Journal of Machine Learning Research 22 (2021) 1-67 Submitted 9/20; Revised 8/21; Published 11/21

# Adversarial Monte Carlo Meta-Learning of Optimal Prediction Procedures

**Alex Luedtke** ALUEDTKE@UW.EDU
**Incheoul Chung** IC247@UW.EDU
*Department of Statistics*
*University of Washington*
*Seattle, WA 98195-4322, USA*

**Oleg Sofrygin** OLEG.SOFRYGIN@KP.ORG
*Division of Research*
*Kaiser Permanente Northern California*
*Oakland, CA 94612-2304, USA*

**Editor:** Philipp Hennig

## Abstract

We frame the meta-learning of prediction procedures as a search for an optimal strategy in a two-player game. In this game, Nature selects a prior over distributions that generate labeled data consisting of features and an associated outcome, and the Predictor observes data sampled from a distribution drawn from this prior. The Predictor's objective is to learn a function that maps from a new feature to an estimate of the associated outcome. We establish that, under reasonable conditions, the Predictor has an optimal strategy that is equivariant to shifts and rescalings of the outcome and is invariant to permutations of the observations and to shifts, rescalings, and permutations of the features. We introduce a neural network architecture that satisfies these properties. The proposed strategy performs favorably compared to standard practice in both parametric and nonparametric experiments.

## 1. Introduction

### 1.1 Problem Formulation

Consider a dataset consisting of $n \geq 2$ observations $(X_1, Y_1), \ldots, (X_n, Y_n)$ drawn independently from a distribution $P$ belonging to some known model $\mathcal{P}$, where each $X_i$ is a continuously distributed feature with support contained in $\mathcal{X} := \mathbb{R}^p$ and each $Y_i$ is an outcome with support contained in $\mathcal{Y} := \mathbb{R}$. This dataset can be written as $\boldsymbol{D} := (\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ is the $n \times p$ matrix for which row $i$ contains $X_i$ and $\boldsymbol{Y}$ is the $n$-dimensional vector for which entry $i$ contains $Y_i$. The support of $\boldsymbol{D}$ is contained in $\mathcal{D} := \mathcal{X}^n \times \mathcal{Y}^n$. The objective is to develop an estimator of the regression function $\mu_P$ that maps from $x_0$ to $\mathrm{E}_P[Y|X = x_0]$. An estimator $T$ belongs to the collection $\mathcal{T}$ of operators that take

The authors thank Devin Didericksen for help in the early stages of this project. Generous support was provided by Amazon through an AWS Machine Learning Research Award and the NIH under award number DP2-LM013340. The content is solely the responsibility of the authors and does not necessarily represent the official views of Amazon or the NIH.

as input a dataset $\boldsymbol{d} := (\boldsymbol{x}, \boldsymbol{y})$ and output a prediction function $T(\boldsymbol{d}) : \mathcal{X} \to \mathbb{R}$, where here and throughout we use $\boldsymbol{d} = (\boldsymbol{x}, \boldsymbol{y})$ to denote a possible realization of the random variable $\boldsymbol{D} = (\boldsymbol{X}, \boldsymbol{Y})$. Examples of estimators include the generalized linear models (Nelder and Wedderburn, 1972), random forests (Breiman, 2001), and gradient boosting machines (Friedman, 2001). We will also refer to estimators as prediction procedures. We focus on the case that the performance of an estimator is quantified via the standardized mean-squared error (MSE), namely

$$R(T, P) := \mathrm{E}_P \left[ \int \frac{[T(\boldsymbol{D})(x_0) - \mu_P(x_0)]^2}{\sigma_P^2} dP_X(x_0) \right], \tag{1}$$

where the expectation above is over the draw of $\boldsymbol{D}$ under sampling from $P$, $P_X$ denotes the marginal distribution of $X$ implied by $P$, and $\sigma_P^2$ denotes the variance of the error $\epsilon_P := Y - \mu_P(X)$ when $(X, Y) \sim P$. Note that $\epsilon_P$ may be heteroscedastic. Throughout we assume that, for all $P \in \mathcal{P}$, $\mathrm{E}_P[Y^2] < \infty$ and $\epsilon_P$ is a continuous random variable. Note that the continuity of $\epsilon_P$ implies that $Y$ is continuous and that $\sigma_P^2 > 0$.

In practice, the distribution $P$ is not known, and therefore the risk $R(T, P)$ of a given estimator $T$ is also not known. We now describe three existing criteria for judging the performance of $T$ that do not rely on knowledge of $P$. The first criterion is the maximal risk $\sup_{P \in \mathcal{P}} R(T, P)$. If $T$ minimizes the maximal risk over $\mathcal{T}$, then $T$ is referred to as a minimax estimator (Wald, 1945). Minimax estimators optimize for the worst-case scenario wherein the distribution $P$ is chosen adversarially in such a way that the selected estimator performs as poorly as possible. The second criterion is Bayesian in nature, namely the average of the risk $R(T, P)$ over draws of $P$ from a given prior $\Pi$ on $\mathcal{P}$. Specifically, this Bayes risk is defined as $r(T, \Pi) := \mathrm{E}_\Pi[R(T, P)]$ (Robert, 2007). A $\Pi$-Bayes estimator optimally incorporates the prior beliefs encoded in $\Pi$ with respect to the Bayes risk $r(\cdot, \Pi)$ — more concretely, an estimator $T$ is referred to as a $\Pi$-Bayes estimator if it minimizes the Bayes risk over $\mathcal{T}$. Though the optimality property of Bayes estimators is useful in settings where $\Pi$ only encodes substantive prior knowledge, its utility is less clear otherwise. Indeed, as the function $r(\cdot, \Pi)$ generally depends on the choice of $\Pi$, it is possible that a $\Pi$-Bayes estimator $T$ is meaningfully suboptimal with respect to some other prior $\Pi'$, that is, that $r(T, \Pi) \gg \inf_{T'} r(T', \Pi')$. This phenomenon can be especially common when the sample size is small or the model is nonparametric. In fact, in the latter case, Bayes estimators against particular priors $\Pi$ can easily be inconsistent even though consistent frequentist estimators are available (Ghosal and Van der Vaart, 2017) — for such priors, Bayes estimators perform poorly even when the sample size is large. Therefore, in settings where there is no substantive reason to favor a particular choice of $\Pi$, it is sensible to seek another approach for judging the performance of $T$. A natural criterion is the worst-case Bayes risk of $T$ over some user-specified collection $\Gamma$ of priors, namely $\sup_{\Pi \in \Gamma} r(T, \Pi)$. This criterion is referred to as the $\Gamma$-maximal Bayes risk of $T$. The collection $\Gamma$ may be restricted to contain all priors that are compatible with available prior information, such as knowledge about the smoothness of a regression function, while being left large enough to acknowledge that prior knowledge may be too vague to encode within a single prior distribution (see Section 3.6 of Robert, 2007, for more possible forms of vague prior information). If $T$ is a minimizer of the $\Gamma$-maximal Bayes risk, then $T$ is referred to as a $\Gamma$-minimax estimator (Berger, 1985). Such estimators can be viewed as the optimal strategy in a sequential two-player game between a Predictor

and Nature, where the Predictor selects an estimator and Nature then selects a prior in $\Gamma$ at which the Predictor's chosen estimator performs as poorly as possible in terms of Bayes risk. Notably, in settings where $\Gamma$ contains all distributions with support in $\mathcal{P}$, the $\Gamma$-maximal Bayes risk is equivalent to the maximal risk. Consequently, in this special case, an estimator is $\Gamma$-minimax if and only if it is minimax. In settings where $\Gamma = \{\Pi\}$, an estimator is $\Gamma$-minimax if and only if it is $\Pi$-Bayes. Therefore, by allowing for a choice of $\Gamma$ as large as the unrestricted set of all possible distributions or as small as a singleton set, $\Gamma$-minimaxity provides a means of interpolating between the minimax and Bayesian criteria.

Though $\Gamma$-minimax estimators represent an appealing compromise between the Bayesian and minimax paradigms, they have seen limited use in practice because they are rarely available in closed form. In this work, we aim to overcome this challenge in the context of prediction by providing an iterative strategy for learning $\Gamma$-minimax prediction procedures. Due to the potentially high computational cost of this iterative scheme, a key focus of our work involves identifying conditions under which we can identify a small subclass of $\mathcal{T}$ that still contains a $\Gamma$-minimax estimator. This then makes it possible to optimize over this subclass, which we show in our experiments can dramatically improve the performance of our iterative scheme given a fixed computational budget.

Hereafter we refer to $\Gamma$-minimax estimators as 'optimal', where it is to be understood that this notion of optimality relies on the choice of $\Gamma$.

### 1.2 Overview of Our Strategy and Our Contributions

Our strategy builds on two key results, each of which will be established later in this work. First, under conditions on $\mathcal{T}$ and $\Gamma$, there exists a $\Gamma$-minimax estimator in the subclass $\mathcal{T}_e \subset \mathcal{T}$ of estimators that are equivariant to shifts and rescalings of the outcome and are invariant to permutations of the observations and to shifts, rescalings, and permutations of the features. Second, under further conditions, there is an equilibrium point $(T^\star, \Pi^\star) \in \mathcal{T}_e \times \Gamma$ such that

$$\sup_{\Pi \in \Gamma} r(T^\star, \Pi) = r(T^\star, \Pi^\star) = \inf_{T \in \mathcal{T}_e} r(T, \Pi^\star). \tag{2}$$

Upper bounding the right-hand side by $\sup_{\Pi \in \Gamma} \inf_{T \in \mathcal{T}_e} r(T, \Pi)$ and applying the max-min inequality shows that $T^\star$ is $\Gamma$-minimax. To find an equilibrium numerically, we propose to use adversarial Monte Carlo meta-learning (AMC) (Luedtke et al., 2020) to iteratively update an estimator in $\mathcal{T}_e$ and a prior in $\Gamma$. AMC is a form of stochastic gradient descent ascent (e.g., Lin et al., 2019) that can be used to learn optimal statistical procedures in general decision problems.

We make the following contributions:

- In Section 2, we characterize several equivariance properties of optimal estimators for a wide range of $(\mathcal{T}, \Gamma)$.
- In Section 3, we present a general framework for adversarially learning optimal prediction procedures.
- In Section 4, we present a novel neural network architecture for parameterizing estimators that satisfy the equivariance properties established in Section 2.

- In Section 5, we apply our algorithm in two settings and learn estimators that outperform standard approaches in numerical experiments. In Section 6, we also evaluate the performance of these learned estimators in data experiments.

All proofs for the results in the above sections can be found in Section 7. Section 8 describes possible extensions and provides concluding remarks.

To maximize the accessibility of our main theoretical results, we do not use group theoretic notation when presenting them in Sections 2 through 4. However, when proving these results, we will heavily rely on tools from group theory; consequently, we adopt this notation in Section 7.

### 1.3 Related Works

The approach proposed in this work is a form of meta-learning (Schmidhuber, 1987; Thrun and Pratt, 1998; Vilalta and Drissi, 2002), where here each task is a regression problem. Most existing works in this area pursue a task-distribution strategy to meta-learning (Hospedales et al., 2020), where the objective is to minimize the average loss (risk) across draws of tasks from some specified distribution. As we will now show, the objective function employed in such strategies in fact corresponds to a Bayes risk. In regression problems, each task is a tuple containing a dataset $\boldsymbol{d}$ and a task-dependent loss $\mathcal{L} : \mathcal{D} \times \mathcal{T} \to \mathbb{R}$. For a given prior $\Pi$, a draw from the task distribution can be obtained by first sampling $P \sim \Pi$, next sampling a dataset $\boldsymbol{D}$ of independent observations from $P$, drawing an evaluation point $X_0 \sim P_X$, and finally defining the loss by $\mathcal{L}(\boldsymbol{d}, T) = [T(\boldsymbol{d})(X_0) - \mu_P(X_0)]^2/\sigma_P^2$ or some related loss, such as a squared error loss that does not standardize by $\sigma_P^2$. The objective function is then equal to $T \mapsto E[\mathcal{L}(\boldsymbol{D}, T)]$, where the expectation is over the draw of $(\boldsymbol{D}, \mathcal{L})$ from the task distribution. This objective function is exactly equal to the Bayes risk function $T \mapsto r(T, \Pi)$. Hence, existing meta-learning approaches for regression problems whose objective functions take this form can be viewed as optimizing a Bayes risk.

We now review existing meta-learning strategies, starting with those that parameterize $\mathcal{T}$ as a neural network class. Hochreiter et al. (2001) advocated parameterizing $\mathcal{T}$ as a collection of long short-term (LSTM) networks (Hochreiter and Schmidhuber, 1997). More recent works have advocated using memory-augmented neural networks (Santoro et al., 2016) or conditional neural processes (CNPs) (Garnelo et al., 2018) rather than LSTMs in meta-learning tasks. There have also been other works on the meta-learning of supervised learning procedures that are parameterized as neural networks (Bosc, 2016; Vinyals et al., 2016; Ravi and Larochelle, 2017). Compared to these works, we *adversarially* learn a prior $\Pi$ from a collection $\Gamma$ of priors, and we also formally characterize equivariance properties that will be satisfied by any optimal prediction procedure in a wide variety of problems. This characterization leads us to develop a neural network architecture designed for the prediction settings that we consider.

Model-agnostic meta-learning (MAML) is another popular meta-learning approach (Finn et al., 2017). In our setting, MAML aims to initialize the weights of a regression function estimate (parameterized as a neural network, for example) in such a way that, on any new task, only a limited number of gradient updates are needed. More recent approaches leverage the fact that, in certain settings, the initial estimate can instead be updated using a convex optimization algorithm (Bertinetto et al., 2018; Lee et al., 2019). To run any of

these approaches, a prespecified prior over tasks is required. In our setting, these tasks take the form of data-generating distributions $P$. In contrast to MAML and related approaches, our proposal adversarially selects a prior from $\Gamma$.

Two recent works (Yin et al., 2018; Goldblum et al., 2019) developed meta-learning procedures that are trained under a different adversarial regime than that studied in the current work, namely under adversarial manipulation of one or both of the dataset $\boldsymbol{d}$ and evaluation point $x_0$ (Dalvi et al., 2004). This adversarial framework appears to be most useful when there truly is a malicious agent that aims to contaminate the data, which is not the case that we consider. In contrast, in our setting, the adversarial nature of our framework allows us to ensure that our procedure will perform well regardless of the true value of $P$, while also taking into account prior knowledge that we may have.

Our approach is also related to existing works in the statistics and econometrics literatures on the numerical learning of minimax and $\Gamma$-minimax statistical decision rules. In finite-dimensional models, early works showed that it is possible to numerically learn minimax rules (Nelson, 1966; Kempthorne, 1987) and, in settings where $\Gamma$ consists of all priors that satisfy a finite number of generalized moment conditions, $\Gamma$-minimax rules (Noubiap and Seidel, 2001). Other works have studied the $\Gamma$-minimax case where $\Gamma$ consists of priors that only place mass on a pre-specified finite set of distributions in $\mathcal{P}$, both for general decision problems (Chamberlain, 2000) and for constructing confidence intervals (Schafer and Stark, 2009). Defining $\Gamma$ in this fashion modifies the statistical model $\mathcal{P}$ to only consist of finitely many distributions, which can be restrictive. A recent work introduced a new approach, termed AMC, for learning minimax procedures for general models $\mathcal{P}$ (Luedtke et al., 2020). In contrast to earlier works, AMC does not require the explicit computation of a Bayes estimator under any given prior, thereby improving the feasibility of this approach in moderate-to-high dimensional models. In their experiments, Luedtke et al. (2020) used neural network classes to define the sets of allowable statistical procedures. Unlike the current work, none of the aforementioned studies identified or leveraged the equivariance properties that characterize optimal procedures. As we will see in our experiments, leveraging these properties can dramatically improve performance.

### 1.4 Notation

We now introduce the notation and conventions that we use. For a function $f : \mathcal{P} \to \mathcal{P}$, we let $\Pi \circ f^{-1}$ denote the pushforward measure that is defined as the distribution of $f(P)$ when $P \sim \Pi$. For any dataset $\boldsymbol{d} = (\boldsymbol{x}, \boldsymbol{y})$ and mapping $f$ with domain $\mathcal{D}$, we let $f(\boldsymbol{x}, \boldsymbol{y}) := f(\boldsymbol{d})$. We take all vectors to be column vectors when they are involved in matrix operations. We write $\odot$ to mean the entrywise product and $a^{\odot 2}$ to mean $a \odot a$. For an $m_1 \times m_2$ matrix $a$, we let $a_{i*}$ denote the $i^{\text{th}}$ row, $a_{*j}$ denote the $j^{\text{th}}$ column, $\bar{a} := \frac{1}{m_1} \sum_{i=1}^{m_1} a_{i*}$, and $s(a)^2 := \frac{1}{m_1} \sum_{i=1}^{m_1} (a_{i*} - \bar{a})^{\odot 2}$. When we standardize a vector $a$ as $[a - \bar{a}]/s(a)$, we always use the convention that $0/0 = 0$. We write $[a \,|\, b]$ to denote the column concatenation of two matrices. For an $m_1 \times m_2 \times m_3$ array $a$, we let $a_{i**}$ denote the $m_2 \times m_3$ matrix with entry $(j, k)$ equal to $a_{ijk}$, $a_{i*k}$ denote the $m_2$-dimensional vector with entry $j$ equal to $a_{ijk}$, etc. For $a \in \mathbb{R}$ and $b \in \mathbb{R}^k$, we write $a + b$ to mean $a\mathbf{1}_k + b$.

## 2. Characterization of Optimal Procedures

### 2.1 Optimality of Equivariant Estimators

We start by presenting conditions that we impose on the collection of priors $\Gamma$. Let $\mathcal{A}$ denote the collection of all $n \times n$ permutation matrices, and let $\mathcal{B}$ denote the collection of all $p \times p$ permutation matrices. We suppose that $\Gamma$ is preserved under the following transformations:

P1. *Permutations of features:* $\Pi \in \Gamma$ and $B \in \mathcal{B}$ implies that $\Pi \circ f_1^{-1} \in \Gamma$, where $f_1(P)$ is the distribution of $(BX, Y)$ when $(X, Y) \sim P$.

P2. *Shifts and rescalings of features:* $\Pi \in \Gamma$, $a \in \mathbb{R}^p$, and $b \in (\mathbb{R}^+)^p$ implies that $\Pi \circ f_2^{-1} \in \Gamma$, where $f_2(P)$ is the distribution of $(a + b \odot X, Y)$ when $(X, Y) \sim P$.

P3. *Shift and rescaling of outcome:* $\Pi \in \Gamma$ and $\tilde{a} \in \mathbb{R}$ and $\tilde{b} > 0$ implies that $\Pi \circ f_3^{-1} \in \Gamma$, where $f_3(P)$ is the distribution of $(X, \tilde{a} + \tilde{b}Y)$ when $(X, Y) \sim P$.

