \Big) \mathrm{d}\gamma(\boldsymbol{x}, \boldsymbol{z}). \tag{20}$$

Taking the reference distribution as the empirical distribution $\widehat{\mathbb{P}} \in \mathscr{P}(\Omega)$, Eq. (20) can be further reformulated to

$$\begin{aligned}
\mathcal{S}_\epsilon(\widehat{\mathbb{P}}, \mathbb{Q}) &= \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}, \mathbb{Q})} \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}} \Big[\int_\Omega \Big(c(\boldsymbol{x}, \boldsymbol{z}) + \epsilon \log \frac{\mathrm{d}\gamma_{\boldsymbol{x}}(\boldsymbol{z})}{\mathrm{d}\nu(\boldsymbol{z})} \Big) \mathrm{d}\boldsymbol{z} \Big] \\
&= \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}, \mathbb{Q})} \epsilon \cdot \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}} \Big[\int_\Omega \log \frac{\mathrm{d}\gamma_{\boldsymbol{x}}(\boldsymbol{z})}{e^{-c(\boldsymbol{x}, \boldsymbol{z})} \cdot \mathrm{d}\nu(\boldsymbol{z})} \mathrm{d}\boldsymbol{z} \Big] \\
&= \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}, \mathbb{Q})} \epsilon \cdot \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}} \Big[\int_\Omega \log \frac{\mathrm{d}\gamma_{\boldsymbol{x}}(\boldsymbol{z})}{\mathrm{d}\mathcal{K}_{\boldsymbol{x},\epsilon}(\boldsymbol{z})} \mathrm{d}\boldsymbol{z} - \log \int_{\mathbb{R}^d} e^{-c(\boldsymbol{x}, \boldsymbol{z})/\epsilon} \mathrm{d}\boldsymbol{z} \Big] \\
&= \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}, \mathbb{Q})} \epsilon \cdot \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}} \Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon}) \Big] - \epsilon \cdot \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}} \Big[\log \int_{\mathbb{R}^d} e^{-c(\boldsymbol{x}, \boldsymbol{z})/\epsilon} \mathrm{d}\boldsymbol{z} \Big],
\end{aligned} \tag{21}$$

where

$$\mathrm{d}\mathcal{K}_{\boldsymbol{x},\epsilon}(\boldsymbol{z}) := \frac{e^{-c(\boldsymbol{x}, \boldsymbol{z})/\epsilon}}{\int_{\mathbb{R}^d} e^{-c(\boldsymbol{x}, \boldsymbol{u})/\epsilon} \cdot \mathrm{d}\boldsymbol{u}} \cdot \mathrm{d}\nu(\boldsymbol{z}).$$

Thus, the ambiguity sets in Eq. (1) is equivalent to

$$\inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \epsilon \cdot \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k} \Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon}) \Big] \le \rho_k + \epsilon \cdot \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k} \Big[\log \int_{\mathbb{R}^d} e^{-c(\boldsymbol{x}, \boldsymbol{z})/\epsilon} \mathrm{d}\boldsymbol{z} \Big], \quad \forall k \in \mathbb{K},$$

that is,

$$\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon) := \{\mathbb{Q} \in \mathscr{P}(\Omega) : \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k}\Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon})\Big] \leq \bar{\rho}_k/\epsilon\}, \quad \forall k \in \mathbb{K},$$

where $\bar{\rho}_k := \rho_k + \epsilon \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k}\Big[\log \int_{\mathbb{R}^d} e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon} \mathrm{d}\boldsymbol{z}\Big]$ and probability density $\mathrm{d}\mathbb{P}_k^\epsilon(\boldsymbol{z}) := \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k}[\mathrm{d}\mathcal{K}_{\boldsymbol{x},\epsilon}(\boldsymbol{z})]$. This completes the proof.

□

B.2 Proof of Theorem 1

To prove Theorem 1, we first establish the weak compactness of the Sinkhorn ambiguity sets and the weak upper semicontinuity of the expected loss.

Based on Eq. (3), the following lemma holds.

Lemma 1 (Weak Compactness of the Ambiguity Sets)**.** *Under Assumption 1, the ambiguity set defined in (3) is weakly compact in $\mathcal{P}(\Omega)$ for any closed set $\Omega \subseteq \mathbb{R}^d$, $k \in \mathbb{K}$, and $\bar{\rho}_k \geq 0$.*

The proof of Lemma 1 is provided in the following Appendix B.2.1. While [45] also confirms the weak compactness of Sinkhorn ambiguity sets based on the properties of Wasserstein balls, our proof follows directly from the KL representation in Proposition 1 and holds under more general assumptions.

Lemma 2 (Weak Upper Semicontinuity of the Expected Loss)**.** *Under Assumption 2(I), $\mathbb{E}_{\boldsymbol{\omega} \sim \mathbb{Q}}[L_k(\phi, \boldsymbol{\omega})]$ is weakly upper semicontinuous in $\mathbb{Q} \in \mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for any $k \in \mathbb{K}$ and $\phi \in \mathcal{F}$.*

The proof of Lemma 2 is provided in the following Appendix B.2.2.

B.2.1 Proof of Lemma 1

Proof of Lemma 1. To prove $\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ is weakly compact, we first show its connection with KL-divergence-based ambiguity set. Since the KL-divergence is a convex function, by Jensen's inequality, we have

$$\begin{aligned}\mathcal{D}_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}_k^\epsilon) &\leq \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k}\Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon})\Big] \\ &\leq \frac{\rho_k}{\epsilon} + \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k}\Big[\log \int_{\mathbb{R}^d} e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon} \mathrm{d}\boldsymbol{z}\Big], \quad \forall k \in \mathbb{K},\end{aligned}$$

where probability distribution $\mathbb{Q} \in \mathscr{P}(\Omega)$. Let

$$\mathbb{B}'_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon) := \{\mathbb{Q} : \mathcal{D}_{\mathrm{KL}}(\mathbb{Q} \| \mathbb{P}_k^\epsilon) \leq \bar{\rho}_k/\epsilon\}, \quad \forall k \in \mathbb{K}.$$

According to Proposition 3.12 in [7], the KL-divergence-based ambiguity sets $\mathbb{B}'_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for any $k \in \mathbb{K}$ are weakly compact, for any closed set $\Omega \subseteq \mathbb{R}^d$ and $k \in \mathbb{K}$, distribution $\mathbb{P}_k^\epsilon \in \mathscr{P}(\Omega)$ and $\bar{\rho}_k \geq 0$.

