

---

# Brenier Meets Adversarial Training: Optimal Transport Geometry for Robust Learning

---

Alireza Abdollahpoorrostam<sup>\*,1</sup>

Ehsan Sharifian<sup>\*,1</sup>

Buse Şen<sup>\*,1</sup>

Marco Cuturi<sup>2</sup>

Daniel Kuhn<sup>1</sup>

<sup>1</sup>EPFL

<sup>2</sup>Apple



## Abstract

Distributionally robust optimization (DRO) provides a principled framework for learning under distribution shift, but its practical use is hindered by the difficulty of evaluating worst-case risks for nonconvex loss functions. We study a penalized DRO formulation in which the adversary may choose any distribution but incurs a Wasserstein penalty for deviating from the empirical distribution. We show that the adversary’s problem can be reformulated as an optimization problem over transport maps that push empirical samples to adversarial ones, and we prove that optimal maps are cyclically monotone. We also show that standard adversarial training—based on per-sample local optimization—violates cyclical monotonicity and wastes transport costs unless the adversary is severely restricted. We propose two remedies. First, we introduce multi-start particle ascent, which alternates parallel gradient ascent with reassignment to enforce cyclical monotonicity across samples. Second, we parameterize adversarial maps as gradients of input-convex neural networks, which guarantees cyclical monotonicity by construction. Experiments on robust regression, image classification, and robust control show that our methods consistently outperform standard adversarial training and state-of-the-art baselines, achieving improved robustness and better generalization under distribution shift.

## 1 Introduction

Adversarial perturbations expose a pronounced fragility of modern machine learning systems, from early evidence of adversarial examples [32, 45, 53, 64] to a broad robustness literature spanning stronger attacks, defense mechanisms [20, 47, 54] and evaluation protocols [5, 14, 17]. Yet, robustness to diverse threat models, common corruptions, and distribution shifts remains an open challenge even for well-trained models and certified defenses [20, 54, 66, 67]. Robust learning is therefore not only about surviving a prescribed attack, but about generalizing under broader distributional variation.

Distributionally robust optimization (DRO) [44] provides a principled framework for adversarial training. Instead of relying on a fixed attack, DRO models a worst-case adversary that perturbs the data distribution. A widely used approach is Wasserstein DRO, which requires the learned model to perform well for all distributions within a prescribed Wasserstein distance of the empirical distribution [10, 22, 29, 43, 50]. This formulation is appealing because the Wasserstein geometry encodes which distribution shifts are plausible, while the radius controls their magnitude.

While conceptually appealing, DRO is not easy to use in practice because the worst-case problem searches over distributions. A common practical variant replaces the hard Wasserstein constraint with a penalized objective, in which the adversary may choose any distribution but incurs a Wasserstein penalty for deviating from the empirical distribution. Evaluating the resulting worst-case risk reduces to a finite-dimensional optimization problem over transported training samples. For standard machine

---

<sup>\*</sup>Equal contribution.

learning models, this problem is typically nonconvex. Existing methods therefore approximate it through independent sample-wise updates or entropic surrogates [60, 73, 75]. A sufficiently strong penalty on moving samples can make this problem better behaved (more concave). But it does so only by making substantial perturbations too expensive, and hence by weakening the adversary itself.

This paper takes a first step toward a theory of penalty-based Wasserstein DRO in the practically relevant regime where the adversary is powerful and thus faces a nonconvex problem. We also aim to develop strong approximate solution methods. The key enabling idea is to recast the adversary’s problem as an optimization problem over transport maps, which give rise to the following main contributions.

- **Section 3:** We argue that the adversary’s optimal transport map is cyclically monotone and that restricting the adversary to cyclically monotone maps benefits *both* the adversary and the learner.
- **Section 4:** We show that standard particle ascent methods for the dual DRO problem generally violate cyclical monotonicity in the nonconvex regime. As a repair, we introduce Multi-start Particle Ascent (MPA), which approximately solves the adversary’s problem by alternating parallel particle ascent over a batch of samples with optimal reassignment of the resulting particles.
- **Section 5:** We also propose an algorithm that models the adversary’s transport map as the gradient of an input-convex neural network (ICNN) [3], thereby enforcing cyclical monotonicity by design.
- **Section 6:** We evaluate both new methods on adversarial training for CIFAR-10 and on robust control tasks with uncertainty in physical parameters. We find that the ICNN-based method delivers the strongest overall robustness, while MPA remains competitive with the DRO baselines.

We do not focus on building a state-of-the-art robust training pipeline, which typically incorporates additional regularization techniques [20, 54, 77]. Instead, our methods (MPA and the ICNN-based approach) are complementary and can be seamlessly integrated with such strategies.

## 2 Background

We first review some basics of optimal transport and robust machine learning.

**Optimal transport.** We use  $\mathcal{P}$  to denote the space of probability measures  $\mathbb{P}$  on  $\mathbb{R}^m$  with finite second moments (i.e.,  $\mathbb{E}_{\mathbb{P}}[\|Z\|_2^2] < \infty$ ), and we equip  $\mathcal{P}$  with the Wasserstein distance.

**Definition 2.1** (Wasserstein distance). The Wasserstein distance of order 2 between two distributions  $\mathbb{P}, \hat{\mathbb{P}} \in \mathcal{P}$  is given by  $W(\mathbb{P}, \hat{\mathbb{P}}) := \inf\{\mathbb{E}_{\gamma}[\|Z - \hat{Z}\|_2^2]^{1/2} : \gamma \in \Gamma(\mathbb{P}, \hat{\mathbb{P}})\}$ , where  $\Gamma(\mathbb{P}, \hat{\mathbb{P}})$  denotes the set of joint distributions or couplings of  $Z$  and  $\hat{Z}$  with marginals  $\mathbb{P}$  and  $\hat{\mathbb{P}}$ , respectively.

The infimum in the definition of  $W(\mathbb{P}, \hat{\mathbb{P}})$  is always attained by some optimal coupling  $\gamma^*$ . Below we use  $\mathcal{T}$  to denote the set of all Borel-measurable functions  $T : \mathbb{R}^m \rightarrow \mathbb{R}^m$ , and we call elements of  $\mathcal{T}$  transport maps.

If  $\hat{\mathbb{P}}$  is absolutely continuous with respect to the Lebesgue measure, then the optimal coupling  $\gamma^*$  is induced by a transport map. Specifically, Brenier’s theorem asserts that there is a unique  $T^* \in \mathcal{T}$  that is cyclically monotone and satisfies  $\gamma^* = (T^*, \text{id})_{\#} \hat{\mathbb{P}}$  and  $\mathbb{P} = T_{\#}^* \hat{\mathbb{P}}$  [12].

**Definition 2.2** (Cyclical monotonicity). Let  $S_k$  denote the set of all permutations of  $\{1, \dots, k\}$ . A set  $\mathcal{C} \subseteq \mathbb{R}^m \times \mathbb{R}^m$  is called cyclical monotone if for any  $k \in \mathbb{N}$ , any collection of pairs  $\{(\hat{z}_1, z_1), \dots, (\hat{z}_k, z_k)\} \subseteq \mathcal{C}$  and any  $\sigma \in S_k$ , the following equivalent conditions hold:

$$\sum_{i=1}^k \|\hat{z}_i - z_i\|_2^2 \leq \sum_{i=1}^k \|\hat{z}_i - z_{\sigma(i)}\|_2^2 \iff \sum_{i=1}^k \langle z_{\sigma(i)} - z_i, \hat{z}_i \rangle \leq 0.$$

We say that a map  $T \in \mathcal{T}$  is cyclically monotone on a set  $\mathcal{Z} \subseteq \mathbb{R}^m$  if its graph  $\{(\hat{z}, T(\hat{z})) : \hat{z} \in \mathcal{Z}\}$  is a cyclically monotone set. If  $\mathcal{Z} = \mathbb{R}^m$ , we simply say that  $T$  is cyclically monotone.

By Rockafellar’s theorem [26, Theorem 2.5.3], a map  $T \in \mathcal{T}$  is cyclically monotone if and only if there exists a convex function  $\psi : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$  such that  $T(z) \in \partial\psi(z)$  for all  $z \in \mathbb{R}^m$ . Any transport map that violates cyclical monotonicity induces a suboptimal coupling between  $\mathbb{P}$  and  $\hat{\mathbb{P}}$ , and the degree of suboptimality is sometimes referred to as the Monge gap [68].

**Definition 2.3** (Monge gap). Given any  $\hat{\mathbb{P}} \in \mathcal{P}$ , the Monge gap of a map  $T \in \mathcal{T}$  is defined as

$$\mathcal{M}_{\hat{\mathbb{P}}}(T) := \mathbb{E}_{\hat{\mathbb{P}}}[\|\hat{Z} - T(\hat{Z})\|_2^2] - W^2(T_{\#} \hat{\mathbb{P}}, \hat{\mathbb{P}}).$$

Note that  $\mathcal{M}_{\hat{\mathbb{P}}}(T) \geq 0$ , and  $\mathcal{M}_{\hat{\mathbb{P}}}(T) = 0$  if and only if  $T$  is cyclically monotone on the support of  $\hat{\mathbb{P}}$ .

**Robust learning.** Let  $f : \Theta \times \mathbb{R}^m \rightarrow \mathbb{R}$  be a function that assigns each model parameter  $\theta$  from within a closed convex set  $\Theta \subseteq \mathbb{R}^d$  and each data point  $z \in \mathbb{R}^m$  a training loss  $f(\theta, z)$ . Throughout the paper we assume that  $f(\theta, z)$  is continuous in  $\theta$  and differentiable as well as Lipschitz continuous in  $z$ . Empirical Risk Minimization (ERM) seeks to minimize the expected loss under some  $\hat{\mathbb{P}} \in \mathcal{P}$ .

$$\min_{\theta \in \Theta} \mathbb{E}_{\hat{\mathbb{P}}} [f(\theta, \hat{Z})] \quad (\text{ERM})$$

The data distribution  $\hat{\mathbb{P}}$  is typically set to the empirical measure  $\hat{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \delta_{\hat{z}_i}$  corresponding to a given training dataset. While computationally convenient, ERM is sensitive to distribution shifts and tends to overfit to the training data. As a remedy, Robust Optimization (RO) minimizes the expected value of the *worst-case* loss with respect to all data realizations in an  $\varepsilon$ -neighborhood of  $\hat{Z}$  [47].

$$\min_{\theta \in \Theta} \mathbb{E}_{\hat{\mathbb{P}}} \left[ \max \left\{ f(\theta, z) : \|z - \hat{Z}\|_2 \leq \varepsilon \right\} \right] \quad (\text{RO})$$

While RO accounts for local data perturbations, it does not provide protection against global distribution shifts. Moreover, unless  $f(\theta, z)$  is concave in  $z$ , the embedded maximization problems in (RO) can only be solved approximately, which is usually done via iterative first-order methods [47]. Distributionally Robust Optimization (DRO) offers a principled approach to adversarial training that models distribution shifts explicitly. In penalty-based Wasserstein DRO a fictitious adversary maximizes the expected loss over all possible data distributions  $\mathbb{P} \in \mathcal{P}$  subject to a penalty proportional to  $W^2(\mathbb{P}, \hat{\mathbb{P}})$  for choosing distributions that differ from  $\hat{\mathbb{P}}$ . Thus, the training goal is to solve

$$\min_{\theta \in \Theta} \left\{ F(\theta) := \max_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} [f(\theta, Z)] - \lambda W^2(\mathbb{P}, \hat{\mathbb{P}}) \right\}, \quad (\text{DRO})$$

where  $\lambda > 0$  captures the strength of the penalty. We refer to  $F(\theta)$  as the *adversarial risk*.

### 3 Reparameterizations of the Adversarial Risk

Evaluating the adversarial risk  $F(\theta)$  amounts to maximizing a concave functional over an infinite-dimensional space of probability measures. Standard duality results in Wasserstein DRO such as [60, Proposition 1] allow us to reformulate  $F(\theta)$  as the empirical expectation of the *adversarial loss*, which is defined as the negative Moreau envelope of  $-f(\theta, z)$ ; see Appendix B for details.

$$F(\theta) = \mathbb{E}_{\hat{\mathbb{P}}} \left[ \max_{z \in \mathbb{R}^m} \{ f(\theta, z) - \lambda \|z - \hat{Z}\|_2^2 \} \right]. \quad (1)$$

Hence, evaluating  $F(\theta)$  reduces to solving an unconstrained global maximization problem over  $z$  for each empirical sample  $\hat{z}$ . The main results of this paper will exploit yet another reformulation of  $F(\theta)$  as the maximum of a functional optimization problem over the space of transport maps  $\mathcal{T}$ . Indeed, by the interchangeability principle [57, Theorem 14.60], we can recast the adversarial risk in (1) as

$$F(\theta) = \max_{T \in \mathcal{T}} \mathbb{E}_{\hat{\mathbb{P}}} \left[ f(\theta, T(\hat{Z})) - \lambda \|T(\hat{Z}) - \hat{Z}\|_2^2 \right]. \quad (2)$$

The reformulation (2) seeks an adversarial transport map  $T^*$  that simultaneously solves the global optimization problems associated with *all* empirical samples in (1). Even though the objective function value of  $T^*$  in problem (2) depends only on its behavior on the support of  $\hat{\mathbb{P}}$ , it is expedient to extend  $T^*$  to the entire ambient space. Formally, this can be done by constructing a set-valued mapping that assigns each empirical sample  $\hat{z} \in \mathbb{R}^m$  the corresponding set of adversarial samples

$$S(\hat{z}) := \arg \max_{z \in \mathbb{R}^m} \{ f(\theta, z) - \lambda \|z - \hat{z}\|_2^2 \}. \quad (3)$$

Next, we define the optimal adversarial map  $T^*$  as a Borel-measurable selector of this set-valued mapping. The existence of such a selector is shown in Proposition B.1. Since  $T^*(\hat{z}) \in S(\hat{z})$  for all  $\hat{z} \in \mathbb{R}^m$ , [57, Theorem 14.60] implies that  $T^*$  solves (2). In addition, the pushforward  $T^*_{\#} \hat{\mathbb{P}}$  constitutes a worst-case distribution for the inner maximization problem in (DRO); see Proposition B.2.

The equivalent reformulations (1) and (2) of the adversarial risk  $F(\theta)$  give rise to two complementary methods for constructing adversarial maps. The pointwise formulation (1) gives rise to methods for constructing *implicit* maps, which are only defined implicitly through simple algorithms for evaluating the map at a given input  $\hat{z} \in \mathbb{R}^m$ . Conversely, the global formulation (2) inspires the design of *explicit* maps, which are modeled as explicit parametric functions such as neural networks.

A simple approach for constructing implicit maps is due to Sinha et al. [60]. They show that if  $\nabla_z f(\theta, z)$  is  $L_{zz}$ -Lipschitz continuous in  $z$  and  $\lambda > L_{zz}/2$ , then problem (3) has a strongly concave objective function and thus a unique maximizer. Hence,  $T^*$  can be evaluated pointwise by solving (3) via gradient ascent. They further show that problem (1) is amenable to standard stochastic gradient descent (SGD) methods. Specifically, for each sample  $\hat{z} \sim \hat{\mathbb{P}}$ , they compute  $T^*(\hat{z})$  by solving (3) and then use Danskin's theorem to differentiate the parametric maximum in (1) with respect to  $\theta$ , thus constructing an unbiased stochastic gradient for  $F(\theta)$ . This naturally leads to an efficient inner-maximization outer-minimization loop. However, this procedure hinges on the condition  $\lambda > L_{zz}/2$ , which is difficult to satisfy in practice if  $f(\theta, z)$  depends on  $z$  through a neural network. As shown in [72], even computing first-order Lipschitz constants is NP-hard, with loose upper bounds reaching the order of  $10^6$  for standard architectures like AlexNet [42]. Bounding  $L_{zz}$ , which relates to the network's Hessian, generally inherits and often exacerbates these first-order intractabilities. When  $\lambda$  is set sufficiently large to guarantee strong concavity, adversarial perturbations are pulled close to the training samples, and the adversarial risk degenerates toward the empirical risk. In other words, the very condition that makes the problem tractable undermines its robustness.

In practice, reasonable values of  $\lambda$  that correspond to a sufficiently powerful adversary are significantly smaller than  $L_{zz}$  and invariably render problem (3) nonconvex. Given the intractability of nonconvex optimization, the best we can thus hope for is to construct (implicit or explicit) adversarial maps that solve (2) approximately. The *adversarial error*  $\varepsilon(\theta, T)$  of a given map  $T \in \mathcal{T}$  in problem (2) corresponding to a fixed parameter  $\theta \in \Theta$  is defined as the degree of suboptimality of  $T$ , that is,

$$\varepsilon(\theta, T) := F(\theta) - \mathbb{E}_{\hat{\mathbb{P}}}[f(\theta, T(\hat{Z})) - \lambda \|T(\hat{Z}) - \hat{Z}\|_2^2] \geq 0. \quad (4)$$

The following proposition shows that access to an oracle that approximately solves (2) with small adversarial error enables one to solve (DRO) to high accuracy via projected gradient descent.

**Proposition 3.1** (Convergence with inexact adversary). *Assume that  $f(\theta, z)$  is convex in  $\theta$  with uniformly bounded gradients  $\|\nabla_\theta f(\theta, z)\|_2 \leq L_\theta$  for all  $\theta \in \Theta$  and  $z \in \mathbb{R}^m$ . Fix any  $\theta_0 \in \Theta$ , let  $\theta^* \in \arg \min_{\theta \in \Theta} F(\theta)$ , and consider  $H$  projected gradient descent iterates satisfying*

$$\theta_{t+1} = \text{Proj}_\Theta(\theta_t - \alpha g_t), \quad \text{where} \quad g_t = \mathbb{E}_{\hat{\mathbb{P}}}[\nabla_\theta f(\theta_t, T_t(\hat{Z}))],$$

where  $T_t \in \mathcal{T}$  is a near-optimal adversarial map that approximately solves (2) at  $\theta = \theta_t$ . If the constant step size is set to  $\alpha = \|\theta_0 - \theta^*\|_2 / L_\theta \sqrt{H}$ , the average iterate  $\bar{\theta}_H := \frac{1}{H} \sum_{t=0}^{H-1} \theta_t$  satisfies

$$F(\bar{\theta}_H) - F(\theta^*) \leq \frac{L_\theta \|\theta_0 - \theta^*\|_2}{\sqrt{H}} + \frac{1}{H} \sum_{t=0}^{H-1} \varepsilon(\theta_t, T_t).$$

Proposition 3.1 adapts a standard convergence result for minimax problems to our setting [9, § 3.3]. For completeness we provide a short proof in Appendix C. It suggests that, in order to compute near-optimal minimizers for (DRO), one needs access to a strong adversarial oracle that outputs near-optimal adversarial maps. While Proposition 3.1 only holds under the simplifying assumption that the loss function  $f(\theta, z)$  is convex in  $\theta$ , we expect the conclusions of Proposition 3.1 to remain valid in the vicinity of local minimizers even if the loss is nonconvex. The next proposition shows that the adversarial error of any transport map is bounded below by its Monge gap.

**Proposition 3.2** (Suboptimality from wasteful transport). *We have  $\varepsilon(\theta, T) \geq \lambda \mathcal{M}_{\hat{\mathbb{P}}}(T)$  for all  $T \in \mathcal{T}$ .*

Note that  $\mathcal{M}_{\hat{\mathbb{P}}}(T) > 0$  indicates that  $T$  transports  $\hat{\mathbb{P}}$  to  $T_{\#}\hat{\mathbb{P}}$  in an inefficient manner. Proposition 3.2 implies that such inefficient transport necessarily leads to suboptimality in (2) and weakens the convergence guarantees of Proposition 3.1. The following result is a direct corollary of Proposition 3.2.

**Proposition 3.3.** *The optimal adversarial map  $T^*$  is cyclically monotone, and thus  $\mathcal{M}_{\hat{\mathbb{P}}}(T^*) = 0$ .*

Proposition 3.3 establishes a necessary optimality condition for Wasserstein DRO problems. It is reminiscent of Brenier's theorem [12], which establishes a similar optimality condition for optimal transport problems. However, it holds even if  $\hat{\mathbb{P}}$  is discrete and  $f(\theta, z)$  fails to be concave in  $z$ .

Evaluating  $F(\theta)$  requires solving a separate optimization problem for every point in the support of  $\hat{\mathbb{P}}$ ; see (1). Unless  $\lambda$  is astronomically large, all of these optimization problems are nonconvex and thus intractable. However, these global optimization problems are *coupled* because their objective functions differ only with regard to the anchor point  $\hat{z}$  of the quadratic penalty term. Accordingly,

**Proposition 3.3** reveals that their respective global maximizers  $T^*(\hat{z})$  are connected through the cyclical monotonicity of  $T^*$ . Even though the exact adversarial map  $T^*$  is hard to compute, the insight that a map  $T$  can only be optimal if it is cyclically monotone reduces the search space for good adversarial maps and guides us, by virtue of **Proposition 3.1**, toward better solutions for (DRO).

#### 4 Implicit Adversarial Maps

*Implicit* adversarial maps are defined in terms of an algorithm for point evaluation. A simple implicit map is obtained by using gradient ascent initialized at  $\hat{z}$  for solving the nonconvex maximization problem (1) to local optimality. This method naturally generalizes the approach by Sinha et al. [60] to nonconvex instances of (1) with moderate penalty parameters  $\lambda \leq L_{zz}/2$ . We first provide a formal definition of this particle ascent method.

**Particle Ascent (PA).** PA approximately solves the maximization problem (1) by using gradient ascent initialized at  $\hat{z}$ , either in continuous or discrete time. In both cases, we use  $T_{\text{pa}}$  to denote the resulting family of implicit maps parametrized by time. The correct interpretation of  $T_{\text{pa}}$  will always be clear from the context. In continuous time,  $T_{\text{pa}} : \mathbb{R}_+ \times \mathbb{R}^m \rightarrow \mathbb{R}^m$  is defined via the ODE

$$\partial_t T_{\text{pa}}(t, \hat{z}) = \nabla_z f(\theta, T_{\text{pa}}(t, \hat{z})) - 2\lambda(T_{\text{pa}}(t, \hat{z}) - \hat{z})$$

initialized by  $T_{\text{pa}}(0, \hat{z}) = \hat{z}$ . In discrete time,  $T_{\text{pa}} : \mathbb{Z}_+ \times \mathbb{R}^m \rightarrow \mathbb{R}^m$  is defined via the recursion

$$T_{\text{pa}}(t+1, \hat{z}) = T_{\text{pa}}(t, \hat{z}) + \eta(\nabla_z f(\theta, T_{\text{pa}}(t, \hat{z})) - 2\lambda(T_{\text{pa}}(t, \hat{z}) - \hat{z}))$$

initialized by  $T_{\text{pa}}(0, \hat{z}) = \hat{z}$ , where  $\eta > 0$  is the step size. When  $t$  is fixed and clear from context, we abbreviate  $T_{\text{pa}}(t, \hat{z})$  as  $T_{\text{pa}}(\hat{z})$ . Note that  $T_{\text{pa}}$  is implicitly defined through an optimization oracle.

The per-sample formulation of PA is computationally appealing, as it allows different particles to be updated in parallel. Even though  $T_{\text{pa}}$  is generically suboptimal in (2) unless  $\lambda > L_{zz}/2$ , it is guaranteed to be cyclically monotone (in fact, monotone) for one-dimensional data.

**Proposition 4.1** (Monotonicity of  $T_{\text{pa}}$  when  $m = 1$ ). *If  $m = 1$  and  $\nabla_z f(\theta, \cdot)$  is  $L_{zz}$ -Lipschitz, then*

- *the continuous-time PA map  $T_{\text{pa}}(t, \cdot)$  is cyclically monotone for all  $t \in \mathbb{R}_+$ ;*
- *the discrete-time PA map  $T_{\text{pa}}(t, \cdot)$  is cyclically monotone for all  $t \in \mathbb{Z}_+$  if  $\eta \leq 1/(L_{zz} + 2\lambda)$ .*

Unfortunately,  $T_{\text{pa}}$  ceases to be cyclically monotone already in two dimensions.

**Example 4.1** ( $T_{\text{pa}}$  violates cyclical monotonicity). *Set  $m = 2$ , and consider the loss function  $f(\theta, z) = \sum_{k=1}^2 a_k e^{-(z - \mu_k)^\top \Sigma_k^{-1} (z - \mu_k)}$  with  $a_1 = 200$ ,  $a_2 = 500$ ,  $\mu_1 = (6, 2)^\top$ ,  $\mu_2 = (-6, -2)^\top$ , and  $\Sigma_1 = \Sigma_2 = \text{diag}(64, 1)$ . We compute  $T_{\text{pa}}$  in discrete time for  $\lambda = 3$  and  $t = 8,000$  steps with step size  $\eta = 0.001$ , initialized at  $\hat{z}_1 = (-1, 1)^\top$  and  $\hat{z}_2 = (1, -1)^\top$ . The two particles converge to different local maxima,  $T_{\text{pa}}(\hat{z}_1) = (2.17, 1.98)^\top$  and  $T_{\text{pa}}(\hat{z}_2) = (-3.96, -2.00)^\top$  (see **Figure 1**), yielding  $\langle T_{\text{pa}}(\hat{z}_1) - T_{\text{pa}}(\hat{z}_2), \hat{z}_1 - \hat{z}_2 \rangle = -4.32 < 0$ , a strict violation of cyclical monotonicity.*

**Example 4.1** shows via **Proposition 3.2** that PA wastes transport costs and generates a suboptimal adversarial map. While we cannot hope to solve the nonconvex problem (2) to global optimality, we can efficiently improve any suboptimal map  $T$  by forcing its Monge gap to 0. To see this, assume that  $\hat{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \delta_{\hat{z}_i}$ . Then, the Monge gap satisfies  $\mathcal{M}_{\hat{\mathbb{P}}}(T) = \max_{\sigma \in \mathcal{S}_N} \frac{2}{N} \sum_{i=1}^N \langle T(\hat{z}_{\sigma(i)}) - T(\hat{z}_i), \hat{z}_i \rangle$  [68]. The optimal permutation  $\sigma$  identifies a reassignment of the empirical samples  $\{\hat{z}_i\}_{i=1}^N$  to the adversarial samples  $\{T(\hat{z}_i)\}_{i=1}^N$ , thus inducing an improved map  $T^\sigma$  with a strictly higher objective function value in (2) while ensuring that  $T_{\#} \hat{\mathbb{P}} = T_{\#}^\sigma \hat{\mathbb{P}}$ . An even better map is obtained by allowing many-to-one assignments rather than bijective permutations, which collapse multiple empirical samples to a common output and thereby modify  $T_{\#} \hat{\mathbb{P}}$  itself. Such an assignment can reduce the adversarial error by strictly more than the Monge gap. From now on, we call a map  $T$  assignment-stationary if  $T(\hat{z}_i)$  attains the largest objective value in problem (3) with  $\hat{z} = \hat{z}_i$  among all candidate solutions  $\{T(\hat{z}_j)\}_{j=1}^N$ . It cannot be improved by changing the output of  $\hat{z}_i$  to that of  $\hat{z}_j$  for any  $j \neq i$ .

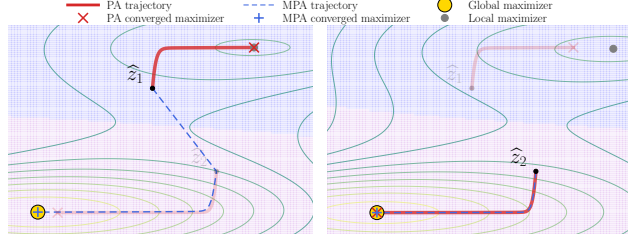


Figure 1: PA and MPA trajectories over the contours of  $f(\theta, z) - \lambda\|z - \hat{z}_i\|_2^2$  for  $i = 1$  (left) and  $i = 2$  (right).

**Definition 4.1** (Assignment-stationarity). A map  $T \in \mathcal{T}$  is *assignment-stationary* on  $\{\hat{z}_i\}_{i=1}^N$  if

$$f(\theta, T(\hat{z}_i)) - \lambda \|T(\hat{z}_i) - \hat{z}_i\|_2^2 \geq f(\theta, T(\hat{z}_j)) - \lambda \|T(\hat{z}_j) - \hat{z}_j\|_2^2 \quad \forall i, j \in [N]. \quad (5)$$

Assignment-stationarity can be viewed as a discrete, batch-wide analogue of standard continuous stationarity. Just as an algorithm for solving a single maximization problem should not stop if the objective function can be improved locally by moving along an ascent direction, an algorithm for solving  $N$  maximization problems in tandem should not stop if the  $i$ -th objective function can be improved by jumping to the candidate solution of the  $j$ -th problem for some  $j \neq i$  (even if the  $i$ -th candidate solution is a local maximizer). Beyond guaranteeing that no sample is trapped in a suboptimal assignment, assignment-stationarity implies cyclical monotonicity.

**Lemma 4.1.** If  $\hat{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \delta_{\hat{z}_i}$  while  $T \in \mathcal{T}$  is assignment-stationary on  $\{\hat{z}_i\}_{i=1}^N$ , then  $T$  is cyclically monotone on  $\{\hat{z}_i\}_{i=1}^N$ .

The converse implication fails. A cyclically monotone map need not be assignment-stationary because cyclical monotonicity ignores the loss  $f$  (counterexamples exist even for  $T = \text{id}$ ).

From [Example 4.1](#) we know that the PA map generically violates cyclical monotonicity and consequently fails to be assignment-stationary ([Lemma 4.1](#)). We thus propose Multi-start Particle Ascent (MPA) as a refinement of PA that enforces assignment-stationarity on the batch to which it is applied.