The above conditions implicitly encode that $f_1(P)$, $f_2(P)$, and $f_3(P)$ all belong to $\mathcal{P}$ whenever $P \in \mathcal{P}$. Section 7.1 provides an alternative characterization of P1, P2, and P3 in terms of the preservation of $\Gamma$ under a certain group action.

Condition P1 ensures that permuting the features during preprocessing will not impact the collection of priors considered. This condition is reasonable in settings where there is only a limited prior understanding of each individual feature under consideration or, if such information is available, there is little anticipated benefit from including it in the analysis. Most commonly used supervised machine learning algorithms similarly do not incorporate specific prior information about individual features, and are instead designed to work across a variety of settings — this is the case, for example, for commonly used implementations of random forests, extreme gradient boosting, and penalized linear models (Pedregosa et al., 2011; Chen and Guestrin, 2016). It is worth noting, however, that P1 still allows information on the features to be incorporated should it be available — for example, prior beliefs on the multivariate feature distribution, such as the number of modes that it has, or the regression function, such as its level of sparsity, can be imposed in the collection $\Gamma$ of prior distributions. Conditions P2 and P3 are imposed to ensure that the $\Gamma$-maximal risk criterion captures the possibility that the data may be preprocessed via affine transformations, such as prestandardization or a change of the unit of measure (Fahrenheit to Celsius, say), before being supplied to the prediction algorithm. By having $\Gamma$ be large enough to ensure that P2 and P3 are satisfied, the $\Gamma$-minimax risk reflects performance in an adversarial setting wherein affine transformations are applied to the features and outcome in such a way as to make the (Bayes) risk as large as possible for a given prediction algorithm. Because it minimizes this adversarial criterion, a $\Gamma$-minimax estimator should be robust to such adversarial transformations, thereby ensuring satisfactory performance regardless of the chosen unit of measure or prestandardization scheme.

We also assume that the signal-to-noise ratio (SNR) is finite — this condition is important in light of the fact that the MSE risk that we consider standardizes by $\sigma_P^2$.

P4. *Finite SNR:* $\sup_{P \in \mathcal{P}} \mathrm{var}_P(\mu_P(X))/\sigma_P^2 < \infty$.

We now present conditions that we impose on the class of estimators $\mathcal{T}$. In what follows we let $\mathcal{D}_0 := \{(\boldsymbol{d}, x_0) \in \mathcal{D} \times \mathcal{X} : s(\boldsymbol{y}) \neq 0, s(\boldsymbol{x}) \neq \mathbf{0}_p\}$. For $(\boldsymbol{d}, x_0) \in \mathcal{D}_0$, we let

$$z(\boldsymbol{d}, x_0) := \left( \frac{\boldsymbol{x} - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}, \frac{\boldsymbol{y} - \bar{\boldsymbol{y}}}{s(\boldsymbol{y})}, \frac{x_0 - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}, \frac{\bar{\boldsymbol{x}}}{s(\boldsymbol{x})}, \frac{\bar{\boldsymbol{y}}}{s(\boldsymbol{y})}, \log s(\boldsymbol{x}), \log s(\boldsymbol{y}) \right),$$

where $\log s(\boldsymbol{x})$ is the vector where log is applied entrywise and where we abuse notation and let $\frac{\boldsymbol{x} - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}$ represent the $n \times p$ matrix for which row $i$ is equal to $[x_i - \bar{\boldsymbol{x}}]/s(\boldsymbol{x})$, and similarly for $\bar{\boldsymbol{x}}/s(\boldsymbol{x})$. We let $\mathcal{Z} := \{z(\boldsymbol{d}, x_0) : (\boldsymbol{d}, x_0) \in \mathcal{D}_0\}$. When it will not cause confusion, we will write $\boldsymbol{z} := z(\boldsymbol{d}, x_0)$. Fix $T \in \mathcal{T}$. Let $S_T : \mathcal{Z} \to \mathbb{R}$ denote the unique function that satisfies

$$T(\boldsymbol{d})(x_0) = \bar{\boldsymbol{y}} + s(\boldsymbol{y}) S_T(\boldsymbol{z}) \quad \text{for all } (\boldsymbol{d}, x_0) \in \mathcal{D}_0. \tag{3}$$

The uniqueness arises because $s(\boldsymbol{y}) \neq 0$ on $\mathcal{D}_0$. Because we have assumed that $X$ and $Y$ are continuous random variables under sampling from any $P \in \mathcal{P}$, it follows that, for all $P \in \mathcal{P}$, the class $\mathcal{S} := \{S_T : T \in \mathcal{T}\}$ uniquely characterizes the functions in $\mathcal{T}$ up to their behavior on subsets of $\mathcal{D} \times \mathcal{X}$ of $P$-probability zero. In what follows, we will impose smoothness constraints on $\mathcal{S}$, which in turn imposes constraints on $\mathcal{T}$. The first three conditions suffice to show that $\mathcal{S}$ is compact in the space $C(\mathcal{Z}, \mathbb{R})$ of continuous $\mathcal{Z} \to \mathbb{R}$ functions equipped with the compact-open topology.

T1. *$\mathcal{S}$ is pointwise bounded:* For all $\boldsymbol{z} \in \mathcal{Z}$, $\sup_{S \in \mathcal{S}} |S(\boldsymbol{z})| < \infty$.

T2. *$\mathcal{S}$ is locally Hölder:* For all compact sets $\mathcal{K} \subset \mathcal{Z}$, there exists an $\alpha \in (0, 1)$ such that

$$\sup_{S \in \mathcal{S}, \boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{K}} \frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|_2^\alpha} < \infty,$$

where $\|\cdot\|_2$ denotes the Euclidean norm. We take the supremum to be zero if $\mathcal{K}$ is a singleton or is empty.

T3. *$\mathcal{S}$ is sequentially closed in the topology of compact convergence:* If $\{S_j\}_{j=1}^\infty$ is a sequence in $\mathcal{S}$ and $S_j \to S$ compactly in the sense that, for all compact $\mathcal{K} \subset \mathcal{Z}$, $\sup_{\boldsymbol{z} \in \mathcal{K}} |S_j(\boldsymbol{z}) - S(\boldsymbol{z})| \to 0$, then $S \in \mathcal{S}$.

The following conditions ensure that $\mathcal{S}$ is invariant to certain preprocessings of the data, in the sense that, for any function $S \in \mathcal{S}$, the function that first preprocesses the data in an appropriate fashion and then applies $S$ to this data is itself in $\mathcal{S}$. When formulating these conditions, we write $z(\boldsymbol{d}, x_0)$ to mean an element of $\mathcal{Z}$. Because $z$ is a bijection between $\mathcal{D}_0$ and $\mathcal{Z}$, it is possible to recover $(\boldsymbol{d}, x_0)$ from $z(\boldsymbol{d}, x_0)$. Below we use this fact to abuse notation and define functions with domain $\mathcal{Z}$ like $z(\boldsymbol{d}, x_0) \mapsto g(\boldsymbol{d}, x_0)$ for functions $g$ with domain $\mathcal{D}_0$, without explicitly introducing notation for the inverse of $z$.

T4. *Permutations:* For all $S \in \mathcal{S}$, $A \in \mathcal{A}$, and $B \in \mathcal{B}$, $z(\boldsymbol{d}, x_0) \mapsto S(z((A\boldsymbol{x}B, A\boldsymbol{y}), B^\top x_0))$ is in $\mathcal{S}$.

T5. *Shifts and rescalings:* For all $S \in \mathcal{S}$, $a \in \mathbb{R}^p$, $b \in (\mathbb{R}^+)^p$, $\tilde{a} \in \mathbb{R}$, and $\tilde{b} > 0$, the function $z(\boldsymbol{d}, x_0) \mapsto S(z((\boldsymbol{x}^{a,b}, \tilde{a} + \tilde{b}\boldsymbol{y}), a + b \odot x_0))$ is in $\mathcal{S}$, where $\boldsymbol{x}^{a,b}$ is the $n \times p$ matrix with row $i$ equal to $a + b \odot \boldsymbol{x}_{i*}$.

In Appendix B, we provide two examples of classes $\mathcal{S}$ that satisfy Conditions T1-T5. One of these classes is finite-dimensional and the other is infinite-dimensional. The infinite-dimensional class takes a particularly simple form. In particular, for some $c, \alpha > 0$ and some function $F : \mathcal{Z} \to \mathbb{R}^+$ that is invariant to permutations, shifts, and rescalings, we consider the class $\mathcal{S}$ to be the collection of all the collection of all $S : \mathcal{Z} \to \mathbb{R}$ such that $|S(\boldsymbol{z})| \leq F(\boldsymbol{z})$ and $|S(\boldsymbol{z}) - S(\boldsymbol{z}')| \leq c\|\boldsymbol{z} - \boldsymbol{z}'\|_2^{\alpha}$ for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$.

Let $\mathcal{T}_e \subseteq \mathcal{T}$ denote the class of estimators that are equivariant to shifts and rescalings of the outcome and are invariant to permutations of the observations and to shifts, rescalings, and permutations of the features. Specifically, $\mathcal{T}_e$ consists of functions in $\mathcal{T}$ satisfying the following properties for all pairs $(\boldsymbol{d}, x_0)$ of datasets and features in $\mathcal{D}_0$, permutation matrices $A \in \mathcal{A}$ and $B \in \mathcal{B}$, shifts $a \in \mathbb{R}^p$ and $\tilde{a} \in \mathbb{R}$, and rescalings $b \in (\mathbb{R}^+)^p$ and $\tilde{b} > 0$:

$$T(A\boldsymbol{x}B, A\boldsymbol{y})(B^\top x_0) = T(\boldsymbol{d})(x_0), \tag{4}$$

$$T(\boldsymbol{x}^{a,b}, \tilde{a} + \tilde{b}\boldsymbol{y})(a + b \odot x_0) = \tilde{a} + \tilde{b}T(\boldsymbol{d})(x_0), \tag{5}$$

The following result shows that the $\Gamma$-maximal risk is the same over $\mathcal{T}$ and $\mathcal{T}_e \subseteq \mathcal{T}$.

**Theorem 1** *Under P1-P4 and T1-T5,*

$$\inf_{T \in \mathcal{T}} \sup_{\Pi \in \Gamma} r(T, \Pi) = \inf_{T \in \mathcal{T}_e} \sup_{\Pi \in \Gamma} r(T, \Pi).$$

The above does not rule out the possibility that there exists a non-equivariant $\Gamma$-minimax estimator, that is, a $\Gamma$-minimax estimator that belongs to $\mathcal{T} \backslash \mathcal{T}_e$. Rather, when paired with additional conditions that ensure that the infimum over $\mathcal{T}_e$ above is achieved (see Theorem 3), the above implies that $\mathcal{T}_e$ contains at least one $\Gamma$-minimax estimator.

Theorem 1 is a variant of the Hunt-Stein theorem (Hunt and Stein, 1946). Our proof, which draws inspiration from Le Cam (2012), consists in showing that our prediction problem is invariant to the action of an amenable group and subsequently applying Day's fixed-point theorem (Day, 1961) to show that, for all $T \in \mathcal{T}$, the collection of $T'$ for which $\sup_{\Pi \in \Gamma} r(T', \Pi) \leq \sup_{\Pi \in \Gamma} r(T, \Pi)$ has nonempty intersection with $\mathcal{T}_e$.

This theorem has a natural analogy to the translation equivariance that is enjoyed by convolutional neural networks in object detection problems, where the goal is to classify and draw a bounding box around objects in an image (Russakovsky et al., 2015). To simplify the discussion, here we focus on the special case where there is only one object class of interest (e.g., humans), so that the goal is simply to draw a bounding box around each object. In object detection settings, a key insight is that an object's class does not change even if its position is shifted. Given this insight, it seems reasonable to expect that any sufficiently rich collection of candidate detectors will be such that, given any object detector $V$, the collection will contain a translation equivariant detector with equal or superior performance to that of $V$. For this to be true, certain requirements are also generally needed of the loss function used to measure performance. In particular, the error accrued by incorrectly bounding or failing to bound an object should not depend on the position of that object in the image — this condition is satisfied by many loss functions that are commonly used in this setting. In our setting, conditions P1-P3, which say that a prior still belongs to $\Gamma$ even after certain transformations are applied to the distributions drawn from that prior, are the analogues of the translation invariance property of an object's class ("a human remains a

human if they are shifted to the left, and the pushforward of a prior in $\Gamma$ remains in $\Gamma$ even if features and outcomes are permuted, shifted, or rescaled"); conditions T4 and T5 are the analogues of the requirement that the collection of detectors be sufficiently rich; and the fact that the standardized squared error $[T(\boldsymbol{d})(x_0) - \mu_P(x_0)]^2/\sigma_P^2$ does not depend on the particular ordering of the features or the centering or scaling of the features or outcomes is analogous to the translation invariance of the loss functions used in object detection.

### 2.2 Focusing Only on Distributions with Standardized Predictors and Outcome

Theorem 1 suggests restricting attention to estimators in $\mathcal{T}_e$ when trying to learn a $\Gamma$-minimax estimator. We now show that, once this restriction has been made, it also suffices to restrict attention to a smaller collection of priors $\Gamma_1$ when identifying a least favorable prior. In fact, we show something slightly stronger, namely that the restriction to $\Gamma_1$ can be made even if optimal estimators are sought over the richer class $\widetilde{\mathcal{T}}_e \supseteq \mathcal{T}_e$ of estimators that satisfy the equivariance property (5) but do not necessarily satisfy (4).

We now define $\Gamma_1$. Let $h(P)$ denote the distribution of

$$\left( \left( \frac{X_j - \mathrm{E}_P[X_j]}{\mathrm{var}_P(X_j)^{1/2}} \right)_{j=1}^{p}, \frac{Y - \mathrm{E}_P[Y]}{\sigma_P} \right)$$

when $(X, Y) \sim P$. Note that here, and here only, we have written $X_j$ to denote the $j^{\text{th}}$ feature rather than the $j^{\text{th}}$ observation. Also let $\Gamma_1 := \{\Pi \circ h^{-1} : \Pi \in \Gamma\}$, which is a collection of priors on $\mathcal{P}_1 := \{h(P) : P \in \mathcal{P}\}$.

**Theorem 2** *If P2 and P3 hold and all $T \in \mathcal{T}$ satisfy* (5)*, then $T^\star$ is $\Gamma$-minimax if and only if it is $\Gamma_1$-minimax.*

We conclude by noting that, under P2 and P3, $\mathcal{P}_1$ consists precisely of those $P \in \mathcal{P}$ that satisfy:

$$\mathrm{E}_P[X] = \mathbf{0}_p, \;\; \mathrm{E}_P[X^{\odot 2}] = \mathbf{1}_p, \;\; \mathrm{E}_P[Y] = 0, \;\; \sigma_P^2 = 1. \tag{6}$$

### 2.3 Existence of an Equilibrium Point

We also make the following additional assumption on $\mathcal{S}$.

T6. *$\mathcal{S}$ is convex:* $S_1, S_2 \in \mathcal{S}$ and $\delta \in (0, 1)$ implies that $\boldsymbol{z} \mapsto \delta S_1(\boldsymbol{z}) + (1 - \delta) S_2(\boldsymbol{z})$ is in $\mathcal{S}$.

The two examples in Appendix B also satisfy T6.

We also impose the following condition on the size of the collection of distributions $\mathcal{P}_1$ and the collection of priors $\Gamma_1$, which in turn imposes restrictions on $\mathcal{P}$ and $\Gamma$.

P5. There exists a metric $\rho$ on $\mathcal{P}_1$ such that (i) $(\mathcal{P}_1, \rho)$ is a complete separable metric space, (ii) $\Gamma_1$ is tight in the sense that, for all $\varepsilon > 0$, there exists a compact set $\mathcal{K}$ in $(\mathcal{P}_1, \rho)$ such that $\Pi(\mathcal{K}) \geq 1 - \varepsilon$ for all $\Pi \in \Gamma_1$, and (iii) for all $T \in \mathcal{T}_e$, $P \mapsto R(T, P)$ is upper semi-continuous and bounded from above on $(\mathcal{P}_1, \rho)$.

In Appendix C, we give examples of parametric and nonparametric settings where P5 is applicable.

So far, the only conditions that we have required on the $\sigma$-algebra $\mathscr{A}$ of $\mathcal{P}$ are that $h$ and $R(T,\cdot)$, $T \in \mathcal{T}$, are measurable. In this subsection, and in this subsection only, we add the assumptions that P5 holds and that $\mathscr{A}$ is such that $\{A \cap \mathcal{P}_1 : A \in \mathscr{A}\}$ equals $\mathscr{B}_1$, where $\mathscr{B}_1$ is the collection of Borel sets on $(\mathcal{P}_1, \rho)$.

We will also assume the following two conditions on $\Gamma_1$.

P6. $\Gamma_1$ *is closed in the topology of weak convergence:* if $\{\Pi_j\}_{j=1}^{\infty}$ is a sequence in $\Gamma_1$ that converges weakly to $\Pi$, then $\Pi \in \Gamma_1$.

P7. $\Gamma_1$ *is convex:* for all $\Pi_1, \Pi_2 \in \Gamma$ and $\alpha \in (0,1)$, the mixture distribution $\alpha\Pi_1 + (1-\alpha)\Pi_2$ is in $\Gamma$.

Under Conditions P5 and P6, Prokhorov's theorem (Billingsley, 1999) can be used to establish that $\Gamma_1$ is compact in the topology of weak convergence. This compactness will be useful for proving the following result, which shows that there is an equilibrium point under our conditions.

**Theorem 3** *If T1-T3, T6, and P2-P7 hold, then there exists* $T^\star \in \mathcal{T}_e$ *and* $\Pi^\star \in \Gamma_1$ *such that, for all* $T \in \mathcal{T}_e$ *and* $\Pi \in \Gamma_1$*, it is true that* $r(T^\star, \Pi) \leq r(T^\star, \Pi^\star) \leq r(T, \Pi^\star)$.

Combining the above with Lemma 14 in Section 7.2.3 establishes (2), that is, that the conclusion of Theorem 3 remains valid if $\Pi$ varies over $\Gamma$ rather than over $\Gamma_1$.

## 3. AMC Meta-Learning Algorithm

We now present an AMC meta-learning strategy for obtaining a $\Gamma$-minimax estimator within some class $\mathcal{T}$. Here we suppose that $\mathcal{T} = \{T_t : t \in \tau\}$, where each $T_t$ is an estimator indexed by a finite-dimensional parameter $t$ that belongs to some set $\tau$. We note that this framework encapsulates: model-based approaches (e.g., Hochreiter et al., 2001), where $T_t$ can be evaluated by a single pass of $(\boldsymbol{d}, x_0)$ through a neural network with weights $t$; optimization-based approaches, where $t$ are the initial weights of some estimate that are subsequently optimized based on $\boldsymbol{d}$ (e.g., Finn et al., 2017); and metric-based approaches, where $t$ indexes a measure of similarity $\alpha_t$ that is used to obtain an estimate of the form $\sum_{i=1}^{n} \alpha_t(x_i, x_0) y_i$ (e.g., Vinyals et al., 2016).