Since $\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon) \subseteq \mathbb{B}'_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for any $k \in \mathbb{K}$, it suffices to prove that $\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ is weakly closed, as weakly closed subsets of weakly compact sets are weakly compact [46, Lemma 1]. Define

$$\mathrm{d}M_{k,\epsilon}(\boldsymbol{x},\boldsymbol{z}) := \mathrm{d}\widehat{\mathbb{P}}_k(\boldsymbol{x}) \cdot \mathrm{d}\mathcal{K}_{\boldsymbol{x},\epsilon}(\boldsymbol{z}) = \frac{e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon}}{\int_{\mathbb{R}^d} e^{-c(\boldsymbol{x},\boldsymbol{u})/\epsilon} \cdot \mathrm{d}\boldsymbol{u}} \cdot \mathrm{d}\widehat{\mathbb{P}}_k(\boldsymbol{x}) \cdot \mathrm{d}v(\boldsymbol{z}), \quad \forall k \in \mathbb{K},$$

According to Eq. (20), we have

$$\inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}\Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon})\Big] = \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \mathcal{D}_{\mathrm{KL}}(\gamma \| M_{k,\epsilon}), \quad \forall k \in \mathbb{K}.$$

According to Proposition 3.12 in [7], to prove the Sinkhorn-based ambiguity set is weakly closed, it suffices to prove that function $f_k(\mathbb{Q}) := \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \mathcal{D}_{\mathrm{KL}}(\gamma \| M_{k,\epsilon})$ is weakly lower semicontinuous in $\mathbb{Q}$ for each $k \in \mathbb{K}$, which implies that any sublevel set of $f(\mathbb{Q})$, i.e., $\{\mathbb{Q} : f(\mathbb{Q}) \leq c\}$ for any $c \in \mathbb{R}$, is weakly closed. Let $\{\mathbb{Q}_j\}_{j\in\mathbb{N}} \subset \mathscr{P}(\Omega)$ be any sequence of distributions that converges weakly to $\mathbb{Q}$. By the definition of infimum, there exists a sequence of couplings $\gamma_n \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q}_n)$ that converges weakly to $\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})$ such that

$$\mathcal{D}_{\mathrm{KL}}(\gamma_n \| M_{k,\epsilon}) \leq f_k(\mathbb{Q}_n) + \frac{1}{n}, \quad \forall k \in \mathbb{K}. \tag{22}$$

Then, for each $k \in \mathbb{K}$, we have

$$f_k(\mathbb{Q}) \leq \mathcal{D}_{\mathrm{KL}}(\gamma \| M_{k,\epsilon}) \leq \liminf_{j\in\mathbb{N}} \mathcal{D}_{\mathrm{KL}}(\gamma_j \| M_{k,\epsilon}) \leq \liminf_{j\in\mathbb{N}} f_k(\mathbb{Q}_j), \quad \forall k \in \mathbb{K},$$

where the second inequality is due to the weak lower semi-continuity of $\mathcal{D}_{\mathrm{KL}}(\gamma \| M_{k,\epsilon})$ in γ [7, Proposition 3.12], and the final inequality is due to Eq. (22). Therefore, for each $k \in \mathbb{K}$, the function $f_k(\mathbb{Q})$ is weakly lower semicontinuous in $\mathbb{Q}$ which implies that the ambiguity set $\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ is weakly closed, and thereby weakly compact. This completes the proof. □

B.2.2 Proof of Lemma 2

Proof of Lemma 2. We prove Lemma 2 in two steps. Specifically, we prove that the loss function L_k is uniformly integrable and $\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\Big[L_k(\phi,\boldsymbol{\omega})\Big]$ is weakly upper semicontinuous with respect to the ambiguity set $\mathbb{Q} \in \mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for each $k \in \mathbb{K}$ and any fixed ϕ, respectively.

Step 1 of Lemma 2: Based on the Donsker-Varadhan variational representation of the KL divergence [47], for any measurable function $f : \Omega \to \mathbb{R}$ and probability distribution $\mathbb{Q}$, we have

$$\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[f] - \log \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}[e^f] \leq \mathcal{D}_{\mathrm{KL}}(\mathbb{Q}, \mathbb{P}_k^\epsilon) \leq \inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{Q})} \mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}\Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon})\Big] \leq \frac{\bar{\rho}_k}{\epsilon}, \quad \forall k \in \mathbb{K}. \tag{23}$$

According to the Donsker-Varadhan variational representation part in relation (23), under Assumption 2(I), for $\beta_k > 0$, we have

$$\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[L_k(\phi,\boldsymbol{\omega})] \leq \frac{1}{\beta}\Big(\mathcal{D}_{\mathrm{KL}}(\mathbb{Q}, \mathbb{P}_k^\epsilon) + \log \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}[e^{\beta_k \cdot L_k(\phi,\boldsymbol{\omega})}]\Big) < \infty, \quad \forall k \in \mathbb{K}, \tag{24}$$

which implies that the inner problem of (SDRHT) is bounded. For a truncation threshold $M > 0$, let $\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}$ be an indicator function that equals to 1 if $L_k(\phi,\boldsymbol{\omega}) > M$ and 0 otherwise. Based on relation (23), we also have

$$\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}] \leq \frac{1}{\beta}\left(\mathcal{D}_{\mathrm{KL}}(\mathbb{Q},\mathbb{P}_k^\epsilon) + \log\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}\left[e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}}\right]\right), \quad \forall k\in\mathbb{K},$$

and it follows that

$$\sup_{\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}] \leq \frac{\bar{\rho}_k}{\epsilon\beta} + \frac{1}{\beta}\log\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}\left[e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}}\right], \quad \forall k\in\mathbb{K}.$$

When $M\to\infty$, we have $\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}\to 0$ for $\mathbb{P}_k^\epsilon$-almost every $\boldsymbol{\omega}$. Moreover, as both β_k and $L_k(\phi,\boldsymbol{\omega})$ are non-negative, we have

$$e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}} \leq e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})}, \quad \forall k\in\mathbb{K},$$

where $e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})}$ is integrable under Assumption 2(I). Hence, by the dominated convergence theorem,

$$\lim_{M\to\infty}\log\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}\left[e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}}\right] = \log\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}\left[e^0\right] = 0, \quad \forall k\in\mathbb{K}.$$

Thus, we have

$$\lim_{M\to\infty}\sup_{\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\left[L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}\right] \leq \frac{\bar{\rho}_k}{\epsilon\beta_k}, \quad \forall k\in\mathbb{K}. \tag{25}$$

Since Assumption 2(I) guarantees that this property holds for any $\beta_k > 0$, let $\beta_k\to\infty$, and then we have

$$\lim_{M\to\infty}\sup_{\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\left[L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}\right] \leq \inf_{\beta>0}\frac{\bar{\rho}_k}{\epsilon\beta_k} = 0, \quad \forall k\in\mathbb{K}, \tag{26}$$

which completes the proof of uniform integrability.