**Multi-Start Particle Ascent (MPA).** MPA computes the right-hand side of [\(1\)](#) approximately by constructing an adversarial sample for each empirical sample in a given batch of size  $B$ . The algorithm alternates between batch-wide reassignment of each empirical sample  $\hat{z}_i$  to the best adversarial sample in the current pool  $\mathcal{Z} = \{z_i\}_{i=1}^B$  and parallel local optimization of the  $B$  adversarial samples in  $\mathcal{Z}$ ; see the inner loop in [Algorithm 1](#), which runs over  $R$  rounds. MPA ends with a final reassignment step to enforce assignment-stationarity and thus cyclical monotonicity on  $\{\hat{z}_i\}_{i=1}^B$ . When  $R = 1$ , the reassignment steps highlighted in blue are disabled, and [Algorithm 1](#) reduces to PA. When  $R > 1$ , the reassignment steps are activated. Both PA and MPA can be embedded into a projected gradient descent algorithm for training  $\theta$ ; see the outer loop in [Algorithm 1](#), which runs over  $E$  epochs.

MPA is reminiscent of multi-start heuristics in non-convex optimization. Indeed, a standard multi-start heuristic launches several local searches from different starting points. Their purpose is to explore different regions (or basins) of the objective landscape. The resulting solutions form a pool of alternatives, from which one can select the best solution. MPA extends this familiar idea from a single optimization problem to a family of  $B$  coupled optimization problems. During reassignment, the local maxima discovered by the different problems using PA are

Algorithm 1: Training with implicit adversarial maps

---

**Input:** distribution  $\hat{\mathbb{P}}$ ; batch size  $B$ ; initial iterate  $\theta$ ; step sizes  $\alpha, \eta$ ; # epochs  $E$ ;  
# inner steps  $K$ ; # rounds  $R$  ( $R=1$ : PA;   $R>1$ : MPA )

**for**  $e \in [E]$  **do**  
  Sample  $\{\hat{z}_i\}_{i=1}^B \stackrel{iid}{\sim} \hat{\mathbb{P}}$ , and set  $z_i \leftarrow \hat{z}_i$  for all  $i \in [B]$   
  **for**  $r \in [R]$  **do**  
      $\mathcal{Z} \leftarrow \{z_i\}_{i=1}^B$    
      $z_i \leftarrow \arg \max_{z \in \mathcal{Z}} f(\theta, z) - \lambda \|z - \hat{z}_i\|_2^2$  for all  $i \in [B]$    
    **do**  $K$  **steps of**  $z_i \leftarrow z_i + \eta (\nabla_z f(\theta, z_i) - 2\lambda(z_i - \hat{z}_i))$  for all  $i \in [B]$   
    **end for**  
      $\mathcal{Z} \leftarrow \{z_i\}_{i=1}^B$    
      $z_i \leftarrow \arg \max_{z \in \mathcal{Z}} f(\theta, z) - \lambda \|z - \hat{z}_i\|_2^2$  for all  $i \in [B]$    
     $\theta \leftarrow \text{Proj}_{\Theta}(\theta - \frac{\alpha}{B} \sum_{i=1}^B \nabla_{\theta} f(\theta, z_i))$   
  **end for**  
**Output:**  $\theta$

---

shared as candidate starting solutions. Thus, each problem is no longer confined to the basin reached from its own empirical sample. Instead, it can jump to promising basins discovered by other samples.

MPA has several striking properties. First, as the inner loop of [Algorithm 1](#) is followed by a reassignment step, all adversarial maps generated by MPA are guaranteed to be assignment-stationary and therefore also cyclically monotone on the batch (see [Lemma 4.1](#)). Second, if each local optimization phase is run over  $K = \infty$  many iterations, which ensures that a stationary point is reached, then the inner loop of [Algorithm 1](#) is guaranteed to converge after finitely many rounds.

**Theorem 4.1** (Finite termination of MPA (informal version of [Theorem D.1](#))). *If  $K = \infty$ , then the adversarial samples  $\{z_i\}_{i=1}^B$  generated by MPA cease to change after finitely many rounds.*

Third, MPA guarantees global optimality whenever the batch of empirical samples is sufficiently large.

**Theorem 4.2** (Global optimality guarantee of MPA (informal version of [Theorem D.2](#))). *Suppose that the empirical samples are drawn independently from a distribution with an everywhere positive density. Then, for every tolerance  $\epsilon > 0$ , the implicit adversarial maps generated by MPA are  $\epsilon$ -optimal in (2) with probability at least  $1 - \beta(B, \epsilon)$ , where  $\beta(B, \epsilon) > 0$  converges to 0 as  $B$  grows.*

Intuitively, [Theorem 4.2](#) holds because MPA resembles a multi-start heuristic and because sufficiently many starting points enable such methods to solve nonconvex optimization problems globally. However, adding a sample to the batch not only enlarges the pool of candidate starting points but also introduces a new global optimization problem, which complicates the proof of [Theorem 4.2](#). Notably, the global optimality guarantee of MPA holds for *every* penalty parameter  $\lambda > 0$ , thereby removing the restriction  $\lambda > L_{zz}/2$  required for PA to be exact [\[60\]](#). See [Appendix D](#) for further details.

## 5 Explicit Adversarial Maps

To overcome the transductive nature of MPA, which constructs an implicit adversarial map restricted to the given batch of training samples and prevents generalization to unseen data, we now construct *explicit* adversarial maps of the form  $T_\omega \in \mathcal{T}$  parametrized by  $\omega \in \Omega$ , where  $\Omega$  is a Euclidean space. Restricting  $\mathcal{T}$  in (2) to this parametric family allows us to work with the parametric objective function

$$\mathcal{L}(\theta; \omega) := \mathbb{E}_{\hat{\mathbb{P}}} \left[ \ell(\theta, \omega, \hat{Z}) := f(\theta, T_\omega(\hat{Z})) - \lambda \|T_\omega(\hat{Z}) - \hat{Z}\|_2^2 \right].$$

To jointly learn the parameters  $\theta$  and  $\omega$  of the model and the adversarial map, respectively, we employ an alternating optimization scheme consisting of two nested loops. The inner loop applies  $K$  gradient-ascent steps in  $\omega$ , while the outer loop applies  $E$  projected gradient descent steps in  $\theta$ . Here, the outer gradient  $\nabla_\theta \max_{\omega \in \Omega} \mathcal{L}(\theta; \omega)$  is computed approximately via Danskin’s theorem, treating the current iterate  $\omega$  as an approximate maximizer. We formalize this procedure in [Algorithm 2](#).

It is tempting to model the explicit adversarial maps as unconstrained neural networks (e.g., generative adversarial networks [\[31\]](#) or neural transport maps for worst-case generation [\[16\]](#)). However, such standard networks typically violate cyclical monotonicity, which is a known property of the globally optimal map. It is thus expedient to enforce cyclical monotonicity in the neural network architecture. By a classical result in convex analysis, a map  $T \in \mathcal{T}$  is cyclically monotone if and only if it lies in the subdifferential of a convex function.

**Theorem 5.1** (Convex-potential representation, [\[26, Theorem 2.5.3\]](#)). *A transport map  $T \in \mathcal{T}$  is cyclically monotone if and only if there exists a convex potential function  $\psi : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  such that  $T(\hat{z}) \in \partial\psi(\hat{z})$  for every  $\hat{z} \in \mathbb{R}^m$ . In particular, if  $\psi$  is differentiable at  $\hat{z}$ , then  $T(\hat{z}) = \nabla\psi(\hat{z})$ .*

[Theorem 5.1](#) is the key to enforce cyclical monotonicity in explicit adversarial maps. Specifically, we can define  $T_\omega := \nabla_z \psi_\omega$ , where the potential function  $\psi_\omega : \mathbb{R}^m \rightarrow \mathbb{R}$  is modeled as an Input Convex Neural Network (ICNN) [\[3\]](#). ICNNs are scalar-valued neural networks guaranteed to be convex in their input  $\hat{z}$  (see, e.g., [\[3, Proposition 1\]](#)). Thus, the map  $T_\omega$  induced by  $\psi_\omega$  is cyclically monotone. We henceforth refer to any such  $T_\omega$  as an *ICNN map*. Note that ICNNs represent universal approximators for convex functions [\[15, Theorem 1\]](#). More precisely, any Lipschitz convex potential  $\psi$  on a compact set can be approximated to arbitrary absolute precision by an ICNN. Below we describe the exact architecture and initialization used to train ICNN maps in our experiments.

**Architecture.** Among existing implementations of ICNN-based Brenier maps (see, e.g., [\[39, 48, 70\]](#)), we adopt the convex-potential construction proposed in [\[70\]](#) and adapt it to our setting. We model only the scalar potential function  $\psi_\omega : \mathbb{R}^m \rightarrow \mathbb{R}$ , and evaluate the ICNN map  $T_\omega = \nabla_z \psi_\omega$  on the fly via automatic differentiation. With  $L$  hidden layers,  $\psi_\omega$  is defined through the recursions

$$\begin{aligned} y_1 &= \sigma(W_0^z z + b_0), \\ y_{\ell+1} &= \sigma(\exp(W_\ell^y y_\ell + W_\ell^z z + b_\ell) \quad \ell = 1, \dots, L-1, \\ \psi_\omega(z) &= \exp(w_L^y)^\top y_L + \frac{1}{2} z^\top (\text{diag}(\delta_L^2) + A_L^\top A_L) z + (w_L^z)^\top z + b_L, \end{aligned} \tag{ICNN}$$

where the exponential function  $\exp(\cdot)$  is applied component-wise. The full parameter vector is

$$\omega = \left( \{W_\ell^z\}_{\ell=0}^{L-1}, w_L^z, \delta_L, A_L, \{W_\ell^y\}_{\ell=1}^{L-1}, w_L^y, \{b_\ell\}_{\ell=0}^L \right)$$

with  $W_\ell^z \in \mathbb{R}^{q_\ell \times m}$ ,  $w_L^z, \delta_L \in \mathbb{R}^m$ ,  $A_L \in \mathbb{R}^{r_{\text{out}} \times m}$ ,  $W_\ell^y \in \mathbb{R}^{q_\ell \times q_{\ell-1}}$ ,  $w_L^y \in \mathbb{R}^{q_L-1}$ , and  $b_\ell \in \mathbb{R}^{q_\ell}$ , where  $q_\ell$  denotes the number of neurons in the  $\ell$ -th layer ( $q_L = 1$ ). By construction,  $\psi_\omega$  is convex in  $z$  for all  $\omega$ . Convexity is preserved across layers through affine transformations  $W_\ell^z z + b_\ell$ , convex non-decreasing activation functions  $\sigma$  (e.g., Softplus), and elementwise positive weights connecting each layer to the next. The quadratic term in the last layer is also convex because  $\text{diag}(\delta_L^2) + A_L^\top A_L \succeq 0$ . Thus, the induced map  $T_\omega = \nabla_z \psi_\omega$  is cyclically monotone for every  $\omega$ . The original formulation proposed in [70] incorporates positive semidefinite quadratic forms from the input to every layer. However, incorporating such quadratic forms at every layer is computationally expensive. We therefore restrict the quadratic terms to the final readout layer, keeping the hidden-layer passthroughs purely affine.

Our architecture’s capacity is primarily controlled by the last-layer readout rank  $r_{\text{out}}$ . As we show in Section G.4, restricting the quadratic term to the last layer while increasing  $r_{\text{out}}$  achieves the best robust performance and substantially reduces runtime compared with incorporating a quadratic term at every layer. Furthermore, ICNNs are notoriously difficult to train because their nonnegative weights can cause standard zero-mean initializations to propagate unstably across layers. We address this challenge with a two-part initialization strategy. First, we use a principled moment-matching scheme to stabilize the variance of the forward and backward signals across depth. Second, our single-quadratic architecture allows us to initialize the overall gradient map  $T_\omega$  near the identity, providing an effective warm start. Finally, to maintain training stability, we use an adaptive inner step size that dynamically adjusts to the local curvature of the network. We detail these training mechanisms in Appendix E.

## 6 Experiments

We benchmark our two new methods, namely the implicit multi-start particle ascent approach (MPA) and the explicit ICNN map approach (ICNN-DRO), against the following six baselines. Empirical Risk Minimization (ERM) solves problem (ERM) using SGD. Robust Optimization (RO) solves (RO) as in [47]. Particle Ascent (PA) solves (DRO) using Algorithm 1 with  $R = 1$ . Neural Network DRO (NN-DRO) solves (DRO) using Algorithm 2 with the explicit transport map modeled as a Multi-Layer Perceptron (MLP) [16] *without* enforcing cyclical monotonicity.<sup>1</sup> Finally, Sinkhorn DRO (SDRO) [73] and Wasserstein-Fisher-Rao DRO (WFR) [75] solve an entropy-regularized optimal transport variant of problem (DRO) (recovering (DRO) as the regularization weight vanishes) using sampling methods. The two methods differ in how they sample from the worst-case distribution, which is known to be a Gibbs distribution [73, Remark 4].

Our experiments span three domains of increasing complexity: feature-space logistic regression in § 6.1; image-space logistic regression under adversarial attacks, domain shifts, and image corruptions in § 6.2; and robust control in § 6.3. Additional results for uncertain least squares and cross-model universality are provided in § G.1 and § G.2, respectively.

### 6.1 Adversarial Multi-class Logistic Regression

In the first experiment, we follow the setup of [75, § 6.3] to assess the adversarial robustness of a multi-class logistic regression model trained on features extracted from the CIFAR-10 dataset [41]. Specifically, each image is mapped to a feature vector  $x \in \mathbb{R}^{m-1}$  with  $m - 1 = 512$  using a frozen ResNet-50 architecture [33] pre-trained on ImageNet [19]. We then train a linear classifier on these features using the negative log-likelihood loss. For a data point  $z = (x, y)$ , the loss is given by  $f(\theta, (x, y)) := -y^\top \theta^\top x + \log(\mathbf{1}^\top \exp(\theta^\top x))$ , where  $\theta = [\theta_1, \dots, \theta_C] \in \mathbb{R}^{(m-1) \times C}$  comprises all model parameters,  $y \in \{0, 1\}^C$  is the one-hot label vector, and  $C$  is the number of classes. The adversary perturbs only the features  $x$ , while the labels  $y$  remain unchanged.

To evaluate robustness under adversarial distribution shift, we subject the test features to an  $\ell_2$ -constrained Projected Gradient Descent (PGD) attack. We sweep the relative

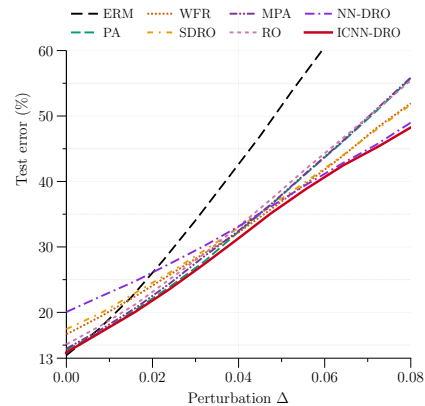


Figure 2: Test error vs. perturbation level  $\Delta$  on multi-class logistic regression.

<sup>1</sup> NN-DRO is related to [16], which models the transport map as an MLP trained *indirectly* via least-squares regression on adversarial samples generated by PA. In contrast, NN-DRO trains the MLP *directly* using Algorithm 2.

Table 1: Accuracy (%) under adversarial attacks, natural domain shifts and image corruptions. The left section reports results for the original CIFAR-10 test set (Clean, PGD, and AA), along with the Monge gap. The center columns evaluate accuracy on the unseen domains of CIFAR-10.1 and CIFAR-10.2. The right section (CIFAR-10-C) details performance across five increasing levels of corruption severity, followed by the mean accuracy across all five levels and the runtime. Bold values indicate the best results, and underlined values denote the second best.

| Method   | CIFAR-10     |              |              |                | CIFAR-10.1   | CIFAR-10.2   | CIFAR-10-C (Common Corruption) |              |              |              |              |              |            | Runtime |
|----------|--------------|--------------|--------------|----------------|--------------|--------------|--------------------------------|--------------|--------------|--------------|--------------|--------------|------------|---------|
|          | Clean        | PGD          | AA           | Monge Gap      | Clean        | Clean        | 1                              | 2            | 3            | 4            | 5            | Avg          | (s)        |         |
| ERM      | <b>93.12</b> | 14.76        | 00.20        | —              | <b>87.12</b> | <b>83.14</b> | <b>91.50</b>                   | <b>87.95</b> | 83.84        | 77.89        | 68.41        | 81.92        | —          |         |
| PA       | 90.08        | 52.41        | 48.86        | 8.01e-4        | 80.27        | 76.16        | 87.11                          | 84.35        | 80.21        | 74.62        | 69.74        | 79.20        | 1320       |         |
| RO       | 90.05        | 53.37        | 50.26        | 8.06e-4        | 80.28        | 76.81        | 87.15                          | 85.06        | 81.75        | 75.35        | 70.12        | 79.89        | <b>659</b> |         |
| WFR      | 91.03        | 53.28        | 50.37        | 5.63e-3        | 81.24        | 77.09        | 88.60                          | 85.34        | 82.93        | 77.56        | 71.09        | 81.10        | 6396       |         |
| SDRO     | 90.00        | 53.12        | 49.61        | 1.37e-3        | 80.20        | 76.48        | 87.12                          | 84.82        | 80.33        | 74.70        | 70.09        | 79.41        | 6241       |         |
| NN-DRO   | 89.46        | 50.34        | 44.27        | 3.25e-3        | 80.01        | 75.71        | 85.90                          | 82.23        | 78.55        | 72.20        | 67.42        | 77.26        | 11051      |         |
| MPA      | 91.39        | <u>55.31</u> | <u>54.47</u> | <u>4.71e-4</u> | 82.23        | 78.45        | <u>89.34</u>                   | <u>87.18</u> | <b>84.47</b> | <b>80.64</b> | <b>74.51</b> | <b>83.23</b> | 3612       |         |
| ICNN-DRO | <u>91.41</u> | <b>58.11</b> | <b>57.71</b> | <b>1.01e-7</b> | <u>82.56</u> | <u>80.12</u> | 89.20                          | 86.79        | <u>84.01</u> | <u>79.92</u> | <u>73.95</u> | <u>82.77</u> | 4037       |         |

attack budget  $\Delta \in [0, 0.08]$ , where the maximum perturbation radius is scaled by the average  $\ell_2$ -norm of the test features. Under this configuration (full training hyperparameters are provided in Section F.1), Figure 2 reports the test error as a function of  $\Delta$ . Both explicit map approaches perform well, with ICNN-DRO slightly outperforming NN-DRO across most perturbation budgets. At extreme perturbation levels, their test errors coincide. Under such large attack budgets, the adversary shifts probability mass so extensively that the optimal transport geometry no longer acts as a limiting factor, making the ICNN’s structural advantages less critical. The key difference emerges on clean samples ( $\Delta = 0$ ), where NN-DRO degrades significantly. In contrast, ICNN-DRO preserves clean accuracy. Section G.3 additionally reports the Monge gaps of the trained adversarial maps.

## 6.2 Robustness to Adversarial and Natural Distribution Shifts

Building on the feature-space analysis of Section 6.1, we now transition to full image-space adversarial training on the CIFAR-10 dataset. We train a deep neural network classifier  $h_\theta : \mathbb{R}^{m-1} \rightarrow \mathbb{R}^C$  against perturbations injected directly into the raw pixel space ( $m - 1 = 32 \times 32 \times 3$ ). For a given data point  $z = (x, y)$ , where  $y \in \{0, 1\}^C$  is the one-hot encoded ground-truth label, we optimize the network parameters  $\theta$  utilizing the multi-class cross-entropy loss. We formulate this objective as  $f(\theta, (x, y)) := -y^\top h_\theta(x) + \log(\mathbf{1}^\top \exp(h_\theta(x)))$ . As in the previous experiment, the adversary exclusively perturbs the covariates  $x$  while the labels  $y$  remain fixed. We subject the trained models to a comprehensive evaluation spanning worst-case adversarial attacks, image corruptions, and natural domain shifts to assess their generalizability, and additionally report each map’s empirical Monge gap and training runtime (implementation details in Section F.2).

**Robustness to adversarial attacks.** We evaluate on CIFAR-10 under  $\ell_2$  attacks with  $\varepsilon=0.5$  (standard budget [17, 18]) using PGD and AutoAttack (AA) [17]. AA is a suite of diverse attacks designed to expose the gradient obfuscation failure mode [5], in which a model appears robust to PGD only because its gradients are uninformative. The left block of Table 1 reports the results. ICNN-DRO preserves clean accuracy at 91.41% (within 1.71% of ERM) and attains 58.11% against PGD and 57.71% against AA, substantially outperforming PA and WFR. Crucially, the PGD–AA gap is only 0.4% for ICNN-DRO compared to 3.55% for PA. This tight agreement certifies that the robustness gain reflects genuine geometric regularity induced by the Brenier-structured map, rather than an artifact of gradient masking [5] (see Section F.2).

**Robustness to unseen domains.** CIFAR-10.1 [55] and CIFAR-10.2 [46] provide independent, natural covariate-shift benchmarks free of algorithmic corruption. Table 1 shows that ICNN-DRO achieves 82.56% on CIFAR-10.1 and 80.12% on CIFAR-10.2. Our MPA follows very closely behind at 82.23% and 78.45%, respectively. Both of our proposed methods consistently outperform the full suite of established robust baselines. Because these datasets were unseen during training, these gains demonstrate genuine generalization rather than overfitting to specific shift patterns.

**Robustness to image corruptions.** We evaluate on CIFAR-10-C [35], which features 15 algorithmic corruptions (e.g., noise, blur, weather effects) at five severity levels. As shown in Table 1, ICNN-DRO attains 82.77% average accuracy, outperforming PA (79.20%) and WFR (81.10%). Notably, it

exceeds ERM (81.92%), showing that robustification via a global transport map improves resilience to corruptions without sacrificing in-distribution accuracy. The gap widens at the highest severity (level 5), where ICNN-DRO reaches 73.95% against 69.74% for PA.

**Monge gap and cost.** ICNN-DRO attains a Monge gap of  $1.01\text{e-}7$ , nearly four orders of magnitude below every baseline, since it is cyclically monotone by construction, while reassignment alone lowers PA’s gap by 38%. The two smallest gaps belong to the two most robust methods, as predicted by [Proposition 3.2](#). Enforcing cyclical monotonicity is also inexpensive: ICNN-DRO trains in 4,037s, faster than every other DRO baseline while surpassing all of them on every robustness metric, and only the first-order RO is faster, at a cost of 7.45 AA points.

### 6.3 Robust Control

We evaluate our methods on the cart-pole control task, where the goal is to balance a pole on a cart by applying horizontal actuation. We have parametric physical uncertainty over the pole mass and length, denoted by  $z = (m_p, \ell) \in \mathcal{Z} \subseteq \mathbb{R}^2$ , where  $\mathcal{Z} = [0.05, 0.2] \times [0.25, 0.75]$ , and draw the training samples uniformly from  $\mathcal{Z}$ . At any time step  $t$ , the agent observes a state  $s_t \in \mathcal{S} \subseteq \mathbb{R}^4$ , capturing the system’s position, linear velocity, angle, and angular velocity, and samples an action  $a_t \in \mathcal{A} = \{0, 1\}$  (corresponding to a left or right impulse) from a parameterized policy  $\pi_\theta(a_t | s_t)$ . The environment then evolves according to known physical dynamics  $s_{t+1} = F(s_t, a_t, z)$  [27]. We define the loss function as the negative expected reward  $f(\theta, z) := -\mathbb{E}_{\pi_\theta}[\sum_{t=0}^{H-1} r(s_t, a_t)]$  over trajectories generated by the dynamics parameterized by  $z$ . The adversary perturbs the nominal uncertainty distribution under the regularization parameter  $\lambda = 1$ . We optimize  $\theta$  using proximal policy optimization (PPO) [59].

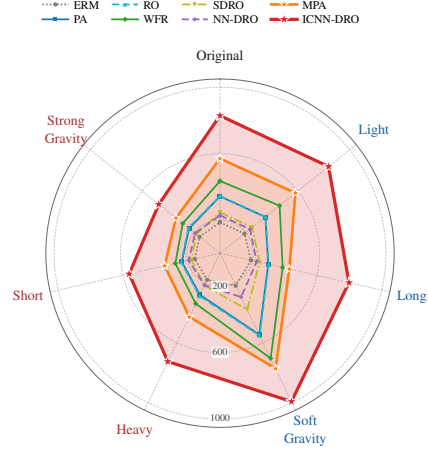


Figure 3: Mean episode length for  $\lambda=1$ .

We evaluate each policy over 1,000 deterministic rollouts, each with maximum horizon  $H = 1,000$ , and report the mean episode length across these trials in [Figure 3](#). The figure reports mean episode lengths only; a more detailed table including standard errors, together with a sensitivity analysis over  $\lambda \in \{0.1, 0.5\}$  and additional training details, is provided in [Section F.3](#). The evaluation grid spans three categories to test both interpolation and extrapolation. The original configuration captures the nominal physics at the center of the training box. The evaluation also includes Easier Environments (Light, Long, Soft Gravity) and Harder Environments (Heavy, Short, Strong Gravity). Crucially, the harder variants lie outside the training uncertainty box, so they probe extrapolation rather than interpolation of the learned robustness. As shown in [Figure 3](#), ICNN-DRO achieves the best episode length on every variant, with especially large gains on the extrapolative Heavy, Short, and Strong gravity shifts. MPA is the consistent runner-up and substantially improves over PA, while WFR is the strongest of the remaining baselines but degrades more noticeably on the harder variants. These results suggest that improving the transport geometry in the inner maximization already yields a substantial benefit, and that learning a single global Brenier-structured adversarial map provides an additional robustness gain beyond batch-wise reassignment alone.

## 7 Conclusions

We studied regularized Wasserstein DRO through the geometry of optimal transport and showed that cyclical monotonicity is the key structural property missing from standard per-sample adversarial training. This yields two remedies: MPA, which restores the transport structure implicitly through ascent and reassignment, and ICNN-based adversaries, which enforce it explicitly by parameterizing a Brenier map. Across regression, adversarial training, and robust control, both approaches improve robustness and out-of-distribution generalization over standard baselines.

**Limitations.** The main limitation is computational cost. MPA requires an  $\mathcal{O}(B^2)$  reassignment step per round, while the ICNN adversary introduces an auxiliary network, making each training iteration more expensive than standard single-step adversarial updates [32]. However, this increased cost is offset by significant robustness benefits.

**Acknowledgments** This work was supported as a part of NCCR Automation, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40\_225155). We thank Yves Rychener for insightful discussions in the early stages of this work.

## References

- [1] Pierre-Antoine Absil, Robert Mahony, and Ben Andrews. [Convergence of the iterates of descent methods for analytic cost functions](#). *SIAM Journal on Optimization*, 16(2):531–547, 2005.
- [2] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Springer, 2005.
- [3] Brandon Amos, Lei Xu, and J Zico Kolter. [Input convex neural networks](#). In *International Conference on Machine Learning*, pages 146–155, 2017.
- [4] Larry Armijo. [Minimization of functions having Lipschitz continuous first partial derivatives](#). *Pacific Journal of Mathematics*, 16(1):1–3, 1966.
- [5] Anish Athalye, Nicholas Carlini, and David Wagner. [Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples](#). In *International Conference on Machine Learning*, pages 274–283, 2018.
- [6] Sheldon Axler. *Measure, Integration & Real Analysis*. Springer, 2020.
- [7] Andrew G Barto, Richard S Sutton, and Charles W Anderson. [Neuronlike adaptive elements that can solve difficult learning control problems](#). *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):834–846, 1983.
- [8] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. [Estimating or propagating gradients through stochastic neurons for conditional computation](#). *arXiv:1308.3432*, 2013.
- [9] Dimitri Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.
- [10] Jose Blanchet and Karthyek Murthy. [Quantifying distributional model risk via optimal transport](#). *Mathematics of Operations Research*, 44(2):565–600, 2019.
- [11] Jose H. Blanchet and Peter W. Glynn. [Unbiased Monte Carlo for optimization and functions of expectations via multi-level randomization](#). In *Winter Simulation Conference*, pages 3656–3667, 2015.
- [12] Yann Brenier. [Polar factorization and monotone rearrangement of vector-valued functions](#). *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991.
- [13] Charlotte Bunne, Andreas Krause, and Marco Cuturi. [Supervised training of conditional Monge maps](#). In *Advances in Neural Information Processing Systems*, pages 6859–6872, 2022.
- [14] Nicholas Carlini and David Wagner. [Towards evaluating the robustness of neural networks](#). In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.
- [15] Yize Chen, Yuanyuan Shi, and Baosen Zhang. [Optimal control via neural networks: A convex approach](#). In *International Conference on Learning Representations*, 2019.
- [16] Xiuyuan Cheng, Yao Xie, Linglingzhi Zhu, and Yunqin Zhu. [Worst-case generation via minimax optimization in Wasserstein space](#). *arXiv:2512.08176*, 2025.
- [17] Francesco Croce and Matthias Hein. [Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks](#). In *International Conference on Machine Learning*, pages 2206–2216, 2020.
- [18] Francesco Croce, Maksym Andriushchenko, Vikash Sehwal, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. [RobustBench: A standardized adversarial robustness benchmark](#). In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

- [19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. [Imagenet: A large-scale hierarchical image database](#). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [20] Junhao Dong, Seyed-Mohsen Moosavi-Dezfooli, Jianhuang Lai, and Xiaohua Xie. [The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training](#). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 24678–24687, 2023.
- [21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*, 2021.
- [22] John C Duchi and Hongseok Namkoong. [Learning models with uniform performance via distributionally robust optimization](#). *Annals of Statistics*, 49(3):1378–1406, 2021.
- [23] Laurent El Ghaoui and Hervé Lebret. [Robust solutions to least-squares problems with uncertain data](#). *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.
- [24] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. [Exploring the landscape of spatial robustness](#). In *International Conference on Machine Learning*, pages 1802–1811, 2019.
- [25] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. [Interpolating between optimal transport and MMD using Sinkhorn divergences](#). In *International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.
- [26] Alessio Figalli and Federico Glaudo. [An Invitation to Optimal Transport, Wasserstein Distances, and Gradient Flows](#). EMS Press Berlin, 2021.
- [27] C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. [Brax - A differentiable physics engine for large scale rigid body simulation](#). In *Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.
- [28] A. M. Gabrielov. [Projections of semi-analytic sets](#). *Functional Analysis and its Applications*, 2(4):282–291, 1968.
- [29] Rui Gao, Xi Chen, and Anton J Kleywegt. [Wasserstein distributionally robust optimization and variation regularization](#). *Operations Research*, 72(3):1177–1191, 2024.
- [30] Xavier Glorot and Yoshua Bengio. [Understanding the difficulty of training deep feedforward neural networks](#). In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.
- [31] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. [Generative adversarial nets](#). In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [32] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. [Explaining and harnessing adversarial examples](#). In *International Conference on Learning Representations*, 2015.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. [Deep residual learning for image recognition](#). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [34] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Yuval Tassa, and Tom Erez. [Learning continuous control policies by stochastic value gradients](#). In *Advances in Neural Information Processing Systems*, pages 2944–2952, 2015.
- [35] Dan Hendrycks and Thomas Dietterich. [Benchmarking neural network robustness to common corruptions and perturbations](#). In *International Conference on Learning Representations*, 2019.
- [36] Pieter-Jan Hoedt and Günter Klambauer. [Principled weight initialisation for input-convex neural networks](#). In *Advances in Neural Information Processing Systems*, pages 46093–46104, 2023.