We suppose that all estimators in $\mathcal{T}$ satisfy the equivariance property (5), which can be arranged by prestandardizing the outcome and features and then poststandardizing the final prediction — see Algorithm 2 for an example. Since all $T \in \mathcal{T}$ satisfy (5), Theorem 2 shows that it suffices to consider a collection $\Gamma_1$ of priors with support on $\mathcal{P}_1$, that is, so that, for all $\Pi \in \Gamma_1$, $P \sim \Pi$ satisfies (6) almost surely. To ensure that the priors are easy to sample from, we parameterize them via generator functions $G_g$ (Goodfellow et al., 2014) that are indexed by a finite-dimensional $g$ that belongs to some set $\gamma$. Each $G_g$ takes as input a source of noise $U$ drawn from a user-specified distribution $\nu_u$ and outputs the parameters indexing a distribution in $\mathcal{P}$ (Luedtke et al., 2020). Though this form of sampling limits to parametric families $\mathcal{P}$, the number of parameters indexing this family may be much larger than the sample size $n$, which can, for all practical purposes, lead to a nonparametric estimation

problem. For each $g$, we let $\Pi_g$ denote the distribution of $G_g(U)$ when $U \sim \nu_u$. We then let $\Gamma_1 = \{\Pi_g : g \in \gamma\}$. It is worth noting that classes $\Gamma_1$ that are defined in this way will not generally satisfy the conditions P5-P7 used in Theorem 3. To iteratively improve the performance of the prior, we require the ability to differentiate realized datasets through the parameters indexing the prior. To do this, we assume that, for each $P \in \mathcal{P}$, the user has access to a generator function $H_P : \mathcal{V} \to \mathbb{R}$ such that $H_P(V)$ has the same distribution as $(X, Y) \sim P$ when noise $V$ is drawn from a user-specified distribution $\nu_v$. We suppose that, for all realizations of the noise $u$ in the support of $\nu_u$ and $v$ in the support of $\nu_v$, the function $g \mapsto H_{G_g(u)}(v)$ is differentiable at each parameter value $g$ indexing the prior.

**Algorithm 1** Adversarially learn an estimator.

1: **Initialize** estimator $T_t$, generator $G_g$, step sizes $\eta_1, \eta_2$.
2: **for** $K$ iterations **do**
3:  **for** $j = 1, 2$ **do**
4:  Independently draw $U \sim \nu_u$ and $V_0, \ldots, V_p \overset{iid}{\sim} \nu_v$.
5:  Let $P = G_g(U)$.
6:  Let $(X_i, Y_i) = H_P(V_i)$, $i = 0, 1, \ldots, n$.
7:  Let $\boldsymbol{D}$ be the dataset containing $(X_i, Y_i)_{i=1}^n$.
8:  Let Loss $= [T_t(\boldsymbol{D})(X_0) - \mu_P(X_0)]^2$
9:  **if** j=1 **then**
10:  Update estimator: $t = t - \eta_1 \nabla_t$Loss.
11:  ▷ Loss depends on $t$ through $T_t$.
12:  **else**
13:  Update prior: $g = g + \eta_2 \nabla_g$Loss.
14:  ▷ Loss depends on $g$ through the definitions of $P$, $(X_i, Y_i)$, and $\boldsymbol{D}$.
15:  **end if**
16:  **end for**
17: **end for**

The AMC learning strategy is presented in Algorithm 1. The algorithm takes stochastic gradient steps on the parameters indexing an estimator and prior generator to iteratively reduce and increase the Bayes risk, respectively. All gradients in the algorithm can be computed via backpropagation using standard software — in our experiments, we used Pytorch for this purpose (Paszke et al., 2019). When computing $\nabla_g$Loss, the dependence of Loss on $g$ is tracked through the dependence of $P$ on $g$ on line 5, the dependence of $X_0$ and $\boldsymbol{D} = (X_i, Y_i)_{i=1}^n$ on $P$ on lines 6 and 7, and the dependence of Loss on $P$, $X_0$, and $\boldsymbol{D}$ on line 8. We caution that, when the outcome or some of the features are discrete, $\nabla_g$Loss will not generally represent an unbiased estimate of the gradient of $g \mapsto r(T_t, \Pi_g)$, which can cause Algorithm 1 to perform poorly. To handle these cases, the algorithm can be modified to instead obtain an unbiased gradient estimate using the likelihood ratio method (Glynn, 1987).

Though studying the convergence properties of the minimax optimization in Algorithm 1 is not the main focus of this work, we now provide an overview of how results from Lin et al. (2019) can be used to provide some guarantees for this algorithm. When doing so, we focus on the special case where there exists some $\ell < \infty$ such that, for all $g$, $t \mapsto r(T_t, G_g)$ is

differentiable with $\ell$-Lipschitz gradient and, for some finite (but potentially large) collection $\mathcal{P}_D := \{P_1, \ldots, P_D\} \subset \mathcal{P}$, $\Gamma$ is the collection of all mixtures of distributions in $\mathcal{P}_D$. We also suppose that the parameter $g$ indexing the generator $G_g$ takes values on the $D-1$ simplex and that this generator is parameterized in such a way that $\nu_u \circ G_g^{-1}$ has the same distribution as the mixture of distributions in $\mathcal{P}_D$ that places mass $g_j$ on distribution $P_j$, $j = 1, \ldots, D$. In this case, provided the learning rates $\eta_1$ and $\eta_2$ are chosen appropriately, Theorem 4.5 in Lin et al. (2019) gives guarantees on the number of iterations required to return an $\epsilon$-stationary point $T_{t_K}$ (idem, Definition 3.7) within a specified number of iterations — this stationary point is such that there exists a $t'$ near $t_K$ at which the function $t \mapsto \sup_{\Pi \in \Gamma} r(T_t, \Pi)$ has at least one small subgradient (idem, Lemma 3.8, for details). If, also, $t \mapsto T_t(\boldsymbol{d})$ is convex for all $\boldsymbol{d}$, then this also implies that $T_{t_K}$ is nearly $\Gamma$-minimax. If, alternatively, the prior update step in Algorithm 1 (line 13) is replaced by an oracle optimizer such that, at each iteration, $g$ is defined as a true maximizer of the Bayes risk $g \mapsto r(T, \Pi_g)$, then Theorem E.4 of Lin et al. (2019) similarly guarantees that an $\epsilon$-stationary point will be reached within a specified number of iterations.

Alternatives to Algorithm 1 are possible. As one example, the stochastic gradient descent ascent optimization scheme could be replaced by an extragadient method (Korpelevich, 1976), which has been shown to perform well in generative adversarial network settings (Gidel et al., 2018). As another example, the prior distribution could, in principle, be specified via its density rather than as the pushforward distribution $\nu_u \circ G_g^{-1}$ defined by the generator. While this density-based parameterization may make it easier to relate the specified priors to commonly used probability distributions, it may also lead to challenges since sampling from a distribution specified by its density is generally a hard problem that necessitates the use of numerical approaches such as Markov chain Monte Carlo methods (Hastings, 1970; Geman and Geman, 1984). Because the prior is updated at each of the $K$ iterations, it seems that many instances of these numerical sampling schemes would need to be run before the termination of the AMC algorithm. Identifying a means to expedite the convergence of this density-based approach is an interesting area for future work.

## 4. Proposed Class of Estimators

### 4.1 Equivariant Estimator Architecture

Algorithm 2 presents our proposed estimator architecture, which relies on four modules. Each module $k$ can be represented as a function $m_k$ belonging to a collection $\mathcal{M}_k$ of functions mapping from $\mathbb{R}^{a_k}$ to $\mathbb{R}^{b_k}$, where the values of $a_k$ and $b_k$ can be deduced from Algorithm 2. For given data $\boldsymbol{d}$, a prediction at a feature $x_0$ can be obtained by sequentially calling the modules and, between calls, either mean pooling across one of the dimensions of the output or concatenating the evaluation point as a new column in the output matrix.

We let $\mathcal{T}_{\mathcal{M}}$ represent the collection of all prediction procedures described by Algorithm 2, where here $(m_k)_{k=1}^4$ varies over $\prod_{k=1}^4 \mathcal{M}_k$. We now give conditions under which the proposed architecture yields an equivariant estimator.

M1) $m_1(AvB)_{**\ell} = A[m_1(v)_{**\ell}]B$ for all $m_1 \in \mathcal{M}_1$, $A \in \mathcal{A}$, $B \in \mathcal{B}$, $v \in \mathbb{R}^{n \times p \times 2}$, and $\ell \in \{1, \ldots, o_1\}$.

M2) $m_2(Bv) = Bm_2(v)$ for all $m_2 \in \mathcal{M}_2$, $B \in \mathcal{B}$, and $v \in \mathbb{R}^{p \times o_1}$.

**Algorithm 2** Use data $\boldsymbol{d}$ to obtain prediction at $x_0$.

1: **Preprocess:** Let $x_0^0 := \frac{x_0 - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}$ and define $\boldsymbol{d}^0 \in \mathbb{R}^{n \times p \times 2}$ so that $\boldsymbol{d}^0_{i*1} = \frac{x_i - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}$ for all $i = 1, \ldots, n$ and $\boldsymbol{d}^0_{*j2} = \frac{\boldsymbol{y} - \bar{\boldsymbol{y}}}{s(\boldsymbol{y})}$ for all $j = 1, \ldots, p$.
2: **Module 1:** $\boldsymbol{d}^1 := m_1(\boldsymbol{d}^0)$. $\boldsymbol{d}^1 \in \mathbb{R}^{n \times p \times o_1}$
3: **Mean Pool:** $\bar{\boldsymbol{d}}^1 := n^{-1} \sum_{i=1}^n \boldsymbol{d}^1_{i**}$. $\bar{\boldsymbol{d}}^1 \in \mathbb{R}^{p \times o_1}$
4: **Module 2:** $\boldsymbol{d}^2 := m_2(\bar{\boldsymbol{d}}^1)$. $\boldsymbol{d}^2 \in \mathbb{R}^{p \times o_2}$
5: **Augment:** $\tilde{\boldsymbol{d}}^2 := [\boldsymbol{d}^2 \mid x_0^0]$. $\tilde{\boldsymbol{d}}^2 \in \mathbb{R}^{p \times (o_2 + 1)}$
6: **Module 3:** $\boldsymbol{d}^3 := m_3(\tilde{\boldsymbol{d}}^2)$. $\boldsymbol{d}^3 \in \mathbb{R}^{p \times o_3}$
7: **Mean Pool:** $\bar{\boldsymbol{d}}^3 := p^{-1} \sum_{j=1}^p \boldsymbol{d}^3_{j*}$. $\bar{\boldsymbol{d}}^3 \in \mathbb{R}^{o_3}$
8: **Module 4:** $\boldsymbol{d}^4 := m_4(\bar{\boldsymbol{d}}^3)$. $\boldsymbol{d}^4 \in \mathbb{R}$
9: **return** $\bar{\boldsymbol{y}} + s(\boldsymbol{y}) \boldsymbol{d}^4$.

M3) $m_3(Bv) = Bm_3(v)$ for all $m_3 \in \mathcal{M}_3$, $B \in \mathcal{B}$, and $v \in \mathbb{R}^{p \times o_2}$.

**Theorem 4** *If M1-M3, then all $T \in \mathcal{T}_{\mathcal{M}}$ satisfy* (4) *and* (5).

### 4.2 Neural Network Parameterization

In our experiments, we choose the four module classes $\mathcal{M}_k$, $k = 1, 2, 3, 4$, indexing our estimator architecture to be collections of neural networks. For each $k$, we let $\mathcal{M}_k$ contain the neural networks consisting of $h_k$ hidden layers of widths $w_k^1, w_k^2, \ldots, w_k^{h_k}$, where the types of layers used depends on the module $k$. When $k = 1$, multi-input-output channel equivariant layers as defined in Hartford et al. (2018) are used. In particular, for $j = 1, \ldots, h_1 + 1$, we let $\mathcal{L}_1^j$ denote the collection of all such layers that map from $\mathbb{R}^{n \times p \times w_1^{j-1}}$ to $\mathbb{R}^{n \times p \times w_1^j}$, where we let $w_1^0 = 2$ and $w_1^{h_1+1} = o_1$. For each $j$, each member $L_1^j$ of $\mathcal{L}_1^j$ is equivariant in the sense that, for all $A \in \mathcal{A}$, $B \in \mathcal{B}$, and $v \in \mathbb{R}^{n \times p \times w_1^{j-1}}$, $L_1^j(AvB)_{**\ell} = AL_1^j(v)_{**\ell}B$ for all $\ell = 1, \ldots, o_1$. When $k = 2, 3$, multi-input-output channel equivariant layers as described in Eq. 22 of Zaheer et al. (2017) are used, except that we replace the sum-pool term in that equation with a mean-pool term (see the next subsection for the rationale). In particular, for $j = 1, \ldots, h_k + 1$, we let $\mathcal{L}_k^j$ denote the collection of all such equivariant layers that map from $\mathbb{R}^{p \times w_k^{j-1}}$ to $\mathbb{R}^{p \times w_k^j}$. For each $j$, each member $L_k^j$ of $\mathcal{L}_k^j$ is equivariant in the sense that, for all $B \in \mathcal{B}$ and $v \in \mathbb{R}^{p \times w_k^{j-1}}$, $L_k^j(Bv) = BL_k^j(v)$. When $k = 4$, standard linear layers mapping from $\mathbb{R}^{w_4^{j-1}}$ to $\mathbb{R}^{w_4^j}$ are used for each $j = 1, \ldots, h_4 + 1$, where $w_4^0 = o_3$ and $w_4^{h_4+1} = 1$. For each $j$, we let $\mathcal{L}_4^j$ denote the collection of all such layers. For a user-specified activation function $q$, we then define the module classes as follows for $k = 1, 2, 3, 4$:

$$\mathcal{M}_k := \{v \mapsto q \circ L_k^{h_k+1} \circ q \circ L_k^{h_k} \circ \ldots \circ q \circ L_k^1(v) : L_k^j \in \mathcal{L}_k^j, j = 1, 2, \ldots, h_k + 1\}.$$

Notably, $\mathcal{M}_1$ satisfies M1 (Ravanbakhsh et al., 2017; Hartford et al., 2018), and $\mathcal{M}_2$ and $\mathcal{M}_3$ satisfy M2 and M3, respectively (Ravanbakhsh et al., 2016; Zaheer et al., 2017). Each element of $\mathcal{M}_4$ is a multilayer perceptron.

The proposed architecture bears some resemblance to CNPs (Garnelo et al., 2018). Like our proposed architecture, CNPs are invariant to permutations of the observations. Nevertheless, CNPs fail to satisfy the other properties imposed on $\mathcal{T}_e$, namely invariance to shifts, rescalings, and permutations of the features and equivariance to shifts and rescalings of the outcome. Moreover, a decision-theoretic rationale for making CNPs invariant to permutations of the observations has not yet been provided in the literature, for example, via a Hunt-Stein-type theorem.

### 4.3 Pros and Cons of Proposed Architecture

A benefit of using the proposed architecture in Algorithm 2 is that Modules 1 and 2 can be evaluated without knowing the feature $x_0$ at which a prediction is desired. As a consequence, these modules can be precomputed before making predictions at new feature values, which can lead to substantial computational savings when the number of values at which predictions will be made is large. Another advantage of the proposed architecture is that it can be evaluated on a dataset that has a different sample size $n$ than did the datasets used during meta-training. In the notation of Eq. 4 from Hartford et al., this corresponds to noting that the weights from an $\mathbb{R}^{N\times M\times k} \to \mathbb{R}^{N\times M\times o}$ multi-input-output channel layer can be used to define an $\mathbb{R}^{N'\times M\times k} \to \mathbb{R}^{N'\times M\times o}$ layer for which the output $Y_{n,m}^{\langle o\rangle}$ is given by the same symbolic expression as that displayed in Eq. 4 from that work, but now with $n$ ranging over $1,\ldots,N'$. We will show in our upcoming experiments that procedures trained using 500 observations can perform well even when evaluated on datasets containing only 100 observations. It is similarly possible to evaluate the proposed architecture on datasets containing a different number of features than did the datasets used during meta-training — again see Eq. 4 in Hartford et al. (2018), and also see Eq. 22 in Zaheer et al. (2017), but with the sum-pool term replaced by a mean-pool term. The rationale for replacing the sum-pool term by a mean-pool term is that this will ensure that the scale of the hidden layers will remain fairly stable when the number of testing features differs somewhat from the number of training features.

A disadvantage of the proposed architecture is that it currently has no established universality guarantees. Such guarantees have been long available for standard multilayer perceptrons (e.g., Cybenko, 1989; Hornik, 1991), and have recently also become available for certain invariant architectures (Maron et al., 2019). In future work, it would be interesting to see if the arguments in Maron et al. (2019) can be modified to provide universality guarantees for our architecture. Establishing such results may also help us to overcome a second disadvantage of our architecture, namely that the resulting neural network classes will not generally satisfy the convexity condition T6 used in Theorem 3. If a network class $\mathcal{T}_{\mathcal{M}}$ that we have proposed can be shown to satisfy a universality result for some appropriate convex class $\mathcal{T}_c$, and if $\mathcal{T}_{\mathcal{M}}$ is itself a subset of $\mathcal{T}_c$, then perhaps it will be possible to invoke Theorem 3 to establish an equilibrium result over the class of estimators $\mathcal{T}_c$, and then to use this result to establish an (approximate) equilibrium result for $\mathcal{T}_{\mathcal{M}}$. To ensure that conditions T1-T3 are satisfied, such an argument will likely require that the weights of the networks in $\mathcal{T}_{\mathcal{M}}$ be restricted to belong to some compact set.

## 5. Numerical Experiments

### 5.1 Overview

In this section, we present the results from two sets of numerical experiments, with the first corresponding to benchmarks from the meta-learning literature and the second consisting of settings designed to evaluate the performance of our method relative to that of analytically-derived estimators that are commonly used in practice for which theoretical performance guarantees are available. In each example, the collection of estimators $\mathcal{T}$ is parameterized as the network architecture introduced in Section 4.2 with $o_1 = o_2 = 50$, $o_3 = 10$, $h_1 = h_3 = 10$, $h_2 = h_4 = 3$, and, for $k = 1, 2, 3, 4$, $w_k = 100$. For each module, we use the leaky ReLU activation $q(z) := \max\{z, 0\} + 0.01 \min\{z, 0\}$. At the end of this section, we report the results of an ablation study that evaluates the extent to which imposing invariance to permutations of the observations and features improves performance.