Step 2 of Lemma 2: We next prove the weak upper semicontinuity of $\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[L_k(\phi,\boldsymbol{\omega})]$ in $\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$. We consider a continuous detector ϕ and a non-negative, non-decreasing, convex, and continuous generating function ℓ. Then, the loss functions L_0 and L_1 are non-negative and upper semicontinuous for all $k\in\mathbb{K}$. By Lemma 1, the ambiguity set $\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ is weakly closed. For each $k\in\mathbb{K}$, let $\mathbb{Q}_j\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for all $j\in\mathbb{N}$ be any sequence of distributions that converges weakly to $\mathbb{Q}$, and then we have

$$\begin{aligned}
&\limsup_{j\in N}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}_j}\left[L_k(\phi,\boldsymbol{\omega})\right] \\
&\quad\leq \limsup_{j\in N}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}_j}\left[\min\{L_k(\phi,\boldsymbol{\omega}),M\}\right] + \limsup_{j\in N}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}_j}\left[(L_k(\phi,\boldsymbol{\omega})-M)\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}\right] \\
&\quad\leq \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\left[\min\{L_k(\phi,\boldsymbol{\omega}),M\}\right] + \limsup_{j\in N}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}_j}\left[(L_k(\phi,\boldsymbol{\omega})-M)\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}\right] \\
&\quad\leq \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\left[\min\{L_k(\phi,\boldsymbol{\omega}),M\}\right] + \sup_{\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)}\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}], \quad \forall k\in\mathbb{K},
\end{aligned} \tag{27}$$

where $M > 0$, the second inequality is due to Proposition 3.3 in [7] for the upper semicontinuous and bounded function $\min\{L_k(\phi,\boldsymbol{\omega}),M\}$, and the third inequality follows because $\mathbb{Q}_j\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for

any $j \in \mathbb{N}$ and $(L_k(\phi,\boldsymbol{\omega}) - M) \cdot \mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}} \le L_k(\phi,\boldsymbol{\omega}) \cdot \mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}$. Taking the limit as $M \to \infty$ on both sides of inequality (27), we obtain

$$\begin{aligned}
&\limsup_{j\in N} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}_j}\Big[L_k(\phi,\boldsymbol{\omega})\Big] \\
&\quad\le \lim_{M\to\infty} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\Big[\min\{L_k(\phi,\boldsymbol{\omega}),M\}\Big] + \lim_{M\to\infty} \sup_{\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}[L_k(\phi,\boldsymbol{\omega})\cdot\mathbb{1}_{\{L_k(\phi,\boldsymbol{\omega})>M\}}] \\
&\quad= \lim_{M\to\infty} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\Big[\min\{L_k(\phi,\boldsymbol{\omega}),M\}\Big] = \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\Big[L_k(\phi,\boldsymbol{\omega})\Big], \quad \forall k\in\mathbb{K},
\end{aligned}$$

where the first equality follows from Eq. (26) and the second equality follows from the monotone convergence theorem. This completes the proof. □

B.2.3 Formal Proof of Theorem 1

Proof of Theorem 1. For Theorem 1(I), the problem (SDRHT) can be reformulated as

$$\inf_{\phi\in\mathcal{F}} \sup_{\mathbb{P}_k\in\mathbb{B}_{\rho_k}(\widehat{\mathbb{P}}_k),\forall k\in\mathbb{K}} \sum_{k\in\mathbb{K}} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k}\Big[L_k(\phi,\boldsymbol{\omega})\Big].$$

Note that $\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ is weakly compact for any $\bar{\rho}_k \ge 0$ thanks to Proposition 1 and Lemma 1, which applies because of Assumption 1. In addition, $\mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{Q}}\Big[L_k(\phi,\boldsymbol{\omega})\Big]$ is non-negative and weakly upper semicontinuous in $\mathbb{Q}\in\mathbb{B}_{\bar{\rho}_k}(\mathbb{P}_k^\epsilon)$ for all $k\in\mathbb{K}$ and $\phi\in\mathcal{F}$, due to Definition 1 and Lemma 2. As $\Phi(\phi;\mathbb{P}_0,\mathbb{P}_1)$ is convex in ϕ and concave in distributions, Theorem 1(II) in [46], an extension of the lopsided minimax theorem [7, Theorem 5.15], ensures that

$$\text{Optimal Value of (SDRHT)} = \max_{\mathbb{P}_0\in\mathbb{B}_{\rho_0}(\widehat{\mathbb{P}}_0),\mathbb{P}_1\in\mathbb{B}_{\rho_1}(\widehat{\mathbb{P}}_1)} \inf_{\phi\in\mathcal{F}} \Phi(\phi;\mathbb{P}_0,\mathbb{P}_1).$$

For Theorem 1(II), the problem (soft-SDRHT) can be reformulated as

$$\inf_{\phi\in\mathcal{F}} \sup_{\mathbb{P}_0,\mathbb{P}_1\in\mathscr{P}(\Omega)} \sum_{k\in\mathbb{K}} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k}\Big[L_k(\phi,\boldsymbol{\omega})\Big] - \lambda_k \mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k,\mathbb{P}_k),$$

where $\lambda_k > 0$ for any $k\in\mathbb{K}$. Define

$$H(\phi,\mathbb{P}_0,\mathbb{P}_1) := \sum_{k\in\mathbb{K}} \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k}\Big[L_k(\phi,\boldsymbol{\omega})\Big] - \lambda_k \mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k,\mathbb{P}_k).$$

We next verify the level compactness of $H(\phi,\mathbb{P}_0,\mathbb{P}_1)$ required for the lopsided minimax theorem [7, Theorem 5.15]. By Eqs. (21), (23), and (24), for any fixed ϕ, we have

$$H(\phi,\mathbb{P}_0,\mathbb{P}_1) \le \sum_{k\in\mathbb{K}} \Big[C_k + \Big(\frac{1}{\beta_k} - \lambda_k\epsilon\Big)\cdot \inf_{\gamma\in\Gamma(\widehat{\mathbb{P}}_k,\mathbb{P}_k)} \mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}}\|\mathcal{K}_{\boldsymbol{x},\epsilon})]\Big], \tag{28}$$

where $\beta_k > 0$ is finite, $C_k := \frac{1}{\beta_k}\log \mathbb{E}_{\boldsymbol{\omega}\sim\mathbb{P}_k^\epsilon}[e^{\beta_k\cdot L_k(\phi,\boldsymbol{\omega})}] + \lambda_k\epsilon\cdot\mathbb{E}_{\boldsymbol{x}\sim\widehat{\mathbb{P}}_k}[\log\int_{\mathbb{R}^d} e^{-c(\boldsymbol{x},\boldsymbol{z})/\epsilon}\mathrm{d}\boldsymbol{z}]$ for any $k\in\mathbb{K}$ is finite by Assumptions 1(II) and 2(I), and is independent of $\mathbb{P}_0$ and $\mathbb{P}_1$.

By choosing $\beta_k \in (\frac{1}{\lambda_k \epsilon}, +\infty)$ for each $k \in \mathbb{K}$, when $\inf_{\gamma \in \Gamma(\widehat{\mathbb{P}}_k, \mathbb{P}_k)} \mathbb{E}_{\boldsymbol{x} \sim \widehat{\mathbb{P}}_k}\Big[\mathcal{D}_{\mathrm{KL}}(\gamma_{\boldsymbol{x}} \| \mathcal{K}_{\boldsymbol{x},\epsilon})\Big] \to \infty$, we have $H(\phi, \mathbb{P}_0, \mathbb{P}_1) \to -\infty$. This implies that for any $\alpha \in \mathbb{R}$, there exist finite radii r_0 and r_1 such that the upper-level set of $H(\phi, \mathbb{P}_0, \mathbb{P}_1)$ satisfies

$$\{(\mathbb{P}_0, \mathbb{P}_1) : H(\phi, \mathbb{P}_0, \mathbb{P}_1) \geq \alpha\} \subseteq \mathbb{B}_{r_0}(\widehat{\mathbb{P}}_0) \times \mathbb{B}_{r_1}(\widehat{\mathbb{P}}_1).$$