- [37] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. [DiffTaichi: Differentiable programming for physical simulation](#). In *International Conference on Learning Representations*, 2020.
- [38] Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. [Wasserstein-2 generative networks](#). *arXiv:1909.13082*, 2019.
- [39] Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. [Do neural optimal transport solvers work? A continuous Wasserstein-2 benchmark](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 14593–14605, 2021.
- [40] Steven G Krantz and Harold R Parks. *A Primer of Real Analytic Functions*. Springer Science & Business Media, 2002.
- [41] Alex Krizhevsky and Geoffrey Hinton. [Learning multiple layers of features from tiny images](#). Technical report, University of Toronto, 2009.
- [42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. [Imagenet classification with deep convolutional neural networks](#). In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [43] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. [Wasserstein distributionally robust optimization: Theory and applications in machine learning](#). *INFORMS TutORials in Operations Research*, pages 130–166, 2019.
- [44] Daniel Kuhn, Soroosh Shafiee, and Wolfram Wiesemann. [Distributionally robust optimization](#). *Acta Numerica*, 34:579–804, 2025.
- [45] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. [Adversarial machine learning at scale](#). In *International Conference on Learning Representations*, 2017.
- [46] Shangyun Lu, Bradley Nott, Aaron Olson, Alberto Todeschini, Hossein Vahabi, Yair Carmon, and Ludwig Schmidt. [Harder or different? A closer look at distribution shift in dataset reproduction](#). In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, page 15, 2020.
- [47] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. [Towards deep learning models resistant to adversarial attacks](#). In *International Conference on Learning Representations*, 2018.
- [48] Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. [Optimal transport mapping via input convex neural networks](#). In *International Conference on Machine Learning*, pages 6672–6681, 2020.
- [49] Luke Metz, C. Daniel Freeman, Samuel S. Schoenholz, and Tal Kachman. [Gradients are not all you need](#). *arXiv:2111.05803*, 2021.
- [50] Peyman Mohajerin Esfahani and Daniel Kuhn. [Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations](#). *Mathematical Programming*, 171(1):115–166, 2018.
- [51] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. [Monte Carlo gradient estimation in machine learning](#). *Journal of Machine Learning Research*, 21(132):1–62, 2020.
- [52] Yurii Nesterov. [A method for solving the convex programming problem with convergence rate  \$O\(1/k^2\)\$](#) . *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.
- [53] Anh Nguyen, Jason Yosinski, and Jeff Clune. [Deep neural networks are easily fooled: High confidence predictions for unrecognizable images](#). In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.
- [54] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. [Reducing excessive margin to achieve a better accuracy vs. robustness trade-off](#). In *International Conference on Learning Representations*, 2022.

- [55] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. [Do ImageNet classifiers generalize to ImageNet?](#) In *International Conference on Machine Learning*, pages 5389–5400, 2019.
- [56] Gareth O Roberts and Richard L Tweedie. [Exponential convergence of Langevin distributions and their discrete approximations](#). *Bernoulli*, 2(4):341–363, 1996.
- [57] R Tyrrell Rockafellar and Roger JB Wets. *Variational Analysis*. Springer, 2009.
- [58] Filippo Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*. Progress in Nonlinear Differential Equations and Their Applications. Birkhäuser, 2015.
- [59] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. [Proximal policy optimization algorithms](#). *arXiv:1707.06347*, 2017.
- [60] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. [Certifying some distributional robustness with principled adversarial training](#). In *International Conference on Learning Representations*, 2018.
- [61] Sarath Sivaprasad, Ankur Singh, Naresh Manwani, and Vineet Gandhi. [The curious case of convex neural networks](#). In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 738–754, 2021.
- [62] Leslie N. Smith and Nicholay Topin. [Super-convergence: Very fast training of neural networks using large learning rates](#). In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, page 1100612, 2019.
- [63] H. J. Terry Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. [Do differentiable simulators give better policy gradients?](#) In *International Conference on Machine Learning*, pages 20668–20696, 2022.
- [64] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. [Intriguing properties of neural networks](#). In *International Conference on Learning Representations*, 2014.
- [65] Conghui Tan, Shiqian Ma, Yu-Hong Dai, and Yuqiu Qian. [Barzilai-Borwein step size for stochastic gradient descent](#). In *Advances in Neural Information Processing Systems*, pages 685–693, 2016.
- [66] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. [Measuring robustness to natural distribution shifts in image classification](#). In *Advances in Neural Information Processing Systems*, pages 18583–18599, 2020.
- [67] Florian Tramèr and Dan Boneh. [Adversarial training and robustness for multiple perturbations](#). In *Advances in Neural Information Processing Systems*, pages 5866–5876, 2019.
- [68] Théo Uscidda and Marco Cuturi. [The Monge gap: A regularizer to learn all transport maps](#). In *International Conference on Machine Learning*, pages 34709–34733, 2023.
- [69] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.
- [70] Nina Vesseron and Marco Cuturi. [On a neural implementation of Brenier’s polar factorization](#). In *International Conference on Machine Learning*, pages 49434–49454, 2024.
- [71] Cédric Villani. *Topics in Optimal Transportation*, volume 58. American Mathematical Society, 2003.
- [72] Aladin Virmaux and Kevin Scaman. [Lipschitz regularity of deep neural networks: Analysis and efficient estimation](#). In *Advances in Neural Information Processing Systems*, pages 3839–3848, 2018.
- [73] Jie Wang, Rui Gao, and Yao Xie. [Sinkhorn distributionally robust optimization](#). *Operations Research*, 74(3):1581–1603, 2026.

- [74] Chen Xu, Jonghyeok Lee, Xiuyuan Cheng, and Yao Xie. [Flow-based distributionally robust optimization](#). *IEEE Journal on Selected Areas in Information Theory*, 5:62–77, 2024.
- [75] Zusen Xu and Jia-Jie Zhu. [Gradient flow sampler-based distributionally robust optimization](#). *arXiv:2510.25956*, 2025.
- [76] Haoxiang You, Yilang Liu, and Ian Abraham. [Accelerating visual-policy learning through parallel differentiable simulation](#). In *Advances in Neural Information Processing Systems*, 2025.
- [77] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. [Theoretically principled trade-off between robustness and accuracy](#). In *International Conference on Machine Learning*, pages 7472–7482, 2019.
- [78] Luhao Zhang, Jincheng Yang, and Rui Gao. [A Short and General Duality Proof for Wasserstein Distributionally Robust Optimization](#). *Operations Research*, 73(4):1723–2295, 2024.
- [79] Jia-Jie Zhu, Wittawat Jitkrittum, Moritz Diehl, and Bernhard Schölkopf. [Kernel distributionally robust optimization: Generalized duality theorem and stochastic approximation](#). In *International Conference on Artificial Intelligence and Statistics*, pages 280–288, 2021.

# Appendix

## Contents

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| <b>A Related Work</b>                                                    | <b>16</b> |
| <b>B Technical Background Results</b>                                    | <b>17</b> |
| <b>C Proofs</b>                                                          | <b>20</b> |
| <b>D Properties of MPA</b>                                               | <b>22</b> |
| D.1 Finite-time Termination of MPA . . . . .                             | 22        |
| D.2 Global Optimality Guarantees for MPA . . . . .                       | 25        |
| D.3 An Illustrative Example where PA Stalls and MPA Progresses . . . . . | 30        |
| <b>E ICNN Training Details</b>                                           | <b>31</b> |
| E.1 Adaptive Inner Step Size Selection . . . . .                         | 32        |
| <b>F Supplementary Details for Section 6</b>                             | <b>33</b> |
| F.1 Supplementary Details for Section 6.1 . . . . .                      | 34        |
| F.2 Supplementary Details for Section 6.2 . . . . .                      | 34        |
| F.3 Supplementary Details for Section 6.3 . . . . .                      | 35        |
| <b>G Additional Experiments</b>                                          | <b>39</b> |
| G.1 Robust Least Squares . . . . .                                       | 39        |
| G.2 Cross-Model Universality . . . . .                                   | 40        |
| G.3 An Empirical Monge Gap . . . . .                                     | 41        |
| G.4 Design Axes of the ICNN Adversary . . . . .                          | 42        |
| G.5 Penalty Annealing for the Global Map . . . . .                       | 43        |

## A Related Work

The literature most closely related to our work falls naturally into three groups.

**Optimal transport and ICNNs.** For any  $\mathbb{P}, \hat{\mathbb{P}} \in \mathcal{P}$ , one can use Kantorovich duality to show that

$$\frac{1}{2}W^2(\mathbb{P}, \hat{\mathbb{P}}) = \frac{1}{2}\mathbb{E}_{\hat{\mathbb{P}}}[\|\hat{Z}\|_2^2] + \frac{1}{2}\mathbb{E}_{\mathbb{P}}[\|Z\|_2^2] - \min_{\psi \text{ convex}} \left\{ \mathbb{E}_{\hat{\mathbb{P}}}[\psi(\hat{Z})] + \mathbb{E}_{\mathbb{P}}[\psi^*(Z)] \right\}, \quad (6)$$

where  $\psi^*$  denotes the convex conjugate of  $\psi$ . The minimum in (6) is always attained. Moreover, if  $\hat{\mathbb{P}}$  is absolutely continuous with respect to Lebesgue measure, then for every minimizer  $\psi$ ,  $\nabla\psi$  coincides  $\hat{\mathbb{P}}$ -almost everywhere with the unique Brenier map from  $\hat{\mathbb{P}}$  to  $\mathbb{P}$  [12]. These observations underlie a rich line of work that parameterizes the convex potential  $\psi$  using an ICNN [3] and optimizes its parameters, with the conjugate  $\psi^*$  evaluated through an inner maximization [13, 38, 39, 48, 70]. Our methods in Section 5 differ from the ICNN-based solution methods for optimal transport problems in several important respects. (i) The target distribution  $\mathbb{P}$  is not fixed as it solves the inner problem in (DRO) and hence varies with  $\theta$ . (ii) Our reference distribution  $\hat{\mathbb{P}}$  is not required to be absolutely continuous (such as a discrete empirical distribution), and thus Brenier’s theorem

does not apply. (iii) Our problem (2) is nonconvex in  $T$  and thus also in the potential  $\psi$ , in contrast to the minimization problem in (6). We show that restricting the adversary to gradients of convex potentials acts as a structural safeguard against selecting poor local optima—a role absent in the context of solving classical optimal transport problems. (iv) We use ICNNs directly to solve a primal optimization problem over transport maps (see (2)) instead of a dual problem over potential functions. (v) For the optimal transport problem underlying (6), cyclical monotonicity of a map is both necessary and sufficient for optimality [71, Theorem 2.12], whereas in our setting it is only necessary.

**Adversarial training with DRO.** For adversarial training based on problem (DRO), the closest approach to ours is [60], which evaluates the adversarial risk by running independent gradient ascent from each training sample. Other approaches quantify the mismatch between the nominal and target distributions using an entropic optimal transport discrepancy. In this setting, the worst-case distribution is a Gibbs measure [73], allowing the adversarial risk to be evaluated by sampling from this measure. An alternative particle-based flow for sampling from the Gibbs measure is proposed in [75]. In all of these approaches, however, adversarial samples are generated independently for each training sample, without enforcing cyclical monotonicity. MPA takes a fundamentally different approach in that it reassigns particles across the batch to enforce this structural property.

**Adversarial transport maps.** The idea of recasting the adversary’s problem as an optimization problem over transport maps was first explored in [16, 60, 74]. When  $\widehat{\mathbb{P}}$  admits a density, [16, 74] justify this formulation via Brenier’s theorem for Monge’s optimal transport problem. In [74], the optimal adversarial map is constructed by training and composing a sequence of neural networks that push the nominal distribution forward to the worst-case distribution. In [16], by contrast, the transport map is represented by a single neural network trained via least-squares regression on adversarial samples generated by particle ascent [60]. Both approaches model the adversarial map using an *unconstrained* neural network and require  $\widehat{\mathbb{P}}$  to admit a density. We instead constrain the map to be the gradient of an ICNN, making it cyclically monotone by construction, while imposing no density assumption that would exclude an empirical nominal distribution. Moreover, none of these prior works recognize that enforcing cyclical monotonicity can steer the adversarial map away from local maxima, thereby improving the quality of the adversarial samples and, ultimately, the effectiveness of adversarial training.

## B Technical Background Results

**Blanket assumptions.** Throughout [Appendix B](#), we fix  $\theta \in \Theta$  and  $\lambda > 0$ , and we assume that  $f(\theta, z)$  is  $L_z$ -Lipschitz continuous in  $z$ . We use  $\mathcal{P}_0(\mathbb{R}^m)$  to denote the family of all Borel probability measures on  $\mathbb{R}^m$ , which contains the family  $\mathcal{P}(\mathbb{R}^m)$  of all Borel probability measures with finite second moments as a subset. Furthermore, we fix an arbitrary reference measure  $\widehat{\mathbb{P}} \in \mathcal{P}(\mathbb{R}^m)$ .

In the remainder of this section, we show that the adversarial map  $T^*$  is well-defined and that the maximum in the definition of  $F(\theta)$  is attained by  $T^*_{\#}\widehat{\mathbb{P}}$ , which is guaranteed to have finite second moments. The main text defines  $F(\theta)$  as a maximum over  $\mathcal{P}(\mathbb{R}^m)$ , which presupposes the existence of a maximizer therein. As establishing this is the goal of the present appendix, we initially replace the maximum with a supremum and reintroduce the maximum only once attainment is shown.

**Proposition B.1** (Measurability and well-definedness of  $T^*$ ). *The set-valued mapping  $S(\widehat{z})$  in (3) is non-empty and closed-valued for all  $\widehat{z} \in \mathbb{R}^m$ . Furthermore, there exists a Borel-measurable map  $T^* : \mathbb{R}^m \rightarrow \mathbb{R}^m$  such that  $T^*(\widehat{z}) \in S(\widehat{z})$  for all  $\widehat{z} \in \mathbb{R}^m$ .*

*Proof.* Define  $g(\widehat{z}, z) := f(\theta, z) - \lambda \|z - \widehat{z}\|_2^2$  to simplify notation. Since  $f(\theta, z)$  is Lipschitz continuous in  $z$ , the quadratic penalty dominates as  $\|z\|_2 \rightarrow \infty$ , and thus the continuous function  $-g(\widehat{z}, z)$  is coercive in  $z$ . Therefore,  $g(\widehat{z}, z)$  attains its maximum over  $z$ , which in turn implies that the set of maximizers  $S(\widehat{z})$  is non-empty and closed for every  $\widehat{z} \in \mathbb{R}^m$ . To establish the measurability of the set-valued mapping  $S(\widehat{z})$ , note first that  $g(\widehat{z}, z)$  is continuous throughout  $\mathbb{R}^m \times \mathbb{R}^m$ . This implies via [57, Examples 14.15 & 14.29] that  $g(\widehat{z}, z)$  is a normal integrand. By [57, Theorem 14.37], the argmax mapping  $S(\widehat{z}) = \arg \max_{z \in \mathbb{R}^m} g(\widehat{z}, z)$  thus constitutes a Borel-measurable set-valued mapping. We can therefore invoke [57, Corollary 14.6] to conclude that  $S$  admits a measurable selection. That is, there exists a Borel-measurable function  $T^* : \mathbb{R}^m \rightarrow \mathbb{R}^m$  with  $T^*(\widehat{z}) \in S(\widehat{z})$  for all  $\widehat{z} \in \mathbb{R}^m$ .  $\square$

We now show that the adversarial risk  $F(\theta)$  can be expressed as in (1). This reformulation is reminiscent of duality results in Wasserstein DRO (see, e.g., [10, 50, 78]) and appeared first in [60, Proposition 1]. We provide a short alternative proof to keep this paper self-contained.

**Proposition B.2** (Adversarial risk). *The adversarial risk satisfies (1), the measure  $\mathbb{P}^* := T_{\#}^* \hat{\mathbb{P}}$  solves the inner maximization problem in (DRO), and  $\gamma^* := (T^*, \text{id})_{\#} \hat{\mathbb{P}}$  is an optimal coupling of  $\hat{\mathbb{P}}$  and  $\mathbb{P}^*$ .*

*Proof.* By rewriting the squared Wasserstein distance appearing in the definition of  $F(\theta)$  as an optimal transport problem over couplings  $\gamma$ , and by eliminating the candidate measure  $\mathbb{P}$  through its representation as the first marginal of  $\gamma$ , we obtain

$$F(\theta) = \sup_{\substack{\gamma \in \mathcal{P}(\mathbb{R}^m \times \mathbb{R}^m) \\ (\pi_2)_{\#} \gamma = \hat{\mathbb{P}}}} \int_{\mathbb{R}^m \times \mathbb{R}^m} \left( f(\theta, z) - \lambda \|z - \hat{z}\|_2^2 \right) d\gamma(z, \hat{z}). \quad (7)$$

Note that the constraint  $\mathbb{P} \in \mathcal{P}(\mathbb{R}^m)$  in the definition of  $F(\theta)$ , which requires  $\mathbb{P}$  to have finite second moments, translates to the requirement  $\gamma \in \mathcal{P}(\mathbb{R}^m \times \mathbb{R}^m)$  in (7), which requires the coupling  $\gamma$  to have finite second moments. Indeed, whenever  $\hat{\mathbb{P}}$  has finite second moments,  $\mathbb{P}$  has finite second moments if and only if every coupling of  $\mathbb{P}$  and  $\hat{\mathbb{P}}$  does. This follows from the relation

$$\mathbb{E}_{\gamma} [\|Z, \hat{Z}\|_2^2] = \mathbb{E}_{(\pi_1)_{\#} \gamma} [\|Z\|_2^2] + \mathbb{E}_{\hat{\mathbb{P}}} [\|\hat{Z}\|_2^2],$$

where  $\pi_1 : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$  is the coordinate projection defined by  $\pi_1(z, \hat{z}) = z$ . Next, we show that couplings with infinite second moments are strictly suboptimal in (7) and that  $\mathcal{P}(\mathbb{R}^m \times \mathbb{R}^m)$  can be relaxed to  $\mathcal{P}_0(\mathbb{R}^m \times \mathbb{R}^m)$ . To this end, we first prove that the optimal value of (7) is finite.

We can start by bounding the integrand in (7). As  $f(\theta, z)$  is  $L_z$ -Lipschitz continuous in  $z$ , we have

$$\begin{aligned} f(\theta, z) - \lambda \|z - \hat{z}\|_2^2 &\leq f(\theta, \hat{z}) + L_z \|z - \hat{z}\|_2 - \lambda \|z - \hat{z}\|_2^2 \\ &= f(\theta, \hat{z}) + \frac{L_z^2}{4\lambda} - \lambda \left( \|z - \hat{z}\|_2 - \frac{L_z}{2\lambda} \right)^2 \end{aligned}$$

for all  $z, \hat{z} \in \mathbb{R}^m$ . Using the elementary inequality  $(a - b)^2 \geq \frac{1}{2}a^2 - b^2$  for any  $a, b \in \mathbb{R}$ , we can further loosen this bound to isolate the quadratic transport cost. We thus obtain

$$\begin{aligned} f(\theta, z) - \lambda \|z - \hat{z}\|_2^2 &\leq f(\theta, \hat{z}) + \frac{L_z^2}{2\lambda} - \frac{\lambda}{2} \|z - \hat{z}\|_2^2 \\ &\leq f(\theta, 0) + L_z \|\hat{z}\|_2 + \frac{L_z^2}{2\lambda} - \frac{\lambda}{2} \|z - \hat{z}\|_2^2. \end{aligned}$$

Integrating both sides with respect to  $\gamma$  and recalling that the second marginal of  $\gamma$  equals  $\hat{\mathbb{P}}$  yields

$$\int (f(\theta, z) - \lambda \|z - \hat{z}\|_2^2) d\gamma(z, \hat{z}) \leq f(\theta, 0) + \frac{L_z^2}{2\lambda} + L_z \mathbb{E}_{\hat{\mathbb{P}}} [\|\hat{Z}\|_2] - \frac{\lambda}{2} \int \|z - \hat{z}\|_2^2 d\gamma(z, \hat{z}) \quad (8a)$$

$$\leq f(\theta, 0) + \frac{L_z^2}{2\lambda} + L_z \mathbb{E}_{\hat{\mathbb{P}}} [\|\hat{Z}\|_2]. \quad (8b)$$

As  $\hat{\mathbb{P}}$  has finite second moments, the inequality (8b) shows that the integral is bounded above by a finite constant uniformly across all couplings  $\gamma$ . Thus, we find  $F(\theta) < \infty$ . Conversely, the trivial coupling  $\gamma = (\text{id}, \text{id})_{\#} \hat{\mathbb{P}}$  is feasible in (7) and attains the finite objective value  $\mathbb{E}_{\hat{\mathbb{P}}} [f(\theta, \hat{Z})]$ . Consequently, we have  $F(\theta) > -\infty$ . In summary, we may conclude that  $F(\theta)$  is indeed finite.

By (8a), the objective function in (7) evaluates to  $-\infty$  for any coupling  $\gamma$  with infinite transport cost. Moreover,  $\gamma$  has infinite transport cost if and only if it has infinite second moments because

$$\int \frac{1}{2} \|z - \hat{z}\|_2^2 d\gamma(z, \hat{z}) \leq \int (\|z\|_2^2 + \|\hat{z}\|_2^2) d\gamma(z, \hat{z}) \leq \int (2\|z - \hat{z}\|_2^2 + 4\|\hat{z}\|_2^2) d\gamma(z, \hat{z}).$$

We may therefore relax  $\mathcal{P}(\mathbb{R}^m \times \mathbb{R}^m)$  to  $\mathcal{P}_0(\mathbb{R}^m \times \mathbb{R}^m)$  without increasing the supremum in (7).

Next, we can decompose any feasible coupling  $\gamma \in \mathcal{P}_0(\mathbb{R}^m \times \mathbb{R}^m)$  with respect to its second marginal. Indeed, by the disintegration theorem on Polish spaces [2, Theorem 5.3.1],  $\gamma$  decomposes as

$$\gamma(dz, d\hat{z}) = \nu_{\hat{z}}(dz) \otimes \hat{\mathbb{P}}(d\hat{z}) \quad (9)$$

for a  $\hat{\mathbb{P}}$ -a.e. uniquely defined Borel family  $(\nu_{\hat{z}})_{\hat{z} \in \mathbb{R}^m} \subseteq \mathcal{P}_0(\mathbb{R}^m)$ . Conversely, any such family defines a feasible  $\gamma$  via (9). Substituting (9) into the exact relaxation of (7) derived above and applying the Fubini-Tonelli theorem, which is justified in view of (8b), we obtain

$$F(\theta) = \sup_{(\nu_{\hat{z}})} \int_{\mathbb{R}^m} \left[ \int_{\mathbb{R}^m} (f(\theta, z) - \lambda \|z - \hat{z}\|_2^2) d\nu_{\hat{z}}(z) \right] d\hat{\mathbb{P}}(\hat{z}).$$

For each fixed  $\hat{z}$ , the inner integral admits the upper bound

$$\int_{\mathbb{R}^m} (f(\theta, z) - \lambda \|z - \hat{z}\|_2^2) d\nu_{\hat{z}}(z) \leq \max_{z \in \mathbb{R}^m} \{f(\theta, z) - \lambda \|z - \hat{z}\|_2^2\},$$

where equality holds whenever  $\nu_{\hat{z}}$  is supported on  $S(\hat{z})$  defined in (3). Note that the right-hand side of the above inequality is a pointwise maximum of continuous functions and thus Borel-measurable. By Proposition B.1, there exists a Borel selector  $T^*$  such that  $T^*(\hat{z})$  is optimal on the right-hand side for every  $\hat{z} \in \mathbb{R}^m$ . Clearly, the choice  $\nu_{\hat{z}}^* := \delta_{T^*(\hat{z})}$  defines a feasible disintegration. Thus, we find

$$F(\theta) = \mathbb{E}_{\hat{\mathbb{P}}} \left[ \max_{z \in \mathbb{R}^m} \{f(\theta, z) - \lambda \|z - \hat{Z}\|_2^2\} \right],$$

which establishes (1). This shows that  $\gamma^* := (\text{id}, T^*)_{\#} \hat{\mathbb{P}}$  solves (7) and that  $\mathbb{P}^* = (\pi_1)_{\#} \gamma^* = T_{\#} \hat{\mathbb{P}}$  solves the inner maximization problem in (DRO). We emphasize that  $\mathbb{P}^* \in \mathcal{P}(\mathbb{R}^m)$  has finite second moments for otherwise  $\gamma^*$  would have infinite second moments and an objective value of  $-\infty$ , thus contradicting its optimality. It remains to be shown that  $\gamma^*$  is an optimal coupling between  $\mathbb{P}^*$  and  $\hat{\mathbb{P}}$ . Suppose for contradiction that there is  $\tilde{\gamma} \in \Gamma(\mathbb{P}^*, \hat{\mathbb{P}})$  with

$$\int_{\mathbb{R}^m \times \mathbb{R}^m} \|z - \hat{z}\|_2^2 d\tilde{\gamma}(z, \hat{z}) < \int_{\mathbb{R}^m \times \mathbb{R}^m} \|z - \hat{z}\|_2^2 d\gamma^*(z, \hat{z}).$$

Since  $\tilde{\gamma}$  and  $\gamma^*$  share the first marginal  $\mathbb{P}^*$ , they yield identical loss integrals, that is,

$$\int f(\theta, z) d\tilde{\gamma}(z, \hat{z}) = \int f(\theta, z) d\gamma^*(z, \hat{z}) = \mathbb{E}_{\mathbb{P}^*}[f(\theta, Z)].$$

Consequently, we find

$$\int (f(\theta, z) - \lambda \|z - \hat{z}\|_2^2) d\tilde{\gamma}(z, \hat{z}) > \int (f(\theta, z) - \lambda \|z - \hat{z}\|_2^2) d\gamma^*(z, \hat{z}) = F(\theta),$$

which contradicts (7). We may thus conclude that

$$W^2(\mathbb{P}^*, \hat{\mathbb{P}}) = \int_{\mathbb{R}^m \times \mathbb{R}^m} \|z - \hat{z}\|_2^2 d\gamma^*(z, \hat{z}) = \mathbb{E}_{\hat{\mathbb{P}}}[\|T^*(\hat{Z}) - \hat{Z}\|_2^2],$$

which confirms that  $\gamma^*$  is indeed an optimal coupling between  $\mathbb{P}^*$  and  $\hat{\mathbb{P}}$ .  $\square$

The proof of Proposition B.2 shows that  $F(\theta)$  is guaranteed to be finite.

**Remark B.1** (Empirical measure). When  $\hat{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \delta_{\hat{z}_i}$  is the empirical measure corresponding to the training samples  $\hat{z}_1, \dots, \hat{z}_N \in \mathbb{R}^m$ , the disintegration (9) reduces to the elementary decomposition

$$\gamma(dz, d\hat{z}) = \frac{1}{N} \sum_{i=1}^N \nu_i(dz) \otimes \delta_{\hat{z}_i}(d\hat{z}) \quad \text{with} \quad \nu_i \in \mathcal{P}_0(\mathbb{R}^m) \quad \forall i \in [N],$$

in which case no measurable selection is required. Specifically, it suffices to pick a maximizer

$$z_i^* \in \arg \max_{z \in \mathbb{R}^m} \{f(\theta, z) - \lambda \|z - \hat{z}_i\|_2^2\}$$

for each  $i \in [N]$ . Proposition B.2 then reduces to the simpler statement

$$\sup_{\mathbb{P} \in \mathcal{P}(\mathbb{R}^m)} \left\{ \mathbb{E}_{\mathbb{P}}[f(\theta, Z)] - \lambda W^2(\mathbb{P}, \hat{\mathbb{P}}) \right\} = \frac{1}{N} \sum_{i=1}^N \sup_{z \in \mathbb{R}^m} \{f(\theta, z) - \lambda \|z - \hat{z}_i\|_2^2\},$$

and an optimizer of the inner maximization problem in (DRO) is given by  $\mathbb{P}^* = \frac{1}{N} \sum_{i=1}^N \delta_{z_i^*}$ .