All experiments were run in Pytorch 1.0.1 on Tesla V100 GPUs using Amazon Web Services. The code used to conduct the experiments can be found at `https://github.com/alexluedtke12/amc-meta-learning-of-optimal-prediction-procedures`. Further experimental details can be found in Appendix D.

### 5.2 Meta-Learning Benchmarks

#### 5.2.1 Preliminaries

We now evaluate the performance of AMC on widely used meta-learning benchmarks. As described in the Introduction, existing meta-learning algorithms tend to be Bayesian in nature, where the goal during meta-training is to learn an estimator with small Bayes risk under a specified prior $\Pi$. Consequently, when adjudicating performance in this study, we will primarily focus on the evaluation of each learned estimator $T$ in terms of its Bayes MSE against this fixed prior $\Pi$, defined as $\int \mathrm{E}_P[\int \{T(\boldsymbol{D})(x_0) - \mu_P(x_0)\}^2 \, dP_X(x_0)] d\Pi(P)$.

Because our method is designed to learn adversarially over a collection of priors $\Gamma$ that satisfies the invariance properties P1, P2, and P3, we define the collection $\Gamma$ used when training our method as the smallest collection of priors that satisfies these three properties and contains $\Pi$. It can be verified that $\Gamma_1$ is a singleton in this case, so that the generator is a constant function and is never updated in these benchmark settings. Though this simplified meta-training may make it appear that AMC will not be robust to an adversarial choice of prior, it is worth noting that the learned estimator in fact is robust to such a choice in the sense that the Bayes risk of the learned estimator will be invariant under permutations of the features and also under shifts and rescalings of the outcomes and features. The main motivation for using a small $\Gamma$ when comparing to these benchmarks is that doing so will help inform on the performance of the estimator architecture that we proposed in Section 4 in Bayesian settings for which existing meta-learning approaches are tailor-made.

We compare the performance of AMC to that of two popular meta-learning methods for which code is readily available: MAML (Finn et al., 2017) and CNPs (Garnelo et al., 2018). Because these algorithms do not prestandardize the features and outcomes, they may have large standardized Bayes MSEs (the Bayes risk derived from Eq. 1) if these quantities are simply shifted or rescaled. To ensure that possible discrepancies in performance between AMC and MAML or CNPs are not solely due to prestandardization, we also compare our

(a) Sinusoid

| | $n$=5 | 10 | 20 |
|---|---|---|---|
| MAML* | 0.22 | 0.10 | 0.03 |
| CNP* | 0.05 | 0.02 | 0.01 |
| MAML-Eq | 2.06 | 0.47 | 0.07 |
| CNP-Eq | 1.13 | 0.13 | 0.04 |
| AMC (ours) | 0.89 | 0.09 | 0.03 |

(b) Gaussian process

| | 1d feature | | 5d feature | |
|---|---|---|---|---|
| | $n$=5 | 50 | 5 | 50 |
| MAML* | 0.85 | 0.13 | 1.00 | 1.00 |
| CNP* | 0.47 | 0.04 | 0.95 | 0.73 |
| MAML-Eq | 0.93 | 0.13 | 1.22 | 1.02 |
| CNP-Eq | 0.56 | 0.04 | 1.12 | 0.73 |
| AMC (ours) | 0.56 | 0.03 | 1.11 | 0.66 |

Table 1: Bayes MSEs of meta-learning approaches in the meta-learning benchmark experiments, where the Bayes MSE is defined as the squared difference between the predictions and true underlying regression function, averaged across draws of the data-generating distribution from the prior and the feature from the feature distribution. Standard errors all $< 0.005$ in the sinusoid experiment and $< 0.001$ in the Gaussian process experiments.
* As these two algorithms do not prestandardize the features or outcomes, their standardized MSEs can be made large by simply shifting or rescaling the features and outcomes. See Figure S5 for more information.

method to natural variants of MAML and CNPs that, like AMC, are robust to such shifts and rescalings. For each method, these variants prestandardize the features and outcomes, and then, in an analogous fashion to line 9 of Algorithm 2, scale the final output by the sample standard deviation of the original training outcomes and shift by their sample mean. These algorithms, which we refer to as MAML-Eq and CNP-Eq, are invariant to shifts and rescalings of the features and equivariant to shifts and rescalings of the outcomes. Details on the MAML and CNP implementations used can be found in Appendix D.1.

#### 5.2.2 Sinusoidal Regression

We start with a benchmark few-shot regression setting from that is commonly used in the meta-learning literature. The prior $\Pi$ is defined as follows. The feature is 1-dimensional and is $\text{Unif}(-5, 5)$ distributed, and the regression function $\mu_P$ takes the form $x \mapsto a \sin(x - b)$, where the parameters $a$ and $b$ are drawn independently from a $\text{Unif}(0.1, 5.0)$ and $\text{Unif}(0, \pi)$ distribution, respectively (Finn et al., 2017). Following related meta-learning benchmarks (Finn et al., 2018; Vuorio et al., 2018), the error $\epsilon_P$ added to the signal $\mu_P(X)$ is distributed as $N(0, 0.3^2)$. We use the same sample sizes as were used in Finn et al. (2017), namely $n = 5$, 10, and 20.

We now report on the performance of the various meta-learning approaches in this setting. In Table 1a, we can see that MAML and CNPs consistently outperform their equivariant counterparts, namely MAML-Eq and CNP-Eq, in this setting. Nevertheless, as we noted earlier, MAML and CNPs are non-robust in that their standardized MSE can be made large by simply shifting or rescaling the outcomes or features. In Figure S5 in the appendix we provide evidence that this is indeed the case. As a particularly striking example, when $n = 5$, scaling the feature down by a factor of 5 leads to 24-fold and 149-fold increases in the MSEs of MAML and CNPs, respectively. The degradation of performance worsens with sample

size. Indeed, when $n = 20$, the same rescaling leads to 144-fold and 487-fold increases in the MSEs of these two methods. Consequently, even seemingly innocuous preprocessings of the data, such as applying an affine transformation to change the unit of measurement, can have a dramatic impact on the performance of MAML and CNPs. In contrast, the standardized MSE performance of MAML-Eq and CNP-Eq is invariant to such preprocessings of the data.

Table 1a also displays results for AMC. AMC consistently outperforms the robust versions of existing algorithms, namely MAML-Eq and CNP-Eq. When compared with the non-robust variants, AMC is outperformed by MAML when $n = 5$, outperforms MAML when $n = 10$, and has about the same performance as MAML when $n = 20$. CNPs perform better than MAML and AMC, though this difference begins to diminish as the sample size increases.

#### 5.2.3 Gaussian Process Regression

We next consider a benchmark Gaussian process regression setting. We consider two cases for the prior. The first is the same as that considered in Garnelo et al. (2018), except that they considered the noise-free case where $\epsilon_P = 0$ almost surely, whereas we consider the noisy case where the errors $\epsilon_P$ are homoscedastic and distributed as $N(0, 0.3^2)$. Considering a noisy case where $\epsilon_P$ is non-degenerate is necessary for the standardized MSE that we consider to be well-defined, and also better reflects real-world regression scenarios where observed outcomes are rarely, if ever, deterministic functions of the features considered. Following Garnelo et al. (2018), the feature is 1-dimensional and follows a $\text{Unif}(-2, 2)$ distribution, and the regression function $\mu_P$ is drawn from a mean-zero Gaussian process with a squared exponential kernel with lengthscale 0.4 and variance 1. We also use the same sample sizes as were used in that work, namely $n = 5$ and 50. The second case that we consider is the same as the first except that the feature $X$ is 5-dimensional, where the entries of $X$ are independent $\text{Unif}(-2, 2)$ random variables, and the lengthscale is taken to be equal to 1.2.

Table 1b displays the performance of the various methods in this setting. Adversarial Monte Carlo noticeably outperforms MAML and MAML-Eq across all settings except the 5-dimensional, $n = 5$ case, where MAML performs slightly better than does AMC. The ordering between AMC and the CNP-based methods varies by sample size. At the smaller sample size considered ($n = 5$), AMC outperforms the robust CNP-based method, namely CNP-Eq, but is outperformed by the non-robust method, namely CNP. In the larger sample size considered ($n = 50$), AMC outperforms both CNP and CNP-Eq. The fact that AMC outperforms CNP in this setting is notable given that CNPs are designed to mimic the desirable properties of Gaussian process regression procedures (Garnelo et al., 2018).

### 5.3 Comparing to (Regularized) Empirical Risk Minimizers

#### 5.3.1 Preliminaries

We now compare the performance of our approach to that of existing estimators that are commonly used in practice for which theoretical performance guarantees are available. The examples differ in the definitions of the model $\mathcal{P}$ and the collection $\Gamma$ of priors on $\mathcal{P}$. In each case, $\Gamma$ satisfies the invariance properties P1, P2, and P3. By the equivariance of the estimators in $\mathcal{T}$, Theorem 2 shows that it suffices to consider a collection of priors $\Gamma_1$ with support on $\mathcal{P}_1$. Hence, it suffices to define the collection $\mathcal{P}_1 \subset \mathcal{P}$ of distributions $P$ satisfying

(6). By P2 and P3, we see that $\mathcal{P} = \cup_{P \in \mathcal{P}_1} \mathcal{P}(P)$, where $\mathcal{P}(P)$ consists of the distributions of $(a + b \odot X, \tilde{a} + \tilde{b}Y)$ when $(X, Y) \sim P$; here, $a$, $b$, $\tilde{a}$, and $\tilde{b}$ vary over $\mathbb{R}^p$, $(\mathbb{R}^+)^p$, $\mathbb{R}$, and $\mathbb{R}^+$, respectively. In each setting, the submodel $\mathcal{P}_1$ takes the form

$$\mathcal{P}_1 := \left\{ P : \mu_P \in \mathcal{R}, P_X \in \mathcal{P}_X, \epsilon_P | X \overset{P}{\sim} N(0,1) \right\}$$

and the $p = 10$ dimensional feature $X$ is known to be drawn from a distribution in the set $\mathcal{P}_X$ of $N(\mathbf{0}_p, \Sigma)$ distributions, where $\Sigma$ varies over all positive-definite $p \times p$ covariance matrices with diagonal equal to $\mathbf{1}_p$. The collections $\mathcal{R}$ of regression functions differ in the examples and are detailed in the coming subsections. These collections are indexed by a sparsity parameter $\mathfrak{s}$ that specifies the number of features that may contribute to the regression function $\mu_P$. In each setting, we considered all four combinations of $\mathfrak{s} \in \{1, 5\}$ and $n \in \{100, 500\}$, where $n$ denotes the number of observations in the datasets $\boldsymbol{d}$ used to evaluate the performance of the final learned estimators. For each $n$, we evaluated the performance of AMC meta-trained with datasets of size $n_{mt} = 100$ observations (AMC100) and $n_{mt} = 500$ observations (AMC500).

### 5.3.2 Sparse Linear Regression

We first considered the setting where $\mu_P$ belongs to a sparse linear model and the feature is $p = 10$ dimensional. In this setting,

$$\mathcal{R} := \{x \mapsto \beta^\top x : \|\beta\|_0 \leq \mathfrak{s}, \|\beta\|_1 \leq 5\}, \tag{7}$$

where $\|a\|_0 := \#\{j : a_j \neq 0\}$ and $\|a\|_1 := \sum_{j=1}^p |a_j|$. The collection $\Gamma$ is described in Appendix D.

For each sparsity level $\mathfrak{s} \in \{1, 5\}$, we evaluated the performance of the prediction procedure trained at sparsity level $\mathfrak{s}$ using two priors. Both priors sample the covariance matrix of the feature distribution $P_X$ from the Wishart prior $\Pi_X$ described in Appendix D.2.1 and let $\beta = (\alpha, 0)$ for a random $\alpha$ satisfying $\|\alpha\|_1 \leq 5$. They differ in how $\alpha$ is drawn. Both make use of a uniform draw $Z$ from $\ell_1$ ball $\{a \in \mathbb{R}^{\mathfrak{s}} : \|a\|_1 = 5\}$. The first sets $\alpha = Z$, whereas the second sets $\alpha = UZ$ for $U \sim \text{Unif}(0, 1)$ drawn independently of $Z$. We will refer to the two settings as 'boundary' and 'interior', respectively. We refer to the $\mathfrak{s} = 1$ and $\mathfrak{s} = 5$ cases as the 'sparse' and 'dense' settings, respectively. Further details can be found in Appendix D.2.2.

In this example, AMC leverages knowledge of the underlying sparse linear regression model by generating synthetic training data from distributions $P$ for which $E_P[Y|X = \cdot]$ belongs to the class $\mathcal{R}$ defined in Eq. 7 (see line 5 of Algorithm 1). Therefore, we aimed to compare AMC's performance to that of estimators that also take advantage of this linearity. Ideally, we would compare AMC's performance to that of the true $\Gamma$-minimax estimator. Unfortunately, as is the case in most problems, the form of this estimator is not known in this sparse linear regression setting. Therefore, we instead compared AMC's performance to ordinary least squares (OLS) and lasso (Tibshirani, 1996) with tuning parameter selected by 10-fold cross-validation, as implemented in `scikit-learn` (Pedregosa et al., 2011).

Table 2a displays performance for the sparse setting. We see that AMC outperformed OLS and lasso for the boundary priors, and was outperformed for the interior priors. Surprisingly, AMC500 outperformed AMC100 for the interior prior when $n = 100$ observations were used

(a) Sparse signal

| | Boundary | | Interior | |
|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 |
| OLS | 0.12 | 0.02 | 0.12 | 0.02 |
| Lasso | 0.06 | 0.01 | 0.06 | 0.01 |
| AMC100 (ours) | 0.02 | <0.01 | 0.11 | 0.09 |
| AMC500 (ours) | 0.02 | <0.01 | 0.07 | 0.04 |

(b) Dense signal

| | Boundary | | Interior | |
|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 |
| OLS | 0.13 | 0.02 | 0.13 | 0.02 |
| Lasso | 0.11 | 0.02 | 0.09 | 0.02 |
| AMC100 (ours) | 0.10 | 0.04 | 0.08 | 0.02 |
| AMC500 (ours) | 0.09 | 0.02 | 0.09 | 0.02 |

Table 2: MSEs based on datasets of size $n$ in the linear regression settings. Standard errors all $< 0.001$.

to evaluate performance. The fact that AMC100 was trained specifically for the $n = 100$ case suggests that a suboptimal equilibrium may have been reached in this setting. Table 2b displays performance for the dense setting. Here AMC always performed at least as well as OLS and lasso when $n_{mt} = n$, and performed comparably even when $n_{mt} \neq n$.

#### 5.3.3 Fused Lasso Additive Model

We next considered the setting where $P$ belongs to a variant of the fused lasso additive model (FLAM) (Petersen et al., 2016) and the feature is $p = 10$ dimensional. This model enforces that $\mu_P$ belong to a generalized additive model, that only a certain number of the components can be different from the zero function, and that the sum of the total variations of the remaining components is not too large. We recall that the total variation $V(f)$ of $f : \mathbb{R} \to \mathbb{R}$ is equal to the supremum of $\sum_{\ell=1}^{k} |f(a_{\ell+1}) - f(a_\ell)|$ over all $(a_\ell)_{\ell=1}^{k+1}$ such that $k \in \mathbb{N}$ and $a_1 < a_2 < \ldots < a_{k+1}$ (Cohn, 2013). Let $v(\mu) := (V(\mu_j))_{j=1}^{p}$. Writing $x_j$ to denote feature $j$, the model we considered imposes that $\mu_P$ falls in

$$\mathcal{R} := \left\{ x \mapsto \sum_{j=1}^{p} \mu_j(x_j) : \|v(\mu)\|_1 \leq M, \|v(\mu)\|_0 \leq \mathfrak{s} \right\}.$$

We take $M = 10$ in the experiments in this section. The collection $\Gamma$ is described in Appendix D.

In this example, we preprocessed the features before supplying them to the estimator. In particular, we replaced each entry with its rank statistic among the $n$ observations so that, for each $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$, we replaced $\boldsymbol{x}_{ij}$ by $\sum_{k=1}^{n} I\{\boldsymbol{x}_{ij} \geq \boldsymbol{x}_{kj}\}$ and $x_{0j}$ by $\sum_{k=1}^{n} I\{x_{0j} \geq \boldsymbol{x}_{kj}\}$. This preprocessing step is natural given that the FLAM estimator (Petersen et al., 2016) also only depends on the features through their ranks. An advantage

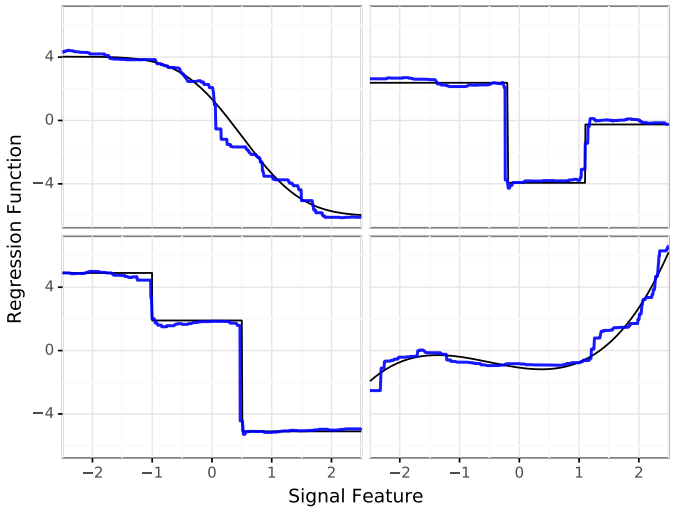



Figure 1: Examples of AMC500 fits (thin blue lines) based on $n = 500$ observations drawn from distributions at sparsity level $\mathfrak{s} = 1$ with four possible signal components (thick black lines). Predictions obtained at different signal feature values with all 9 other features set to zero.

of making this restriction is that, by the homoscedasticity of the errors and the invariance of the rank statistics and total variation to strictly increasing transformations, the learned estimators should perform well even if the feature distributions do not belong to a Gaussian model, but instead belong to a much richer Gaussian copula model.

We evaluated the performance of the learned estimators using variants of simulation scenarios 1-4 from Petersen et al. (2016). The level of smoothness varies across the settings (see Fig. 2 in that work). In the variants we considered, the true regression function either contains $\mathfrak{s}_0 = 1$ ('sparse') or $\mathfrak{s}_0 = 4$ ('dense') nonzero components. In the sparse setting, we evaluated the performance of the estimators that were meta-trained at sparsity level $\mathfrak{s} = 1$, and, in the dense setting, we evaluated the performance of the estimators that were meta-trained at $\mathfrak{s} = 5$. Further details can be found in Appendix D.2.3.