By Lemma 1, the set $\mathbb{B}_{r_0}(\widehat{\mathbb{P}}_0) \times \mathbb{B}_{r_1}(\widehat{\mathbb{P}}_1)$ is weakly compact. For each $k \in \mathbb{K}$, the Sinkhorn discrepancy is convex and weakly lower semicontinuous (see the proof of Lemma 1), and $\mathbb{E}_{\boldsymbol{\omega} \sim \mathbb{P}_k}\Big[L_k(\phi, \boldsymbol{\omega})\Big]$ is weakly upper semicontinuous in $\mathbb{P}_k$ by Lemma 2. Hence, $H(\phi, \mathbb{P}_0, \mathbb{P}_1)$ is weakly upper semicontinuous and its upper-level set $\{(\mathbb{P}_0, \mathbb{P}_1) : H(\phi, \mathbb{P}_0, \mathbb{P}_1) \geq \alpha\}$ is weakly closed. Therefore, for every $\alpha \in \mathbb{R}$ and every fixed $\phi \in \mathcal{F}$, the upper-level set $\{(\mathbb{P}_0, \mathbb{P}_1) : H(\phi, \mathbb{P}_0, \mathbb{P}_1) \geq \alpha\}$, a closed subset of a weakly compact set, is also weakly compact. In addition, $-H(\phi, \mathbb{P}_0, \mathbb{P}_1)$ is convex, weakly lower semicontinuous, and thus weakly closed in $\mathbb{P}_0$ and $\mathbb{P}_1$.

Finally, $H(\cdot, \mathbb{P}_0, \mathbb{P}_1)$ is convex in ϕ, and Assumption 2(II) ensures that

$$\sup_{\mathbb{P}_0, \mathbb{P}_1} \inf_{\phi \in \mathcal{F}} \quad H(\phi, \mathbb{P}_0, \mathbb{P}_1) > -\infty.$$

Therefore, as all conditions of the lopsided minimax theorem are satisfied, the strong duality of the problem (soft-SDRHT) holds, i.e.,

$$\text{Optimal Value of (soft-SDRHT)} = \sup_{\mathbb{P}_0, \mathbb{P}_1 \in \mathscr{P}(\Omega)} \inf_{\phi \in \mathcal{F}} \Phi(\phi; \mathbb{P}_0, \mathbb{P}_1) - \sum_{k \in \mathbb{K}} \lambda_k \mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k, \mathbb{P}_k). \tag{29}$$

Further, since $\inf_{\phi \in \mathcal{F}} H(\phi, \mathbb{P}_0, \mathbb{P}_1)$ is weakly upper semicontinuous in $\mathbb{P}_0$ and $\mathbb{P}_1$, has a finite upper bound by Eq. (28) under proper β, and has weakly compact upper-level sets, the supremum in Eq. (29) is attained by Weierstrass' theorem, i.e.,

$$\text{Optimal Value of (soft-SDRHT)} = \max_{\mathbb{P}_0, \mathbb{P}_1 \in \mathscr{P}(\Omega)} \inf_{\phi \in \mathcal{F}} \Phi(\phi; \mathbb{P}_0, \mathbb{P}_1) - \sum_{k \in \mathbb{K}} \lambda_k \mathcal{S}_\epsilon(\widehat{\mathbb{P}}_k, \mathbb{P}_k).$$

This completes the proof. □

B.3 Proof of Theorem 3

Proof of Theorem 3. As a preparation, when $\mathbb{D}$ has a finite diameter D, for any $\upsilon \in (0,1)$, there exists a υ-net $\mathbb{D}_\upsilon \subset \mathbb{D}$ containing J points such that $\forall \boldsymbol{x} \in \mathbb{D}$, $\exists \boldsymbol{x}_j \in \mathbb{D}_\upsilon$, $\|\boldsymbol{x} - \boldsymbol{x}_j\| \leq \upsilon$. According to [48], the size of the υ-net is bounded by $J \leq \mathcal{O}((D/\upsilon)^d)$.

We prove Theorem 3(I) in two steps. According to Lemma 1 in [40], all continuous Lipschitz convex functions over convex compact sets can be approximated using the maximum of affine functions. Thus, in Step 1, we control the discretization error by adjusting the size J of the υ-net. In Step 2, we prove that a HyCNN with maxout activation functions can exactly represent a maximum of affine functions and determine the quantitative relationship between J and the network hyperparameters (L, h).

Step 1 of Theorem 3(I): For any L_f-Lipschitz function f, we approximate it by the maximum of supporting hyperplanes at points in the υ-net. The supporting hyperplane at $(\boldsymbol{x}_j, f(\boldsymbol{x}_j))$ is defined as

$$s_j(\boldsymbol{x}) := \boldsymbol{g}_j^\top(\boldsymbol{x} - \boldsymbol{x}_j) + f(\boldsymbol{x}_j).$$

where $\boldsymbol{g}_j \in \partial f(\boldsymbol{x}_j)$. For any $\boldsymbol{x} \in \mathbb{D}$, let $\boldsymbol{x}_{j_0} \in \mathbb{D}_\upsilon$ be the nearest point to $\boldsymbol{x}$ in the υ-net. The pointwise error satisfies

$$\begin{aligned}
\sup_{\boldsymbol{x}\in\mathbb{D}} \left| f(\boldsymbol{x}) - \max_{j\in[J]} s_j(\boldsymbol{x}) \right| &= \sup_{\boldsymbol{x}\in\mathbb{D}} \left(f(\boldsymbol{x}) - \max_{j\in[J]} s_j(\boldsymbol{x}) \right) \\
&\le \sup_{\boldsymbol{x}\in\mathbb{D}} \left(f(\boldsymbol{x}) - s_{j_0}(\boldsymbol{x}) \right) \\
&= \sup_{\boldsymbol{x}\in\mathbb{D}} \left(f(\boldsymbol{x}) - f(\boldsymbol{x}_{j_0}) - \boldsymbol{g}_j^\top(\boldsymbol{x} - \boldsymbol{x}_{j_0}) \right) \\
&\le \sup_{\boldsymbol{x}\in\mathbb{D}} \left(|f(\boldsymbol{x}) - f(\boldsymbol{x}_{j_0})| + \|\boldsymbol{g}_j\| \cdot \|\boldsymbol{x} - \boldsymbol{x}_{j_0}\| \right) \\
&\le 2L_f \|\boldsymbol{x} - \boldsymbol{x}_{j_0}\| \le 2L_f \upsilon.
\end{aligned}$$

Here, the first equality holds since f is convex, implying $s_j(\boldsymbol{x}) \le f(\boldsymbol{x})$ for all j. The first inequality follows because $\max_{j\in[J]} s_j(\boldsymbol{x}) \ge s_{j_0}(\boldsymbol{x})$. The second inequality uses the triangle inequality and Cauchy-Schwarz inequality. The third inequality applies the L_f-Lipschitz continuity of f, which bounds both $|f(\boldsymbol{x}) - f(\boldsymbol{x}_{j_0})| \le L_f \|\boldsymbol{x} - \boldsymbol{x}_{j_0}\|$ and $\|\boldsymbol{g}_j\| \le L_f$. The last inequality holds as $\|\boldsymbol{x} - \boldsymbol{x}_{j_0}\| \le \upsilon$. To ensure $\sup_{\boldsymbol{x}\in\mathbb{D}} |f(\boldsymbol{x}) - \max_{j\in[J]} s_j(\boldsymbol{x})| \le \varepsilon$ for any $\varepsilon > 0$, we have $\upsilon \le \min\{\frac{\varepsilon}{2L_f}, 1\}$. Taking $\upsilon = \varepsilon/2L_f$. By $J \le \mathcal{O}((\frac{D}{\upsilon})^d)$, the number of points in the υ-net is given by $J \le \mathcal{O}((\frac{DL_f}{\varepsilon})^d)$.