## C Proofs

*Proof of Proposition 3.1.* We first show that  $g_t$  is an  $\varepsilon(\theta_t, T_t)$ -subgradient of  $F$  at  $\theta_t$ . As  $f(\theta, z)$  is convex in  $\theta$ , we may use the first-order condition of convexity to obtain

$$f(\theta, T_t(\hat{z})) - \lambda \|T_t(\hat{z}) - \hat{z}\|_2^2 \geq f(\theta_t, T_t(\hat{z})) - \lambda \|T_t(\hat{z}) - \hat{z}\|_2^2 + \langle \nabla_\theta f(\theta_t, T_t(\hat{z})), \theta - \theta_t \rangle$$

for any fixed  $\theta \in \Theta$  and all  $\hat{z} \in \mathbb{R}^m$ . Taking expectations with respect to  $\hat{\mathbb{P}}$  on both sides implies that

$$\mathbb{E}_{\hat{\mathbb{P}}}[f(\theta, T_t(\hat{Z})) - \lambda \|T_t(\hat{Z}) - \hat{Z}\|_2^2] \geq \mathbb{E}_{\hat{\mathbb{P}}}[f(\theta_t, T_t(\hat{Z})) - \lambda \|T_t(\hat{Z}) - \hat{Z}\|_2^2] + \langle g_t, \theta - \theta_t \rangle.$$

By the definition of the adversarial risk, the left-hand side is no larger than  $F(\theta)$ , and by the definition of the adversarial error, the first term on the right-hand side equals  $F(\theta_t) - \varepsilon(\theta_t, T_t)$ . Hence, we find

$$F(\theta) \geq F(\theta_t) + \langle g_t, \theta - \theta_t \rangle - \varepsilon(\theta_t, T_t). \quad (10)$$

As  $\theta \in \Theta$  was chosen freely, this shows that  $g_t$  is indeed an  $\varepsilon(\theta_t, T_t)$ -subgradient of  $F$  at  $\theta_t$ .

Next, we use the nonexpansiveness of the Euclidean projection operator ‘ $\text{Proj}_\Theta(\cdot)$ ’ and the update rule  $\theta_{t+1} = \text{Proj}_\Theta(\theta_t - \alpha g_t)$  of the projected gradient descent algorithm to conclude that

$$\|\theta_{t+1} - \theta^*\|_2^2 \leq \|\theta_t - \alpha g_t - \theta^*\|_2^2 \leq \|\theta_t - \theta^*\|_2^2 - 2\alpha \langle g_t, \theta_t - \theta^* \rangle + \alpha^2 \|g_t\|_2^2. \quad (11)$$

As  $g_t$  is defined at the expected value of a random variable whose norm is uniformly bounded by  $L_\theta$ , it is clear that  $\|g_t\|_2 \leq L_\theta$ . Combining (10) for  $\theta = \theta^*$  with (11) and rearranging terms we then find

$$F(\theta_t) - F(\theta^*) \leq \frac{1}{2\alpha} (\|\theta_t - \theta^*\|_2^2 - \|\theta_{t+1} - \theta^*\|_2^2) + \frac{\alpha L_\theta^2}{2} + \varepsilon(\theta_t, T_t).$$

Summing over  $t = 0, \dots, H-1$  and discarding the last term  $-\|\theta_H - \theta^*\|_2^2/(2\alpha) \leq 0$  in the telescoping sum then yields the estimate

$$\frac{1}{H} \sum_{t=0}^{H-1} (F(\theta_t) - F(\theta^*)) \leq \frac{\|\theta_0 - \theta^*\|_2^2}{2\alpha H} + \frac{\alpha L_\theta^2}{2} + \frac{1}{H} \sum_{t=0}^{H-1} \varepsilon(\theta_t, T_t).$$

The step size  $\alpha = \|\theta_0 - \theta^*\|_2 / (L_\theta \sqrt{H})$  balances the first two terms, ensuring that their sum reduces to  $L_\theta \|\theta_0 - \theta^*\|_2 / \sqrt{H}$ . Since  $F(\theta)$  is convex (as the pointwise supremum of convex functions), Jensen’s inequality implies that  $F(\bar{\theta}_H) \leq \frac{1}{H} \sum_{t=0}^{H-1} F(\theta_t)$ , which yields the claimed bound.  $\square$

*Proof of Proposition 3.2.* Since  $F(\theta)$  is finite by the proof of Proposition B.2, we may assume without loss of generality that  $T_{\#}\hat{\mathbb{P}}$  has finite second moments. Otherwise, the same proof yields  $\varepsilon(\theta, T) = \infty$ , in which case the claim holds trivially. Hence,  $T_{\#}\hat{\mathbb{P}}$  is feasible in the inner maximization problem in (DRO), and we have

$$F(\theta) \geq \mathbb{E}_{T_{\#}\hat{\mathbb{P}}}[f(\theta, Z)] - \lambda W^2(T_{\#}\hat{\mathbb{P}}, \hat{\mathbb{P}}).$$

Subtracting  $\mathbb{E}_{\hat{\mathbb{P}}}[f(\theta, T(\hat{Z})) - \lambda \|T(\hat{Z}) - \hat{Z}\|_2^2]$  from both sides and using the measure-theoretic change of variables formula  $\mathbb{E}_{T_{\#}\hat{\mathbb{P}}}[f(\theta, Z)] = \mathbb{E}_{\hat{\mathbb{P}}}[f(\theta, T(\hat{Z}))]$ , we obtain

$$\varepsilon(\theta, T) \geq \lambda \mathbb{E}_{\hat{\mathbb{P}}}[\|T(\hat{Z}) - \hat{Z}\|_2^2] - \lambda W^2(T_{\#}\hat{\mathbb{P}}, \hat{\mathbb{P}}) = \lambda \mathcal{M}_{\hat{\mathbb{P}}}(T),$$

where the equality follows from the definition of the Monge gap. This concludes the proof.  $\square$

*Remark C.1* (Monge gap of the empirical distribution). When  $\hat{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \delta_{\hat{z}_i}$  is the empirical distribution over  $N$  training samples, the Monge gap admits the more explicit representation

$$\mathcal{M}_{\hat{\mathbb{P}}}(T) = \max_{\sigma \in \mathcal{S}_N} \frac{2}{N} \sum_{i=1}^N \langle T(\hat{z}_{\sigma(i)}) - T(\hat{z}_i), \hat{z}_i \rangle,$$

where  $\mathcal{S}_N$  denotes the family of all permutations of  $N$  elements [68]. By Proposition 3.2, any lack of cyclical monotonicity of  $T$  on the support of  $\hat{\mathbb{P}}$  thus leads to a strictly positive lower bound on  $\varepsilon(\theta, T)$ .

*Proof of Proposition 3.3.* Select any  $k \in \mathbb{N}$  and any collection of points  $\{\hat{z}_i\}_{i=1}^k$  in  $\mathbb{R}^m$ . By the definition of the optimal adversarial map  $T^*$ , we have

$$T^*(\hat{z}_i) \in \arg \max_{z \in \mathbb{R}^m} \{f(\theta, z) - \lambda \|z - \hat{z}_i\|_2^2\} \quad \forall i \in [k].$$

Therefore, the following inequality holds for any permutation  $\sigma \in \mathcal{S}_k$ :

$$f(\theta, T^*(\hat{z}_i)) - \lambda \|T^*(\hat{z}_i) - \hat{z}_i\|_2^2 \geq f(\theta, T^*(\hat{z}_{\sigma(i)})) - \lambda \|T^*(\hat{z}_{\sigma(i)}) - \hat{z}_i\|_2^2 \quad \forall i \in [k].$$

Summing over  $i \in [k]$ , the terms involving  $f$  on both sides of the inequality coincide and thus cancel. As the points  $\{\hat{z}_i\}_{i=1}^k$  were chosen arbitrary, this implies that  $T^*$  is cyclically monotone on  $\mathbb{R}^m$ . In particular,  $T^*$  is cyclically monotone on the support of  $\mathbb{P}$ , which implies that  $\mathcal{M}_{\mathbb{P}}(T^*) = 0$ .  $\square$

*Proof of Proposition 4.1.* Since  $m = 1$ , the map  $T_{\text{pa}}(t, \cdot)$  is cyclical monotone if and only if it is monotonically non-decreasing [58, Chapter 2]. We therefore prove that the PA map  $T_{\text{pa}}(t, \cdot)$  is monotonically non-decreasing both in the continuous-time and discrete-time settings.

**Continuous-time PA.** By the Picard-Lindelöf theorem, the defining ODE of the continuous-time PA map has a unique solution that depends continuously on the initial conditions. This implies that  $T_{\text{pa}}(t, \hat{z})$  is continuously differentiable in  $t$  and continuous in  $\hat{z}$ . In the remainder, we fix  $t > 0$ .

Suppose now for the sake of contradiction that  $T_{\text{pa}}(t, \cdot)$  fails to be monotonically non-decreasing. Thus, there exist  $\hat{z}_1 < \hat{z}_2$  such that  $T_{\text{pa}}(t, \hat{z}_1) > T_{\text{pa}}(t, \hat{z}_2)$ . For ease of notation, we define the gap function  $\delta(t') := T_{\text{pa}}(t', \hat{z}_2) - T_{\text{pa}}(t', \hat{z}_1)$ , which is continuously differentiable. By assumption, we have  $\delta(0) = \hat{z}_2 - \hat{z}_1 > 0$  and  $\delta(t) < 0$ . The intermediate value theorem then implies that there exists  $t^* < t$  with  $\delta(t^*) = 0$  and  $\delta'(t^*) \leq 0$ . Next, an elementary calculation reveals that

$$\begin{aligned} \delta'(t^*) &= \frac{d}{dt} T_{\text{pa}}(t^*, \hat{z}_2) - \frac{d}{dt} T_{\text{pa}}(t^*, \hat{z}_1) \\ &= [\nabla_z f(\theta, T_{\text{pa}}(t^*, \hat{z}_2)) - 2\lambda(T_{\text{pa}}(t^*, \hat{z}_2) - \hat{z}_2)] \\ &\quad - [\nabla_z f(\theta, T_{\text{pa}}(t^*, \hat{z}_1)) - 2\lambda(T_{\text{pa}}(t^*, \hat{z}_1) - \hat{z}_1)]. \end{aligned}$$

Since  $T_{\text{pa}}(t^*, \hat{z}_1) = T_{\text{pa}}(t^*, \hat{z}_2)$ , the terms involving  $\nabla_z f$  and  $2\lambda T_{\text{pa}}(t^*, \cdot)$  cancel. Thus, we obtain

$$\delta'(t^*) = 2\lambda(\hat{z}_2 - \hat{z}_1) > 0,$$

which contradicts the defining property  $\delta'(t^*) \leq 0$  of the point  $t^*$ . Consequently, we may conclude that the continuous-time PA map  $T_{\text{pa}}(t, \cdot)$  must be monotonically non-decreasing.

**Discrete-time PA.** Fix  $\hat{z}_1 < \hat{z}_2$  and define the gap  $\delta_t := T_{\text{pa}}(t, \hat{z}_2) - T_{\text{pa}}(t, \hat{z}_1)$ . We use induction on  $t$  to show that  $\delta_t \geq 0$  for every  $t \in \mathbb{N}$ . As  $\hat{z}_1$  and  $\hat{z}_2$  with  $\hat{z}_1 < \hat{z}_2$  were chosen arbitrarily, this readily implies that  $T_{\text{pa}}(t, \hat{z})$  is monotonically non-decreasing in  $\hat{z}$ . The base case corresponding to  $t = 0$  holds trivially because  $T_{\text{pa}}(0, \cdot)$  is the identity map, which implies that  $\delta_0 = \hat{z}_2 - \hat{z}_1 > 0$ . As for the induction step, assume now that  $\delta_t \geq 0$ , and note that the gap obeys the recursion

$$\begin{aligned} \delta_{t+1} &= \delta_t + \eta [\nabla_z f(\theta, T_{\text{pa}}(t, \hat{z}_2)) - \nabla_z f(\theta, T_{\text{pa}}(t, \hat{z}_1))] - 2\lambda\eta\delta_t + 2\lambda\eta(\hat{z}_2 - \hat{z}_1) \\ &= (1 - 2\lambda\eta)\delta_t + \eta [\nabla_z f(\theta, T_{\text{pa}}(t, \hat{z}_2)) - \nabla_z f(\theta, T_{\text{pa}}(t, \hat{z}_1))] + 2\lambda\eta(\hat{z}_2 - \hat{z}_1). \end{aligned}$$

We have  $\nabla_z f(\theta, T_{\text{pa}}(t, \hat{z}_2)) - \nabla_z f(\theta, T_{\text{pa}}(t, \hat{z}_1)) \geq -L_{zz}(T_{\text{pa}}(t, \hat{z}_2) - T_{\text{pa}}(t, \hat{z}_1)) = -L_{zz}\delta_t$  because  $\nabla_z f(\theta, z)$  is  $L_{zz}$ -Lipschitz continuous in  $z$ . Substituting this bound into the recursion yields

$$\delta_{t+1} \geq (1 - \eta(L_{zz} + 2\lambda))\delta_t + 2\lambda\eta(\hat{z}_2 - \hat{z}_1).$$

The step size condition  $\eta \leq (L_{zz} + 2\lambda)^{-1}$  and the induction hypothesis ensure that the first term is nonnegative, while the second term is positive by construction. We may thus conclude that  $\delta_{t+1} \geq 0$ , which completes the induction step. Thus, the claim follows.  $\square$

*Proof of Lemma 4.1.* The proof widely parallels the second part of the proof of Proposition 3.3, with the optimality of  $T^*$  replaced by the assignment-stationarity inequality (5).  $\square$

## D Properties of MPA

In this appendix, we establish two key properties of MPA. This discussion focuses on the inner loop of Algorithm 1, which evaluates the adversarial risk of a fixed model parameter  $\theta$ . First, we prove that an ideal variant of MPA with  $K = \infty$  is guaranteed to terminate in finitely many rounds (Section D.1). Second, we prove that if the batch size  $B$  is sufficiently large, then MPA is guaranteed to find a near-optimal adversarial map with high probability (Section D.2). To showcase the benefits of MPA *vis-à-vis* PA, we also demonstrate that the outer loop of Algorithm 1 may get trapped at the initial iterate unless the current adversarial map is cyclically monotone (Section D.3).

### D.1 Finite-time Termination of MPA

Recall our standing assumption that  $f(\theta, z)$  is differentiable and  $L_z$ -Lipschitz continuous with respect to  $z$ . Some of the results in this section require one or both of the following additional assumptions.

**Assumption D.1.** The gradient  $\nabla_z f(\theta, z)$  is  $L_{zz}$ -Lipschitz continuous in  $z$  uniformly over all  $\theta \in \Theta$ .

**Assumption D.2.** The loss function  $f(\theta, z)$  is real analytic in  $z$  uniformly across all  $\theta \in \Theta$ .

For example, Assumption D.2 is satisfied when  $f$  is the cross-entropy loss of a multilayer perceptron, which consists of compositions of linear transformations and analytic activation functions, such as sigmoid, GELU, or Softplus. Throughout this section we focus on one epoch of Algorithm 1 and thus consider a fixed batch of  $B$  training samples  $\{\hat{z}_i\}_{i=1}^B$  generated from  $\hat{\mathbb{P}}$ .

Focusing on one epoch also means that  $\lambda > 0$  and  $\theta \in \Theta$  are kept fixed. For ease of notation, we then define

$$f_i(z) := f(\theta, z) - \lambda \|z - \hat{z}_i\|_2^2 \quad (12)$$

as the penalized objective function for each sample  $i \in [B]$ .

In the following we use boldface symbols to denote stacked vectors of per-sample quantities, such as  $\mathbf{z} = (z_1, \dots, z_B)$ . Accordingly, we assemble the per-sample penalized objectives into a vector field  $\mathbf{f} := (f_1, \dots, f_B)$ , such that  $\mathbf{f}(\mathbf{z}) = (f_1(z_1), \dots, f_B(z_B))$ .

Since the objective function in (2) depends on the transport map  $T \in \mathcal{T}$  only through its evaluations at the empirical samples, we introduce the shorthand  $z_i := T(\hat{z}_i)$ . This allows us to identify the map  $T$  restricted to the empirical samples with the tuple  $\mathbf{z} \in \mathbb{R}^{Bm}$ . Finally, for each  $i \in [B]$ , let  $\mathcal{M}_i \subseteq \mathbb{R}^m$  denote the set of stationary points of  $f_i$ , and define the joint set  $\mathcal{M} := \mathcal{M}_1 \times \dots \times \mathcal{M}_B$ .

**Lemma D.1.** *If Assumption D.2 holds, then  $\mathcal{M}_i$  is a compact analytic subset of  $\mathbb{R}^m$  for each  $i \in [B]$ .*

*Proof.* We first show that  $\mathcal{M}_i$  is compact. By definition, any  $z \in \mathcal{M}_i$  satisfies

$$0 = \nabla_z f_i(z) = \nabla_z f(\theta, z) - 2\lambda(z - \hat{z}_i),$$

which implies  $2\lambda\|z - \hat{z}_i\|_2 = \|\nabla_z f(\theta, z)\|_2$ . Lipschitz continuity of  $f(\theta, \cdot)$  ensures that  $\|\nabla_z f(\theta, \cdot)\|_2 \leq L_z$ . As  $\lambda > 0$ , we thus find  $\|z - \hat{z}_i\|_2 \leq \frac{L_z}{2\lambda}$ . Consequently,  $\mathcal{M}_i$  is bounded. Because  $\nabla_z f_i(z)$  is continuous, its zero-set  $\mathcal{M}_i$  is also closed, making  $\mathcal{M}_i$  a compact subset of  $\mathbb{R}^m$ .

Next, note that the penalized objective  $f_i(z)$  inherits real analyticity from  $f(\theta, \cdot)$ . Therefore, the squared norm of its gradient,  $\|\nabla_z f_i(z)\|_2^2$ , is also a real analytic function on  $\mathbb{R}^m$ . The set  $\mathcal{M}_i$  of stationary points can be uniquely identified as the zero locus of this single analytic function, that is,

$$\mathcal{M}_i = \{z \in \mathbb{R}^m : \|\nabla_z f_i(z)\|_2^2 = 0\}.$$

By definition, this makes  $\mathcal{M}_i$  an analytic subset of  $\mathbb{R}^m$  [40, Chapter 6].  $\square$

**Corollary D.1.** *If Assumption D.2 holds, then  $\mathcal{M}_i$  has finitely many connected components, and the set of critical values  $f_i(\mathcal{M}_i)$  is finite for each  $i \in [B]$ .*

*Proof.* By Lemma D.1,  $\mathcal{M}_i$  is a compact analytic subset of  $\mathbb{R}^m$ , which inherently belongs to the broader class of semianalytic sets. A fundamental topological property of semianalytic sets is that they are locally connected and that the family of their connected components is locally finite [40, Theorem 6.5.12]. Local finiteness means that for every point  $z \in \mathcal{M}_i$  there exists an open neighborhood  $U_z$  that intersects only a finite number of connected components. The collection of

all such neighborhoods,  $\{U_z\}_{z \in \mathcal{M}_i}$ , naturally forms an open cover of  $\mathcal{M}_i$ . Because  $\mathcal{M}_i$  is compact, this open cover admits a finite subcover, say  $\{U_{z_1}, \dots, U_{z_K}\}$ . Since the entirety of  $\mathcal{M}_i$  lies within the union of these  $K$  finite neighborhoods, and each neighborhood intersects only finitely many components, the total number of connected components in  $\mathcal{M}_i$  is finite.

Let  $C \subseteq \mathcal{M}_i$  be one such connected component. By [40, Theorem 6.5.12], the connected component  $C$  is itself a semianalytic set. A standard result in semianalytic geometry is that any two points of a connected semianalytic set can be joined by a semianalytic arc, which is real analytic except at finitely many points [28]. Therefore, for any two points  $z, z' \in C$ , there exists a piecewise smooth path  $\gamma : [0, 1] \rightarrow C$  connecting them. Because every point on  $\gamma$  belongs to  $\mathcal{M}_i$ , we have  $\nabla_z f_i(\gamma(t)) = 0$  for all  $t \in [0, 1]$ . By the fundamental theorem of calculus along curves, we thus have

$$f_i(z') - f_i(z) = \int_0^1 \langle \nabla_z f_i(\gamma(t)), \frac{d}{dt} \gamma(t) \rangle dt = 0.$$

Therefore,  $f_i$  is constant on  $C$ . Because  $\mathcal{M}_i$  consists of a finite number of connected components, it follows that the set of critical values  $f_i(\mathcal{M}_i)$  is finite.  $\square$

We now show that an ideal instance of MPA (Algorithm 1 with  $K = \infty$ ) terminates after finitely many rounds. The argument relies on the properties of certain ascent and reassignment operators.

**Definition D.1** (Ascent Operators). For a fixed step size  $0 < \eta < 2/(L_{zz} + 2\lambda)$ , we define the

- (i)  **$i$ -th one-step ascent operator:**  $\mathcal{A}_i : \mathbb{R}^m \rightarrow \mathbb{R}^m, z \mapsto z + \eta \nabla f_i(z)$ ;
- (ii)  **$i$ -th  $t$ -step ascent operator:**  $\mathcal{A}_i^t := \mathcal{A}_i \circ \mathcal{A}_i^{t-1}$  for all  $t \in \mathbb{N}$ , where  $\mathcal{A}_i^0 := \text{id}$ ;
- (iii)  **$i$ -th asymptotic ascent operator:**  $\mathcal{A}_i^\infty : \mathbb{R}^m \rightarrow \mathbb{R}^m, z \mapsto \lim_{t \rightarrow \infty} \mathcal{A}_i^t(z)$ ;
- (iv)  **$t$ -step ascent operator:**  $\mathcal{A}^t : \mathbb{R}^{Bm} \rightarrow \mathbb{R}^{Bm}, \mathbf{z} \mapsto (\mathcal{A}_1^t(z_1), \dots, \mathcal{A}_B^t(z_B))$  for all  $t \in \mathbb{N} \cup \{\infty\}$ .

The  $t$ -step ascent operator implements one full round of  $t$  ascent steps in parallel across all  $B$  samples. Note that the discrete-time PA map satisfies  $T_{\text{pa}}(t, \hat{z}_i) = \mathcal{A}_i^t(\hat{z}_i)$  and is thus fully determined by the  $t$ -step ascent operator  $\mathcal{A}_i^t$ . However,  $\mathcal{A}_i^t$  itself is more general because its input  $z_i$  may differ from the empirical sample  $\hat{z}_i$ , which is hard-coded in the definition of the penalized objective  $f_i$ . The next lemma shows that all asymptotic ascent operators are well-defined.

**Lemma D.2.** *If Assumptions D.1 and D.2 hold and  $0 < \eta < 2/(L_{zz} + 2\lambda)$ , then the sequence  $z^t = \mathcal{A}_i^t(z^0)$  converges to a stationary point  $z^* \in \mathcal{M}_i$  for every  $i \in [N]$  and initialization  $z^0 \in \mathbb{R}^m$ .*

*Proof.* Fix  $i \in [N]$ , and set  $L := L_{zz} + 2\lambda$ . Since  $\nabla f_i$  is  $L$ -Lipschitz by Assumption D.1, the ascent lemma [9, Proposition 6.1.2] and the update rule  $z^{t+1} = \mathcal{A}_i(z^t) = z^t + \eta \nabla f_i(z^t)$  yield

$$\begin{aligned} f_i(z^{t+1}) - f_i(z^t) &\geq \left( \frac{1}{\eta} - \frac{L}{2} \right) \|z^{t+1} - z^t\|_2^2 \\ &= \left( 1 - \frac{\eta L}{2} \right) \|\nabla f_i(z^t)\|_2 \|z^{t+1} - z^t\|_2. \end{aligned} \tag{13}$$

As  $\eta < 2/L$ , a standard telescoping sum argument implies that  $f_i(z^t) \geq f_i(z^0)$  for all  $t \geq 0$ . Moreover, the  $L_z$ -Lipschitz continuity of  $f(\theta, \cdot)$  implies that  $f_i(z)$  is coercive, i.e.,  $f_i(z) \rightarrow -\infty$  as  $\|z\|_2 \rightarrow \infty$ . Thus, the superlevel set  $\{z \in \mathbb{R}^m \mid f_i(z) \geq f_i(z^0)\}$  is compact and contains the entire sequence  $\{z^t\}_{t \in \mathbb{N}}$ . To establish convergence, we note that the sequence satisfies the strong ascent conditions of [1, Definition 3.1]. This is an immediate consequence of (13). In particular, the inequality in the first line of (13) ensures that  $f_i(z^{t+1}) = f_i(z^t)$  implies  $z^{t+1} = z^t$ . Because the sequence  $\{z^t\}_{t \in \mathbb{N}}$  is bounded and  $f_i$  is real analytic by Assumption D.2, [1, Theorem 3.2] guarantees that the sequence converges and thus has a limit  $z^*$ . Finally, the update rule ensures that

$$\nabla f_i(z^t) = \frac{z^{t+1} - z^t}{\eta} \rightarrow 0 \quad \text{as } t \rightarrow \infty.$$

By the continuity of  $\nabla f_i$ , we thus have  $\nabla f_i(z^*) = 0$ . This implies that  $z^* \in \mathcal{M}_i$ .  $\square$

**Lemma D.2** shows that the asymptotic ascent operator  $\mathcal{A}_i^\infty$  maps any input  $z^0$  to a unique stationary point  $z^*$  and is therefore indeed well-defined for every  $i \in [B]$ . Hence,  $\mathcal{A}^\infty$  is well-defined, too.

To mathematically analyze the MPA algorithm, we also define a reassignment operator.

**Definition D.2** (Reassignment Operator). The reassignment operator  $\mathcal{R} : \mathbb{R}^{Bm} \rightarrow \mathbb{R}^{Bm}$  is defined through  $\mathcal{R}(\mathbf{z}) = \mathbf{z}'$ , where  $z'_i = z_{\sigma(i)}$  and  $\sigma(i) \in \arg \max_{j \in [B]} f_i(z_j)$  for all  $i \in [B]$ . Ties among the maximizers are resolved using a self-favoring rule, i.e. if  $i \in \arg \max_j f_i(z_j)$ , we set  $\sigma(i) = i$ . Otherwise, ties are broken by choosing the maximizer with the smallest index.

Given a list  $\mathbf{z} = (z_1, \dots, z_B)$  of  $B$  samples, the reassignment operator  $\mathcal{R}$  produces a new list  $\mathbf{z}' = (z'_1, \dots, z'_B)$ , where, for each  $i \in [B]$ , the sample  $z'_i$  is a maximizer of  $f_i$  over the set  $\{z_1, \dots, z_B\}$ . Equivalently, for every  $i \in [B]$ ,  $z'_i$  solves the restriction of problem (3) with  $\hat{z} = z_i$ , in which the uncountable original feasible set  $\mathbb{R}^m$  is replaced by the finite set  $\{z_1, \dots, z_B\}$ .

By the construction of the penalized objective functions  $f_i, i \in [B]$ , a map  $T$  is assignment-stationary on  $\{\hat{z}_i\}_{i=1}^B$  in the sense of **Definition 4.1** if and only if its outputs  $z_i := T(\hat{z}_i), i \in [B]$ , satisfy  $f_i(z_i) \geq f_i(z_j)$  for all  $i, j \in [B]$ . Thus, each output  $z_i$  maximizes the corresponding objective function  $f_i$  over the set of outputs  $\{z_1, \dots, z_B\}$ . By a slight abuse of terminology, we call the list  $\mathbf{z} = (z_1, \dots, z_B)$  of output samples *assignment-stationary*. We call  $\mathbf{z}$  simply *stationary* if  $\mathbf{z} \in \mathcal{M}$ .

**Lemma D.3** (Properties of the Ascent and Reassignment Operators). *If **Assumptions D.1** and **D.2** hold and  $0 < \eta < 2/(L_{zz} + 2\lambda)$ , the ascent and reassignment operators have the following properties.*

- (i) **Stationarity:**  $\mathcal{A}^\infty(\mathbf{z})$  is stationary, and  $\mathcal{R}(\mathbf{z})$  is assignment-stationary for all  $\mathbf{z} \in \mathbb{R}^{Bm}$ .
- (ii) **Idempotence:**  $\mathcal{A}^\infty \circ \mathcal{A}^\infty = \mathcal{A}^\infty$  and  $\mathcal{R} \circ \mathcal{R} = \mathcal{R}$ .
- (iii) **Strict monotonicity:**  $\mathbf{f}(\mathcal{A}^t(\mathbf{z})) \geq \mathbf{f}(\mathbf{z})$  and  $\mathbf{f}(\mathcal{R}(\mathbf{z})) \geq \mathbf{f}(\mathbf{z})$  for all  $\mathbf{z} \in \mathbb{R}^{Bm}$  and  $t \in \mathbb{N} \cup \{\infty\}$ , and these inequalities are strict in at least one coordinate if  $\mathcal{A}^t(\mathbf{z}) \neq \mathbf{z}$  or  $\mathcal{R}(\mathbf{z}) \neq \mathbf{z}$ , respectively.

*Proof.* **Stationarity:** Fix an arbitrary  $\mathbf{z} \in \mathbb{R}^{Bm}$ . **Lemma D.2** readily implies that  $\mathcal{A}^\infty(\mathbf{z}) \in \mathcal{M}$  is stationary. Similarly, the definition of the reassignment operator readily implies that  $\mathbf{z}' := \mathcal{R}(\mathbf{z})$  is such that  $z'_i$  maximizes  $f_i$  over the candidate set  $\{z_1, \dots, z_B\}$ . As  $\{z'_1, \dots, z'_B\} \subseteq \{z_1, \dots, z_B\}$ , it immediately follows that  $f_i(z'_i) \geq f_i(z'_j)$  for all  $i, j \in [B]$ , meaning that  $\mathbf{z}'$  is assignment-stationary.