Similarly to as in the previous example, AMC leverages knowledge of the possible forms of the regression function that is imposed by $\mathcal{R}$ — in this case, the model for the regression function is nonparametric but does impose that this function belongs to a particular sparse generalized additive model. Though there does not exist a competing estimator that is designed to optimize over $\mathcal{R}$, the FLAM estimator (Petersen et al., 2016) optimizes over the somewhat larger, non-sparse model where $\mathfrak{s} = p$. We, therefore, compared the performance of AMC to this estimator as a benchmark, with the understanding that AMC is slightly advantaged in that it has knowledge of the underlying sparsity pattern. Nevertheless, we view this experiment as an important proof-of-concept, as it is the first, to our knowledge, to evaluate whether it is feasible to adversarially meta-learn a prediction procedure within a nonparametric regression model.

To illustrate the kinds of functions that AMC can approximate, Fig. 1 displays examples of AMC500 fits from scenario 3 when $(n, \mathfrak{s}) = (500, 1)$. Table 3 provides a more comprehensive

(a) Sparse signal

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | |
|---|---|---|---|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 | 100 | 500 | 100 | 500 |
| FLAM | 0.44 | 0.12 | 0.47 | 0.17 | 0.38 | 0.11 | 0.51 | 0.19 |
| AMC100 (ours) | 0.34 | 0.20 | 0.18 | 0.08 | 0.27 | 0.14 | 0.17 | 0.08 |
| AMC500 (ours) | 0.48 | 0.12 | 0.19 | 0.06 | 0.35 | 0.10 | 0.23 | 0.08 |

(b) Dense signal

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | |
|---|---|---|---|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 | 100 | 500 | 100 | 500 |
| FLAM | 0.59 | 0.17 | 0.65 | 0.24 | 0.53 | 0.16 | 0.76 | 0.36 |
| AMC100 (ours) | 1.20 | 0.91 | 0.47 | 0.39 | 0.87 | 0.57 | 0.30 | 0.30 |
| AMC500 (ours) | 0.58 | 0.15 | 0.37 | 0.08 | 0.46 | 0.12 | 0.36 | 0.09 |

Table 3: MSEs based on datasets of size $n$ in the FLAM settings. Standard errors for FLAM all $< 0.04$ and for AMC all $< 0.01$.

view of the performance of AMC and compares it to that of FLAM. Table 3a displays performance for the sparse setting. The AMC procedures meta-trained with $n_{mt} = n$ observations outperformed FLAM for all of these settings. Interestingly, AMC procedures meta-trained with $n_{mt} \neq n$ also outperformed FLAM in a majority of these settings, suggesting that learned procedures can perform well even at different sample sizes from those at which they were meta-trained. In the dense setting (Table 3b), AMC500 outperformed both AMC100 and FLAM in all but one setting (scenario 4, $n = 100$), and in this setting both AMC100 and AMC500 dramatically outperformed FLAM. The fact that AMC500 also sometimes outperformed AMC100 when $n = 100$ in the linear regression setting suggests that there may be some benefit to training a procedure at a larger sample size than that at which it will be evaluated. We leave an investigation of the generality of this phenomenon to future work.

### 5.4 Ablation Study to Evaluate the Performance of Permutation Invariance

We numerically evaluated the utility of imposing invariance in the architecture in Algorithm 2. To do this, we repeated the $n = n_{mt} = 100$ and $n = n_{mt} = 500$ FLAM settings, separately modifying the architecture to remove invariance to permutations of the observations and the features. In the case where the architecture was not invariant to permutations of the observations, we weakened M1 to the condition that $m_1(vB)_{**\ell} = [m_1(v)_{**\ell}]B$ for all $m_1 \in \mathcal{M}_1$, $B \in \mathcal{B}$, $v \in \mathbb{R}^{n \times p \times 2}$, and $\ell = 1, \ldots, o_1$. We used the same architecture as was used in our earlier experiment, except that each layer in Module 1 was replaced by a multi-input-output channel layer that is equivariant to permutations of the $p$ features (Zaheer et al., 2017), and the output of the final layer was of dimension $\mathbb{R}^{p \times o_1}$ so that the subsequent mean pooling layer could be removed. In the case where the architecture was not invariant to permutations of the features, we removed conditions M2 and M3 and also weakened M1 to the condition that $m_1(Av)_{**\ell} = A[m_1(v)_{**\ell}]$ for all $m_1 \in \mathcal{M}_1$, $A \in \mathcal{A}$, $v \in \mathbb{R}^{n \times p \times 2}$, and $\ell = 1, \ldots, o_1$. We used the same architecture as in our earlier experiment except that

(a) Sparse signal

| | | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | $n$=100 | 500 | 100 | 500 | 100 | 500 | 100 | 500 |
| Not invariant to | observations | 6.98 | 38.29 | 5.82 | 29.93 | 5.03 | 27.58 | 4.29 | 13.08 |
| permutations of: | features | 1.01 | 0.95 | 1.16 | 1.09 | 1.02 | 0.98 | 1.01 | 0.99 |

(b) Dense signal

| | | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | $n$=100 | 500 | 100 | 500 | 100 | 500 | 100 | 500 |
| Not invariant to | observations | 1.86 | 14.68 | 1.69 | 8.60 | 1.97 | 14.20 | 1.51 | 4.70 |
| permutations of: | features | 1.05 | 2.55 | 0.99 | 1.98 | 1.09 | 3.02 | 1.04 | 1.67 |

Table 4: Fold-change in MSEs for modifications of AMC in the FLAM settings with $n = 100$, as compared to the performances of FLAM listed in Table 3. Standard errors all $\leq 0.03$ times the fold-change in the MSE.

Modules 2 and 3 were replaced by multilayer perceptrons and each layer in Module 1 was replaced by a multi-input-output channel layer that is equivariant to permutations of the $n$ observations.

Table 4 displays the results. In every setting considered, removing invariance to permutations of the observations led to a marked increase in the MSE of the estimator, with the degradation of performance tending to be worse at the larger sample size. In the most extreme scenario, the MSE of the non-invariant estimator was 38 times higher than that of the invariant estimator. Removing invariance to permutations of the features also tended to worsen performance, sometimes by a factor of 2 or 3, though there were a few settings where performance improved slightly (no more than 5%). Taken together, these results suggest that *a priori* enforcing that the estimator is invariant to permutations of the features and observations can dramatically improve performance.

## 6. Data Experiments

We also used real datasets to evaluate the performance of AMC100 estimators meta-trained in sparse linear regression settings (Section 5.3.2) or fused lasso additive model settings (Section 5.3.3). We compared the performance of our estimators to the estimators from our numerical experiments, namely, the OLS, lasso, and FLAM estimators. These estimators are natural comparators because they assume the same or similar models as do our AMC estimators; consequently, comparing to these estimators allows us to focus our discussion on differences in the performance of existing estimation strategies as compared to that of new meta-learned strategies, rather than on differences in underlying assumptions that could potentially be resolved by training a new AMC estimator under a different model.

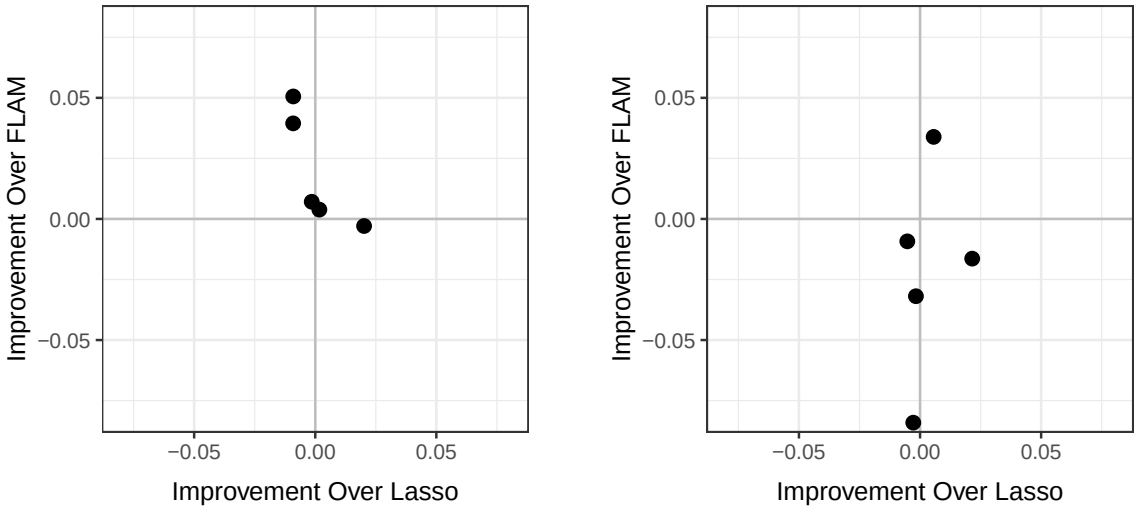
Because the implementations of lasso and FLAM that we compared to both use 10-fold cross-validation to select tuning parameters, we also used 10-fold cross-validation to select tuning parameters for the AMC100 estimators. The first of these estimators, which we refer to as "AMC Linear", selects a tuning parameter $\mathfrak{s} \in \{1, 2, \ldots, 10\}$ by finding the value of $\mathfrak{s}$ for which the cross-validated MSE of an AMC100 estimator trained in the sparse linear

regression setting with sparsity level $\mathfrak{s}$ is minimal. The final prediction then corresponds to that returned by the AMC100 estimator trained in the model with this selected value of $\mathfrak{s}$. The second, which we refer to as "AMC FLAM", selects two tuning parameters, one of which reflects the sparsity level $\mathfrak{s}$ of the problem and the other of which corresponds to the bound $M$ on the sum of the variation norms of the $\mu_j$ components in the fused lasso additive model. In particular, the tuning parameters $(\mathfrak{s}, M) \in \{1, 2, \ldots, 10\} \times \{5, 10, 20\}$ are chosen to be those that minimize the cross-validated MSE of an AMC100 estimator trained in the fused lasso additive model with parameters $(\mathfrak{s}, M)$. Notably, each candidate estimator considered by AMC Linear and AMC FLAM only has access to 90, rather than 100, observations when selecting tuning parameter values using 10-fold cross-validation on a dataset of size $n = 100$. This does not pose a problem because, as was noted in Section 4.3, the trained estimators can be evaluated at different sample sizes than those at which they were trained.

In settings where both AMC-trained estimators and other estimators are available, it is natural to wonder whether there is a way to capitalize on the availability of both types of methods. Ensemble algorithms provide a natural means to do this, with stacked ensembles representing an especially appealing option given theoretical guarantees that adding base learners will not typically degrade performance (Van der Vaart et al., 2006; Van der Laan et al., 2007) and existing experiments showing that they often outperform all included base learners (e.g., Polley and Van der Laan, 2010). We, therefore, evaluate the performance of three stacked ensembles in these experiments. The first includes only the AMC Linear and AMC FLAM estimators as base learners. The second only includes the OLS, lasso, and FLAM estimators. The third includes all five of these estimators. Predictions of the base learners were combined using 10-fold cross-validation. Following the recommendation of Breiman (1996), we employed a non-negative least squares estimator for this combination step.

Our experiments make use of ten datasets. Six of these datasets are available through the University of California, Irvine (UCI) Machine Learning Repository (Dua and Graff, 2017), three were used to illustrate supervised learning machines in popular statistical learning textbooks (Friedman et al., 2001; James et al., 2013), and one was used as an illustrative example in the paper that introduced FLAM (Petersen et al., 2016). All of these datasets contain more than 100 observations. Five of them have at least 10 features and the others have fewer (5, 6, 6, 7, and 9). All outcomes are standardized to have empirical variance 1 so that, for each dataset, the cross-validated MSE performance of a sample mean for predicting the outcome is approximately 1. Further details on these datasets can be found in Appendix E.1.

We evaluated our learned estimators in three settings. First, we considered the case where the number of features in the datasets matched the number that they saw during training, namely 10. In particular, we evaluated the performance of AMC Linear and AMC FLAM in the 5 datasets that have 10 or more features by randomly selecting 100 observations and 10 features from each dataset and evaluating MSE on the held out observations. This and all other Monte Carlo evaluations of MSE described in what follows were repeated 200 times and averaged across the replications. Second, we evaluated the robustness of our learned estimators to a key assumption used during training. In particular, we evaluated the performance of our estimators on the 5 datasets that have fewer features than the 10



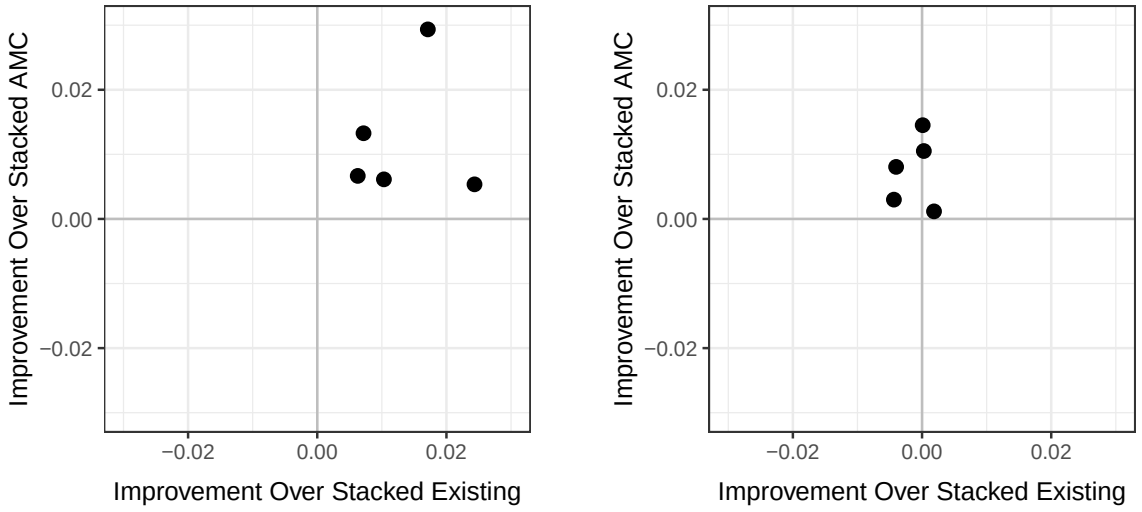

(a) Datasets with same number of features as used during meta-training

(b) Datasets with fewer features than used during meta-training

Figure 2: Improvement of AMC estimators over existing estimators, in terms of differences of cross-validated MSEs of FLAM and AMC FLAM (x-axis) and Lasso and AMC Linear (y-axis). Positive values indicate that AMC outperformed the comparator. AMC performed similarly to or better than existing estimators in settings where the number of features in the dataset was the same as were used in meta-training. As expected, the performance was somewhat worse for datasets that had fewer features than were used during meta-training, though, surprisingly, it was still sometimes better than that of existing methods.

used during meta-training, again sampling 100 observations and evaluating MSE on the held out observations. Third, we evaluated the relative performance of our estimators at varying levels of signal sparsity for each of the ten datasets. In particular, for each training-test split of the data, we selected $\mathfrak{s}$ total features from the dataset, removed the remaining features, and then included $(10 - \mathfrak{s})$ Gaussian noise features so that the dimension of the feature was always $p = 10$.

We first discuss performance on datasets with the same number of features as were used during meta-training. Complete numerical results for estimator performance can be found in Table S5 in Appendix E.2. Here, we focus on graphical summaries of performance to communicate the key trends that we saw. Figure 2a shows that AMC FLAM performed similarly to or better than FLAM across all settings, and AMC Linear performed similarly to lasso across all settings. We have compared AMC Linear to lasso as a baseline in this figure because lasso performed similarly to or better than OLS across all settings. Figure 3a shows that stacking all available base learners consistently yielded better performance than did only stacking only the existing estimators or the AMC estimators. This stacked ensemble also outperformed all base learners considered. These results suggest that incorporating AMC estimators into regression pipelines can reliably lead to improved predictions even in settings where performant learners are already available.




(a) Datasets with same number of features as used during meta-training

(b) Datasets with fewer features than used during meta-training

Figure 3: Improvement of the stacked ensemble algorithm that includes all base learners over those which only include a subset (existing learners or AMC learners), in terms of differences of cross-validated MSEs. Including both AMC and existing estimators as base learners always outperformed only including a subset when the dataset contained the same number of features as were used during training. Adding AMC base learners did not tend to improve performance when the dataset had fewer features than were used during meta-training, though any degradation in performance was minimal.

We now discuss performance on datasets with fewer features than were used during meta-training. Figure 2b displays performance on datasets that have fewer features than were used during meta-training. Unsurprisingly, performance was somewhat less desirable than it was for datasets with the same number of features as were used during meta-training. AMC FLAM tended to be somewhat outperformed by FLAM, though did outperform FLAM in one setting. AMC Linear continued to perform similarly to lasso across all settings. Figure 3 shows that stacking all available base learners outperformed stacking only AMC estimators, and performed similarly to stacking existing estimators.