Step 2 of Theorem 3(I): Here, we show that the maximum of J affine functions can be represented by a HyCNN with L layers and width h. As an intuitive example, we first consider a maximum of two affine functions and reformulate it as a HyCNN with two layers $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ (see Figure 2) by introducing an extra affine term:

$$\max\left\{ \boldsymbol{a}_1^\top \boldsymbol{x} + b_1, \boldsymbol{a}_2^\top \boldsymbol{x} + b_2 \right\} = \underbrace{(\boldsymbol{a}_3^\top \boldsymbol{x} + b_3) + \overbrace{\max\left\{ (\boldsymbol{a}_1 - \boldsymbol{a}_3)^\top \boldsymbol{x} + (b_1 - b_3), (\boldsymbol{a}_2 - \boldsymbol{a}_3)^\top \boldsymbol{x} + (b_2 - b_3) \right\}}^{\boldsymbol{y}_1}}_{\boldsymbol{y}_2}$$

We extend this intuition to a maximum of J affine functions $\max\{\boldsymbol{a}_1^\top \boldsymbol{x} + b_1, \cdots, \boldsymbol{a}_J^\top \boldsymbol{x} + b_J\}$. We consider a simple case in which the width of each hidden layer is $h = 1$. In the following, we first construct a HyCNN with maxout activation functions that use both positive and negative weights, representing a maximum over affine functions. Then, all weights can be restricted to be nonnegative by the method in the proof of Theorem 1 in [40] without changing the width and depth of the network. Let $s_j := \boldsymbol{a}_j^\top \boldsymbol{x} + b_j$ for any $j \in [2J]$, extra affine terms $r_j := \boldsymbol{c}_j^\top \boldsymbol{x} + d_j$ for any

$j \in [2,\cdots,J]$, $s'_j = s_j - r_j$ for any $j \in [2,\cdots,J]$, and $\widehat{\sigma}: \mathbb{R}^2 \to \mathbb{R}$ be a maxout activation function. Then

$$\begin{aligned}
&\max\{s_1,\cdots,s_J\}\\
=&\max\{\max\{s_1,\cdots,s_{J-1}\},s_J\}\\
=&r_J+\widehat{\sigma}(\max\{s_1,\cdots,s_{J-1}\}-r_J,s'_J)\\
=&r_J+\widehat{\sigma}(\max\{\max\{s_1,\cdots,s_{J-2}\},s_{J-1}\}-r_J,s'_J)\\
=&r_J+\widehat{\sigma}(r_{J-1}+\widehat{\sigma}(\max\{s_1,\cdots,s_{J-2}\}-r_{J-1},s'_{J-1})-r_J,s'_J)\\
=&\cdots\\
=&r_J+\widehat{\sigma}(r_{J-1}+\widehat{\sigma}(r_{J-2}+\widehat{\sigma}(\cdots+r_3+\widehat{\sigma}(r_2+\widehat{\sigma}(s_1-r_2,s'_2)-r_3,s'_3)-\cdots)-r_{J-2},s'_{J-1})-r_J,s'_J).
\end{aligned} \tag{30}$$

The last equation describes a J-layer HyCNN with width $h=1$, where the layers are

$$\begin{aligned}
y_1 &= \widehat{\sigma}(s_1-r_2,s_2-r_2),\\
y_2 &= \widehat{\sigma}(1\cdot y_1+r_2-r_3,0\cdot y_1+s_3-r_3),\\
&\cdots\\
y_j &= \widehat{\sigma}(1\cdot y_{j-1}+r_j-r_{j+1},0\cdot y_{j-1}+s_{j+1}-r_{j+1}),\\
&\cdots\\
y_J &= r_J+y_{J-1}.
\end{aligned}$$

To ensure the coefficients of previous layers in this neural network are nonnegative, following [40], we expand the original input $\boldsymbol{x}\in\mathbb{R}^d$ in each layer to $\widehat{\boldsymbol{x}}\in\mathbb{R}^{2d}$ that includes $\boldsymbol{x}$ and $-\boldsymbol{x}$, i.e., $\widehat{\boldsymbol{x}} = [\boldsymbol{x}^\top,-\boldsymbol{x}^\top]^\top$. When all coefficients of $\boldsymbol{x}$ are nonnegative, the coefficients of $-\boldsymbol{x}$ are 0. When $h_j<0$, we set the coefficient of $\widehat{x}_{j+d}$ to $-h_j$ and that of x_j to 0. Therefore, without loss of generality, we can limit all weights between layers to be nonnegative, thereby ensuring that the neural network is input-convex.

The construction in Eq. (30) represents $\max_{j\in[J]} s_j(\boldsymbol{x})$ exactly by a HyCNN with width $h=1$ and depth $L=\mathcal{O}(J)$. By Step 1, choosing $h=1$ and $L=\mathcal{O}(J)$ gives

$$L\cdot h = \mathcal{O}((\frac{DL_f}{\varepsilon})^d),$$

which proves Theorem 3(I).

For Theorem 3(II), we first focus on the property of the smooth maxout activation function. Without loss of generality, suppose $a_1\geq a_2$, we have $\bar{\sigma}(a_1,a_2)=a_1$ and

$$\bar{\sigma}_\tau(a_1,a_2)=\tau\log\left(e^{a_1/\tau}(1+e^{(a_2-a_1)/\tau})\right)=a_1+\tau\log\left(1+e^{(a_2-a_1)/\tau}\right)\leq a_1+\tau\log 2,$$

where $\tau>0$ and the last inequality is due to $e^{(a_2-a_1)/\tau}\in(0,1]$. This is equivalent to

$$\bar{\sigma}(a_1,a_2)<\bar{\sigma}_\tau(a_1,a_2)\leq\bar{\sigma}(a_1,a_2)+\tau\log 2.$$

Consequently, for a HyCNN and its smooth version, we have

$$\sup_{\boldsymbol{x}\in\mathbb{D}}|\widehat{\mathsf{HyCNN}}^\tau(\boldsymbol{x})-\widehat{\mathsf{HyCNN}}(\boldsymbol{x})|\leq C\cdot\tau\log 2$$

where $C > 0$ is a constant depending on the number of maxout activation functions and layers in the network. By Theorem 3(I), for any $\varepsilon > 0$, there exists a HyCNN such that

$$\sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathrm{HyCNN}}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{\varepsilon}{2}.$$

By selecting a sufficiently small τ, e.g., $\tau \leq \frac{\varepsilon}{2C\log 2}$, we have

$$\sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathrm{HyCNN}}^{\tau}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathrm{HyCNN}}^{\tau}(\boldsymbol{x}) - \widehat{\mathrm{HyCNN}}(\boldsymbol{x})| + \sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathrm{HyCNN}}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \varepsilon.$$

Note that the smooth HyCNN and standard HyCNN share the same order of network architecture complexity. Hence, the hyperparameters of smooth HyCNN can also be chosen as $L\cdot h = \mathcal{O}((\frac{DL_f}{\varepsilon})^d)$ to guarantee the ε-approximation. This completes the proof of Theorem 3(II).