**Idempotence:** Set  $z_i^* := \mathcal{A}_i^\infty(z_i)$ . As  $\nabla f_i(z_i^*) = 0$  by virtue of **Lemma D.2**, the point  $z_i^*$  is a fixed point of the one-step operator  $\mathcal{A}_i$ , meaning that  $\mathcal{A}_i^\infty(z_i^*) = z_i^*$ . As this is true for every  $i \in [B]$ , we have  $\mathcal{A}^\infty \circ \mathcal{A}^\infty = \mathcal{A}^\infty$ . as for the reassignment operator, set  $\mathbf{z}' := \mathcal{R}(\mathbf{z})$ . As  $\mathbf{z}'$  is assignment-stationary, we have  $i \in \arg \max_{j \in [B]} f_i(z'_j)$  for all  $i \in [B]$ . The self-favoring tie-breaking rule in **Definition D.2** thus dictates that  $\mathcal{R}(\mathbf{z}')_i = z'_i$  for every  $i \in [B]$ . Consequently, we have  $\mathcal{R} \circ \mathcal{R} = \mathcal{R}$ .

**Strict monotonicity:** Equation (13) in the proof of **Lemma D.2** and the assumptions on  $\eta$  imply that  $f_i(\mathcal{A}_i^t(z_i)) \geq f_i(z_i)$  and that this inequality is strict if  $\mathcal{A}_i^t(z_i) \neq z_i$ , for every  $i \in [B]$ . Hence, the  $t$ -step ascent operator  $\mathcal{A}^t$  is strictly monotone for every  $t \in \mathbb{N} \cup \{\infty\}$ . The reassignment operator  $\mathcal{R}$  is strictly monotone by construction. In particular, if  $\mathbf{z}' := \mathcal{R}(\mathbf{z})$  and there exists  $i \in [B]$  with  $z'_i \neq z_i$ , then the self-favoring tie-breaking rule ensures that  $f_i(z'_i) > f_i(z_i)$ .  $\square$

The ascent and reassignment operators enable us to provide an abstract mathematical description of the MPA algorithm. Each epoch is initialized at the empirical samples  $\mathbf{z}^{(0)} := (\hat{z}_1, \dots, \hat{z}_B)$ , and the iterates  $\mathbf{z}^{(r)} := \mathcal{A}^K(\mathcal{R}(\mathbf{z}^{(r-1)}))$  are constructed recursively for all rounds  $r \in [R]$ . Recall that this approach simultaneously solves  $B$  coupled nonconvex optimization problems—one for each empirical sample  $\hat{z}_i, i \in [B]$ . The strict monotonicity of the ascent and reassignment operators implies that, unless the MPA algorithm has converged to a fixed point of  $\mathcal{A}^K \circ \mathcal{R}$ , at least one of the  $B$  objective functions is improved and *no* objective function deteriorates in each round.

A fundamental question is whether the inner iterates produced by **Algorithm 1** eventually stabilize when the algorithm is run over  $R = \infty$  rounds. Specifically, it is of interest to investigate whether there exists a finite round  $R$  after which the iterates remain unchanged. If so, we say that the algorithm *terminates in finite time*. For finite  $K$ , finite-time termination can occur only in exceptional cases, since performing additional gradient ascent steps generically leads to a strict improvement. For  $K = \infty$ , by contrast, the algorithm is guaranteed to terminate after finitely many rounds. This follows from the idempotence of both the asymptotic ascent operator  $\mathcal{A}^\infty$  and the reassignment

operator  $\mathcal{R}$  (see [Theorem D.1](#) below). Although performing infinitely many gradient ascent steps is not computationally feasible, the resulting *exact MPA algorithm* is nevertheless of theoretical interest. Besides being more amenable to mathematical analysis, it provides an accurate approximation for the practical algorithm with a finite (sufficiently large) number of gradient ascent steps per round.

In any epoch, the exact MPA algorithm terminates at the first round  $r \in \mathbb{N}$  with  $\mathcal{A}^\infty(\mathcal{R}(\mathbf{z}^{(r)})) = \mathbf{z}^{(r)}$ . Since every iterate  $\mathbf{z}^{(r)}$  constitutes a fixed point of  $\mathcal{A}^\infty$ , this termination condition can be expressed more concisely as  $\mathcal{R}(\mathbf{z}^{(r)}) = \mathbf{z}^{(r)}$ . The output of the exact MPA algorithm is thus guaranteed to be stationary as well as assignment-stationary. These insights are formalized in the following theorem.

**Theorem D.1** (Finite termination of exact MPA). *If [Assumptions D.1](#) and [D.2](#) hold,  $0 < \eta < 2/(L_{zz} + 2\lambda)$  and  $K = \infty$ , then the full-batch MPA algorithm terminates after  $R \leq 1 + \sum_{i=1}^N (|f_i(\mathcal{M}_i)| - 1)$  rounds in each epoch. The returned output  $\mathbf{z}^{(R)}$  is both stationary and assignment-stationary.*

*Proof.* The first iterate  $\mathbf{z}^{(1)} = \mathcal{A}^\infty(\mathcal{R}(\mathbf{z}^{(0)}))$  is stationary by [Lemma D.3 \(i\)](#), and thus [Corollary D.1](#) implies that  $f_i(z_i^{(1)})$  belongs to the finite set  $f_i(\mathcal{M}_i)$  for every  $i \in [B]$ . For any subsequent non-terminal round  $r \geq 2$ , we have  $\mathbf{z}^{(r)} \neq \mathbf{z}^{(r-1)}$ . The definition of the exact MPA algorithm then yields

$$\mathbf{f}(\mathbf{z}^{(r)}) = \mathbf{f}(\mathcal{A}^\infty(\mathcal{R}(\mathbf{z}^{(r-1)}))) \geq \mathbf{f}(\mathbf{z}^{(r-1)}),$$

where the inequality is strict for at least one coordinate  $i \in [B]$  by virtue of [Lemma D.3 \(iii\)](#). Because both  $\mathbf{z}^{(r)}$  and  $\mathbf{z}^{(r-1)}$  belong to  $\mathcal{M}$ , this represents a strict jump between two values inside  $f_i(\mathcal{M}_i)$ . For each coordinate,  $f_i$  can jump to a strictly higher critical value at most  $|f_i(\mathcal{M}_i)| - 1$  times. Summing this bound over all  $N$  coordinates, and adding 1 to account for the first round, yields the desired upper bound on the total number of rounds  $R$  until termination:  $1 + \sum_{i=1}^N (|f_i(\mathcal{M}_i)| - 1)$ .

The terminal iterate  $\mathbf{z}^{(R)}$  is generated by the exact ascent operator  $\mathcal{A}^\infty$ , which guarantees stationarity by [Lemma D.3 \(i\)](#). The termination condition thus simplifies to  $\mathcal{R}(\mathbf{z}^{(R)}) = \mathbf{z}^{(R)}$ , which ensures via [Lemma D.3 \(i\)](#) that  $\mathbf{z}^{(R)}$  is also assignment-stationary. This observation completes the proof.  $\square$

*Remark D.1* (Cyclical monotonicity). In practice, an epoch of [Algorithm 1](#) may be interrupted before a fixed point of  $\mathcal{A}^K \circ \mathcal{R}$  is reached. In this case, the last iterate may fail to be assignment-stationary. Indeed, while  $\mathcal{R}$  induces assignment-stationarity,  $\mathcal{A}^K$  can fail to preserve assignment-stationarity even if  $K = \infty$ . In practice, it is therefore expedient to conclude the MPA algorithm with an extra reassignment step to enforce assignment-stationarity and cyclical monotonicity. We also emphasize that assignment-stationarity is enforced only locally on the given mini-batch.

## D.2 Global Optimality Guarantees for MPA

The goal of this section is to show that, for any fixed model parameter  $\theta \in \Theta$  and tolerance  $\epsilon > 0$ , the inner loop of [Algorithm 1](#) computes the adversarial risk  $F(\theta)$  with absolute error at most  $\epsilon$  with high probability, provided that the batch size  $B$  is sufficiently large. Thus, MPA essentially solves the nonconvex functional optimization problem (2) to global optimality irrespective of the penalty parameter  $\lambda > 0$ . In contrast, recall that the state-of-the-art PA algorithm by Sinha et al. [60] provides such a global optimality guarantee only for unrealistically large penalty parameters  $\lambda > L_{zz}/2$ .

Since our analysis of MPA is probabilistic, we model the empirical samples explicitly as independent and identically distributed (i.i.d.) random vectors denoted by  $\hat{Z}_1, \dots, \hat{Z}_B$ . We study a generic instance of MPA with an arbitrary number of local ascent steps  $K \in \mathbb{N} \cup \{\infty\}$  and rounds  $R \in \mathbb{N}$  and with any step size  $\eta \in (0, 2/(L_{zz} + 2\lambda))$ ; see [Algorithm 1](#). Recall from [Section 4](#) that MPA generates an *implicit* adversarial map  $T_{\text{mpa}}$ , which is only defined on the given batch of empirical samples. Using the notation from [Section D.1](#), this map can be constructed as follows. Initialized at  $\mathbf{Z}^{(0)} := (\hat{Z}_1, \dots, \hat{Z}_B)$ , MPA computes iterates  $\mathbf{Z}^{(r)} := \mathcal{A}^K(\mathcal{R}(\mathbf{Z}^{(r-1)}))$  for all  $r \in [R]$  and outputs  $\mathcal{R}(\mathbf{Z}^{(R)})$ , where  $\mathcal{A}^K$  and  $\mathcal{R}$  represent the ascent and reassignment operators of [Definitions D.1](#) and [D.2](#). Thus, we can define  $T_{\text{mpa}}(\hat{Z}_i)$  as the  $i$ -th sample in the last batch of adversarial samples  $\mathcal{R}(\mathbf{Z}^{(R)})$ .

Throughout this section, we continue to denote the penalized objective associated with the  $i$ -th empirical sample by  $f_i$  as in (12), bearing in mind that this sample now constitutes a random vector. We further assume that the empirical samples are governed by a sufficiently regular distribution.

**Assumption D.3.** The empirical samples  $\widehat{Z}_1, \dots, \widehat{Z}_B$  are drawn independently from a distribution  $\mathbb{P}_0 \in \mathcal{P}(\mathbb{R}^m)$  that admits an everywhere positive density with respect to Lebesgue measure on  $\mathbb{R}^m$ .

In the following, we set  $\rho^2 := \mathbb{E}_{\mathbb{P}_0}[\|Z\|_2^2]$  and note that  $\rho^2 < \infty$  because  $\mathbb{P}_0 \in \mathcal{P}(\mathbb{R}^m)$ . **Assumption D.3** implies that any optimal adversarial map is square-integrable with respect to  $\mathbb{P}_0$ .

**Lemma D.4** (Square integrability of the optimal adversarial map). *If **Assumption D.3** holds and  $M^2 := 2\rho^2 + L_z^2/(2\lambda^2)$ , any optimal adversarial map  $T^*$  in the sense of **Proposition B.1** satisfies*

$$\mathbb{E}_{\mathbb{P}_0}[\|T^*(\widehat{Z}_i)\|_2^2] \leq M^2 \quad \forall i \in [B].$$

*Proof.* By construction, the random vector  $Z^* := T^*(\widehat{Z}_i)$  is a global maximizer of the random function  $f_i$  pointwise for every uncertainty realization. The stationarity condition  $\nabla_z f(\theta, Z^*) = 2\lambda(Z^* - \widehat{Z}_i)$  thus implies via the proof of **Lemma D.1** that  $\|Z^* - \widehat{Z}_i\|_2 \leq \frac{L_z}{2\lambda}$ . Hence, we obtain

$$\|Z^*\|_2^2 \leq \left(\|\widehat{Z}_i\|_2 + \|Z^* - \widehat{Z}_i\|_2\right)^2 \leq \left(\|\widehat{Z}_i\|_2 + \frac{L_z}{2\lambda}\right)^2 \leq 2\|\widehat{Z}_i\|_2^2 + \frac{L_z^2}{2\lambda^2}.$$

The claim then follows by replacing  $Z^*$  with its definition  $T^*(\widehat{Z}_i)$ , taking expectations with respect to  $\mathbb{P}_0$  on both sides of the resulting inequality, and invoking the definition of  $M^2$ .  $\square$

To establish the claimed optimality guarantee of MPA, we fix any tolerance  $\epsilon > 0$  and optimal adversarial map  $T^*$  in the sense of **Proposition B.1**, and we introduce two positive radii

$$r_\epsilon := \sqrt{\frac{\epsilon}{L_{zz} + 2\lambda}} \quad \text{and} \quad R_\epsilon := \frac{2L_z M}{\sqrt{2\lambda\epsilon}}, \quad (14)$$

where  $M^2$  stands for the second-order moment bound from **Lemma D.4**. Using this notation, we partition the index set  $[B]$  of the empirical samples into three disjoint random sets

$$O_\epsilon := \{i \in [B] : \|T^*(\widehat{Z}_i)\|_2 > R_\epsilon\}, \quad (15a)$$

$$U_\epsilon := \{i \in [B] : i \notin O_\epsilon, \|\widehat{Z}_j - T^*(\widehat{Z}_i)\|_2 > r_\epsilon \forall j \neq i\}, \quad (15b)$$

$$C_\epsilon := \{i \in [B] : i \notin O_\epsilon, i \notin U_\epsilon\}. \quad (15c)$$

Samples with indices in  $O_\epsilon$  are referred to as ‘outliers,’ whereas samples with indices in  $U_\epsilon$  and  $C_\epsilon$  are termed ‘uncovered inliers’ and ‘covered inliers’, respectively. We will see below that the numbers of outliers and uncovered inliers are small with high probability and that if  $\widehat{z}$  is any covered inlier, then there exists an empirical sample that is near-optimal in (3). Together, these insights imply that  $T_{\text{mpa}}$  is near-optimal in problem (2) if  $\widehat{\mathbb{P}} = \frac{1}{B} \sum_{i=1}^B \delta_{\widehat{Z}_i}$  is the empirical distribution on the given batch.

We first prove that if  $B$  is large, then the fraction of outliers is unlikely to exceed  $\mathcal{O}(\epsilon)$ .

**Lemma D.5** (Outliers are rare). *If **Assumption D.3** holds, then we have*

$$\mathbb{P}_0^B \left( \frac{|O_\epsilon|}{B} > \frac{\lambda\epsilon}{L_z^2} \right) \leq \exp(-B\lambda\epsilon/6L_z^2).$$

*Proof.* The indicator variables  $Y_i := \mathbf{1}_{\{\|T^*(\widehat{Z}_i)\|_2 > R_\epsilon\}}$ ,  $i \in [B]$ , form a sequence of i.i.d. Bernoulli trials. We denote their common success probability by  $p$  and assume temporarily that  $p > 0$ . We also set  $\bar{p} := \lambda\epsilon/(2L_z^2)$ . Markov’s inequality together with **Lemma D.4** then implies that

$$p = \mathbb{E}_{\mathbb{P}_0}[Y_i] = \mathbb{P}_0 \left( \|T^*(\widehat{Z}_i)\|_2^2 > R_\epsilon^2 \right) \leq \frac{M^2}{R_\epsilon^2} = \bar{p},$$

where the last equality exploits the definition of  $R_\epsilon$  in (14). Re-expressing the number of outliers  $|O_\epsilon|$  as the sum of all Bernoulli trials and setting  $\delta := 2\bar{p}/p - 1 \geq 1$  therefore yields

$$\begin{aligned} \mathbb{P}_0^B \left( \frac{|O_\epsilon|}{B} > \frac{\lambda\epsilon}{L_z^2} \right) &= \mathbb{P}_0^B \left( \sum_{i=1}^B Y_i > 2B\bar{p} \right) = \mathbb{P}_0^B \left( \sum_{i=1}^B Y_i > (1 + \delta)Bp \right) \\ &\leq \left( \frac{e^\delta}{(1 + \delta)^{1+\delta}} \right)^{Bp} \leq e^{-Bp\delta/3} \leq e^{-B\bar{p}/3}. \end{aligned}$$

Here, the first two equalities follow from the definitions of  $\bar{p}$  and  $\delta$ , respectively. The first inequality exploits Chernoff's inequality for sums of independent Bernoulli random variables [69, Theorem 2.3.1], and the second inequality holds because  $(1 + \delta) \ln(1 + \delta) \geq 4\delta/3$  for all  $\delta \geq 1$ . The last inequality holds because  $p\delta = 2\bar{p} - p \geq \bar{p}$ . Replacing  $\bar{p}$  with its definition then yields the claim for  $p > 0$ . If  $p = 0$ , on the other hand,  $\mathbb{P}_0^B(\sum_{i=1}^B Y_i > 0) = 0$ , and the claim trivially holds.  $\square$

In order to show that the fraction of uncovered inliers is unlikely to exceed  $\mathcal{O}(\epsilon)$ , it is expedient to define the minimum probability mass of a ball of radius  $r_\epsilon$  centered within the closed ball  $B_{R_\epsilon}(0)$  as

$$\underline{p}(\epsilon) := \inf_{z \in B_{R_\epsilon}(0)} \mathbb{P}_0(B_{r_\epsilon}(z)).$$

**Lemma D.6** (Strict positivity of  $\underline{p}(\epsilon)$ ). *If Assumption D.3 holds, then we have  $\underline{p}(\epsilon) > 0$ .*

*Proof.* Introduce an auxiliary function  $\kappa : \mathbb{R}^m \rightarrow [0, 1]$  defined via  $\kappa(z) = \mathbb{P}_0(B_{r_\epsilon}(z))$ . As  $\epsilon > 0$  and  $\mathbb{P}_0$  admits an everywhere positive density with respect to Lebesgue measure, we have  $\kappa(z) > 0$  for every  $z \in \mathbb{R}^m$ . To prove the lemma, we must show that  $\kappa(z)$  is bounded away from 0 on  $B_{R_\epsilon}(0)$ .

By Assumption D.3, the data-generating distribution  $\mathbb{P}_0$  is absolutely continuous with respect to Lebesgue measure on  $\mathbb{R}^m$ . Thus, for any fixed  $z \in \mathbb{R}^m$ , the boundary of  $B_{r_\epsilon}(z)$  has  $\mathbb{P}_0$ -measure zero, and the function  $h_z(z) := \mathbf{1}_{\{\hat{z} \in B_{r_\epsilon}(z)\}}$  is continuous (in fact, constant) at  $z$  for  $\mathbb{P}_0$ -almost every  $\hat{z} \in \mathbb{R}^m$ . Consequently,  $\kappa(z)$  is continuous thanks to the dominated convergence theorem [6, Theorem 3.31]. As the closed ball  $B_{R_\epsilon}(0)$  is compact, there exists a minimizer  $z^* \in B_{R_\epsilon}(0)$  with  $\underline{p}(\epsilon) = \kappa(z^*) > 0$  by Weierstrass' extreme value theorem. Thus, the claim follows.  $\square$

Armed with Lemma D.6, we can prove that the fraction of uncovered inliers is unlikely to exceed  $\mathcal{O}(\epsilon)$ .

**Lemma D.7** (Uncovered inliers are rare). *If Assumption D.3 holds, then we have*

$$\mathbb{P}_0^B \left( \frac{|U_\epsilon|}{B} > \frac{\lambda\epsilon}{L_z^2} \right) \leq \frac{L_z^2}{\lambda\epsilon} \exp(-(B-1)\underline{p}(\epsilon)). \quad (16)$$

*Proof.* Fix any index  $i \in [B]$ , and note that

$$\begin{aligned} \mathbb{P}_0^B(i \in U_\epsilon) &= \mathbb{P}_0^B(i \in O_\epsilon) \mathbb{P}_0^B(i \in U_\epsilon | i \in O_\epsilon) + \mathbb{P}_0^B(i \notin O_\epsilon) \mathbb{P}_0^B(i \in U_\epsilon | i \notin O_\epsilon) \\ &\leq \mathbb{P}_0^B(i \in U_\epsilon | i \notin O_\epsilon) = \mathbb{E}_{\mathbb{P}_0^B} [\mathbb{P}_0^B(i \in U_\epsilon | \hat{Z}_i, i \notin O_\epsilon) | i \notin O_\epsilon], \end{aligned}$$

where the two equalities follow from the law of total probability, and the inequality holds because  $O_\epsilon \cap U_\epsilon = \emptyset$ , which implies that  $\mathbb{P}_0^B(i \in U_\epsilon | i \in O_\epsilon) = 0$ . The event  $i \notin O_\epsilon$  is completely determined by  $\hat{Z}_i$  (that is, it belongs to the  $\sigma$ -algebra generated by  $\hat{Z}_i$ ). As the empirical samples  $\hat{Z}_j$ ,  $j \neq i$ , are mutually independent as well as independent of  $\hat{Z}_i$ , the definition of  $U_\epsilon$  in (15b) then implies that

$$\begin{aligned} \mathbb{P}_0^B(i \in U_\epsilon | \hat{Z}_i, i \notin O_\epsilon) &= \prod_{j \in [B]: j \neq i} \mathbb{P}_0^B(\|\hat{Z}_j - T^*(\hat{Z}_i)\|_2 > r_\epsilon | \hat{Z}_i, i \notin O_\epsilon) \\ &\leq (1 - \underline{p}(\epsilon))^{B-1} \leq \exp(-(B-1)\underline{p}(\epsilon)). \end{aligned}$$

Here, the first inequality follows from the definition of  $\underline{p}(\epsilon)$ , which ensures that

$$\mathbb{P}_0 \left( \|\hat{Z}_j - T^*(\hat{Z}_i)\|_2 \leq r_\epsilon \mid \hat{Z}_i, i \notin O_\epsilon \right) \geq \underline{p}(\epsilon).$$

Combining the above estimates finally yields

$$\mathbb{P}_0^B(i \in U_\epsilon) \leq \exp(-(B-1)\underline{p}(\epsilon)) \quad \forall i \in [B].$$

As  $|U_\epsilon| = \sum_{i=1}^B \mathbf{1}_{\{i \in U_\epsilon\}}$ , the expected number of uncovered inliers satisfies

$$\mathbb{E}_{\mathbb{P}_0^B}[|U_\epsilon|] = \sum_{i=1}^B \mathbb{P}_0^B(i \in U_\epsilon) \leq B \exp(-(B-1)\underline{p}(\epsilon)).$$

Note that  $|U_\epsilon|$  is a non-negative random variable. By Markov's inequality, we thus obtain

$$\mathbb{P}_0^B \left( \frac{|U_\epsilon|}{B} > \frac{\lambda\epsilon}{L_z^2} \right) \leq \frac{\mathbb{E}_{\mathbb{P}_0^B}[|U_\epsilon|/B]}{\lambda\epsilon/L_z^2} \leq \frac{L_z^2}{\lambda\epsilon} \exp(-(B-1)\underline{p}(\epsilon)).$$

This observation completes the proof.  $\square$

Next, we prove that every point in the  $r_\epsilon$ -neighborhood of a global maximizer of  $f_i$  constitutes an  $\epsilon/2$ -optimal point. For ease of notation, we define  $f_i^* := \max_{z \in \mathbb{R}^m} f_i(z)$  for every  $i \in [N]$ .

**Lemma D.8** (Superlevel sets). *Suppose that [Assumption D.1](#) holds, and define  $r_\epsilon$  as in (14). If  $z^*, z \in \mathbb{R}^m$  with  $f_i(z^*) = f_i^*$  and  $\|z - z^*\|_2 \leq r_\epsilon$ , then  $f_i(z) \geq f_i^* - \epsilon/2$  for all  $i \in [B]$ .*

*Proof.* Note that both  $f_i$  as well as  $-f_i$  are  $(L_{zz} + 2\lambda)$ -smooth by virtue of [Assumption D.1](#). Hence, the standard quadratic upper bound for the smooth function  $-f_i$  [9, Proposition 6.1.2] yields

$$\begin{aligned} f_i(z) &\geq f_i(z^*) + \langle \nabla f_i(z^*), z - z^* \rangle - \frac{L_{zz} + 2\lambda}{2} \|z - z^*\|_2^2 \\ &\geq f_i^* - \frac{L_{zz} + 2\lambda}{2} r_\epsilon^2 = f_i^* - \frac{\epsilon}{2}, \end{aligned}$$

where the second inequality holds because  $z^*$  is a global maximizer of  $f_i$  satisfying  $\nabla f_i(z^*) = 0$  and because  $\|z - z^*\|_2 \leq r_\epsilon$ , and the equality follows from the definition of  $r_\epsilon$ .  $\square$

If  $\hat{z} = \hat{Z}_i$  for  $i \in C_\epsilon$  is a covered inlier and  $z^* = T^*(\hat{z})$ , then there exists  $j \neq i$  such that  $z = \hat{Z}_j$  satisfies  $\|z - z^*\|_2 \leq r_\epsilon$ ; see (15c). [Lemma D.8](#) thus guarantees that  $z$  constitutes an  $\epsilon/2$ -optimal solution for problem (3). This reasoning implies that, for every covered inlier, the adversary can find a near-optimal adversarial sample among the finite set of empirical samples. This insight and the earlier results of this section culminate in the following main theorem, which proves that the MPA map  $T_{\text{mpa}}$  is near-optimal in problem (2) with high probability. To formalize this result, we recall that the adversarial error  $\varepsilon(\theta, T)$  of a map  $T$  is defined as the suboptimality of  $T$  in (2); see (4).

**Theorem D.2** (Global optimality guarantee of MPA). *If [Assumptions D.1](#) and [D.3](#) hold and  $0 < \eta < 2/(L_{zz} + 2\lambda)$ , then for any  $K \in \mathbb{N} \cup \{\infty\}$ ,  $R \in \mathbb{N}$ , and tolerance  $\epsilon > 0$ , we have*

$$\mathbb{P}_0^B(\varepsilon(\theta, T_{\text{mpa}}) \leq \epsilon) \geq 1 - \beta(B, \epsilon),$$

where  $\varepsilon(\theta, T)$  is the adversarial error of a map  $T$  with respect to  $\hat{\mathbb{P}} = \frac{1}{B} \sum_{i=1}^B \delta_{\hat{Z}_i}$ , and where

$$\beta(B, \epsilon) := \exp(-B\lambda\epsilon/6L_z^2) + \frac{L_z^2}{\lambda\epsilon} \exp(-(B-1)\underline{p}(\epsilon)).$$

*Proof.* We first rewrite the adversarial error of  $T_{\text{mpa}}$  by expanding the empirical expectation. Using the definition (4) of the adversarial error and the per-sample formulation (1) of  $F(\theta)$ , we can write

$$\begin{aligned} \varepsilon(\theta, T_{\text{mpa}}) &= \frac{1}{B} \sum_{i=1}^B \left( \max_{z \in \mathbb{R}^m} \{f(\theta, z) - \lambda\|z - \hat{Z}_i\|_2^2\} - \left( f(\theta, T_{\text{mpa}}(\hat{Z}_i)) - \lambda\|T_{\text{mpa}}(\hat{Z}_i) - \hat{Z}_i\|_2^2 \right) \right) \\ &= \frac{1}{B} \sum_{i=1}^B \left( f_i^* - f_i(T_{\text{mpa}}(\hat{Z}_i)) \right). \end{aligned}$$

Next, we introduce  $G_i := f_i^* - f_i(T_{\text{mpa}}(\hat{Z}_i))$  as shorthand for the per-sample suboptimality gap of  $T_{\text{mpa}}$  for every  $i \in [B]$ . Using the batch partition (15), the adversarial error can thus be recast as

$$\varepsilon(\theta, T_{\text{mpa}}) = \frac{1}{B} \sum_{i \in O_\epsilon} G_i + \frac{1}{B} \sum_{i \in U_\epsilon} G_i + \frac{1}{B} \sum_{i \in C_\epsilon} G_i. \quad (17)$$

By [Lemma D.5](#), the probability of the event  $|O_\epsilon|/B > \lambda\epsilon/L_z^2$  is bounded by  $\exp(-B\lambda\epsilon/6L_z^2)$ . Similarly, [Lemma D.7](#) ensures that the probability of the event  $|U_\epsilon|/B > \lambda\epsilon/L_z^2$  is bounded by  $L_z^2/(\lambda\epsilon) \exp(-(B-1)\underline{p}(\epsilon))$ . Using the union bound, one thus verifies that the desirable event in which both  $|O_\epsilon|/B \leq \lambda\epsilon/L_z^2$  and  $|U_\epsilon|/B \leq \lambda\epsilon/L_z^2$  occurs with probability at least  $1 - \beta(B, \epsilon)$ .