We conclude by discussing the performance of the estimators when we induce varying levels of signal sparsity. Figure 4 shows that AMC FLAM outperformed FLAM for the vast majority of datasets and sparsity patterns. The only exception to this trend occurred for the yacht dataset and the LAozone dataset for denser signals (7, 8, or 9 signal features), where AMC FLAM was slightly outperformed by FLAM. Figure S6 in the appendix shows that AMC Linear consistently outperformed OLS and performed comparably to or slightly better than lasso in most settings. Figure S7 shows that there was not a major difference between the cross-validated MSE of the three stacking algorithms. Nevertheless, it is worth noting that stacking all available base learners did outperform the other two stacking schemes in 53% of the 83 dataset-sparsity settings considered, with the stacking scheme that only

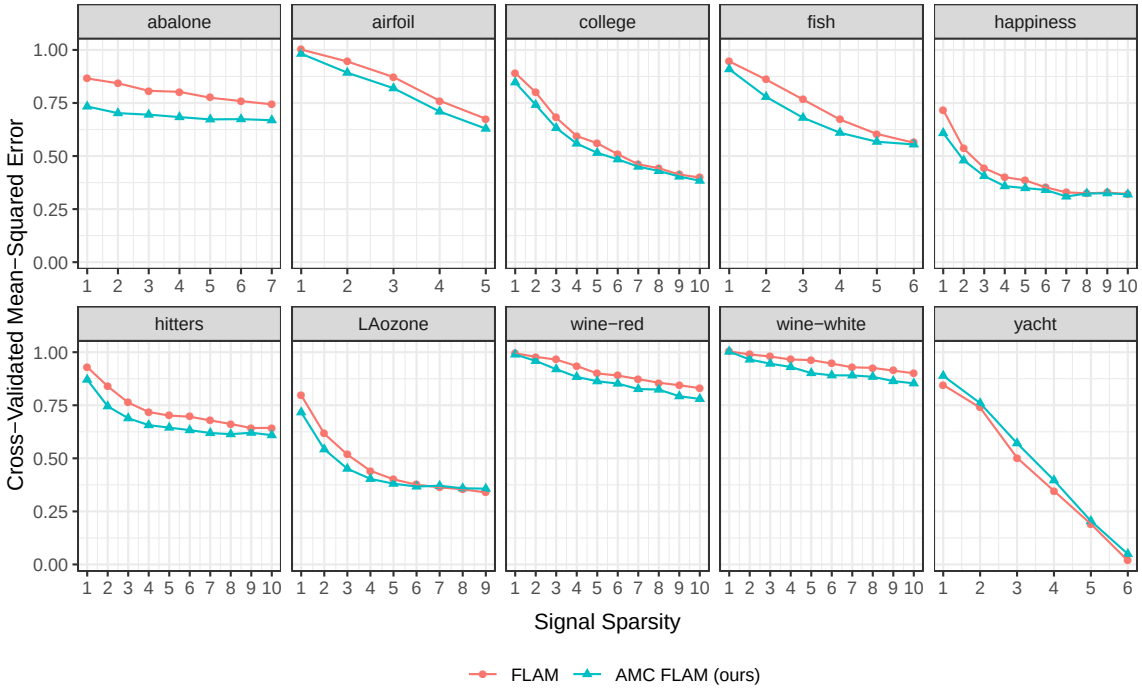



Figure 4: Performance of FLAM and AMC FLAM at different sparsity levels. For each training-validation split of the data, between 1 and $q$ features are selected at random from the original dataset (x-axis), where $q$ is the minimum of 10 and the total number of features in the dataset, and Gaussian noise features are then added so that there are 10 total features. Therefore, the signal is expected to become denser and stronger as the x-axis value increases. AMC FLAM outperforms FLAM in most settings.

included AMC algorithms performing best in 39% of the settings and the scheme that only included existing algorithms performing best in only 8% of these settings. Thus, we again see evidence that including AMC base learners in a stacked ensemble can improve performance, even when other learners are already available.

## 7. Proofs

### 7.1 A Study of Group Actions that are Useful for Our Setting

To prove Theorem 1, it will be convenient to use tools from group theory to describe and study the behavior of our estimation problem under the shifts, rescalings, and permutations that we consider. For $k \in \mathbb{N}$, let $\mathrm{Sym}(k)$ be the symmetric group on $k$ symbols. Let $\mathbb{R} \rtimes \mathbb{R}^+$ be the semidirect product of the real numbers with the positive real numbers with the group multiplication

$$(a_1, b_1)(a_2, b_2) = (a_1 + b_1 a_2, b_1 b_2).$$

Define $\mathcal{G}_0 := (\mathbb{R} \rtimes \mathbb{R}^+) \times [(\mathbb{R} \rtimes \mathbb{R}^+)^p \rtimes \mathrm{Sym}(p)] \times \mathrm{Sym}(n)$. Let $\mathcal{O}_n := \{a \in \mathbb{R}^n : \bar{a} = 0, s(a) = 1\}$. Throughout we equip $\mathcal{G}_0$ with the product topology.

We note that the quantity $\mathcal{Z}$ defined in Section 2.1 writes as

$$\mathcal{Z} = \mathcal{O}_n^p \times \mathcal{O}_n \times \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \times \mathbb{R}. \tag{8}$$

Denote the generic group element $g = ((g^{j+}, g^{j\times})_{j=0}^p, \tau_g, \eta_g)$ where $(g^{j+}, g^{j\times}) \in \mathbb{R} \rtimes \mathbb{R}^+$, $\tau_g \in \mathrm{Sym}(p)$, and $\eta_g \in \mathrm{Sym}(n)$. Denote the generic element $\boldsymbol{z} \in \mathcal{Z}$ by

$$\boldsymbol{z} = ((z^{x,1,j}, \cdots, z^{x,n,j})_{j=1}^p, (z^{y,1}, \cdots, z^{y,n}), (z^{x,0,j})_{j=1}^p, (z^{\bar{x},j})_{j=1}^p, z^{\bar{y}}, (z^{s(x),j})_{j=1}^p, z^{s(y)}).$$

For $g_1 = ((g_1^{j+}, g_1^{j\times})_{j=0}^p, \tau_1, \eta_1)$, $g_2 = ((g_2^{j+}, g_2^{j\times})_{j=0}^p, \tau_2, \eta_2)$, two arbitrary elements in $\mathcal{G}_0$, define the group multiplication as

$$g_1 g_2 = \left( g_1^{0+} g_2^{0\times} + g_1^{0\times} g_2^{0+}, g_1^{0\times} g_2^{0\times}, \left(g_1^{j+} g_2^{\tau_1^{-1}(j)\times} + g_1^{j\times} g_2^{\tau_1^{-1}(j)+}, g_1^{j\times} g_2^{\tau_1^{-1}(j)\times}\right)_{j=1}^p, \tau_1\tau_2, \eta_1\eta_2 \right).$$

Define the group action $\mathcal{G}_0 \times \mathcal{Z} \to \mathcal{Z}$ by

$$\begin{aligned}
(g \cdot \boldsymbol{z})^{x,i,j} &= z^{x,\eta_g^{-1}(i),\tau_g^{-1}(j)} \\
(g \cdot \boldsymbol{z})^{y,i} &= z^{y,\eta_g^{-1}(i)} \\
(g \cdot \boldsymbol{z})^{x,0,j} &= z^{x,0,\tau_g^{-1}(j)} \\
(g \cdot \boldsymbol{z})^{\bar{x},j} &= \frac{g^{j+}}{g^{j\times}} + z^{\bar{x},\tau_g^{-1}(j)} \\
(g \cdot \boldsymbol{z})^{\bar{y}} &= \frac{g^{0+}}{g^{0\times}} + z^{\bar{y}} \\
(g \cdot \boldsymbol{z})^{s(x),j} &= \log g^{j\times} + z^{s(x),\tau_g^{-1}(j)} \\
(g \cdot \boldsymbol{z})^{s(y)} &= \log g^{0\times} + z^{s(y)},
\end{aligned}$$

where $i \in \{1, 2, \cdots, n\}$ and $j \in \{1, 2, \cdots, p\}$.

We make use of the below result without statement in the remainder of this section.

**Lemma 5** *The map defined above is a left group action.*

**Proof** The identity axiom, namely that $e \cdot \boldsymbol{z} = \boldsymbol{z}$ when $e$ is the identity element of $\mathcal{G}_0$, is straightforward to verify and so we omit the arguments. Fix $g_1, g_2 \in \mathcal{G}_0$ and $\boldsymbol{z} \in \mathcal{Z}$. We establish compatibility by showing that $g_1 g_2 \cdot \boldsymbol{z} = g_1 \cdot (g_2 \cdot \boldsymbol{z})$. To see that this is indeed the case, note that, for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$:

$$\begin{aligned}
(g_1 g_2 \cdot \boldsymbol{z})^{y,i} &= z^{y,(\eta_1\eta_2)^{-1}(i)} = z^{y,\eta_2^{-1}\eta_1^{-1}(i)} = (g_2 \cdot \boldsymbol{z})^{y,\eta_1^{-1}(i)} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{y,i} \\
(g_1 g_2 \cdot \boldsymbol{z})^{x,i,j} &= z^{x,\eta_2^{-1}\eta_1^{-1}(i),\tau_2^{-1}\tau_1^{-1}(j)} = (g_2 \cdot \boldsymbol{z})^{x,\eta_1^{-1}(i),\tau_1^{-1}(j)} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{x,i,j} \\
(g_1 g_2 \cdot \boldsymbol{z})^{x,0,j} &= z^{x,0,\tau_2^{-1}\tau_1^{-1}(j)} = (g_2 \cdot \boldsymbol{z})^{x,0,\tau_1^{-1}(j)} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{x,0,j} \\
(g_1 g_2 \cdot \boldsymbol{z})^{\bar{x},j} &= \frac{g_1^{j+} g_2^{\tau_1^{-1}(j)\times} + g_1^{j\times} g_2^{\tau_1^{-1}(j)+}}{g_1^{j\times} g_2^{\tau_1^{-1}(j)\times}} + z^{\bar{x},\tau_2^{-1}\tau_1^{-1}(j)} = \frac{g_1^{j+}}{g_1^{j\times}} + (g_2 \cdot z)^{\bar{x},\tau_1^{-1}(j)} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{\bar{x},j} \\
(g_1 g_2 \cdot \boldsymbol{z})^{\bar{y}} &= \frac{g_1^{0+} g_2^{0\times} + g_1^{0\times} g_2^{0+}}{g_1^{0\times} g_2^{0\times}} + z^{\bar{y}} = \frac{g_1^{0+}}{g_1^{0\times}} + (g_2 \cdot \boldsymbol{z})^{\bar{y}} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{\bar{y}}
\end{aligned}$$

$$(g_1 g_2 \cdot \boldsymbol{z})^{s(x),j} = \log(g_1^{j\times} g_2^{j\times}) + z^{s(x),\tau_2^{-1}\tau_1^{-1}(j)} = \log g_1^{j\times} + (g_2 \cdot \boldsymbol{z})^{s(x),\tau_1^{-1}(j)} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{s(x),j}$$
$$(g_1 g_2 \cdot \boldsymbol{z})^{s(y)} = \log(g_1^{0\times} g_2^{0\times}) + z^{(s(y)} = \log g_1^{0\times} + (g_2 \cdot \boldsymbol{z})^{s(y)} = (g_1 \cdot (g_2 \cdot \boldsymbol{z}))^{s(y)}.$$

■

We now introduce several group actions that we will make heavy use of in our proof of Theorem 1 and in the lemmas that precede it. We first define $\mathcal{G}_0 \times \mathcal{S} \to \mathcal{S}$. For $S \in \mathcal{S}$ and $g \in \mathcal{G}_0$, define $g \cdot S$ to be $(g \cdot S)(\boldsymbol{z}) = S(g \cdot \boldsymbol{z})$. Conditions T4 and T5 can be restated as $g \cdot S \in \mathcal{S}$ for all $g \in \mathcal{G}_0$ and $S \in \mathcal{S}$. It can then readily be shown that, under these conditions, the defined map is a left group action. For $T \in \mathcal{T}$, we will write $g \cdot T$ to denote the $\mathcal{D} \to (\mathcal{X} \to \mathbb{R})$ operator defined so that

$$(g \cdot T)(\boldsymbol{d}) : x_0 \mapsto \begin{cases} \bar{\boldsymbol{y}} + s(\boldsymbol{y})(g \cdot S_T)(z(\boldsymbol{d}, x_0)), & \text{if } (\boldsymbol{d}, x_0) \in \mathcal{D}_0, \\ 0, & \text{otherwise.} \end{cases}$$

It is possible that $g \cdot T$ does not belong to $\mathcal{T}$ due to its behavior when $(\boldsymbol{d}, x_0) \notin \mathcal{D}_0$, and therefore that the defined map is not a group action. Nonetheless, because $\mathcal{D}_0$ has $P$-probability one for any $P \in \mathcal{P}$, this fact will not pose any difficulties in our arguments.

We now define the group action $\mathcal{G}_0 \times (\mathcal{Y} \times \mathcal{X}) \to (\mathcal{Y} \times \mathcal{X})$. For $(y, x) \in \mathbb{R} \times \mathbb{R}^p$, define $g \cdot (y, x)$ as

$$g \cdot (y, x) = (g^{0+} + g^{0\times} y, (g^{i+} + g^{i\times} x^{\tau_g^{-1}(i)})_{i=1}^p).$$

Similar arguments to those used to prove Lemma 5 show that the map defined above is a left group action. We now define the group action $\mathcal{G}_0 \times \mathcal{P} \to \mathcal{P}$. For $P \in \mathcal{P}$, $g \in \mathcal{G}_0$, define $g \cdot P = P \circ g^{-1}$ by $(g \cdot P)(U) = P(g^{-1}(U))$, where

$$g^{-1}(U) = \{(y, x) \in \mathbb{R}^{p+1} : g \cdot (y, x) \in U\}.$$

Under P1, P2, and P3, which, as noted in the Section 2.1, implicitly encode that $P \circ g^{-1} \in \mathcal{P}$, it can readily be shown that the defined map is a left group action. Finally, we define the group action $\mathcal{G}_0 \times \Gamma \to \Gamma$. For $\Pi \in \Gamma$, $g \in \mathcal{G}_0$, define $g \cdot \Pi = \Pi \circ g^{-1}$ by $(g \cdot \Pi)(U) = \Pi(g^{-1}(U))$ where

$$g^{-1}(U) = \{P \in \mathcal{P} : g \cdot P \in U\}.$$

We can restate P1, P2, and P3 as $\Pi \circ g^{-1} \in \Gamma$ for all $\Pi \in \Gamma$, $g \in \mathcal{G}_0$. Under these conditions, it can be shown that the defined map is a left group action.

We now show that $\mathcal{G}_0$ is amenable — see Appendix A for a review of this concept. Establishing this fact will allow us to apply Day's fixed point theorem (Theorem S3 in Appendix A) in the upcoming proof of Theorem 1.

**Lemma 6** *$\mathcal{G}_0$ is amenable.*

**Proof** Because $\mathrm{Sym}(p)$ and $\mathrm{Sym}(n)$ are finite groups, they are compact, and therefore amenable. Because $\mathbb{R}$ and $\mathbb{R}^+$ are Abelian, they are also amenable. By Theorem S6, group extensions of amenable groups are amenable. ■

## 7.2 Proofs of Theorems 1 through 4

This section is organized as follows. Section 7.2.1 introduces three general lemmas that will be useful in proving the results from the main text. Section 7.2.2 proves several lemmas, proves the variant of the Hunt-Stein theorem from the main text (Theorem 1), and concludes with a discussion of the relation of this result to those in Le Cam (2012). Section 7.2.3 establishes a preliminary lemma and then proves that, when the class of estimators is equivariant, it suffices to restrict attention to priors in $\Gamma_1$ when aiming to learn a $\Gamma$-minimax estimator (Theorem 2). Section 7.2.4 establishes several lemmas, including a minimax theorem for our setting, before proving the existence of an equilibrium point (Theorem 3). Section 7.2.5 establishes the equivariance of our proposed neural network architecture (Theorem 4).

In this section, we always equip $C(\mathcal{Z}, \mathbb{R})$ with the topology of compact convergence and, whenever T2 holds so that $\mathcal{S} \subset C(\mathcal{Z}, \mathbb{R})$, we equip $\mathcal{S}$ with the subspace topology. For a fixed compact $\mathcal{K} \subset \mathcal{Z}$ and a function $h \in C(\mathcal{Z}, \mathbb{R})$, we also let $\|h\|_{\infty,\mathcal{K}} := \sup_{\boldsymbol{z} \in \mathcal{K}} |h(\boldsymbol{z})|$.

### 7.2.1 Preliminary lemmas

We now prove three lemmas that will be used in our proofs of Theorems 1 and 3.

**Lemma 7** *$C(\mathcal{Z}, \mathbb{R})$ with the compact-open topology is metrizable.*

**Proof** See Example IV.2.2 in Conway (2010). ■

As a consequence of the above, we can show that a subset of $C(\mathcal{Z}, \mathbb{R})$ is closed by showing that it is sequentially closed, and we can show that a subset of $C(\mathcal{Z}, \mathbb{R})$ is continuous by showing that it is sequentially continuous.

**Lemma 8** *If T1, T2, and T3 hold, then $\mathcal{S}$ is a compact subset of $C(\mathcal{Z}, \mathbb{R})$.*

**Proof** By T1, $\mathcal{S}$ is pointwise bounded. Moreover, the local Hölder condition T2 implies that $\mathcal{S}$ is equicontinuous, in the sense that, for every $\epsilon > 0$ and every $\boldsymbol{z} \in \mathcal{Z}$ there exists an open neighborhood $\mathcal{U} \subset \mathcal{Z}$ of $\boldsymbol{z}$ such that, for all $S \in \mathcal{S}$ and all $\boldsymbol{z}' \in \mathcal{U}$, it holds that $|S(\boldsymbol{z}) - S(\boldsymbol{z}')| < \epsilon$. Hence, by the Arzelà-Ascoli theorem (see Theorem 47.1 in Munkres, 2000 for a convenient version), $\mathcal{S}$ is a relatively compact subset of $C(\mathcal{Z}, \mathbb{R})$. By T3, $\mathcal{S}$ is closed, and therefore $\mathcal{S}$ is compact. ■

We now show that the group action $\mathcal{G}_0 \times \mathcal{S} \to \mathcal{S}$ is continuous under conditions that we assume in Theorem 1. Establishing this continuity condition is necessary for our use of Day's fixed point theorem in the upcoming proof of that result.

**Lemma 9** *If T2, T4, and T5 hold, then the group action $\mathcal{G}_0 \times \mathcal{S} \to \mathcal{S}$ is continuous.*

**Proof** By T4 and T5, $\mathcal{G}_0 \times \mathcal{S} \to \mathcal{S}$ is indeed a group action. Also, by T2 and Lemma 7, $\mathcal{S}$ is metrizable. Recall the expression for $\mathcal{Z}$ given in (8) and that

$$\mathcal{G}_0 := (\mathbb{R} \rtimes \mathbb{R}^+) \times [(\mathbb{R} \rtimes \mathbb{R}^+)^p \rtimes \mathrm{Sym}(p)] \times \mathrm{Sym}(n).$$

The product topology is compatible with semidirect products, and so the fact that each multiplicand is a metric space implies that $\mathcal{G}_0$ is a metric space. Hence, it suffices to show

sequential continuity. Let $\{(g_k, S_k)\}_{k=1}^{\infty}$ be a sequence in $\mathcal{G}_0 \times \mathcal{S}$ such that $(g_k, S_k) \to (g, S)$, where $(g, S) \in \mathcal{G}_0 \times \mathcal{S}$. By the definition of the product metric, $g_k \to g$ and $S_k \to S$. Let $\mathcal{K}_1 \subseteq \mathcal{O}_n^p$, $\mathcal{K}_2 \subseteq \mathcal{O}_n$, $\mathcal{K}_3 \subset \mathbb{R}^p$, $\mathcal{K}_4 \subset \mathbb{R}^p$, $\mathcal{K}_5 \subset \mathbb{R}$, $\mathcal{K}_6 \subset \mathbb{R}^p$, and $\mathcal{K}_7 \subset \mathbb{R}$ be compact spaces. Since each compact space $\mathcal{K} \subset \mathcal{Z}$ is contained in such a $\prod_{i=1}^{7} \mathcal{K}_i$, it suffices to show that

$$\sup_{\boldsymbol{z} \in \prod_{i=1}^{7} \mathcal{K}_i} |(g_k \cdot S_k)(\boldsymbol{z}) - (g \cdot S)(\boldsymbol{z})| = \|g_k \cdot S_k - g \cdot S\|_{\infty, \prod_{i=1}^{7} \mathcal{K}_i} \to 0$$

for arbitrary compact sets $\mathcal{K}_1, \cdots, \mathcal{K}_7$. To show this, we will use the decomposition $g_k = (g_{k,1}, g_{k,2}, g_{k,3}, g_{k,4})$, where $g_{k,1} \in \mathbb{R} \rtimes \mathbb{R}^+$, $g_{k,2} \in (\mathbb{R} \rtimes \mathbb{R}^+)^p$, $g_{k,3} \in \mathrm{Sym}(p)$, and $g_{k,4} \in \mathrm{Sym}(n)$. We similarly use the decomposition $g = (g_1, g_2, g_3, g_4)$. For all $N$ large enough, all of the statements are true for all $k > N$: $g_{k,3} = g_3$, $g_{k,4} = g_4$, $g_{k,1}$ is contained in a compact neighbourhood $C_1$ of $g_1$, and $g_{k,2}$ is contained in a compact neighbourhood $C_2$ of $g_2$.