For Theorem 3(III), we select a sequence of smooth HyCNNs on a compact domain $[-n,n]^d$ with sufficiently small τ and each satisfying $L_n \cdot h_n = \mathcal{O}((\frac{DL_f}{\varepsilon})^d)$. By setting $\varepsilon_n = 1/n$, according to Theorem 3(II), we have $\sup_{\boldsymbol{x}\in\mathbb{D}} |\widehat{\mathrm{HyCNN}}^{\tau}_{\varepsilon_n}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \frac{1}{n}$ where the right-hand-side term converges to 0 as $n \to \infty$. Hence, we can prove $\widehat{\mathrm{HyCNN}}^{\tau}_n$ converges to f uniformly on $\mathbb{D}$. This completes the proof. □

B.4 Proof of Theorem 4

Proof of Theorem 4. For Theorem 4(I), one can see Theorem 2 in [34] for a detailed proof.

For Theorem 4(II), let the function f be M-Lipschitz smooth, and then we have

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \leq \nabla f(\boldsymbol{x})^{\top}(\boldsymbol{y}-\boldsymbol{x}) + \frac{M}{2}\cdot\|\boldsymbol{y}-\boldsymbol{x}\|^2. \tag{31}$$

For any $\widehat{\mathrm{HyCNN}}^{\tau}_n$ in the sequence $\{\widehat{\mathrm{HyCNN}}^{\tau}_n\}_{n=1}^{\infty}$ with hyperparameters satisfying $L_n \cdot h_n = \mathcal{O}({\varepsilon_n}^{-d})$ for any $\varepsilon_n > 0$, we have

$$\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{y}) - \widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) \geq \nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x})^{\top}(\boldsymbol{y}-\boldsymbol{x}), \quad \forall \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d. \tag{32}$$

By Eqs. (31) and (32), we have

$$\left(\nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\right)^{\top}(\boldsymbol{y}-\boldsymbol{x}) \leq \left(\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{y}) - f(\boldsymbol{y})\right) - \left(\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - f(\boldsymbol{x})\right) + \frac{M}{2}\cdot\|\boldsymbol{y}-\boldsymbol{x}\|^2.$$

Let $\boldsymbol{y}$ be obtained by moving $\boldsymbol{x}$ along the direction $\nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - \nabla f(\boldsymbol{x})$ with a proper step length η, i.e., $\boldsymbol{y}-\boldsymbol{x} = \eta\frac{\nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - \nabla f(\boldsymbol{x})}{\|\nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|}$ and $\boldsymbol{y}\in\mathbb{D}$. Then, for the point-wise supremum value of $\nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - \nabla f(\boldsymbol{x})$ on compact set $\mathbb{D}\subset\mathbb{R}^d$, we have

$$\begin{aligned}\sup_{\boldsymbol{x}\in\mathbb{D}}\left\|\nabla\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\right\| &\leq \frac{1}{\eta}\cdot\sup_{\boldsymbol{x}\in\mathbb{D}}\left(\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{y}) - f(\boldsymbol{y}) - \left(\widehat{\mathrm{HyCNN}}^{\tau}_n(\boldsymbol{x}) - f(\boldsymbol{x})\right)\right) + \frac{M}{2\eta}\cdot\|\boldsymbol{y}-\boldsymbol{x}\|^2 \\ &\leq \frac{2\varepsilon_n}{\eta} + \frac{M\eta}{2} \leq 2\sqrt{M\varepsilon_n} = \mathcal{O}(\sqrt{\varepsilon_n}),\end{aligned}$$

where the second inequality is due to Theorem 3(II) and $\|\boldsymbol{y}-\boldsymbol{x}\|^2 = \eta^2$, the last inequality is due to the basic inequality, and the last equality is due to $L_n\cdot h_n = \mathcal{O}({\varepsilon_n}^{-d})$ for any $\varepsilon_n > 0$. Hence, the gradient sequence $\nabla\widehat{\mathrm{HyCNN}}_n$ converges to ∇f uniformly at a rate of $\mathcal{O}\left(\sqrt{\varepsilon_n}\right)$ when function f is Lipschitz smooth. This completes the proof. □

B.5 Proof of Theorem 5

Proof of Theorem 5. Since μ is absolutely continuous and both μ and ν have finite second moments, Brenier's theorem yields a finite convex function $f : \mathbb{R}^d \to \mathbb{R}$ such that

$$(\nabla f)_{\#}\mu = \nu.$$

Since every finite convex function on $\mathbb{R}^d$ is locally Lipschitz, for every $n \in \mathbb{N}$, f is Lipschitz on a neighborhood of the compact set

$$\mathbb{D}_n = [-n, n]^d.$$

By Theorem 3 (II), there exists a smooth HyCNN $\widehat{\mathsf{HyCNN}}_n^{\tau_n}$ satisfying

$$\sup_{\boldsymbol{x}\in\mathbb{D}_n} \left|\widehat{\mathsf{HyCNN}}_n^{\tau_n}(\boldsymbol{x}) - f(\boldsymbol{x})\right| \le \frac{1}{n}.$$

For any fixed compact set $\mathbb{K} \subset \mathbb{R}^d$, there exists N such that $\mathbb{K} \subseteq \mathbb{D}_n$ for every $n \ge N$. Hence,

$$\sup_{\boldsymbol{x}\in\mathbb{K}} \left|\widehat{\mathsf{HyCNN}}_n^{\tau_n}(\boldsymbol{x}) - f(\boldsymbol{x})\right| \le \frac{1}{n} \to 0.$$

Thus, $\widehat{\mathsf{HyCNN}}_n^{\tau_n} \to f$ locally uniformly, and in particular pointwise on $\mathbb{R}^d$.

By Theorem 4 (I),

$$\nabla\widehat{\mathsf{HyCNN}}_n^{\tau_n}(\boldsymbol{x}) \to \nabla f(\boldsymbol{x})$$

at almost every $\boldsymbol{x} \in \mathbb{R}^d$. Since μ is absolutely continuous with respect to the Lebesgue measure, this implies the weak convergence of the pushforward measure holds μ-almost everywhere, i.e., for $\boldsymbol{X} \sim \mu$,

$$\nabla\widehat{\mathsf{HyCNN}}_n^{\tau_n}(\boldsymbol{X}) \to \nabla f(\boldsymbol{X}) \qquad \text{almost surely.}$$

Consequently,

$$(\nabla\widehat{\mathsf{HyCNN}}_n^{\tau_n})_{\#}\mu \Rightarrow (\nabla f)_{\#}\mu = \nu,$$

where $\Rightarrow$ denotes weak convergence of probability measures on $\mathbb{R}^d$. This completes the proof. □

C Technical Note for Training and Testing HyCNN

In this section, we provide details to compute the gradients of the problem (7) in Algorithm 1.