It remains to be shown that the adversarial error is bounded by  $\epsilon$  in this desirable event. To this end, recall first that the ascent and reassignment operators are strictly monotone thanks to [Lemma D.3](#). Hence, the objective value of  $T_{\text{mpa}}(\hat{Z}_i)$  dominates that of  $\hat{Z}_i$ , that is,  $f_i(T_{\text{mpa}}(\hat{Z}_i)) \geq f_i(\hat{Z}_i)$  for all  $i \in [B]$ . As  $f(\theta, z)$  is  $L_z$ -Lipschitz continuous in  $z$ , we may thus conclude that

$$\begin{aligned} G_i &\leq f_i^* - f_i(\hat{Z}_i) = \max_{z \in \mathbb{R}^m} \{f(\theta, z) - f(\theta, \hat{Z}_i) - \lambda\|z - \hat{Z}_i\|_2^2\} \\ &\leq \max_{z \in \mathbb{R}^m} \{L_z\|z - \hat{Z}_i\|_2 - \lambda\|z - \hat{Z}_i\|_2^2\} = \frac{L_z^2}{4\lambda}. \end{aligned}$$

Using this coarse estimate, we obtain

$$\frac{1}{B} \sum_{i \in O_\epsilon} G_i \leq \frac{|O_\epsilon|}{B} \frac{L_z^2}{4\lambda} \leq \frac{\epsilon}{4} \quad \text{and} \quad \frac{1}{B} \sum_{i \in U_\epsilon} G_i \leq \frac{|U_\epsilon|}{B} \frac{L_z^2}{4\lambda} \leq \frac{\epsilon}{4}. \quad (18a)$$

If  $i \in C_\epsilon$ , then we have  $\|T^*(\hat{Z}_i)\|_2 \leq R_\epsilon$ , and there is  $j \neq i$  with  $\|\hat{Z}_j - T^*(\hat{Z}_i)\|_2 \leq r_\epsilon$ ; see (15c). By Lemma D.8, we thus find  $f_i(\hat{Z}_j) \geq f_i^* - \epsilon/2$  and  $G_i \leq \epsilon/2$  for all  $i \in C_\epsilon$ , which implies that

$$\frac{1}{B} \sum_{i \in C_\epsilon} G_i \leq \frac{|C_\epsilon|}{B} \frac{\epsilon}{2} \leq \frac{\epsilon}{2}. \quad (18b)$$

Substituting (18a) and (18b) into (17) yields  $\varepsilon(\theta, T_{\text{mpa}}) \leq \epsilon$ , and thus the claim follows.  $\square$

**Corollary D.2** (Sample complexity of MPA). *If all assumptions of Theorem D.2 hold,  $\beta \in (0, 1)$ , and the batch size satisfies*

$$B \geq \max \left\{ \frac{6L_z^2}{\lambda\epsilon} \log \left( \frac{2}{\beta} \right), 1 + \frac{1}{\underline{p}(\epsilon)} \log \left( \frac{2L_z^2}{\lambda\epsilon\beta} \right) \right\},$$

*then we have  $\varepsilon(\theta, T_{\text{mpa}}) \leq \epsilon$  with probability at least  $1 - \beta$ .*

*Proof.* To guarantee that the failure probability  $\beta(B, \epsilon)$  from Theorem D.2 falls below  $\beta$ , it suffices to bound each of its two additive terms by  $\beta/2$ . As for the first term, we thus require

$$\exp \left( -\frac{B\lambda\epsilon}{6L_z^2} \right) \leq \frac{\beta}{2} \quad \implies \quad B \geq \frac{6L_z^2}{\lambda\epsilon} \log \left( \frac{2}{\beta} \right).$$

Similarly, we require the second term to satisfy

$$\frac{L_z^2}{\lambda\epsilon} \exp(-(B-1)\underline{p}(\epsilon)) \leq \frac{\beta}{2} \quad \implies \quad B \geq 1 + \frac{1}{\underline{p}(\epsilon)} \log \left( \frac{2L_z^2}{\lambda\epsilon\beta} \right).$$

Taking  $B$  to be the maximum of these two lower bounds ensures both conditions hold simultaneously. This observation completes the proof.  $\square$

As is typical of global optimality guarantees for nonconvex optimization, the guarantees established in Theorem D.2 and Corollary D.2 come at the price of a curse of dimensionality.

**Remark D.2** (Curse of dimensionality). The sample complexity bound in Corollary D.2 is governed primarily by the inverse covering mass  $1/\underline{p}(\epsilon)$ , which conceals an exponential dependence on the ambient dimension  $m$ . Indeed, one can construct at least  $(R_\epsilon/2r_\epsilon)^m$  disjoint balls of radius  $r_\epsilon$  with centers in  $B_{R_\epsilon}(0)$ . As their masses sum to at most one, at least one of them has mass at most  $(2r_\epsilon/R_\epsilon)^m$ , and since  $R_\epsilon$  grows when  $\epsilon$  decreases, we conclude that  $p(\epsilon) = \mathcal{O}(r_\epsilon^m) = \mathcal{O}(\epsilon^{m/2})$  irrespective of the data-generating distribution  $\mathbb{P}_0 \in \mathcal{P}(\mathbb{R}^m)$ . The factor  $\epsilon^{m/2}$  thus reflects the volume of the covering ball rather than any tail behavior of  $\mathbb{P}_0$ . One can show that if  $\mathbb{P}_0$  is a standard normal distribution, then  $p(\epsilon) = \Omega(\epsilon^{m/2})$ . Even in this best possible case, Corollary D.2 requires a batch of size  $B = \tilde{\mathcal{O}}(\max\{\epsilon^{-1}, \epsilon^{-m/2}\})$ , which grows exponentially with the ambient dimension  $m$ .

**Remark D.3** (MPA versus PA). The global optimality guarantee of Corollary D.2 requires a large batch of sufficiently diverse empirical samples. In the absence of such a batch, MPA may not output a globally optimal adversarial map. In fact, as each round of MPA starts with a reassignment step, the adversarial error of the resulting MPA map can even exceed that of the corresponding PA map. To see this, suppose that  $f_i(\theta, \hat{z}_i) < f_i(\theta, \hat{z}_j)$ , while the stationary point in the basin containing  $\hat{z}_j$  has a lower objective value than the stationary point in the basin containing  $\hat{z}_i$ . To construct the  $i$ -th adversarial sample, PA applies gradient ascent initialized at  $\hat{z}_i$ , whereas MPA moves from  $\hat{z}_i$  to  $\hat{z}_j$  due to the first reassignment step and then applies gradient ascent initialized at  $\hat{z}_j$ . In this case, PA ends up in the better stationary point and thus finds a better adversarial example than MPA.

To ensure that MPA dominates PA, one could reverse the order of the reassignment and gradient ascent steps in the inner loop of Algorithm 1. The first gradient ascent of this alternative variant of MPA replicates the PA map exactly. Moreover, since both the reassignment operator and the  $t$ -step ascent operator are monotone by Lemma D.3 (iii), every subsequent MPA update can only

improve upon PA, regardless of the batch size. The price of this guarantee, however, is the loss of a global optimality result analogous to [Theorem D.2](#). In particular, one can construct problem instances for which the alternative variant of MPA produces a strictly suboptimal adversarial map for every possible batch of empirical samples. We tested both variants of MPA in our experiments and found that the original version of Algorithm 1 consistently attains the lower adversarial error. Hence, we do not pursue the ascent-first variant of MPA further.

### D.3 An Illustrative Example where PA Stalls and MPA Progresses

[Example 4.1](#) presents an instance of problem (1) in which  $\hat{\mathbb{P}}$  is a discrete distribution supported on only two atoms and the discrete PA map fails to satisfy cyclical monotonicity after a fixed (albeit large) number of iterations. We now extend this example by embedding it into an outer minimization problem over  $\theta$  and demonstrate that the loss of cyclical monotonicity leads to a critical failure of the outer optimization dynamics. Specifically, the algorithm becomes trapped at the initial outer iterate, while a single MPA reassignment produces a nonzero and informative outer gradient, thereby enabling further progress of the optimization algorithm.

Note that the loss function in [Example 4.1](#) is constant in  $\theta$ . In the following, we construct a closely related loss function that displays a nontrivial dependence on  $\theta$ . Specifically, we set

$$f(\theta, z) := \theta(f_0(z) - b), \quad \text{where} \quad f_0(z) := \sum_{k=1}^2 a_k \exp(-(z - \mu_k)^\top \Sigma_k^{-1}(z - \mu_k)).$$

The constants  $a_1, a_2, \mu_1, \mu_2, \Sigma_1$  and  $\Sigma_2$  are chosen as in [Example 4.1](#), and  $\theta \in \Theta = [0, 1]$  is a scalar model parameter. The value of  $b \in \mathbb{R}$  will be specified later. Note that  $f(\theta, z)$  is affine in  $\theta$  with uniformly bounded gradients, and thus [Proposition 3.1](#) applies. If we define  $\hat{\mathbb{P}}$  and  $\lambda$  as in [Example 4.1](#), then [Proposition B.2](#) implies that the adversarial risk can be represented as

$$F(\theta) = \max_{T \in \mathcal{T}} \frac{1}{2} \sum_{i=1}^2 \left( f(\theta, T(\hat{z}_i)) - \lambda \|T(\hat{z}_i) - \hat{z}_i\|_2^2 \right). \quad (19)$$

Next, we approximately minimize  $F(\theta)$  over  $\Theta$  by means of projected gradient descent with step size  $\alpha \in (0, 1)$ , using inexact gradient estimates. The iterates of this algorithm satisfy

$$\theta_{e+1} = \text{Proj}_{[0,1]}(\theta_e - \alpha g_e), \quad \text{where} \quad g_e = \frac{1}{2} \sum_{i=1}^2 (f_0(T_e(\hat{z}_i)) - b) \quad (20)$$

is an inexact gradient of  $F(\theta)$  at  $\theta = \theta_e$ , and  $T_e$  is a near-optimal solution of (19) for each epoch  $e \in [E]$ . Note that if  $T_e$  were a globally optimal adversarial map, then  $g_e$  would coincide with the exact subgradient of  $F(\theta)$  at  $\theta = \theta_e$  by Danskin's theorem. We initialize  $\theta_1 = 1$ .

**PA stalls.** Assume first that the adversary's problem in (19) is addressed with discrete PA such that  $T_1(\cdot) = T_{\text{pa}}(t, \cdot)$  for  $t = 8,000$  steps. As  $f(\theta_1, z)$  coincides with the loss function of [Example 4.1](#) up to the additive constant  $-b$ , discrete PA initialized at the empirical samples of  $\hat{\mathbb{P}}$  yields

$$T_1(\hat{z}_1) = (2.17, 1.98)^\top \quad \text{and} \quad T_1(\hat{z}_2) = (-3.96, -2.00)^\top.$$

Hence,  $T_1$  fails to be cyclically monotone as shown in [Example 4.1](#), which implies via [Lemma 4.1](#) that  $T_1$  also fails to be assignment-stationary. Next, set

$$b := \frac{1}{2} (f_0(T_1(\hat{z}_1)) + f_0(T_1(\hat{z}_2)))$$

such that the inexact gradient  $g_1$  vanishes. One can then prove by induction that  $\theta_e = 1$  for every  $e \in [E]$ . This shows that PA-based adversarial training gets trapped at the initial outer iterate  $\theta_1 = 1$  even though  $\theta_1$  is neither a minimizer nor a stationary point of  $F(\theta)$ .

**MPA progresses.** Assume next that the adversary's problem in (19) is addressed with a full-batch ( $B = 2$ ) version of MPA over  $R = 1$  rounds; see Algorithm 1.

To obtain a near-optimal adversarial map for (19) with  $\theta = \theta_1$ , MPA applies one reassignment step followed by one discrete PA step, and ends with a final reassignment step. The near-optimal

adversarial samples  $\{z_i\}_{i=1}^2$  corresponding to the empirical samples  $\{\hat{z}_i\}_{i=1}^n$  are thus constructed iteratively as follows. In the first round, MPA sets  $\mathbf{z} \leftarrow \mathcal{R}(\hat{\mathbf{z}})$ , where  $\mathcal{R}$  is the reassignment operator from [Definition D.2](#). One readily verifies that

$$f(\theta_1, \hat{z}_2) - \lambda \|\hat{z}_2 - \hat{z}_1\|_2^2 > f(\theta_1, z_1) \quad \text{and} \quad f(\theta_1, \hat{z}_2) > f(\theta_1, \hat{z}_1) - \lambda \|\hat{z}_1 - \hat{z}_2\|_2^2,$$

suggesting that the adversary is better off mapping *both* empirical samples to  $\hat{z}_2$  instead of keeping the identity initialization. Hence, in the first reassignment step, the adversary sets  $z_1 \leftarrow \hat{z}_2$  and  $z_2 \leftarrow \hat{z}_2$ . The first PA step is then warm-started at the same point  $z_1 = z_2 = \hat{z}_2$  for *both* empirical samples. In this step MPA sets  $z_i \leftarrow \mathcal{A}_i^t(z_i)$  where  $\mathcal{A}_i^t$  is the  $i$ -th  $t$ -step ascent operator from [Definition D.1](#). As in [Example 4.1](#), we set  $t = 8,000$ , which ensures that  $z_i$  closely approximates a stationary point. After this PA step, both samples have almost reached the corresponding global maximizers, and one can readily verify that the adversary would not benefit from reassigning the adversarial samples in the final step, that is, the trivial reassignment is optimal. At this stage, MPA has essentially converged. Subsequent PA iterations would merely refine the values of  $\{z_i\}_{i=1}^2$  by adding insignificant digits, while subsequent reassignment steps would have no effect. See [Figure 1](#) for the trajectory that each sample takes in the MPA algorithm. The adversarial map  $T_{\text{mpa}} \in \mathcal{T}$  corresponding to two MPA rounds can thus be defined by setting  $T_{\text{mpa}}(\hat{z}_i) = z_i$  for  $i = 1, 2$  and arbitrarily setting  $T_{\text{mpa}}(z) = 0$  for all  $z \notin \{\hat{z}_i\}_{i=1}^2$ , which ensures Borel measurability. By construction,  $T_{\text{mpa}}$  is assignment-stationary and cyclically monotone on  $\{\hat{z}_i\}_{i=1}^2$ . A direct calculation reveals that

$$\sum_{i=1}^2 f_0(T_{\text{mpa}}(\hat{z}_i)) > \sum_{i=1}^2 f_0(T_{\text{pa}}(\hat{z}_i)) \implies g_1 = \frac{1}{2}(f_0(T_{\text{mpa}}(\hat{z}_1)) + f_0(T_{\text{mpa}}(\hat{z}_2))) - b > 0.$$

This shows that MPA-based adversarial training using the iterative scheme (20) escapes from the spurious fixed point  $\theta_1 = 1$  that traps PA. That is, MPA generates a strictly better iterate  $\theta_2 < 1$ .

This example showcases that the reassignment step in MPA is essential. Its purpose is to warm-start the next PA phase at the best adversarial sample output by the previous phase. In particular, if the output of a PA phase happens to lie within the basin of attraction of the global maximizer associated with an empirical sample  $\hat{z}_i$ , then, after the subsequent PA phase, the corresponding adversarial sample  $z_i$  will be driven arbitrarily close to a global maximizer, provided that a sufficiently large number of PA iterations is performed. Through this mechanism, MPA exploits information about the global optimization landscape that remains inaccessible to PA alone and incorporates it into the outer gradient. As a consequence, the outer minimization over  $\theta$  can continue to make progress under MPA even in situations where PA yields a vanishing gradient.

## E ICNN Training Details

To guarantee convexity of the parametric potential  $\psi_\omega(z)$  in  $z$ , the hidden-to-hidden and output weights of the ICNN must remain non-negative throughout training. We enforce this constraint by reparametrizing these weights as exponentials,  $\exp(W_\ell^y)$  and  $\exp(w_L^y)$ , so that non-negativity holds by construction regardless of the values of the underlying unconstrained parameters  $W_\ell^y$  and  $w_L^y$ . A natural starting point would be to initialize  $W_\ell^y$  and  $w_L^y$  according to a standard zero-mean scheme such as Xavier initialization [30]. However, even when the pre-exponential weights are zero-mean, the exponentiated weights  $\exp(W_\ell^y)$  and  $\exp(w_L^y)$  possess a strictly positive mean. This positive-mean bias accumulates across layers, distorting both the forward signal and the backpropagated gradient, and has been identified as a source of the optimization instabilities and representational bottlenecks reported in prior work [39, 48, 61]. To mitigate this effect, we adopt the principled initialization scheme of Hoedt and Klambauer [36], which extends the signal-propagation analysis underlying Xavier initialization to the setting of weights with non-zero means, yielding initialization scales that are explicitly corrected for the positive-mean offset introduced by the exponential reparametrization.

Following [36], for a layer with  $n_{\text{in}}$  inputs, we set the target mean and variance of the exponentials to

$$\mu_w = \sqrt{\frac{6\pi}{n_{\text{in}}[6(\pi - 1) + (n_{\text{in}} - 1)(3\sqrt{3} + 2\pi - 6)]}} \quad \text{and} \quad \sigma_w^2 = \frac{1}{n_{\text{in}}}, \quad (21)$$

respectively. To achieve these target values, we sample the original, pre-exponential weights independently from a normal distribution  $W_{\ell,ij}^y \sim \mathcal{N}(\mu, \sigma^2)$ . This makes the resulting positive

---

**Algorithm 3** ADAPTIVE INNER STEP SIZE (BB + ARMIJO)

---

**Input:** Initial iterate  $z^{-1} = z^0$ , step size bounds  $[\eta_{\min}, \eta_{\max}]$ , initial step size  $\eta_0 \in [\eta_{\min}, \eta_{\max}]$ , Armijo sufficient ascent  $c \in (0, 1)$ , shrinkage  $\tau \in (0, 1)$ , # max backtracking steps  $J$ , # steps  $K$

```
for  $k = 0, \dots, K - 1$  do
   $s \leftarrow z^k - z^{k-1}$  and  $y \leftarrow \nabla_z f_{\hat{z}}(z^{k-1}) - \nabla_z f_{\hat{z}}(z^k)$ 
  if  $\langle s, y \rangle > 0$  then
     $\eta_k \leftarrow \text{Proj}_{[\eta_{\min}, \eta_{\max}]} \eta_k^{\text{BB}}$ , with  $\eta_k^{\text{BB}}$  as in (25)
  else
     $\eta_k \leftarrow \eta_0$ 
  end if
   $j \leftarrow 0$ 
  while  $j < J$  and  $f_{\hat{z}}(z^k + \eta_k \nabla_z f_{\hat{z}}(z^k)) - f_{\hat{z}}(z^k) < c \eta_k \|\nabla_z f_{\hat{z}}(z^k)\|_2^2$  do
     $\eta_k \leftarrow \tau \eta_k$ ,  $j \leftarrow j + 1$ 
  end while
   $z^{k+1} \leftarrow z^k + \eta_k \nabla_z f_{\hat{z}}(z^k)$ 
end for
return  $z^K$ 
```

---

weights independent and follow a lognormal distribution  $\exp(W_{\ell,ij}^y) \sim \text{LogNormal}(\mu, \sigma^2)$ . To find the exact  $\mu$  and  $\sigma^2$ , we match the lognormal moments  $\mathbb{E}[\exp W_{\ell,ij}^y] = \exp(\mu + \frac{1}{2}\sigma^2)$  and  $\text{Var}(\exp W_{\ell,ij}^y) = (\exp \sigma^2 - 1) \exp(2\mu + \sigma^2)$  to the target moments  $(\mu_w, \sigma_w^2)$  in (21). This yields

$$\mu = \ln(\mu_w^2) - \frac{1}{2} \ln(\sigma_w^2 + \mu_w^2) \quad \text{and} \quad \sigma^2 = \ln(\sigma_w^2 + \mu_w^2) - \ln(\mu_w^2). \quad (22)$$

**Cancelling the positive-mean shift.** The target mean  $\mu_w > 0$  still contributes a systematic bias  $\mu_w \mathbf{1}^\top y_{\ell-1}$  to each pre-activation. Two designs cancel this contribution at initialization. The original recipe in [36] attaches a bias term to each exp-reparameterized layer and initializes it at  $-\mu_b \mathbf{1}$  with

$$\mu_b = \sqrt{\frac{3 n_{\text{in}}}{6(\pi - 1) + (n_{\text{in}} - 1)(3\sqrt{3} + 2\pi - 6)}}. \quad (23)$$

The construction used throughout our experiments takes the second design: the exp-reparameterized layers carry no bias, and the positive-mean contribution is instead absorbed by the unconstrained passthrough  $W_\ell^z z + b_\ell$  at each hidden layer and by  $(w_L^z)^\top z + b_L$  at the readout. The two designs are equivalent at initialization, but the second has a structural advantage during training. The passthrough is  $z$ -dependent, so the offset it applies adapts to the input, whereas a fixed scalar bias on  $\exp(W_\ell^y)$  cannot. This cleanly separates roles, so that convexity-preserving parameters are confined to the exponential reparameterization (22), while all expressive, sign-free degrees of freedom are concentrated in the well-initialized passthrough. Simpler ICNN variants without a passthrough still require the explicit offset (23); the codebase supports both conventions for ablation.

### E.1 Adaptive Inner Step Size Selection

The training dynamics of the adversarial map depend heavily on the inner step size  $\eta$ . For instance, preserving the monotonicity of the PA map in the one-dimensional case requires  $\eta \leq (L_{zz} + 2\lambda)^{-1}$  (Proposition 4.1). Since a neural network  $f(\theta, z)$  is highly non-linear with respect to its inputs, the local curvature  $\nabla_{zz}^2 f$  varies significantly across the domain. Consequently, a single fixed step size makes the training process brittle. A conservative small step size safely avoids instability but wastes the inner-loop computational budget. Conversely, a large step size can cause the optimization to diverge. To dynamically adapt to the varying local geometry, we propose to use the Barzilai-Borwein (BB) method [65] paired with an Armijo backtracking [4] safeguard (Algorithm 3). The BB method acts as a computationally cheap quasi-Newton step to estimate the effective inverse curvature along the search direction, while the Armijo condition serves as a safety net to guarantee sufficient objective improvement. We detail both components next. To this end, let  $f_{\hat{z}}(z) = f(\theta, z) - \lambda \|z - \hat{z}\|_2^2$  denote the local objective to be maximized for a fixed sample  $\hat{z}$ .

**Barzilai-Borwein method.** Quasi-Newton methods, which motivate the Barzilai-Borwein approach, accelerate gradient ascent on  $\max_z f_{\hat{z}}(z)$  by replacing the step size with a local curvature estimate.

The update takes the form

$$z^{k+1} = z^k + (B^k)^{-1} \nabla_z f_{\hat{z}}(z^k), \quad (24)$$

where  $B^k$  serves as an approximation of the exact Hessian matrix of  $-f_{\hat{z}}$  evaluated at the current iterate  $z^k$ . After taking a step from  $z^{k-1}$  to  $z^k$ , the algorithm observes the secant pair  $s = z^k - z^{k-1}$  and  $y = \nabla_z f_{\hat{z}}(z^{k-1}) - \nabla_z f_{\hat{z}}(z^k)$ , which encodes local curvature along the most recent step. A natural requirement is that  $B^k$  satisfies the so-called secant equation  $B^k s = y$ , ensuring the approximation is consistent with the observed curvature. In high dimensions, however, maintaining a dense matrix  $B^k$  that satisfies this condition and solving the resulting linear system becomes computationally prohibitive. The BB method alleviates this by approximating the Hessian with a simple scalar multiple of the identity matrix,  $B^k \approx (\eta_k)^{-1} \text{id}$ , where the scalar  $\eta_k$  is chosen to minimize the squared norm of the secant equation residual. Solving this least-squares problem,  $\min_{\eta} \|(\eta)^{-1} s - y\|_2^2$ , yields the BB step size

$$\eta_k^{\text{BB}} = \frac{\|s\|_2^2}{\langle s, y \rangle}. \quad (25)$$

Substituting  $(B^k)^{-1}$  in (24) with  $\eta_k^{\text{BB}}$  bypasses the matrix operations and yields an efficient scalar update that scales with the effective inverse curvature. Since neural network landscapes are highly non-concave, the secant product  $\langle s, y \rangle$  may occasionally be non-positive. In such cases where the curvature estimate is invalid, we fall back to a default initial step size  $\eta_0$  (Algorithm 3).

**Armijo backtracking.** While the BB rule provides an inexpensive and often accurate estimate of the local inverse curvature, it does not by itself guarantee that the resulting step yields an improvement in the objective. This issue is particularly pronounced in our setting, where  $f_{\hat{z}}$  is generally non-concave, and the BB estimate may occasionally be overly aggressive. To ensure stability of the inner maximization, we complement the BB proposal with an Armijo backtracking line search, which enforces a sufficient ascent condition at each iteration.

Given the current iterate  $z^k$  and ascent direction  $\nabla_z f_{\hat{z}}(z^k)$ , we initialize a trial step size with the BB proposal, *i.e.*,  $\eta_k \leftarrow \eta_k^{\text{BB}}$ . The Armijo condition requires that the achieved improvement in the objective is at least a fixed fraction of the linearized gain predicted by the gradient, namely

$$f_{\hat{z}}(z^k + \eta_k \nabla_z f_{\hat{z}}(z^k)) - f_{\hat{z}}(z^k) \geq c \eta_k \|\nabla_z f_{\hat{z}}(z^k)\|_2^2,$$

where  $c \in (0, 1)$  controls the required level of sufficient ascent. This condition compares the actual improvement against the increase predicted by a first-order approximation of  $f_{\hat{z}}$  at  $z^k$ . When the trial step size is too large, curvature effects cause this approximation to overestimate the gain, and the condition fails. In that case, we reduce the step size to  $\eta_k \leftarrow \tau \eta_k$  with  $\tau \in (0, 1)$ , and repeat the test until the condition is satisfied or the maximum number of backtracking steps  $J$  is reached. This procedure prevents overly aggressive updates while preserving the efficiency of the BB initialization.

In practice, we operate on mini-batches  $\{\hat{z}_i\}_{i=1}^B$  rather than a single sample. Accordingly, the single-sample objective  $f_{\hat{z}}(z) = f(\theta, z) - \lambda \|z - \hat{z}\|_2^2$  is replaced by its batch counterpart

$$f_{\hat{\mathbf{z}}}(z) = \frac{1}{B} \sum_{i=1}^B (f(\theta, z_i) - \lambda \|z_i - \hat{z}_i\|_2^2) \quad \text{with} \quad \mathbf{z} = (z_1, \dots, z_B).$$

All steps are applied to this averaged objective, and the same analysis carries over directly. For notational simplicity, we present the algorithm and derivations in the single-sample form above.

## F Supplementary Details for Section 6

Appendices F.1, F.2, and F.3 provide full hyperparameter settings and implementation details for the experiments of Sections 6.1, 6.2, and 6.3, respectively. The latter two appendices additionally report gradient-obfuscation checks and a sensitivity analysis in the penalty parameter  $\lambda$ . For all methods requiring an inner maximization step—namely PA, RO, WFR, NN-DRO, MPA, and ICNN-DRO—we conduct a comprehensive search over the inner step size  $\eta$ , evaluating both constant step sizes and adaptive Barzilai–Borwein step sizes coupled with an Armijo line search (Algorithm 3), and report the best-performing configuration for each method. All experiments were run on an NVIDIA A100-SXM4-80GB GPU with dual AMD EPYC 7543 32-core CPUs.

We briefly outline the configurations of our baselines, with exact hyperparameter values deferred to the subsequent experimental sections. To ensure a fair comparison among parametrized-map methods, the network architectures of NN-DRO and ICNN-DRO are matched in depth and width. For the entropy-regularized baselines, we introduce a regularization parameter  $\varepsilon_{\text{ent}} > 0$ . SDRO [73] exploits the fact that the entropy-regularized worst-case distribution admits a closed-form Gibbs representation [73, Remark 4] to reformulate the inner maximization as a nested log-expectation [73, Theorem 1], whose gradient is then estimated via randomized multilevel Monte Carlo with importance samples drawn from a truncated geometric distribution [11], parametrized by a success probability  $p \in (0, 1)$  and truncation level  $M \in \mathbb{N}_+$ . WFR [75], by contrast, targets the adversarial distribution directly by evolving a cloud of  $n \in \mathbb{N}_+$  auxiliary particles per data point under an interacting particle dynamics. Unlike PA, which perturbs the empirical samples themselves, WFR keeps the training data fixed and uses these auxiliary particles to approximate the local adversarial distribution. The parameters  $\varepsilon_{\text{ent}}$ ,  $p$ ,  $M$ , and  $n$  are specified in each experimental section.

### F.1 Supplementary Details for Section 6.1

This section provides additional details for the adversarial multi-class logistic regression experiment of Section 6.1. All methods are trained for 10 epochs with batch size  $B = 128$  and penalty parameter  $\lambda = 10$ . The outer minimization uses step size  $\alpha = 5 \times 10^{-3}$  for all robust benchmarks and  $\alpha = 10^{-3}$  for the ERM baseline. The inner maximization runs for  $K = 20$  ascent iterations for ICNN-DRO and for  $K = 100$  iterations for every other method, with step sizes selected by the BB+Armijo procedure with  $(\eta_0, \eta_{\min}, \eta_{\max}) = (5 \times 10^{-4}, 10^{-6}, 1)$  and  $(c, \tau, J) = (0.1, 0.5, 10)$ . For method-specific parameters, MPA performs  $R = 5$  rounds of  $K = 20$  iterations, hence 100 ascent iterations in total and, counting the terminal reassignment, six reassignments. RO has an  $\ell_2$ -perturbation budget of  $\varepsilon = 0.04$ . The entropy-regularized baselines (SDRO and WFR) use a regularization strength of  $\varepsilon_{\text{ent}} = 0.2$ . Moreover, WFR utilizes  $n = 16$  auxiliary particles per data point. Additionally, SDRO employs a maximum truncation level of  $M = 5$  with success probability  $p = 0.5$  for its multilevel Monte Carlo estimator. The  $\ell_2$ -constrained PGD attack budget ( $\Delta \in [0, 0.08]$ ) applied to the covariates and the feature embedding process remains exactly as detailed in the main text.

### F.2 Supplementary Details for Section 6.2

Across all benchmarks, the classifier  $h_\theta$  is a ResNet-18 [33], pre-trained on clean CIFAR-10 for 200 epochs and then adversarially trained under each method for a further 50 epochs with batch size  $B = 256$  and penalty parameter  $\lambda = 30$ . The outer minimization uses SGD with Nesterov momentum parameter 0.9 [52] under a one-cycle schedule with cosine annealing [62] and peak learning rate 0.1.

For all gradient-based inner solvers, the inner maximization runs for  $K = 20$  iterations with the step size set by the Barzilai-Borwein rule with Armijo backtracking (Algorithm 3): initial step  $\eta_0 = 2 \times 10^{-4}$ , clipping interval  $[\eta_{\min}, \eta_{\max}] = [10^{-7}, 2.5 \times 10^{-1}]$ , sufficient-ascent constant  $c = 10^{-4}$ , shrinkage factor  $\tau = 0.5$ , and at most  $J = 10$  backtracking steps. The RO inner attack is  $\ell_2$ -constrained PGD with budget  $\varepsilon = 0.5$  under the same BB+Armijo schedule, so RO and PA differ only in the  $\ell_2$  feasibility projection, not in step-size selection.