Since permutations are continuous, $g_4 \mathcal{K}_1 g_3 := \{g_4 w g_3 : w \in \mathcal{K}_1\}$, $g_4 \mathcal{K}_2 := \{g_4 w : w \in \mathcal{K}_2\}$, and $\mathcal{K}_j g_3 := \{w g_3 : w \in \mathcal{K}_j\}$, $j = 3, 4, 6$, are compact. In the following we use the decomposition $g' := (g_1', g_2', g_3', g_4')$ for an arbitrary element $g' \in \mathcal{G}$. Since addition and multiplication are continuous, $C_2 \odot (\mathcal{K}_3 g_3) := \{g_2' \cdot w : g_2' \in C_2, w \in \mathcal{K}_3 g_3\}$, $C_2 \odot (\mathcal{K}_4 g_3) := \{g_2' \cdot w : g_2' \in C_2, w \in \mathcal{K}_4 g_3\}$, $C_1 \odot \mathcal{K}_5 := \{g_1' \cdot w : g_1' \in C_1, w \in \mathcal{K}_5\}$, $C_2 \odot (\mathcal{K}_6 g_3) := \{g_2' \cdot w : g_2' \in C_2, w \in \mathcal{K}_6 g_3\}$, and $C_1 \odot \mathcal{K}_7 := \{g_1' \cdot w : g_1' \in C_1, w \in \mathcal{K}_7\}$ are compact. Define $\mathcal{K}^\circ$ to be the compact set

$$\mathcal{K}^\circ = g_4 \mathcal{K}_1 g_3 \times g_4 \mathcal{K}_2 \times C_2 \odot (\mathcal{K}_3 g_3) \times C_2 \odot (\mathcal{K}_4 g_3) \times C_1 \odot \mathcal{K}_5 \times C_2 \odot (\mathcal{K}_6 g_3) \times C_1 \odot \mathcal{K}_7$$

Then,

$$\|g_k \cdot S_k - g \cdot S\|_{\infty, \prod_{i=1}^{7} \mathcal{K}_i} \leq \|S_k - S\|_{\infty, \mathcal{K}^\circ} \to 0.$$

■

### 7.2.2 Proof of Theorem 1

We begin this subsection with four lemmas and then we prove Theorem 1. Following this proof, we briefly describe how the argument relates to that given in Le Cam (2012). In the proof of Theorem 1, we will use notation that we established about the group $\mathcal{G}_0$ in Section 7.1. We refer the reader to that section for details.

**Lemma 10** *For any $g \in \mathcal{G}_0, T \in \mathcal{T}$, and $P \in \mathcal{P}$, $R(g \cdot T, P) = R(T, g \cdot P)$*

**Proof** Fix $T \in \mathcal{T}$ and $P \in \mathcal{P}$, and let $S := S_T$, where $S_T$ is defined in (3). By the change-of-variables formula,

$$\begin{aligned} R(g \cdot T, P) &= \mathrm{E}_P \left[ \int \sigma_P^{-2} \left\{ \bar{\boldsymbol{Y}} + s(\boldsymbol{Y}) S(g \cdot \boldsymbol{Z}) - \mu_P(x_0) \right\}^2 dP_X(x_0) \right] \\ &= \mathrm{E}_{P \circ g^{-1}} \left[ \int \sigma_P^{-2} \left\{ g^{-1} \cdot \bar{\boldsymbol{Y}} + s(g^{-1} \cdot \boldsymbol{Y}) S(\boldsymbol{Z}) - \mu_P(g^{-1} \cdot x_0) \right\}^2 d(P_X \circ g^{-1})(x_0) \right]. \end{aligned}$$

Plugging the fact that $g^{-1} \cdot \boldsymbol{y} = (\boldsymbol{y} - g^{0+})/g^{0\times}$ and that

$$\mu_P(g^{-1} \cdot x_0) = \mathrm{E}_P[Y | X_0 = g^{-1} \cdot x_0] = \mathrm{E}_P[Y | g \cdot X_0 = x_0]$$

$$= \frac{\mathrm{E}_P[g \cdot Y | g \cdot X_0 = x_0] - g^{0+}}{g^{0\times}} = \frac{\mu_{P \circ g^{-1}}(x_0) - g^{0+}}{g^{0\times}}$$

into the right-hand side of the preceding display yields that

$$\begin{aligned}
&R(g \cdot T, P) \\
&= \mathrm{E}_{P \circ g^{-1}} \left[ \int \sigma_P^{-2} \left\{ \frac{\bar{\boldsymbol{Y}} - g^{0+}}{g^{0\times}} + s\left( \frac{\bar{\boldsymbol{Y}} - g^{0+}}{g^{0\times}} \right) S(\boldsymbol{Z}) - \frac{\mu_{P \circ g^{-1}}(x_0) - g^{0+}}{g^{0\times}} \right\}^2 d(P_X \circ g^{-1})(x_0) \right] \\
&= \mathrm{E}_{P \circ g^{-1}} \left[ \int \sigma_P^{-2} \left\{ \frac{\bar{\boldsymbol{Y}}}{g^{0\times}} + s\left( \frac{\bar{\boldsymbol{Y}} - g^{0+}}{g^{0\times}} \right) S(\boldsymbol{Z}) - \frac{\mu_{P \circ g^{-1}}(x_0)}{g^{0\times}} \right\}^2 d(P_X \circ g^{-1})(x_0) \right].
\end{aligned}$$

By the shift and scale properties of the standard deviation and variance, the above continues as

$$\begin{aligned}
&= \mathrm{E}_{P \circ g^{-1}} \left[ \int \sigma_P^{-2} \left\{ \frac{\bar{\boldsymbol{Y}}}{g^{0\times}} + \frac{s\left(\bar{\boldsymbol{Y}}\right)}{g^{0\times}} S(\boldsymbol{Z}) - \frac{\mu_{P \circ g^{-1}}(x_0)}{g^{0\times}} \right\}^2 d(P_X \circ g^{-1})(x_0) \right] \\
&= \mathrm{E}_{P \circ g^{-1}} \left[ \int \sigma_{P \circ g^{-1}}^{-2} \left\{ \bar{\boldsymbol{Y}} + s\left(\bar{\boldsymbol{Y}}\right) S(\boldsymbol{Z}) - \mu_{P \circ g^{-1}}(x_0) \right\}^2 d(P_X \circ g^{-1})(x_0) \right] \\
&= R(T, g \cdot P).
\end{aligned}$$

■

**Lemma 11** *For any $g \in \mathcal{G}_0$, $T \in \mathcal{T}$, and $\Pi \in \Gamma$, it holds that $r(g \cdot T, \Pi) = r(T, g \cdot \Pi)$.*

**Proof** This result follows quickly from Lemma 10. Indeed, for any $g \in \mathcal{G}_0$, $T \in \mathcal{T}$, and $\Pi \in \Gamma$,

$$\begin{aligned}
r(g \cdot T, \Pi) &= \int R(g \cdot T, P) d\Pi(P) = \int R(T, g \cdot P) d\Pi(P) \\
&= \int R(T, P) d(\Pi \circ g^{-1})(P) = r(T, g \cdot \Pi).
\end{aligned}$$

■

Let $\mathcal{S}_e := \{S \in \mathcal{S} : g \cdot S = S \text{ for all } g \in \mathcal{G}_0\}$ consists of the $\mathcal{G}_0$-invariant elements of $\mathcal{S}$. The following fact will be useful when proving Theorem 1, and also when proving results in the upcoming Section 7.2.3.

**Lemma 12** *It holds that $\mathcal{S}_e = \{S_T : T \in \mathcal{T}_e\}$.*

**Proof** Fix $S \in \mathcal{S}_e$ and $g \in \mathcal{G}_0$. By the definition of $\mathcal{S} := \{S_T : T \in \mathcal{T}\}$, there exists a $T \in \mathcal{T}$ such that $S = S_T$. For this $T$, the fact that $S_T(\boldsymbol{z}) = S_T(g \cdot \boldsymbol{z})$ implies that

$$T(g \cdot \boldsymbol{z}) = (g^{0+} + g^{0\times} \bar{\boldsymbol{y}}) + g^{0\times} s(\boldsymbol{y}) S_T(g \cdot \boldsymbol{z}) = (g^{0+} + g^{0\times} \bar{\boldsymbol{y}}) + g^{0\times} s(\boldsymbol{y}) S_T(\boldsymbol{z})$$

$$= g^{0+} + g^{0\times}[\bar{\boldsymbol{y}} + s(\boldsymbol{y})S_T(\boldsymbol{z})] = g^{0+} + g^{0\times}T(\boldsymbol{z}).$$

As $g$ was arbitrary, $T \in \mathcal{T}_e$. Hence, $\mathcal{S}_e \subseteq \{S_T : T \in \mathcal{T}_e\}$.

Now fix $T \in \mathcal{T}_e$ and $g \in \mathcal{G}_0$. Note that $S_T(\boldsymbol{z}) = [T(\boldsymbol{z}) - \bar{\boldsymbol{y}}]/s(\boldsymbol{y})$. Using that $T \in \mathcal{T}_e$ implies that $T(g \cdot \boldsymbol{z}) = g^{0+} + g^{0\times}T(\boldsymbol{z})$, we see that

$$\begin{aligned} S_T(g \cdot \boldsymbol{z}) &= \frac{T(g \cdot \boldsymbol{z}) - g^{0+} - g^{0\times}\bar{\boldsymbol{y}}}{s(g \cdot \boldsymbol{y})} = \frac{T(g \cdot \boldsymbol{z}) - g^{0+} - g^{0\times}\bar{\boldsymbol{y}}}{g^{0\times}s(\boldsymbol{y})} \\ &= \frac{g^{0+} + g^{0\times}T(\boldsymbol{z}) - g^{0+} - g^{0\times}\bar{\boldsymbol{y}}}{g^{0\times}s(\boldsymbol{y})} = \frac{T(\boldsymbol{z}) - \bar{\boldsymbol{y}}}{s(\boldsymbol{y})} = S_T(\boldsymbol{z}). \end{aligned}$$

As, $g$ was arbitrary, $S_T \in \mathcal{S}_e$, and so $\mathcal{S}_e \supseteq \{S_T : T \in \mathcal{T}_e\}$. ■

We define $r_0 : \mathcal{S} \times \Gamma \to [0, \infty)$ as follows:

$$r_0(S, \Pi) := \int \mathrm{E}_P \left[ \int_{x_0:(\boldsymbol{D},x_0)\in\mathcal{D}_0} \frac{\{\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S(z(\boldsymbol{D}, x_0)) - \mu_P(x_0)\}^2}{\sigma_P^2} dP_X(x_0) \right] d\Pi(P). \quad (9)$$

Because $\mathcal{D}_0$ occurs with $P$-probability one (for any $P \in \mathcal{P}$), it holds that $r(T, \Pi) = r_0(S_T, \Pi)$ for any $T \in \mathcal{T}$.

**Lemma 13** *Fix $\Pi \in \Gamma$. If T1, T2, and P4 hold, then $r_0(\cdot, \Pi) : \mathcal{S} \to \mathbb{R}$ is lower semicontinuous.*

**Proof** Fix $\Pi \in \Gamma$. For any compact $\mathcal{K} \subset \mathcal{Z}$, we define $f_{\mathcal{K}} : \mathcal{S} \to \mathbb{R}$ by

$$f_{\mathcal{K}}(S) := \int \mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} \sigma_P^{-2} \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 dP_X(x_0) \right] d\Pi(P),$$

where here and throughout in this proof we let $\boldsymbol{Z} := z(\boldsymbol{D}, x_0)$ and $\mathcal{X}_{\boldsymbol{D},\mathcal{K}} := \{x_0 : (\boldsymbol{D}, x_0) \in \mathcal{D}_0, z(\boldsymbol{D}, x_0) \in \mathcal{K}\} \subseteq \mathcal{X}$. Recalling that there exists an increasing sequence of compact subsets $\mathcal{K}_1 \subset \mathcal{K}_2 \subset \cdots$ such that $\bigcup_{j=1}^{\infty} \mathcal{K}_j = \mathcal{Z}$, we see that $\sup_{j\in\mathbb{N}} f_{\mathcal{K}_j}(\cdot) = r_0(\cdot, \Pi)$ by the monotone convergence theorem. Moreover, as suprema of collections of continuous functions are lower semicontinuous, we see that $f$ is lower semicontinuous if $f_{\mathcal{K}}$ is continuous for every $\mathcal{K}$. In the remainder of this proof, we will show that this is indeed the case.

By Lemma 7, it suffices to show that $f_{\mathcal{K}}$ is sequentially continuous. Fix $S_1, S_2 \in \mathcal{S}$. By Jensen's inequality,

$$\begin{aligned} &|f_{\mathcal{K}}(S_1) - f_{\mathcal{K}}(S_2)| \\ &= \left| \int \mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} \sigma_P^{-2} \Big( \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S_1(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 \right. \right. \\ &\qquad \left. \left. - \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S_2(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 \Big) dP_X(x_0) \right] d\Pi(P) \right| \\ &\leq \int \sigma_P^{-2} \, \mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} \Big| \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S_1(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 \right. \end{aligned}$$

$$- \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S_2(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 \Big| dP_X(x_0) \Bigg] d\Pi(P). \tag{10}$$

In what follows, we will bound the right-hand side above by some finite constant times $\|S_1 - S_2\|_{\mathcal{K},\infty}$. We start by noting that, for any $(\boldsymbol{d}, x_0) \in \mathcal{D}_0$ such that $z(\boldsymbol{d}, x_0) \in \mathcal{K}$,

$$\begin{aligned}
&\Big| \left[\bar{\boldsymbol{y}} + s(\boldsymbol{y})S_1(\boldsymbol{z}) - \mu_P(x_0)\right]^2 - \left[\bar{\boldsymbol{y}} + s(\boldsymbol{y})S_2(\boldsymbol{z}) - \mu_P(x_0)\right]^2 \Big| \\
&= \Big| s(\boldsymbol{y}) \left[2\bar{\boldsymbol{y}} + s(\boldsymbol{y})\{S_1(\boldsymbol{z}) + S_2(\boldsymbol{z})\} - 2\mu_P(x_0)\right] \left[S_1(\boldsymbol{z}) - S_2(\boldsymbol{z})\right] \Big| \\
&\leq \|S_1 - S_2\|_{\infty,\mathcal{K}} s(\boldsymbol{y}) \Big| 2\bar{\boldsymbol{y}} + s(\boldsymbol{y})\{S_1(\boldsymbol{z}) + S_2(\boldsymbol{z})\} - 2\mu_P(x_0) \Big| \\
&\leq \|S_1 - S_2\|_{\infty,\mathcal{K}} \left( s(\boldsymbol{y})^2 [\|S_1\|_{\mathcal{K},\infty} + \|S_2\|_{\mathcal{K},\infty}] + 2s(\boldsymbol{y}) |\bar{\boldsymbol{y}} - \mu_P(x_0)| \right) \\
&\leq \|S_1 - S_2\|_{\infty,\mathcal{K}} \left( s(\boldsymbol{y})^2 [\|S_1\|_{\mathcal{K},\infty} + \|S_2\|_{\mathcal{K},\infty}] + 2s(\boldsymbol{y}) |\bar{\boldsymbol{y}} - \mathrm{E}_P[Y]| + 2s(\boldsymbol{y}) |\mu_P(x_0) - \mathrm{E}_P[Y]| \right) \\
&\leq 2\|S_1 - S_2\|_{\infty,\mathcal{K}} \left( C_1 s(\boldsymbol{y})^2 + s(\boldsymbol{y}) |\bar{\boldsymbol{y}} - \mathrm{E}_P[Y]| + s(\boldsymbol{y}) |\mu_P(x_0) - \mathrm{E}_P[Y]| \right),
\end{aligned}$$

where $C_1 := \sup_{S \in \mathcal{S}} \|S\|_{\mathcal{K},\infty}$ is finite by T1 and T2. Integrating both sides shows that

$$\begin{aligned}
&\mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} \Big| \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S_1(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 - \left[\bar{\boldsymbol{Y}} + s(\boldsymbol{Y})S_2(\boldsymbol{Z}) - \mu_P(x_0)\right]^2 \Big| dP_X(x_0) \right] \\
&\quad \leq 2\|S_1 - S_2\|_{\infty,\mathcal{K}} \Bigg( C_1 \, \mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} s(\boldsymbol{Y})^2 dP_X(x_0) \right] + \mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} s(\boldsymbol{Y}) |\bar{\boldsymbol{Y}} - \mathrm{E}_P[Y]| dP_X(x_0) \right] \\
&\qquad\qquad + \mathrm{E}_P \left[ \int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}} s(\boldsymbol{Y}) |\mu_P(x_0) - \mathrm{E}_P[Y]| dP_X(x_0) \right] \Bigg) \\
&\quad \leq 2\|S_1 - S_2\|_{\infty,\mathcal{K}} \Bigg( C_1 \, \mathrm{E}_P \left[ s(\boldsymbol{Y})^2 \right] + \mathrm{E}_P \left[ s(\boldsymbol{Y}) |\bar{\boldsymbol{Y}} - \mathrm{E}_P[Y]| \right] \\
&\qquad\qquad + \mathrm{E}_P \left[ s(\boldsymbol{Y}) \int |\mu_P(x_0) - \mathrm{E}_P[Y]| dP_X(x_0) \right] \Bigg).
\end{aligned} \tag{11}$$