C.1 Gradient of the Risk Function Estimation

In this subsection, we take the $k = 0$ term in Eq. (6) as an example. For brevity, we omit the superscript (i). Denote the parameters of the neural network HyCNN_k as $\boldsymbol{\theta}_k$ for each $k \in \mathbb{K}$. Since our approximated convex function $\widetilde{\varphi}_k$ is strongly convex, we have $\det \nabla^2\widetilde{\varphi}_k(\boldsymbol{z}) > 0$ for any k and $\boldsymbol{z}$.

(I) **Gradient with respect to $\boldsymbol{\theta}_0$.**

For brevity, let $h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1) := \exp\left[\frac{1}{2}(\log p_1(\boldsymbol{\omega}_0) - \log p_0(\boldsymbol{\omega}_0))\right]$, which is the same as $h_0(\boldsymbol{\omega}_0,\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)$ in Algorithm 3. Then, we have

$$\frac{\partial h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_0} = \frac{h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{2}\cdot\left[\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_0} - \frac{\partial \log p_0(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_0}\right]. \tag{33}$$

For the second item on the right-hand side of Eq. (33), based on Eq. (8), we have

$$\frac{\partial \log p_0(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_0} = \frac{\partial}{\partial \boldsymbol{\theta}_0}\log p_0^{\epsilon}(\boldsymbol{z}_0) - \frac{\partial}{\partial \boldsymbol{\theta}_0}\log\det\nabla^2\widetilde{\varphi}_0(\boldsymbol{z}_0) = -\frac{\partial}{\partial \boldsymbol{\theta}_0}\log\det\nabla^2\widetilde{\varphi}_0(\boldsymbol{z}_0), \tag{34}$$

where $\boldsymbol{z}_0 = \nabla\varphi_0^{-1}(\boldsymbol{\omega}_0)$, the first equality is due to the strong convexity of $\widetilde{\varphi}_0$, and the second equality is due to $\frac{\partial}{\partial \boldsymbol{\theta}_0}\log p_0^{\epsilon}(\boldsymbol{z}_0) = 0$. The log-determinant-Hessian in Eq. (34) can be efficiently computed by Algorithm 2.

For the first item on the right-hand side of Eq. (33), based on the chain rule, we have

$$\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_0} = \left(\frac{\partial \boldsymbol{\omega}_0}{\partial \boldsymbol{\theta}_0}\right)^{\top}\left(\frac{\partial \bar{\boldsymbol{z}}_0}{\partial \boldsymbol{\omega}_0}\right)^{\top}\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \bar{\boldsymbol{z}}_0}. \tag{35}$$

Let vector

$$\boldsymbol{g} := \frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \bar{\boldsymbol{z}}_0} = \frac{\partial\left[\log p_1^{\epsilon}(\bar{\boldsymbol{z}}_0) - \log\det\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)\right]}{\partial \bar{\boldsymbol{z}}_0},$$

and this derivative can be computed by automatic differentiation. Let $\bar{\boldsymbol{z}}_0 = \nabla\widetilde{\varphi}_1^{-1}(\boldsymbol{\omega}_0)$. As $\frac{\partial \bar{\boldsymbol{z}}_0}{\partial \boldsymbol{\omega}_0} = [\nabla^2\varphi_1(\bar{\boldsymbol{z}}_0)]^{-1}$, Eq.(35) can be reformulated as

$$\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_0} = \left(\frac{\partial \boldsymbol{\omega}_0}{\partial \boldsymbol{\theta}_0}\right)^{\top}[\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)]^{-1}\boldsymbol{g}, \tag{36}$$

where $[\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)]^{-1}\boldsymbol{g}$ can be efficiently solved as the minimizer of a quadratic optimization problem similar to (15) without explicitly computing the inverse of the Hessian matrix. Then, $\left(\frac{\partial \boldsymbol{\omega}_0}{\partial \boldsymbol{\theta}_0}\right)^{\top}[\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)]^{-1}\boldsymbol{g}$ can be computed by vector-Jacobian product in an automatic differentiation framework. In summary, the gradient of $h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)$ with respect to $\boldsymbol{\theta}_0$ is given by

$$\frac{\partial h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_0} = \frac{h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{2}\cdot\left[\left(\frac{\partial \boldsymbol{\omega}_0}{\partial \boldsymbol{\theta}_0}\right)^{\top}[\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)]^{-1}\boldsymbol{g} + \frac{\partial}{\partial \boldsymbol{\theta}_0}\log\det\nabla^2\widetilde{\varphi}_0(\boldsymbol{z}_0)\right].$$

(II) **Gradient with respect to $\boldsymbol{\theta}_1$.**

Similar to Eq. (33), we have

$$\frac{\partial h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} = \frac{h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{2}\cdot\left[\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_1} - \frac{\partial \log p_0(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_1}\right] = \frac{h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{2}\cdot\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_1}. \tag{37}$$

Since $\bar{\boldsymbol{z}}_0 = \nabla\widetilde{\varphi}_1^{-1}(\boldsymbol{\omega}_0)$, we have

$$\begin{aligned}\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_1} &= \frac{\partial}{\partial \boldsymbol{\theta}_1}\left[\log p_1^{\epsilon}(\bar{\boldsymbol{z}}_0) - \log\det\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)\right] \\ &= \left(\frac{\partial \bar{\boldsymbol{z}}_0}{\partial \boldsymbol{\theta}_1}\right)^{\top}\frac{\partial \log p_1^{\epsilon}(\bar{\boldsymbol{z}}_0) - \log\det\nabla^2\widetilde{\varphi}_1(\bar{\boldsymbol{z}}_0)}{\partial \bar{\boldsymbol{z}}_0} - \frac{\partial}{\partial \boldsymbol{\theta}_1}\log\det\nabla^2\widetilde{\varphi}_1(\boldsymbol{z})\Big|_{\boldsymbol{z}=\bar{\boldsymbol{z}}_0},\end{aligned} \tag{38}$$

where the second equality is based on the total derivative. To evaluate the Jacobian $\frac{\partial \bar{z}_0}{\partial \boldsymbol{\theta}_1}$ in the first term, we take the total derivative with respect to $\boldsymbol{\theta}_1$ on both sides of $\nabla\widetilde{\varphi}_1(\bar{z}_0) = \boldsymbol{\omega}_0$:

$$\nabla^2\widetilde{\varphi}_1(\bar{z}_0)\cdot\frac{\partial \bar{z}_0}{\partial \boldsymbol{\theta}_1} + \frac{\partial \nabla\widetilde{\varphi}_1(z)}{\partial \boldsymbol{\theta}_1}\Big|_{z=\bar{z}_0} = \mathbf{0},$$

and it follows that

$$\Big(\frac{\partial \bar{z}_0}{\partial \boldsymbol{\theta}_1}\Big)^\top = -\Big[\nabla^2\widetilde{\varphi}_1(\bar{z}_0)\Big]^{-1}\frac{\partial \nabla\widetilde{\varphi}_1(\bar{z}_0)}{\partial \boldsymbol{\theta}_1}.$$