MPA performs  $R = 3$  rounds of  $K = 10$  ascent iterations, amounting to 30 ascent iterations and, counting the terminal reassignment, four reassignments in total. Its step size is constant rather than adaptive, set to 0.1 in the first round and to 0.05 thereafter, which consistently outperformed BB+Armijo in our experiments.

For SDRO, we replace the multilevel Monte Carlo estimator used in the lower-dimensional experiments of § G.1 and § 6.1 by direct sampling from the Gibbs worst-case distribution, as that estimator showed high-variance in this high-dimensional regime. We use the Metropolis-adjusted Langevin algorithm [56], which repeatedly displaces each particle by a step of size  $h$  along the gradient of the log-density, perturbs it by Gaussian noise, and accepts or rejects the resulting move to correct for discretization bias. We run 20 such steps with  $h = 5 \times 10^{-3}$  on  $n = 8$  particles per data point and set  $\varepsilon_{\text{ent}} = 0.05$ . WFR utilizes  $n = 32$  auxiliary particles per data point with the same  $\varepsilon_{\text{ent}} = 0.05$ .

**Per-epoch runtime comparison.** We record the per-epoch wall-clock time of the full min-max training loop across all methods in Table 1. All runs use PyTorch Distributed Data Parallel (DDP) across  $4 \times$  A100 GPUs. The DDP communication backend, the large-shared memory dataloader configuration, the random seed, and the per-GPU local batch size are held fixed across methods,

so that the reported per-epoch times are directly comparable wall-clock measurements at matched effective batch size and matched all-reduce overhead. In particular, the per epoch times in Table 1 are full DDP wall-clock times (forward, backward, gradient all-reduce, optimizer step, and inner-loop ascent), not single-GPU compute times.

**Certifying the absence of gradient obfuscation.** A reliable adversarial training algorithm must avoid *obfuscated gradients* [5]. These are superficial defenses that resist first-order attacks not through true robustness, but by disrupting the gradient signal via shattered, stochastic, or exploding/vanishing gradients. Hence, robustness against PGD alone is therefore insufficient as a robustness certificate. We provide two complementary checks confirming that the robustness of ICNN-DRO reflects genuine geometric regularity rather than gradient masking.

AutoAttack (AA) is a parameter-free collection of attacks, each designed specifically to expose the problems described above. Its adaptive variants APGD-CE and APGD-DLR replace the fixed step size of standard PGD with a schedule adapted to the local curvature, and so navigate ill-conditioned or shattered landscapes on which fixed-step PGD stagnates. Square Attack, by contrast, is a score-based random-search procedure that never queries the gradient of the loss function, and is therefore manifestly immune to shattered, stochastic, and vanishing-gradient effects. Reporting robustness under AA thus provides one evidence for the absence of gradient obfuscation. The tight PGD-AA gap of 0.4% for ICNN-DRO (vs. 3.55% for PA) is the quantitative signature of this consistency (see Table 1).

We complement the AutoAttack (AA) evaluation with a qualitative inspection of the input-space loss landscape, following [5, 24]. For a test sample  $z = (x, y)$ , let  $x_{\text{adv}}$  be the adversarial input obtained by  $\ell_2$ -PGD at budget  $\varepsilon = 0.5$ , using 20 steps and five random restarts. Define the adversarial displacement  $r = x_{\text{adv}} - x \in \mathbb{R}^{m-1}$  and draw a random direction  $u \in \mathbb{R}^{m-1}$  orthogonal to  $r$ . Writing  $\hat{r} = r/\|r\|_2$  and  $\hat{u} = u/\|u\|_2$ , we probe the classifier on the affine plane

$$x(\alpha, \beta) = x + \alpha \hat{r} + \beta \hat{u}, \quad (\alpha, \beta) \in [-\rho, \rho]^2, \quad \rho = 1.$$

Perturbations are measured in pixel-space  $\ell_2$  units, with standard CIFAR-10 normalization applied before classifier evaluation. We evaluate the affine plane without additional pixel clipping, recording on a  $101 \times 101$  grid both the cross-entropy  $f(\theta, (x(\alpha, \beta), y))$  and the classification correctness  $\mathbb{1}\{\arg \max_k h_\theta(x(\alpha, \beta))_k = \arg \max_k y_k\}$ .

Figure 4 presents the resulting surfaces and decision regions for two uniformly sampled test examples. The horizontal and vertical axes correspond to  $\beta \hat{u}$  and  $\alpha \hat{r}$ , respectively; the dashed circle  $\alpha^2 + \beta^2 = \varepsilon^2$  marks the attack budget. Orange and blue denote correct and incorrect classification, while the red curve traces the loss along  $\beta = 0$ . For both samples, the loss rises smoothly and monotonically from the clean input to the PGD endpoint, with no visible oscillations or plateaus along this segment. Together with the AA results, these local observations provide complementary evidence against gradient masking.

### F.3 Supplementary Details for Section 6.3

This section collects the experimental details of the robust control experiment: the CartPole environment, hyperparameter configurations, differentiable surrogate formulations, and additional results.

**Setting.** We model the CartPole environment with a continuous state  $s = (x, \dot{x}, \vartheta, \dot{\vartheta}) \in \mathbb{R}^4$ , capturing the cart’s position and velocity alongside the pole’s angle and angular velocity. The agent selects discrete actions  $a \in \{0, 1\}$ , which dictate a horizontal control force  $u \in \{-u_{\max}, u_{\max}\}$  applied directly to the cart’s center of mass. The system state then evolves according to known dynamics  $s_{t+1} = F(s_t, a_t, z)$  [7, 27] that are parameterized by the uncertainty vector  $z = (m_p, \ell) \in \mathcal{Z} \subseteq \mathbb{R}^2$ , where  $m_p$  and  $\ell$  denote the mass and half-length of the pole and  $\mathcal{Z} = [0.05, 0.20] \times [0.25, 0.75]$ . The agent receives a reward  $r_t := r(s_t, a_t) = 1$  at every non-terminal step. Episodes terminate early if the cart position or pole angle exceeds the physical bounds  $x_{\max}$  or  $\vartheta_{\max}$ , and are otherwise truncated at a maximum horizon of  $H = 1,000$  steps. Hence, the total undiscounted reward corresponds exactly to the episode length censored at 1,000. The loss function  $f(\theta, z) := -\mathbb{E}_{\pi_\theta}[\sum_{t=0}^{H-1} r(s_t, a_t)]$  is the expected negative undiscounted return under policy  $\pi_\theta$  and physics  $z$ ; see § 6.3.

Since the physical parameters  $z = (m_p, \ell)$  have incommensurable units (kilograms and meters), a standard Euclidean penalty would skew the optimization based on numerical scale. We resolve this by normalizing each coordinate by its domain width. Defining the lower and upper bounds as

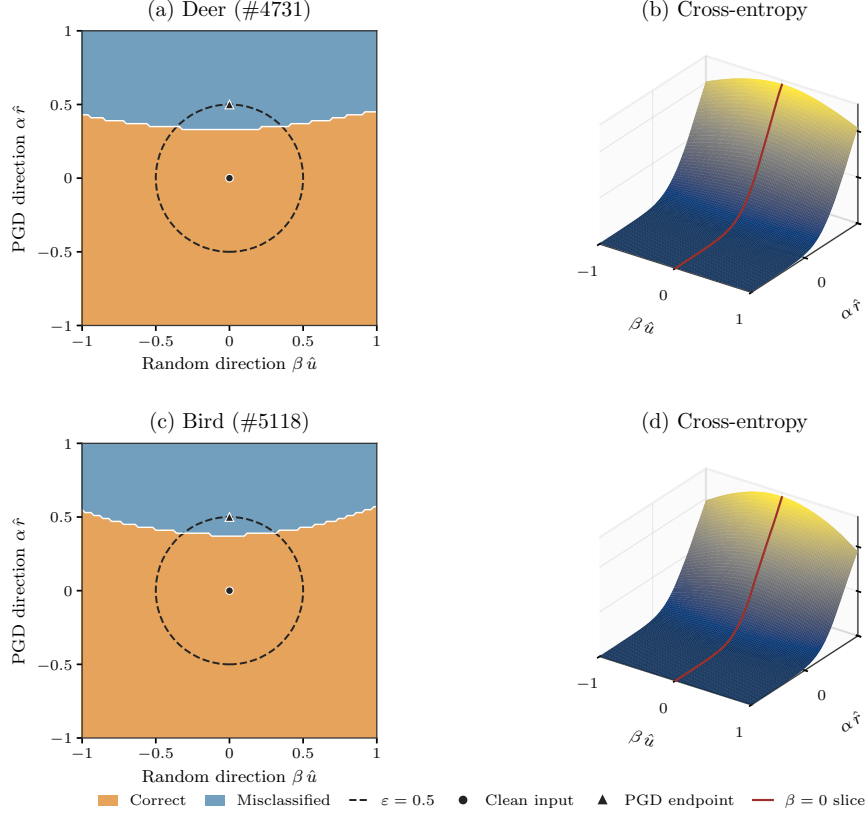


Figure 4: **Loss landscape of ICNN-DRO on CIFAR-10.** Panels (a) and (b) correspond to test sample #4731 (Deer), while panels (c) and (d) detail test sample #5118 (Bird). For each sample, the left panel (a, c) displays the 2D decision regions. The dashed circle marks the  $\ell_2$  budget  $\varepsilon = 0.5$ ; orange and blue indicate correct and incorrect classification, respectively; and point markers specify the exact coordinates of the initial clean image (black circle) and the final adversarial image (black triangle). The right panel (b, d) plots the corresponding 3D cross-entropy surface where the red curve traces the loss along the adversarial direction.

$\underline{z} = (0.05, 0.25)$  and  $\bar{z} = (0.20, 0.75)$  respectively, we employ the width-normalized metric

$$\|z - \hat{z}\|_w^2 := \sum_{k=1}^2 \left( \frac{z_k - \hat{z}_k}{w_k} \right)^2 \quad \text{with} \quad w := \bar{z} - \underline{z} = (0.15, 0.50).$$

Replacing the Euclidean norm with  $\|\cdot\|_w^2$  in the inner maximization objectives (1) and (2) is equivalent to solving the standard Euclidean DRO problem in the rescaled coordinates  $z_k \leftarrow z_k/w_k$ .

**Hyperparameters.** All methods in the robust control experiment are trained for  $E = 100$  epochs. For the outer minimization, we optimize the policy  $\pi_\theta$  using Proximal Policy Optimization (PPO) [59]. We parameterize PPO’s underlying actor (policy) and critic (value) networks as MLPs. Denoting by  $g_\theta : \mathbb{R}^4 \rightarrow \mathbb{R}^2$  the two action logits produced by the actor, the policy is  $\pi_\theta(a|s) = \exp(g_{\theta,a}(s)) / \sum_{b=0}^1 \exp(g_{\theta,b}(s))$  for  $a \in \{0, 1\}$ . We update the weights of both networks via SGD with Nesterov momentum [52] using a momentum parameter of 0.9, while keeping all other PPO settings at their standard defaults (including the outer step size  $\alpha$ ). At each iteration, we draw a batch of  $B = 64$  nominal uncertainty anchors  $\{\hat{z}_i\}_{i=1}^B$  independently from the uniform distribution on  $\mathcal{Z}$  and generate one adversarial counterpart for each anchor, yielding a combined batch of  $2B = 128$  environments. Updates are performed on this mixture of nominal and adversarial conditions so that the agent remains capable on standard tasks while learning to handle worst-case scenarios. We report results for penalty parameters  $\lambda \in \{0.1, 0.5, 1\}$ .

MPA performs  $R = 3$  rounds of  $K = 15$  ascent iterations, amounting to 45 ascent iterations in total and, including the terminal reassignment, four reassignment steps. For RO, we tune the  $\ell_2$ -

Table 2: Mean episode length  $\pm$  standard error over 1,000 deterministic rollouts across different  $\lambda$ . Panel (a) reports  $\lambda = 0.1$ ; panel (b) reports  $\lambda = 0.5$  and panel (c) reports  $\lambda = 1$ . The leftmost column corresponds to the original training configuration, the middle group to easier physics variants (Light, Long, Soft Gravity), and the right group to harder variants (Heavy, Short, Strong Gravity). Bold entries indicate the best result per column.

| Method                              | Original               | Easier Environments    |                        |                        | Harder Environments    |                        |                        |
|-------------------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
|                                     | Original               | Light                  | Long                   | Soft Gravity           | Heavy                  | Short                  | Strong Gravity         |
| <i>(a) <math>\lambda=0.1</math></i> |                        |                        |                        |                        |                        |                        |                        |
| ERM                                 | 186.4 $\pm$ 1.9        | 188.7 $\pm$ 1.9        | 191.4 $\pm$ 1.2        | 217.3 $\pm$ 1.9        | 178.6 $\pm$ 1.5        | 155.6 $\pm$ 0.8        | 157.6 $\pm$ 1.1        |
| PA                                  | 382.9 $\pm$ 7.4        | 395.1 $\pm$ 7.4        | 266.9 $\pm$ 4.1        | 543.3 $\pm$ 7.1        | 289.0 $\pm$ 5.8        | 205.1 $\pm$ 1.7        | 207.0 $\pm$ 2.8        |
| RO                                  | 380.1 $\pm$ 5.1        | 398.3 $\pm$ 5.5        | 270.5 $\pm$ 3.4        | 556.6 $\pm$ 5.3        | 295.1 $\pm$ 2.1        | 209.0 $\pm$ 1.6        | 212.4 $\pm$ 1.2        |
| WFR                                 | 507.3 $\pm$ 7.6        | 532.9 $\pm$ 8.0        | 434.5 $\pm$ 5.8        | 756.8 $\pm$ 7.2        | 387.0 $\pm$ 4.6        | 313.4 $\pm$ 2.5        | 323.9 $\pm$ 3.0        |
| SDRO                                | 239.5 $\pm$ 3.3        | 241.6 $\pm$ 4.7        | 238.2 $\pm$ 2.9        | 355.8 $\pm$ 6.0        | 213.5 $\pm$ 2.4        | 173.6 $\pm$ 1.4        | 182.7 $\pm$ 1.6        |
| NN-DRO                              | 225.3 $\pm$ 1.8        | 228.2 $\pm$ 1.6        | 221.0 $\pm$ 1.3        | 289.6 $\pm$ 3.1        | 210.3 $\pm$ 1.7        | 184.1 $\pm$ 0.6        | 185.3 $\pm$ 1.4        |
| MPA                                 | 516.4 $\pm$ 7.5        | 549.3 $\pm$ 7.0        | 446.3 $\pm$ 4.3        | 763.3 $\pm$ 8.1        | 390.6 $\pm$ 5.8        | 323.3 $\pm$ 2.7        | 337.4 $\pm$ 3.4        |
| ICNN-DRO                            | <b>771.3</b> $\pm$ 7.3 | <b>779.4</b> $\pm$ 7.5 | <b>616.3</b> $\pm$ 6.2 | <b>978.2</b> $\pm$ 1.6 | <b>665.2</b> $\pm$ 7.8 | <b>478.3</b> $\pm$ 4.3 | <b>415.3</b> $\pm$ 5.5 |
| <i>(b) <math>\lambda=0.5</math></i> |                        |                        |                        |                        |                        |                        |                        |
| ERM                                 | 186.4 $\pm$ 1.9        | 188.7 $\pm$ 1.9        | 191.4 $\pm$ 1.2        | 217.3 $\pm$ 1.9        | 178.6 $\pm$ 1.5        | 155.6 $\pm$ 0.8        | 157.6 $\pm$ 1.1        |
| PA                                  | 255.1 $\pm$ 2.0        | 259.0 $\pm$ 2.0        | 252.9 $\pm$ 1.1        | 354.2 $\pm$ 3.4        | 239.1 $\pm$ 1.3        | 219.9 $\pm$ 0.9        | 217.6 $\pm$ 1.0        |
| RO                                  | 260.3 $\pm$ 2.1        | 267.4 $\pm$ 3.2        | 258.8 $\pm$ 4.1        | 366.6 $\pm$ 7.5        | 245.1 $\pm$ 5.2        | 227.0 $\pm$ 1.2        | 221.3 $\pm$ 2.5        |
| WFR                                 | 481.9 $\pm$ 7.2        | 491.7 $\pm$ 7.3        | 375.9 $\pm$ 4.6        | 656.3 $\pm$ 7.2        | 400.6 $\pm$ 5.9        | 301.8 $\pm$ 2.2        | 311.9 $\pm$ 3.4        |
| SDRO                                | 227.6 $\pm$ 4.1        | 231.8 $\pm$ 3.8        | 216.2 $\pm$ 3.2        | 303.5 $\pm$ 6.0        | 210.7 $\pm$ 2.6        | 181.8 $\pm$ 1.3        | 184.5 $\pm$ 1.6        |
| NN-DRO                              | 221.0 $\pm$ 1.3        | 224.4 $\pm$ 1.7        | 213.0 $\pm$ 1.4        | 285.4 $\pm$ 2.5        | 207.6 $\pm$ 1.2        | 187.2 $\pm$ 0.6        | 188.2 $\pm$ 1.3        |
| MPA                                 | 641.1 $\pm$ 7.7        | 659.8 $\pm$ 7.3        | 481.4 $\pm$ 6.4        | 835.2 $\pm$ 5.0        | 488.6 $\pm$ 7.0        | 421.7 $\pm$ 3.6        | 381.6 $\pm$ 3.5        |
| ICNN-DRO                            | <b>773.6</b> $\pm$ 7.6 | <b>781.6</b> $\pm$ 7.8 | <b>618.2</b> $\pm$ 6.9 | <b>984.3</b> $\pm$ 1.8 | <b>656.3</b> $\pm$ 8.1 | <b>468.3</b> $\pm$ 4.1 | <b>401.3</b> $\pm$ 5.3 |
| <i>(c) <math>\lambda=1.0</math></i> |                        |                        |                        |                        |                        |                        |                        |
| ERM                                 | 186.4 $\pm$ 1.9        | 188.7 $\pm$ 1.9        | 191.4 $\pm$ 1.2        | 217.3 $\pm$ 1.9        | 178.6 $\pm$ 1.5        | 155.6 $\pm$ 0.8        | 157.6 $\pm$ 1.1        |
| PA                                  | 342.4 $\pm$ 5.4        | 349.9 $\pm$ 5.4        | 298.0 $\pm$ 3.1        | 545.2 $\pm$ 7.4        | 278.8 $\pm$ 3.0        | 236.4 $\pm$ 1.3        | 238.4 $\pm$ 1.8        |
| RO                                  | 345.0 $\pm$ 4.2        | 351.6 $\pm$ 4.3        | 300.6 $\pm$ 2.7        | 551.2 $\pm$ 2.3        | 286.3 $\pm$ 1.2        | 240.0 $\pm$ 3.8        | 243.7 $\pm$ 1.6        |
| WFR                                 | 435.0 $\pm$ 6.8        | 459.6 $\pm$ 7.3        | 387.4 $\pm$ 5.2        | 703.8 $\pm$ 7.7        | 337.5 $\pm$ 4.1        | 276.2 $\pm$ 2.0        | 287.3 $\pm$ 2.8        |
| SDRO                                | 246.4 $\pm$ 4.5        | 248.7 $\pm$ 4.5        | 244.6 $\pm$ 3.5        | 378.1 $\pm$ 7.0        | 215.9 $\pm$ 2.9        | 176.7 $\pm$ 1.1        | 183.3 $\pm$ 1.8        |
| NN-DRO                              | 228.0 $\pm$ 1.7        | 231.4 $\pm$ 1.9        | 227.0 $\pm$ 1.1        | 293.3 $\pm$ 2.7        | 214.8 $\pm$ 1.5        | 193.4 $\pm$ 0.9        | 192.6 $\pm$ 1.1        |
| MPA                                 | 570.3 $\pm$ 8.4        | 581.6 $\pm$ 8.5        | 428.6 $\pm$ 5.9        | 772.1 $\pm$ 6.8        | 426.3 $\pm$ 7.0        | 342.3 $\pm$ 2.5        | 339.8 $\pm$ 3.6        |
| ICNN-DRO                            | <b>829.3</b> $\pm$ 6.6 | <b>837.4</b> $\pm$ 6.3 | <b>797.5</b> $\pm$ 6.7 | <b>992.7</b> $\pm$ 0.6 | <b>725.2</b> $\pm$ 7.7 | <b>561.7</b> $\pm$ 6.4 | <b>472.5</b> $\pm$ 7.6 |

perturbation budget to the regularization level, using  $\varepsilon = 0.75$  for  $\lambda = 0.1$ ,  $\varepsilon = 0.5$  for  $\lambda = 0.5$ , and  $\varepsilon = 0.25$  for  $\lambda = 1.0$ . Both entropy-regularized baselines, SDRO and WFR, use a regularization strength of  $\varepsilon_{\text{ent}} = 0.01$ . WFR approximates the adversarial distribution using  $n = 8$  auxiliary particles per data point, whereas SDRO uses a multilevel Monte Carlo gradient estimator based on a truncated geometric distribution with success probability  $p = 0.5$  and truncation level  $M = 3$ .

**Additional results.** To evaluate the sensitivity of the learned robustness to the regularization parameter  $\lambda$ , Table 2 reports results for three values, with all other training and evaluation settings kept fixed: panel (a) uses  $\lambda = 0.1$ , panel (b) uses  $\lambda = 0.5$ , and panel (c) uses  $\lambda = 1.0$ , corresponding to the experiment reported in Figure 3. ICNN-DRO attains the highest mean episode length on every evaluation variant at every penalty level, with MPA second throughout. MPA’s advantage over the remaining baselines is substantial at  $\lambda = 0.5$  and  $\lambda = 1.0$ , but narrows at  $\lambda = 0.1$ , where it stays ahead of WFR on all seven variants by only a small margin. WFR is consistently the strongest remaining baseline, but falls substantially behind ICNN-DRO on the harder extrapolative shifts, particularly Heavy and Short. Overall, performance varies non-monotonically with  $\lambda$  for several methods, while the ordering of the two strongest approaches remains stable throughout.

**Differentiable rollouts and pathwise gradients.** To ascend  $\mathcal{L}(\theta; \omega)$  with respect to  $\omega$ , we need  $\nabla_{\omega} f(\theta, T_{\omega}(\hat{z}))$ . Applying the chain rule, the gradient of the objective  $f(\theta, T_{\omega}(\hat{z}))$  decomposes as

$$\nabla_{\omega} f(\theta, T_{\omega}(\hat{z})) = \left[ \frac{\partial T_{\omega}(\hat{z})}{\partial \omega} \right]^{\top} \nabla_z f(\theta, z) \Big|_{z=T_{\omega}(\hat{z})}.$$

We must resolve three distinct non-differentiability challenges to ensure both components of this chain rule permit continuous gradient flow. Specifically, the environment gradient  $\nabla_z f(\theta, z)$  presents two discontinuities within the standard MDP formulation, while the map Jacobian  $\partial T_\omega(\hat{z})/\partial \omega$  introduces a third related to physical boundaries. We address these in sequence.

Evaluating  $\nabla_z f(\theta, z)$ , which is also required by particle-based baselines like PA and SDRO to optimize their adversarial perturbations, requires differentiating the episodic return directly through the unrolled environment dynamics; *i.e.*,  $\nabla_z r_t$ . Following the decoupled formulation in [76, Appendix A.1], the gradient of the per-step reward  $r_t$  with respect to  $z$  propagates recursively

$$\nabla_z r_t = (\nabla_z a_t)^\top \nabla_{a_t} r_t + (\nabla_z s_t)^\top \nabla_{s_t} r_t,$$

where the state and action Jacobians are given recursively by

$$\nabla_z s_t = \frac{\partial s_t}{\partial z} + \frac{\partial s_t}{\partial s_{t-1}} \nabla_z s_{t-1} + \frac{\partial s_t}{\partial a_{t-1}} \nabla_z a_{t-1}, \quad \nabla_z a_{t-1} = \frac{\partial a_{t-1}}{\partial s_{t-1}} \nabla_z s_{t-1}.$$

Two terms in this standard discrete MDP formulation break the gradient flow. First, the original action  $a_t \in \{0, 1\}$  is discrete, rendering  $\nabla_z a_t = 0$  almost everywhere and completely severing the gradient. We address this with a straight-through estimator [8]. Specifically, we define a vector of available physical forces  $\mathbf{u} := (-u_{\max}, u_{\max})^\top$ . In the forward pass, the simulator applies the hard action  $a_t = \arg \max_{a \in \{0, 1\}} g_{\theta, a}(s_t)$ , so that the realized force is  $-u_{\max}$  if  $a_t = 0$  and  $u_{\max}$  otherwise. In the backward pass, derivatives are instead taken through the relaxed control signal  $u_t := \text{softmax}(g_\theta(s_t))^\top \mathbf{u}$ , where

$$[\text{softmax}(\mathbf{v})]_a = \frac{e^{v_a/\tau}}{\sum_{a' \in \{0, 1\}} e^{v_{a'}/\tau}}$$

for a temperature  $\tau > 0$ , so that the degenerate term  $\nabla_z a_t$  is replaced by  $\nabla_z u_t$ . Second, an episode terminates once  $|x_t| > x_{\max}$  or  $|\vartheta_t| > \vartheta_{\max}$ , at which point the per-step reward switches from one to zero. As a function of  $s_t$ , this reward has zero gradient away from the termination boundaries and is nondifferentiable on the boundaries. We circumvent this pathology by replacing the hard indicator with a smooth survival function  $p_t(s_t)$ , which captures the probability that the agent remains within physical bounds at time  $t$ . Given the limits  $x_{\max}$  and  $\vartheta_{\max}$  and the logistic sigmoid function  $\sigma(x) = (1 + e^{-\beta x})^{-1}$  with a sharpness parameter  $\beta > 0$ , we define

$$p_t(s_t) = \sigma(x_{\max} - |x_t|) \cdot \sigma(\vartheta_{\max} - |\vartheta_t|).$$

The loss  $f(\theta, z)$  is then approximated by the sum of cumulative survival probabilities over the horizon

$$f(\theta, z) \approx - \sum_{t=0}^{H-1} \prod_{k=0}^t p_k(s_k).$$

As the parameters  $\beta \rightarrow \infty$  and  $\tau \rightarrow 0^+$ , this formulation exactly recovers the true undiscounted return and the discrete action logic of the original MDP. By operating away from these asymptotic limits in practice (we set  $\beta = 20$  and  $\tau = 1$ ), these relaxations provide a smooth approximation that carries informative gradients throughout the rollout. This aligns with established practices in differentiable simulation, where smoothing discrete transitions is essential to prevent chaotic or vanishing gradients [27, 37, 49, 63]. With these relaxations, we estimate the environment gradient  $\nabla_z f(\theta, z)$  by backpropagating the surrogate loss through each simulated trajectory while holding its sampled initial state fixed, and then averaging the resulting derivatives over the sampled initial states. This procedure is known as pathwise, or reparameterization, gradient estimation [34]; see also [51, §5]. Because the smooth survival score modifies the reward and the straight-through action rule substitutes relaxed derivatives for discrete actions, the resulting estimator is generally biased relative to the original discrete-MDP objective [8].

With the issues related to  $\nabla_z f(\theta, z)$  resolved, we finally address the third challenge regarding the map Jacobian  $\frac{\partial T_\omega(\hat{z})}{\partial \omega}$ . The physical parameter  $z$  must satisfy the box constraints  $\underline{z} \leq z \leq \bar{z}$  for the dynamics to remain well-defined. While particle-based baselines (such as PA and WFR) can enforce this via projected gradient ascent, which does not require differentiating through the projection operator itself, optimizing  $\omega$  requires backpropagating through the entire mapping mechanism. If we bounded the map’s output using a hard projection operator, the Jacobian  $\partial T_\omega(\hat{z})/\partial \omega$  would evaluate

to zero everywhere outside the valid domain (*i.e.*, whenever  $z > \bar{z}$  or  $z < \underline{z}$ ), severing the gradient flow. To guarantee a non-vanishing Jacobian, we reparameterize  $T_\omega$  such that its output is naturally bounded. We lift the normalized nominal parameters into an unconstrained latent space using the logistic sigmoid function  $\sigma(x) = (1 + e^{-x})^{-1}$ , apply a neural mapping  $H_\omega$ , and then map the result back to the physical bounds via the logistic sigmoid function  $\sigma$ . Expressed as a single composition, the full adversarial map becomes

$$T_\omega(\hat{z}) = \underline{z} + (\bar{z} - \underline{z}) \odot \sigma \left( H_\omega \left( \sigma^{-1} \left( \frac{\hat{z} - \underline{z}}{\bar{z} - \underline{z}} \right) \right) \right),$$

where both  $\sigma$  and  $\sigma^{-1}$  are applied element-wise and  $\odot$  denotes the Hadamard product. For the NN-DRO baseline,  $H_\omega$  is parameterized as a standard Multi-Layer Perceptron. For our ICNN-DRO approach,  $H_\omega$  is explicitly modeled as the Brenier gradient  $\nabla\psi_\omega$ . This formulation respects the physical box constraints while remaining differentiable end-to-end.

## G Additional Experiments

This section collects experiments and analyses that complement the main text. Section G.1 presents a robust least-squares experiment; Section G.2 examines the transferability of adversarial perturbations across classifiers; Section G.3 audits empirical Monge gaps across penalty levels. Two further studies probe the geometry of ICNN-DRO itself: Section G.4 reveals that the rank  $r_{\text{out}}$  of the quadratic term in the ICNN’s output layer—rather than its depth or width—is the binding constraint on adversarial capacity, and Section G.5 investigates the effect of annealing the penalty  $\lambda$  during training.