Therefore, Eq. (38) can be reformulated as

$$\frac{\partial \log p_1(\boldsymbol{\omega}_0)}{\partial \boldsymbol{\theta}_1} = -\Big(\frac{\partial \nabla\widetilde{\varphi}_1(\bar{z}_0)}{\partial \boldsymbol{\theta}_1}\Big)^\top\Big[\nabla^2\widetilde{\varphi}_1(\bar{z}_0)\Big]^{-1}\boldsymbol{g} - \frac{\partial}{\partial \boldsymbol{\theta}_1}\log\det\nabla^2\widetilde{\varphi}_1(\bar{z}_0),$$

where the first term can be calculated in the same way as Eq. (36) and the second term can be computed by Algorithm 2 as $\bar{z}_0$ is a constant here, thanks to the total derivative. In summary, the gradient of $h(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ with respect to $\boldsymbol{\theta}_1$ is given by

$$\frac{\partial h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} = -\frac{h(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1)}{2}\cdot\Big[\Big(\frac{\partial \nabla\widetilde{\varphi}_1(\bar{z}_0)}{\partial \boldsymbol{\theta}_1}\Big)^\top[\nabla^2\widetilde{\varphi}_1(\bar{z}_0)]^{-1}\boldsymbol{g} + \frac{\partial}{\partial \boldsymbol{\theta}_1}\log\det\nabla^2\widetilde{\varphi}_1(\bar{z}_0)\Big].$$

C.2 Gradient of the Sinkhorn Discrepancy Estimation

When the potential f is optimally solved, the total derivative of our Sinkhorn discrepancy estimator (18) with respect to the HyCNN parameters is given by

$$\frac{\partial \widetilde{\mathcal{S}}_{\epsilon,(n_k,N)}(\widehat{\mathbb{P}}_k, \mathbb{P}_k)}{\partial \boldsymbol{\theta}_k} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial g^c_{\epsilon,k}(\boldsymbol{\omega}_k^{(i)})}{\partial \boldsymbol{\theta}_k}\Big|_{f=f_{\epsilon,k,(n_k,N)}} = \frac{1}{N}\sum_{i=1}^{N}\Big(\frac{\partial \boldsymbol{\omega}_k^{(i)}}{\partial \boldsymbol{\theta}_k}\Big)^\top\cdot\frac{\partial g^c_{\epsilon,k}(\boldsymbol{\omega}_k^{(i)})}{\partial \boldsymbol{\omega}_k^{(i)}}\Big|_{f=f_{\epsilon,k,(n_k,N)}}. \tag{39}$$

where the first equality is due to the envelope theorem, $\frac{\partial \boldsymbol{\omega}_k^{(i)}}{\partial \boldsymbol{\theta}_k}$ is the Jacobian of the generated sample with respect to the HyCNN parameters, and the gradient of $g^c_{\epsilon,k}$ with respect to $\boldsymbol{\omega}_k^{(i)}$ can be explicitly computed based on Eq. (17).

C.3 Testing

In the testing phase, for a testing sample $\boldsymbol{\omega} \in \Omega$, we first trace its position in distributions $\mathbb{P}_0^\epsilon$ and $\mathbb{P}_1^\epsilon$ by $\bar{z}_0 = \nabla\widetilde{\varphi}_0^{-1}(\boldsymbol{\omega})$ and $\bar{z}_1 = \nabla\widetilde{\varphi}_1^{-1}(\boldsymbol{\omega})$, respectively. Then, we evaluate the log-density of least-favorable distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ at $\boldsymbol{\omega}$ by

$$\log p_k(\boldsymbol{\omega}) = \log p_k^\epsilon(\bar{z}_k) - \log\det\nabla^2\widetilde{\varphi}_k(\bar{z}_k), \quad \forall k \in \mathbb{K}.$$

Finally, we make decisions by the selected optimal detector $\phi_{\text{opt}} = \frac{1}{2}(\log p_0(\boldsymbol{\omega}) - \log p_1(\boldsymbol{\omega}))$.

D Wasserstein Distributionally Robust Hypothesis Testing Model

In this section, we provide the model of the WDRO method that is used as a baseline in our experimental part. The WDRO method is the soft-constraint version of the Wasserstein distributionally robust hypothesis testing model proposed by [17] (assuming $n_0 = n_1 = n$):

$$\begin{aligned} \max_{\boldsymbol{p}_0,\boldsymbol{p}_1\in\mathbb{R}_+^n,\gamma_0,\gamma_1\in\mathbb{R}_+^{n\times n}} \quad & \sum_{i\in[n]} \min\{\boldsymbol{p}_0^i,\boldsymbol{p}_1^i\} - \sum_{k\in\mathbb{K}} \lambda_k \cdot \sum_{i\in[n]}\sum_{j\in[n]} \gamma_{k,i,j}\cdot c(\boldsymbol{x}_k^{(i)},\boldsymbol{x}_k^{(j)}) \\ \text{s.t.} \quad & \sum_{j\in[n]} \gamma_{k,i,j} = \widehat{\mathbb{P}}_k^i \qquad \forall i\in[n], k\in\mathbb{K}, \\ & \sum_{i\in[n]} \gamma_{k,i,j} = \boldsymbol{p}_k^j, \qquad \forall j\in[n], k\in\mathbb{K}, \end{aligned} \tag{40}$$

where $\lambda_k \geq 0$ is a penalty parameter for each $k \in \mathbb{K}$, $\gamma_{k,i,j}$ represents the ij-th entry of γ_k, and the i-th entry of $\boldsymbol{p}_k$ (respectively, $\widehat{\mathbb{P}}_k$) is specified by $\boldsymbol{p}_k^i$ (respectively, $\widehat{\mathbb{P}}_k^i$). This model assumes that the samples in the testing set have the same support as the empirical distribution. To extend the resulting least-favorable distribution to the entire space, a kernel smoothing method is introduced. Let $\boldsymbol{p}_k^*$ be an optimal solution of (40). This method convolves the discrete distribution with a kernel function $G_h : \mathbb{R}^d \to \mathbb{R}$ parameterized by a bandwidth $h > 0$:

$$\boldsymbol{p}_k^h(\boldsymbol{\omega}) = \sum_{i=1}^n \boldsymbol{p}_k^*(\boldsymbol{x}_k^{(i)}) \cdot G_h(\boldsymbol{\omega} - \boldsymbol{x}_k^{(i)}), \quad \forall \boldsymbol{\omega}\in\Omega, k\in\mathbb{K},$$

where we choose the Gaussian kernel function $G_h(\boldsymbol{x}) = \exp(-\|\boldsymbol{x}\|^2/2h^2)$. Note that the number of decision variables of this linear program is $\mathcal{O}(n_0 \cdot n_1)$. Thus, this finite-dimensional convex program is not sensitive to dimension d, but is sensitive to the number of training data points of the empirical distribution.