### G.1 Robust Least Squares

We evaluate robustness on a synthetic *uncertain least squares* problem, following the setup in [75], which in turn builds on [23, 79]. The goal is to learn a parameter vector  $\theta \in \Theta = \mathbb{R}^{10}$  for a least-squares problem in which uncertainty enters through the system matrix. Specifically, the uncertainty variable is one-dimensional, with values restricted to the uncertainty domain  $[-1, 1]$ . The per-sample loss is defined as  $f(\theta, z) := \|A(z)\theta - b\|_2^2$ , where  $A(z) := A_0 + zA_1 \in \mathbb{R}^{10 \times 10}$ . The matrices  $A_0, A_1 \in \mathbb{R}^{10 \times 10}$  and the vector  $b \in \mathbb{R}^{10}$  are fixed realizations with entries sampled independently from  $\mathcal{N}(0, 1)$ . To simulate a limited observation window, we generate  $N = 10$  nominal training samples  $\{\hat{z}_i\}_{i=1}^N$  independently from the uniform distribution on  $[-0.5, 0.5]$  to construct the empirical reference measure  $\hat{\mathbb{P}} = \frac{1}{N} \sum_{i=1}^N \delta_{\hat{z}_i}$ . Robustness is then evaluated by testing the learned models on a shifted distribution  $\mathbb{P}_\Delta := \text{Unif}([-0.5(1 + \Delta), 0.5(1 + \Delta)])$  for  $\Delta \in [0, 10]$ , where larger  $\Delta$  corresponds to a greater departure from the training distribution.

All methods are trained for  $E = 10$  epochs using full-batch gradient descent ( $N = B = 10$ ) with an outer step size of  $\alpha = 0.01$  and a penalty parameter of  $\lambda = 0.1$ . Every inner maximization is given the same budget of 3,000 ascent iterations, with step sizes selected by the BB+Armijo procedure (Algorithm 3) under the best configuration from our search, namely  $[\eta_{\min}, \eta_{\max}] = [10^{-6}, 10^1]$ ,  $\eta_0 = 10^{-1}$ , and  $(c, \tau, J) = (10^{-4}, 0.5, 10)$ . MPA splits this budget into  $R = 5$  rounds of  $K = 600$  iterations, which amounts to six reassignments once the terminal one is counted, and RO has an  $\ell_2$ -perturbation budget of  $\varepsilon = 0.316$ . Both entropy-regularized baselines, SDRO and WFR, utilize a regularization strength of  $\varepsilon_{\text{ent}} = 0.1$ . Within this group, WFR approximates the adversarial distribution using  $n = 8$  auxiliary particles per data point, while SDRO employs a multilevel Monte Carlo gradient estimator based on a truncated geometric distribution with success probability  $p = 0.5$  and a truncation level of  $M = 4$ .

The results in Figure 5, averaged over 10 independent runs, compare ICNN-DRO against ERM and the established robust baselines. As  $\Delta$  increases, all DRO variants substantially outperform

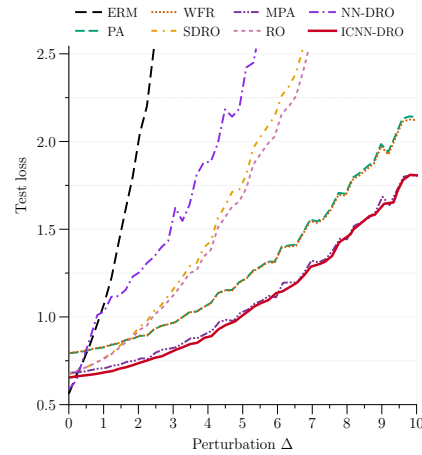


Figure 5: Test loss vs. perturbation level  $\Delta$  on uncertain least squares.

ERM, confirming that robust training improves stability under distribution shift. Among the DRO baselines, the SDRO method degrades the fastest, likely due to the bias inherent in its nested Monte Carlo estimation. Since  $m = 1$ , [Proposition 4.1](#) guarantees that PA is already cyclically monotone, so one might expect PA and MPA to perform similarly. Yet MPA still outperforms PA, showing that its benefit extends beyond restoring cyclical monotonicity. Even when cyclical monotonicity already holds, the reassignment step acts as an implicit multi-start mechanism, letting each particle escape a weak local maximum by inheriting a stronger point discovered by a neighboring sample. ICNN-DRO attains the best performance in this experiment, followed closely by MPA, while PA and WFR trail by a wider margin. This behavior is expected because, in one dimension, the inner maximization pushes worst-case samples toward the boundaries of the uncertainty set, collapsing to either  $-1$  or  $1$ . As a result, the various DRO formulations ultimately discover the same worst-case support, leaving limited room for our structured map-based approach to demonstrate a significant advantage.

## G.2 Cross-Model Universality

When solving (DRO) with an ICNN as described in [Section 5](#), the resulting adversarial perturbations are constructed with respect to a specific trained model  $\theta$ . This raises the question of whether these perturbations are tailored to that model or whether they generalize across different models. In this appendix, we investigate the extent to which perturbations generated against a given *source model* remain effective when applied to other *target models* that differ from it in training or in architecture. Here, “universality” should be understood in the *cross-model* sense: the ICNN still produces an *input-dependent* perturbation for each test example, but we ask whether that perturbation continues to fool models other than the one used during adversary training.

We consider a population of CIFAR-10 classifiers consisting of five ResNet-18 models (R1–R5) and four Vision Transformers (ViT1–ViT4) [\[21\]](#), where models within each group differ only in their training hyperparameters. We take R1 (the first ResNet-18) as the *source model*, train an ICNN adversary against it, and denote the resulting pair by R1-ICNN. The remaining models, namely R2-R5 and ViT1-ViT4, serve as *target models*. This distinction is important. Specifically, the source model is the one that appears inside the minimax problem during adversary training, whereas the target models are used only at evaluation time to test transfer.

We evaluate on a fixed subset of  $N = 9,085$  CIFAR-10 test points. For each point, we generate one perturbation against the source model R1 using its trained ICNN

adversary, and then apply that perturbed input unchanged to every target model. The source model is misclassified on 2,336 of these perturbed examples. In [Table 3](#), the accuracy block reports clean accuracy, adversarial accuracy, and their gap on this same fixed subset. The transfer block then measures how source failures propagate to each target: *Ovlp.* counts perturbed examples misclassified by both source and target (and is therefore at most 2,336), *% src* is this overlap as a percentage of the 2,336 source adversarial failures, and *Lbl mis.%* is the percentage of overlap examples on which source and target predict different incorrect labels.

It is clear that transfer is much stronger within the ResNet family than across architectures. For the held-out ResNets R2-R5, adversarial accuracy drops by 39.26 to 55.80 points and the transferred attack recovers 82%–91% of the source errors. For ViT1-ViT4, the drops are smaller, 24.68 to 39.98 points, and the recovery rate falls to 73%–81%. This indicates that the learned perturbations are not merely exploiting one particular set of weights; rather, they capture vulnerabilities that are shared most strongly within the architectural class of the source model. ResNet targets are not only fooled more often (*% src*: 82%–91% vs. 73%–81% for ViTs), but also more often agree with the source on the wrong label (*Lbl mis. (%)*: 36.76%–47.85% vs. 54.13%–56.96%). This suggests that perturbations learned

Table 3: **Cross-architecture transfer of the ICNN adversary trained against R1.** *Drop*: Clean-Adv. gap; *Ovlp*: counts examples fooled by both the source and the target; *% src err*: percentage of source-model failures that transfer to the target; *Lbl mis.%*: percentage of overlap examples on which source and target predict different incorrect labels.

| Model                  | Accuracy (%)     |                 |                   | Transfer |       |           |
|------------------------|------------------|-----------------|-------------------|----------|-------|-----------|
|                        | Clean $\uparrow$ | Adv. $\uparrow$ | Drop $\downarrow$ | Ovlp.    | % src | Lbl mis.% |
| R1-ICNN ( <i>src</i> ) | 92.38            | 74.29           | 18.09             | 2,336    | —     | —         |
| ViT1                   | 89.37            | 49.39           | 39.98             | 1,903    | 81.46 | 56.96     |
| ViT2                   | 78.22            | 53.54           | 24.68             | 1,696    | 72.60 | 54.13     |
| ViT3                   | 82.32            | 54.38           | 27.95             | 1,763    | 75.47 | 55.30     |
| ViT4                   | 92.42            | 57.29           | 35.12             | 1,839    | 78.72 | 55.41     |
| R2                     | 98.14            | 42.34           | 55.80             | 2,120    | 90.75 | 42.78     |
| R3                     | 91.54            | 50.99           | 40.55             | 2,043    | 87.46 | 36.76     |
| R4                     | 89.99            | 46.35           | 43.64             | 2,029    | 86.86 | 42.34     |
| R5                     | 91.71            | 52.45           | 39.26             | 1,927    | 82.49 | 47.85     |

Table 4: Monge gap on CIFAR10.

| Method   | Regularization strength $\lambda$ |         |         |         |         |
|----------|-----------------------------------|---------|---------|---------|---------|
|          | 2                                 | 5       | 10      | 15      | 30      |
| ERM      | 0                                 | 0       | 0       | 0       | 0       |
| PA       | 1.24e-2                           | 1.15e-2 | 3.01e-3 | 7.93e-4 | 7.97e-4 |
| WFR      | 1.13e-2                           | 1.02e-2 | 1.73e-3 | 5.68e-4 | 5.58e-4 |
| SDRO     | 1.83                              | 1.04    | 4.52e-3 | 1.27e-3 | 1.29e-3 |
| NN-DRO   | 1.37e-2                           | 1.21e-2 | 3.54e-3 | 3.28e-3 | 3.16e-3 |
| Ours     |                                   |         |         |         |         |
| MPA      | 1.22e-3                           | 1.08e-3 | 9.03e-4 | 5.89e-4 | 5.07e-4 |
| ICNN-DRO | 1.52e-7                           | 1.33e-7 | 1.28e-7 | 1.25e-7 | 1.12e-7 |

against R1 transfer more faithfully within the ResNet family. This is a desirable failure mode for a robust training method: transfer persists beyond the source model, but degrades across architectures.

### G.3 An Empirical Monge Gap

The analysis in § 3 shows that cyclical monotonicity is tied to the quality of the inner maximization. By Proposition 3.2, transport waste induced by a non-monotone admissible map appears as a positive Monge gap, and hence as suboptimality in problem (2). In this appendix, we examine this effect empirically by estimating the Monge gap in two settings: the feature-space logistic regression experiment in § 6.1 and the robust CartPole control experiment in § 6.3.

Recall from Definition 2.3 that, for any  $\hat{\mathbb{P}} \in \mathcal{P}$ , the Monge gap of a map  $T \in \mathcal{T}$  is given by

$$\mathcal{M}_{\hat{\mathbb{P}}}(T) = \mathbb{E}_{\hat{\mathbb{P}}}[\|\hat{Z} - T(\hat{Z})\|_2^2] - W^2(T_{\#}\hat{\mathbb{P}}, \hat{\mathbb{P}}).$$

In this appendix we specialize this definition to the empirical measure  $\hat{\mathbb{P}} = \frac{1}{B} \sum_{i=1}^B \delta_{\hat{z}_i}$ . Since  $W^2(T_{\#}\hat{\mathbb{P}}, \hat{\mathbb{P}})$  is costly to evaluate directly for high-dimensional empirical measures, we follow the approach in [68] and replace it with the Sinkhorn divergence. Specifically, for  $\mathbb{P}, \hat{\mathbb{P}} \in \mathcal{P}$ , we first define the entropy-regularized optimal transport cost as

$$\text{OT}(\mathbb{P}, \hat{\mathbb{P}}) := \inf \{ \mathbb{E}_{\gamma}[\|Z - \hat{Z}\|_2^2] + \rho \text{KL}(\gamma \| \mathbb{P} \otimes \hat{\mathbb{P}}) : \gamma \in \Gamma(\mathbb{P}, \hat{\mathbb{P}}) \},$$

and we write  $S(\mathbb{P}, \hat{\mathbb{P}})$  for the associated Sinkhorn divergence as  $S(\mathbb{P}, \hat{\mathbb{P}}) := \text{OT}(\mathbb{P}, \hat{\mathbb{P}}) - \frac{1}{2} \text{OT}(\mathbb{P}, \mathbb{P}) - \frac{1}{2} \text{OT}(\hat{\mathbb{P}}, \hat{\mathbb{P}})$  [25]. We then estimate the Monge gap  $\mathcal{M}_{\hat{\mathbb{P}}}(T)$  by

$$\widehat{\mathcal{M}}_{\hat{\mathbb{P}}}(T) := \mathbb{E}_{\hat{\mathbb{P}}}[\|T(\hat{Z}) - \hat{Z}\|_2^2] - S(T_{\#}\hat{\mathbb{P}}, \hat{\mathbb{P}}).$$

The parameter  $\rho > 0$  is used only in this Sinkhorn computation and is distinct from the training-time entropic regularization  $\varepsilon_{\text{ent}}$  used in SDRO and WFR. Moreover, we have  $\widehat{\mathcal{M}}_{\hat{\mathbb{P}}}(T) \geq \mathcal{M}_{\hat{\mathbb{P}}}(T)$ , and the two coincide when  $\rho = 0$ ; see [68, Appendix A.3]. Thus, we obtain an upper bound on the actual Monge gap. In all experiments we set

$$\rho = 0.01 \cdot \text{median}\{\|u - v\|_2^2 : u, v \in \{\hat{z}_i\}_{i=1}^B \cup \{T(\hat{z}_i)\}_{i=1}^B, u \neq v\},$$

that is,  $\rho$  is 1% of the median pairwise squared distance between distinct points in the combined set of empirical and adversarial samples. We solve the resulting Sinkhorn problem for 1,000 iterations to solver tolerance  $10^{-7}$ . Next, we report the (approximate) Monge gap for the experiments in § 6.1 and § 6.3; see Table 4 and Table 5. A common pattern in both tables is that, as  $\lambda$  increases, the quadratic penalty in (1) and (2) becomes more dominant, and thus the objective function of the inner maximization problem becomes more concave. Hence, the adversarial map is forced closer to the identity, and the problem moves closer to ERM. Accordingly, the Monge gap decreases across methods. The ERM row is identically zero in both tables, since its map is the identity.

**Adversarial multi-class logistic regression (§ 6.1).** We evaluate  $\widehat{\mathcal{M}}_{\hat{\mathbb{P}}}(T)$  on the full CIFAR-10 dataset, so that  $N = B = 60,000$ , and report the results in Table 4 for  $\lambda \in \{2, 5, 10, 15, 30\}$ ; the main-text experiment in § 6.1 uses  $\lambda = 10$ . At  $\lambda = 10$ , ICNN-DRO is the best-performing method in Figure 2 and also attains the smallest Monge gap estimate, essentially at numerical precision. This is consistent with the role of transport waste in the inner-maximization error. Still, the Monge gap

Table 5: Monge gap on robust control (CartPole).

|           | Method   | Regularization strength $\lambda$ |         |         |
|-----------|----------|-----------------------------------|---------|---------|
|           |          | 0.1                               | 0.5     | 1       |
| Baselines | ERM      | 0                                 | 0       | 0       |
|           | PA       | 4.87e-2                           | 2.01e-2 | 1.99e-2 |
|           | WFR      | 1.87e-2                           | 1.41e-2 | 1.13e-2 |
|           | SDRO     | 2.18e-2                           | 2.03e-2 | 1.93e-3 |
|           | NN-DRO   | 2.17e-2                           | 2.07e-2 | 1.98e-3 |
| Ours      | MPA      | 6.16e-6                           | 1.16e-6 | 1.15e-6 |
|           | ICNN-DRO | 6.92e-7                           | 6.82e-7 | 8.81e-8 |

does not determine the full empirical ranking: MPA has the second-smallest Monge gap at  $\lambda = 10$  and improves over PA by roughly an order of magnitude (see Table 4), yet it is not the second-best method empirically. This is not surprising, since the Monge gap appears only in a lower bound on the adversarial suboptimality; cf. Proposition 3.2.

**Robust control (§ 6.3).** For this experiment, we evaluate  $\widehat{\mathcal{M}}_{\mathbb{P}}(T)$  on a set of  $B = 32,768$  points sampled uniformly from the CartPole box  $(m_p, \ell) \in [0.05, 0.20] \times [0.25, 0.75]$ , using adversaries trained for  $H = 1,000$  episode steps. The resulting estimates are reported in Table 5 for  $\lambda \in \{0.1, 0.5, 1\}$ . Across all three values of  $\lambda$ , ICNN-DRO attains the smallest Monge gap estimate, while MPA is consistently second and improves over PA by roughly three orders of magnitude. The method with the smallest gap throughout, ICNN-DRO, yields also the best-performing controller in § 6.3. At  $\lambda = 1$  (i.e., the value used in Figure 3 in § 6.3), ICNN-DRO achieves the best control performance, with MPA as the runner-up; these are also the two methods with the smallest Monge gap estimates in Table 5. Thus, although the Monge gap enters only in a lower bound on the adversarial error, it still appears to be a useful indicator of empirical performance in this low-dimensional control setting.

#### G.4 Design Axes of the ICNN Adversary

Three design choices govern the effective capacity of the ICNN adversary with architecture (ICNN). The first is the depth and the width of the hidden layers. The second is the placement of the quadratic injections, either at every hidden layer or only at the readout. The third is the rank  $r_{\text{out}}$  of the matrix  $A_L \in \mathbb{R}^{r_{\text{out}} \times m}$  that generates the quadratic term of the readout. Depth, width, and injection placement are the conventional levers, whereas  $r_{\text{out}}$  has no counterpart in standard network design. We now isolate the contribution of each axis and show that the third one dominates.

**Design grid.** For the depth and width axes, we consider three backbones,  $[64, 64, 64, 64]$  (simple),  $[128, 128, 128, 128]$  (moderate) and  $[1024, 512, 512, 256, 128, 64]$  (complex). For the quadratic injections, we consider two variants: quadratic only in the readout (i.e., ICNN-DRO) and quadratic at all layers (i.e., ICNN-DRO-Q). For the readout rank we sweep  $r_{\text{out}} \in \{32, 128, 512\}$  in the readout-only variant and  $r_{\text{out}} \in \{8, 32, 128\}$  in the all-layers variant. We repeat the robust image classification experiment in Section 6.2 with all of these network architectures. The results are collected in Table 6. We observe that a simple ICNN equipped with a high-rank readout outperforms substantially larger ICNNs with low-rank readouts, while requiring markedly less computation. Hence,  $r_{\text{out}}$  serves as the effective capacity knob of our architecture and determines the strength of the learned adversarial map more directly than network width or depth, and than the injection of quadratics.

**Why the readout rank is the binding constraint.** To understand why the expressiveness of the adversarial map is mainly determined by  $r_{\text{out}}$ , we isolate the readout quadratic by setting  $\delta_L = 0$  in (ICNN). With  $A_L = [a_1, \dots, a_{r_{\text{out}}}]^\top$ , the transport map  $T_\omega(z) = \nabla_z \psi_\omega(z)$  expands to

$$T_\omega(z) = \nabla_z \phi_\omega(z) + \sum_{k=1}^{r_{\text{out}}} (a_k a_k^\top) z + w_L^z,$$

where  $\phi_\omega(z) := \exp(w_L^y)^\top y_L(z)$  collects the output of the hidden layers. The adversary perturbs nominal samples via two main components. The first component is the neural network gradient  $\nabla_z \phi_\omega(z)$ , and the second is a linear transformation of the input sample  $\sum_{k=1}^{r_{\text{out}}} (a_k a_k^\top) z$ . Crucially,

Table 6: **Adversary capacity versus robustness.** Best results per column are in bold, and second-best results are underlined. PGD cells are shaded green for configurations satisfying our robustness criterion (PGD accuracy  $\geq 56\%$ ), with darker green indicating stronger robustness, and red otherwise, with darker red indicating a larger shortfall. Runtime cells are shaded green only for configurations satisfying this robustness criterion, with darker green indicating lower computational cost. A runtime marked with  $\dagger$  corresponds to a configuration with PGD accuracy below 56%. Although such configurations may be substantially faster, we do not regard them as successful accelerations of ICNN training because they fail our minimum robustness criterion. The objective of this ablation is specifically to reduce the approximately  $10^4$  s wall-clock training time of the full-width quadratic ICNN while retaining at least 56% PGD accuracy.

| Method                                                 | $r_{\text{out}}$ | Total<br>params | CIFAR-10     |                          | Wall-clock<br>Runtime (s) |
|--------------------------------------------------------|------------------|-----------------|--------------|--------------------------|---------------------------|
|                                                        |                  |                 | Clean        | PGD $_{\varepsilon=0.5}$ |                           |
| <i>Complex backbone</i> [1024, 512, 512, 256, 128, 64] |                  |                 |              |                          |                           |
| ICNN-DRO-Q                                             | 8                | 31,693,314      | 91.37        | 57.36                    | 10854                     |
| ICNN-DRO-Q                                             | 32               | 31,767,042      | 91.30        | 57.41                    | 10908                     |
| ICNN-DRO-Q                                             | 128              | 32,061,954      | 91.21        | 57.58                    | 11325                     |
| ICNN-DRO                                               | 8                | 8,665,602       | <b>91.45</b> | 57.28                    | 6439                      |
| ICNN-DRO                                               | 32               | 8,739,330       | <u>91.42</u> | 57.40                    | 6813                      |
| ICNN-DRO                                               | 128              | 9,034,242       | 91.35        | 57.51                    | 6909                      |
| ICNN-DRO                                               | 512              | 10,213,890      | 91.35        | <b>58.46</b>             | 7095                      |
| <i>Moderate backbones</i> [128, 128, 128, 128]         |                  |                 |              |                          |                           |
| ICNN-DRO-Q                                             | 8                | 6,402,690       | 90.61        | 43.38                    | 6271 <sup>†</sup>         |
| ICNN-DRO-Q                                             | 32               | 6,476,418       | 90.81        | 53.48                    | 6548 <sup>†</sup>         |
| ICNN-DRO-Q                                             | 128              | 6,771,330       | 91.10        | 55.70                    | 6729 <sup>†</sup>         |
| ICNN-DRO                                               | 32               | 1,733,250       | 90.85        | 53.46                    | 4089 <sup>†</sup>         |
| ICNN-DRO                                               | 128              | 2,028,162       | 91.07        | 55.68                    | 4135 <sup>†</sup>         |
| ICNN-DRO                                               | 512              | 3,207,810       | 91.36        | <u>58.22</u>             | 4473                      |
| <i>Simple backbones</i> [64, 64, 64, 64]               |                  |                 |              |                          |                           |
| ICNN-DRO-Q                                             | 8                | 3,219,778       | 90.60        | 43.40                    | 4462 <sup>†</sup>         |
| ICNN-DRO-Q                                             | 32               | 3,293,506       | 90.81        | 52.44                    | 4483 <sup>†</sup>         |
| ICNN-DRO-Q                                             | 128              | 3,588,418       | 90.93        | 55.25                    | 4665 <sup>†</sup>         |
| ICNN-DRO                                               | 16               | 860,482         | 90.67        | 47.56                    | 3118 <sup>†</sup>         |
| ICNN-DRO                                               | 32               | 909,634         | 90.81        | 52.42                    | 3415 <sup>†</sup>         |
| ICNN-DRO                                               | 64               | 1,007,938       | 90.83        | 53.17                    | 3568 <sup>†</sup>         |
| ICNN-DRO                                               | 128              | 1,204,546       | 90.92        | 55.21                    | 3713 <sup>†</sup>         |
| ICNN-DRO                                               | 256              | 1,597,762       | 91.32        | 56.81                    | <b>3867</b>               |
| ICNN-DRO                                               | 512              | 2,384,194       | 91.41        | 58.11                    | <u>4037</u>               |

the second term is confined to  $\text{span}\{a_1, \dots, a_{r_{\text{out}}}\}$ , meaning it can only ever explore a fixed set of at most  $r_{\text{out}}$  directions. While  $\nabla_z \phi_\omega(z)$  could theoretically supply additional directions, it is empirically dominated by the second term, meaning that the relative norm  $\|\nabla_z \phi_\omega(z)\|_2 / \|A_L^\top A_L z\|_2$  is empirically small. More precisely, across 4,096 training samples, this relative norm is merely 0.0711 for  $r_{\text{out}} = 1$  and 0.0910 for  $r_{\text{out}} = 2$ . Because the second term dictates the overall perturbation magnitude,  $r_{\text{out}}$  effectively governs the map’s total directional capacity. This explains why increasing the hidden backbone width or depth (Table 6) cannot overcome a low readout rank, and increasing  $r_{\text{out}}$  serves as the main component of the expressivity of the ICNN.

## G.5 Penalty Annealing for the Global Map

The penalty parameter  $\lambda$  in (DRO) controls the strength of the adversary. A large  $\lambda$  severely constrains the adversary, forcing the worst-case map toward the identity and reducing the objective to standard ERM. Conversely, a small  $\lambda$  permits a highly potent adversary. Rather than committing to a single  $\lambda$ , we anneal it along a decreasing schedule  $\lambda_1 > \lambda_2 > \dots > \lambda_S$  and, crucially, *never reset the adversarial map between stages*. The ICNN weights  $\omega$  and the classifier  $\theta$  learned under  $\lambda_s$  initialize the next phase under  $\lambda_{s+1}$ . This defines a smooth sequence of penalized DRO problems, one per penalty level. Note that warm-starting each stage from the previous solution is coherent

Table 7: **Penalty annealing versus fixed-penalty adversarial training on CIFAR-10.** The  $K = 20$  fixed- $\lambda$  baseline is reproduced from Table 1 for comparison. Best results per column are in bold, and the second best are underlined.

| Method                           | $K$ | CIFAR-10     |                          |                         | CIFAR-10.1   | CIFAR-10.2   | CIFAR-10-C (Common Corruption) |              |              |              |              |              |
|----------------------------------|-----|--------------|--------------------------|-------------------------|--------------|--------------|--------------------------------|--------------|--------------|--------------|--------------|--------------|
|                                  |     | Clean        | PGD $_{\varepsilon=0.5}$ | AA $_{\varepsilon=0.5}$ | Clean        | Clean        | 1                              | 2            | 3            | 4            | 5            | Avg          |
| ICNN-DRO (fixed $\lambda = 30$ ) | 50  | 91.20        | <u>59.83</u>             | <u>58.81</u>            | <u>82.57</u> | 79.98        | <u>89.28</u>                   | <u>86.85</u> | <u>84.10</u> | <u>80.00</u> | 74.06        | <u>82.85</u> |
| ICNN-DRO (fixed $\lambda = 30$ ) | 20  | <u>91.41</u> | 58.11                    | 57.71                   | 82.56        | <b>80.12</b> | 89.20                          | 86.79        | 84.01        | 79.92        | 73.95        | 82.77        |
| ICNN-DRO + $\lambda$ -annealing  | 20  | <b>91.48</b> | <b>59.94</b>             | <b>58.92</b>            | <b>82.60</b> | <u>80.10</u> | <b>89.31</b>                   | <b>86.93</b> | <b>84.15</b> | <b>80.04</b> | <b>74.12</b> | <b>82.91</b> |

because our adversary is a single global map  $T_\omega = \nabla_z \psi_\omega$  defined on all of  $\mathbb{R}^m$ , *i.e.*, what carries across stages is the entire transport, not a set of transductive particles.

This scheduling serves two main purposes. First, it creates an adversarial curriculum for the classifier. The adversary begins near the ERM baseline at  $\lambda_1$  and steadily strengthens as  $\lambda$  decreases. Second, it stabilizes the ICNN optimization by allowing the map’s displacement to grow incrementally. Because the worst-case displacement scales as  $1/\lambda$ , warm-starting avoids the unstable jumps of a cold solve, instead expanding the transport map by a modest factor at each stage. Note that  $T_\omega$  is always the gradient of a convex potential, the refined map remains cyclically monotone throughout the entire schedule.

**Setup.** We evaluate this schedule using the CIFAR-10 image-space pipeline from Table 1 (§ 6.2). We retain the exact ICNN architecture used in the main text (a  $[64, 64, 64, 64]$  backbone, readout-only quadratic injection, and  $r_{\text{out}} = 128$ ). Consequently, this run differs from the fixed-penalty baseline solely in its penalty schedule. The penalty decreases along  $\lambda \in \{60, 50, 40, 35, 30\}$  over  $S = 5$  stages. Each stage lasts for  $E = 10$  adversarial epochs (50 in total, preceded by a 2-epoch adversary-only warmup at  $\lambda_1 = 60$ ). Because the schedule terminates at the pipeline’s standard operating point ( $\lambda_S = 30$ ), it steadily transitions from a mild, near-ERM adversary down to our standard robust formulation. In each stage, we perform  $K = 20$  BB+Armijo inner ascent steps per batch. Crucially, neither the classifier  $\theta$  nor the ICNN weights  $\omega$  are reset at stage boundaries, allowing the transport map to be continuously refined across penalty levels rather than re-solved from scratch.

**Results.** As shown in Table 7, penalty annealing attains the best performance across nearly every metric. Compared to the matched fixed- $\lambda$  baseline at  $K = 20$ , the curriculum improves every reported metric except CIFAR-10.2, where it is lower by 0.02 percentage points. Remarkably, it even yields a higher PGD accuracy than the more computationally expensive fixed- $\lambda$  configuration (59.94% vs. 59.83%), despite using  $2.5\times$  fewer inner ascent steps ( $K = 20$  vs. 50). This demonstrates that the curriculum discovers a stronger adversary at a fraction of the inner-loop cost. Crucially, this avoids the typical robustness-accuracy trade-off: the annealed model simultaneously achieves the highest clean CIFAR-10 accuracy (91.48%) and superior performance on the natural shifts of CIFAR-10.1, CIFAR-10.2, and CIFAR-10-C. These gains are consistent with the warm-started curriculum successfully stabilizing the classifier  $\theta$  before the adversary reaches full strength.