We now bound the three expectations on the right-hand side by finite constants that do not depend on $S_1$ or $S_2$. All three bounds make use of the bound on the first expectation, namely $\mathrm{E}_P \left[ s(\boldsymbol{Y})^2 \right] = \frac{n-1}{n} \mathrm{Var}_P(Y) \leq \frac{n-1}{n} C_2 \sigma_P^2$, where $C_2 := \sup_{P \in \mathcal{P}} \mathrm{Var}_P(Y)/\sigma_P^2$. We note that (P4) can be used to show that $C_2 < \infty$. Indeed,

$$\mathrm{E}_P \left[ \mathrm{Var}_P(Y \mid X) \right] = \mathrm{E}_P \left[ \mathrm{Var}_P(\epsilon_P \mid X) \right] = \mathrm{E}_P[\epsilon_P^2] = \sigma_P^2,$$

and so, by the law of total variance and (P4), $C_2 = 1 + \sup_{P \in \mathcal{P}} \mathrm{Var}_P(\mu_P(X))/\sigma_P^2 < \infty$. By Cauchy-Schwarz, the second expectation on the right-hand side of (11) bound as

$$\begin{aligned}
\mathrm{E}_P \left[ s(\boldsymbol{Y}) |\boldsymbol{Y} - \mathrm{E}_P[Y]| \right] &\leq \mathrm{E}_P \left[ s(\boldsymbol{Y})^2 \right]^{1/2} \mathrm{E}_P \left[ \{\boldsymbol{Y} - \mathrm{E}_P[Y]\}^2 \right]^{1/2} = \mathrm{E}_P \left[ s(\boldsymbol{Y})^2 \right]^{1/2} \sigma_P \\
&= \sqrt{\frac{n-1}{n}} \sqrt{C_2} \sigma_P^2,
\end{aligned}$$

and the third expectation bounds as

$$\begin{aligned}\mathrm{E}_P\left[s(\boldsymbol{Y})|\mu_P(x_0)-\mathrm{E}_P[Y]|\right] &\leq \mathrm{E}_P\left[s(\boldsymbol{Y})^2\right]^{1/2}\mathrm{E}_P\left[\int\{\mu_P(x_0)-\mathrm{E}_P[Y]\}^2 dP_{X_0}\right]^{1/2}\\ &\leq \mathrm{E}_P\left[s(\boldsymbol{Y})^2\right]^{1/2}\mathrm{Var}_P(Y)^{1/2} \leq \sqrt{\frac{n-1}{n}}\sqrt{C_2}\sigma_P\mathrm{Var}_P(Y)^{1/2}\\ &\leq \sqrt{\frac{n-1}{n}}C_2\sigma_P^2.\end{aligned}$$

Plugging these bounds into (11), we see that

$$\begin{aligned}&\mathrm{E}_P\left[\int_{\mathcal{X}_{\boldsymbol{D},\mathcal{K}}}\left|\left[\bar{\boldsymbol{Y}}+s(\boldsymbol{Y})S_1(\boldsymbol{Z})-\mu_P(x_0)\right]^2-\left[\bar{\boldsymbol{Y}}+s(\boldsymbol{Y})S_2(\boldsymbol{Z})-\mu_P(x_0)\right]^2\right|dP_X(x_0)\right]\\ &\leq 2\|S_1-S_2\|_{\infty,\mathcal{K}}\sigma_P^2\sqrt{\frac{n-1}{n}}C_2^{1/2}\left(C_1C_2^{1/2}\sqrt{\frac{n-1}{n}}+C_2^{1/2}+1\right).\end{aligned}$$

Plugging this into (10), we have shown that

$$|f_{\mathcal{K}}(S_1)-f_{\mathcal{K}}(S_2)| \leq 2\|S_1-S_2\|_{\infty,\mathcal{K}}\sqrt{\frac{n-1}{n}}C_2^{1/2}\left(C_1C_2^{1/2}\sqrt{\frac{n-1}{n}}+C_2^{1/2}+1\right).$$

We now conclude the proof by showing that the above implies that $f_{\mathcal{K}}$ is sequentially continuous at every $S\in\mathcal{S}$, and therefore is sequentially continuous on $\mathcal{S}$. Fix $S$ and a sequence $\{S_j\}$ such that $S_j\to S$ compactly. This implies that $\|S_j-S\|_{\infty,\mathcal{K}}\to 0$, and so the above display implies that $f_{\mathcal{K}}(S_j)\to f_{\mathcal{K}}(S)$, as desired. ■

We now prove Theorem 1.

**Proof of Theorem 1** Fix $T_0\in\mathcal{T}$ and let $S_0:=S_{T_0}\in\mathcal{S}$. Let $\mathcal{K}$ be the set of all elements $S\in\mathcal{S}$ that satisfy

$$\sup_{\Pi\in\Gamma} r_0(S,\Pi) \leq \sup_{\Pi\in\Gamma} r_0(S_0,\Pi).$$

For fixed $\Pi_0\in\Gamma$, the set of $S\in\mathcal{S}$ that satisfy $r_0(S,\Pi_0)\leq\sup_{\Pi\in\Gamma} r_0(S_0,\Pi)$ is closed due to the lower semicontinuity of the risk function (Lemma 13) and contains $S_0$. The intersection of such sets is closed and contains $S_0$ so that $\mathcal{K}$ is a nonempty closed subset of the compact Hausdorff set $\mathcal{S}$, implying that $\mathcal{K}$ is compact. By the convexity of $x\mapsto\left(\frac{x-a}{b}\right)^2$, the risk function $S\mapsto r_0(S,\Pi)$ is convex. Hence, $\mathcal{K}$ is convex. If $S\in\mathcal{K}$, then Lemma 11 shows that, for any $g\in\mathcal{G}_0$,

$$r_0(g\cdot S,\Pi_0)=r_0(S,g\cdot\Pi_0)\leq\sup_{\Pi\in\Gamma} r_0(S_0,\Pi).$$

Thus, $g\cdot S\in\mathcal{K}$ and $\mathcal{G}_0\times\mathcal{K}\to\mathcal{K}$ is an affine group action on a nonempty, convex, compact subset of a locally compact topological vector space. Combining this with the fact that $\mathcal{G}_0$ is amenable (Lemma 6) shows that we may apply Day's fixed point theorem (Theorem S3) to see that there exists an $S_e\in\mathcal{S}$ such that, for all $g\in\mathcal{G}_0, g\cdot S_e=S_e$ and

$$\sup_{\Pi\in\Gamma} r_0(S_e,\Pi)\leq\sup_{\Pi\in\Gamma} r_0(S_0,\Pi).$$

The conclusion is at hand. By Lemma 12, there exists a $T_e \in \mathcal{T}_e$ such that $S_e = S_{T_e}$. Furthermore, as noted below (9), $r_0(S_{T_e}, \Pi) = r(T_e, \Pi)$ and $r_0(S_{T_0}, \Pi) = r(T_0, \Pi)$ for all $\Pi \in \Gamma$. Recalling that $S_0 := S_{T_0}$, the above shows that $\sup_{\Pi\in\Gamma} r(T_e, \Pi) \leq \sup_{\Pi\in\Gamma} r(T_0, \Pi)$. As $T_0 \in \mathcal{T}$ was arbitrary and $T_e \in \mathcal{T}_e$, we have shown that $\inf_{T_e\in\mathcal{T}_e} \sup_{\Pi\in\Gamma} r(T_e, \Pi) \leq \inf_{T_0\in\mathcal{T}} \sup_{\Pi\in\Gamma} r(T_0, \Pi)$. ■

The proof of Theorem 1 is inspired by that of the Hunt-Stein theorem given in Le Cam (2012). Establishing this result in our context required making meaningful modifications to these earlier arguments. Indeed, Le Cam (2012) uses transitions, linear maps between L-spaces, to characterize the space of decision procedures. This more complicated machinery makes it possible to broaden the set of procedures under consideration. Indeed, with this characterization, it is possible to describe decision procedures that cannot even be represented as randomized decision procedures via a Markov kernel, but instead come about as limits of such decision procedures. Despite the richness of the space of decision procedures considered, Le Cam is still able to show that this space is compact by using a coarse topology, namely the topology of pointwise convergence. Unfortunately, this topology appears to generally be too coarse for our Bayes risk function $r_0(\cdot, \Pi)$ to be lower semi-continuous, which is a fact that we used at the beginning of our proof of Theorem 1. Another disadvantage to this formulation is that it makes it difficult to enforce any natural conditions or structure, such as continuity, on the set of estimators. It is unclear whether it would be possible to implement a numerical strategy optimizing over a class of estimators that lacks such structure. In contrast, we showed that, under appropriate conditions, it is indeed possible to prove a variant of the Hunt-Stein theorem in our setting even once natural structure is imposed on the class of estimators. To show the compactness of the space of estimators that we consider, we applied the Arzelà-Ascoli theorem.

#### 7.2.3 Proof of Theorem 2

We provide one additional lemma before proving Theorem 2. The lemma relates to the class $\widetilde{\mathcal{T}}_e$ of estimators in $\mathcal{T}$ that satisfy the equivariance property (5) but do not necessarily satisfy (4). Note that $\mathcal{T}_e \subseteq \widetilde{\mathcal{T}}_e \subseteq \mathcal{T}$.

**Lemma 14** *If P2 and P3 hold, then, for all $T \in \widetilde{\mathcal{T}}_e$,*

$$r(T, \Pi) = r(T, \Pi \circ h^{-1}) \quad \text{for all } \Pi \in \Gamma,$$

*and so* $\sup_{\Pi\in\Gamma} r(T, \Pi) = \sup_{\Pi\in\Gamma_1} r(T, \Pi)$.

**Proof of Lemma 14** Let $e$ be the identity element in $\mathrm{Sym}(n) \times \mathrm{Sym}(p)$. For each $P \in \mathcal{P}$, define $g_P \in \mathcal{G}_0$ to be

$$g_P := \left( -\frac{E_P[Y]}{\sigma_P}, \frac{1}{\sigma_P}, \left( -\frac{E_P[X_j]}{\sqrt{\mathrm{Var}_P(X_j)}} \right)_{j=1}^{p}, \left( \frac{1}{\sqrt{\mathrm{Var}_P(X_j)}} \right)_{j=1}^{p}, e \right).$$

It holds that

$$R(T, \Pi \circ h^{-1}) = \int R(T, P) d(\Pi \circ h^{-1})(P)$$

$$= \int R(T, P \circ g_P^{-1}) d\Pi(P) \quad \text{by the definition of } h$$
$$= \int R(g_P \cdot T, P) d\Pi(P) \quad \text{by Lemma 10}$$
$$= \int R(T, P) d\Pi(P) = r(T, \Pi) \quad \text{since } T \in \tilde{\mathcal{T}}_e.$$

■

We conclude by proving Theorem 2.

**Proof of Theorem 2** Under the conditions of the theorem, $\widetilde{\mathcal{T}}_e = \mathcal{T}$. Recalling that $\Gamma_1 := \{\Pi \circ h^{-1} : \Pi \in \Gamma\}$, Lemma 14 yields that, for any $T \in \mathcal{T}$, $\sup_{\Pi \in \Gamma} r(T, \Pi) = \sup_{\Pi \in \Gamma} r(T, \Pi \circ h^{-1}) = \sup_{\Pi \in \Gamma_1} r(T, \Pi)$. Hence, an estimator $T \in \mathcal{T}$ is $\Gamma$-minimax if and only if it is $\Gamma_1$-minimax. ■

### 7.2.4 Proof of Theorem 3

In this subsection, we assume (without statement) that all $\Pi \in \Gamma$ are defined on the measurable space $(\mathcal{P}, \mathscr{A})$, where $\mathscr{A}$ is such that $\{A \cap \mathcal{P}_1 : A \in \mathscr{A}\}$ equals $\mathscr{B}_1$, where $\mathscr{B}_1$ is the collection of Borel sets on the metric space $(\mathcal{P}_1, \rho)$ described in P5. Under P2 and P3, which we also assume without statement throughout this subsection, it then follows that each $\Pi_1 \in \Gamma_1$ is defined on the measurable space $(\mathcal{P}_1, \mathscr{B}_1)$, where $\mathscr{B}_1$ is the collection of Borel sets on $(\mathcal{P}_1, \rho)$. Let $\Gamma_0$ denote the collection of all distributions on $(\mathcal{P}_1, \mathscr{B}_1)$. For each $A \in \mathscr{B}_1$, define the $\epsilon$-enlargement of $A$ by $A^\epsilon := \{P \in \mathcal{P}_1 : \exists P' \in A \text{ such that } \rho(P, P') < \epsilon\}$. Further let $\xi$ denote the Lévy-Prokhorov metric on $\Gamma_0$, namely

$$\xi(\Pi, \Pi') := \inf\left\{\epsilon > 0 : \Pi(A) \leq \Pi'(A^\epsilon) + \epsilon \text{ and } \Pi'(A) \leq \Pi(A^\epsilon) + \epsilon \text{ for all } A \in \mathscr{B}_1\right\}.$$

**Lemma 15** *If P5 and P6, then $(\Gamma_1, \xi)$ is a compact metric space.*

**Proof of Lemma 15** By Prokhorov's theorem (see Theorem 5.2 in van Gaans, 2003 for a convenient version, or see Theorems 1.5.1 and 1.6.8 in Billingsley, 1999), P5 implies that $\Gamma_1$ is relatively compact in $(\Gamma_0, \xi)$. The fact that $\Gamma_1$ is closed (P6) implies the result. ■

We now define $r_1 : \mathcal{S}_e \times \Gamma_1 \to [0, \infty)$, which is the analogue of $r_0 : \mathcal{S} \times \Gamma \to [0, \infty)$ from Section 7.2.2:

$$r_1(S, \Pi) := \int \mathrm{E}_P \left[ \int_{x_0 : (\boldsymbol{D}, x_0) \in \mathcal{D}_0} \{\bar{\boldsymbol{Y}} + s(\boldsymbol{Y}) S(z(\boldsymbol{D}, x_0)) - \mu_P(x_0)\}^2 dP_X(x_0) \right] d\Pi(P). \quad (12)$$

Note that, because each distribution in $\mathcal{P}$ is continuous, each distribution in $\mathcal{P}_1$ is also continuous. Hence, $\mathcal{D}_0$ occurs with $P$-probability one for all $P \in \mathcal{P}_1$, and so the definition of $r_1$ combined with Lemma 12 shows that $r(T, \Pi) = r_1(S_T, \Pi)$ for any $T \in \mathcal{T}_e$ and $\Pi \in \Gamma_1$.

**Lemma 16** *If P5, then, for each $S \in \mathcal{S}_e$, $r_1(S, \cdot)$ is upper semicontinuous on $(\Gamma_1, \xi)$.*

**Proof of Lemma 16** Fix $S \in \mathcal{S}_e$, and note that, by Lemma 12, there exists a $T \in \mathcal{T}_e$ such that $S = S_T$. Let $\{\Pi_j\}_{j=1}^{\infty}$ be such that $\Pi_j \overset{k\to\infty}{\longrightarrow} \Pi$ in $(\Gamma_1, \xi)$ for some $\Pi \in \Gamma_1$. Because $\xi$ metrizes weak convergence (Theorem 1.6.8 in Billingsley, 1999), the Portmanteau theorem shows that $\limsup_{k\to\infty} \mathrm{E}_{\Pi_j}[f(P)] \leq \mathrm{E}_{\Pi}[f(P)]$ for every $f : \mathcal{P}_1 \to \mathbb{R}$ that is upper semicontinuous and bounded from above on $(\mathcal{P}_1, \rho)$. By part (iii) of P5, we can apply this result at $f : P \mapsto R(T, P)$ to see that $\limsup_{k\to\infty} r(T, \Pi_j) \leq r(T, \Pi)$. As $\{\Pi_j\}_{j=1}^{\infty}$ was arbitrary, $r(T, \cdot)$ is upper semicontinuous on $(\Gamma_1, \xi)$. Because $r(T, \cdot) = r_1(S_T, \cdot)$ and $S = S_T$, we have this shown that $r_1(S, \cdot)$ is upper semicontinuous on $(\Gamma_1, \xi)$. ■

**Lemma 17** *Under the conditions of Lemma 8, $\mathcal{S}_e$ is a compact subset of $C(\mathcal{Z}, \mathbb{R})$.*

**Proof** By Lemma 8, $\mathcal{S}_e \subset \mathcal{S}$ is relatively compact. Hence, it suffices to show that $\mathcal{S}_e$ is closed. By Lemma 7, a subset of $C(\mathcal{Z}, \mathbb{R})$ is closed in the topology of compact convergence if it is sequentially closed. Let $\{S_j\}_{j=1}^{\infty}$ be a sequence on $\mathcal{S}_e$ such that $S_j \to S$ compactly. Because $\mathcal{S}_e \subset \mathcal{S}$ and $\mathcal{S}$ is closed by T3, we see that $S \in \mathcal{S}$. We now wish to show that $S \in \mathcal{S}_e$. Fix $\boldsymbol{z} \in \mathcal{Z}$ and $g \in \mathcal{G}_0$. Because the doubleton set $\{\boldsymbol{z}, g \cdot \boldsymbol{z}\}$ is compact, $S_j(\boldsymbol{z}) \to S(\boldsymbol{z})$ and $S_j(g \cdot \boldsymbol{z}) \to S(g \cdot \boldsymbol{z})$, and thus $S_j(\boldsymbol{z}) - S_j(g \cdot \boldsymbol{z}) \to S(\boldsymbol{z}) - S(g \cdot \boldsymbol{z})$. Moreover, because $S_j \in \mathcal{S}_e$, $S_j(g \cdot \boldsymbol{z}) = S_j(\boldsymbol{z})$ for all $j$. Hence, $S_j(\boldsymbol{z}) - S_j(g \cdot \boldsymbol{z}) \to 0$. As these two limits must be equal, we see that $S(\boldsymbol{z}) = S(g \cdot \boldsymbol{z})$. Because $\boldsymbol{z} \in \mathcal{Z}$ and $g \in \mathcal{G}_0$ were arbitrary, $S \in \mathcal{S}_e$. ■

**Lemma 18** *Fix $\Pi \in \Gamma_1$. If T1, T2, and P4 hold, then $r_1(\cdot, \Pi) : \mathcal{S}_e \to \mathbb{R}$ is lower semicontinuous.*

**Proof** The proof is similar to that of Lemma 13 and is therefore omitted. ■

**Lemma 19** *If T6, then $\mathcal{S}_e$ is convex.*

**Proof** Fix $S_1, S_2 \in \mathcal{S}_e$ and $\delta \in (0, 1)$. For any $\boldsymbol{z} \in \mathcal{Z}$ and $g \in \mathcal{G}_0$,

$$g \cdot (\delta S_1 + [1-\delta]S_2)(\boldsymbol{z}) = \delta S_1(g \cdot \boldsymbol{z}) + [1-\delta]S_2(g \cdot \boldsymbol{z}) = \delta S_1(\boldsymbol{z}) + [1-\delta]S_2(\boldsymbol{z}),$$

where the latter equality holds since $S_1, S_2 \in \mathcal{S}_e$. Hence, $g \cdot (\delta S_1 + [1-\delta]S_2) = \delta S_1 + [1-\delta]S_2$ for all $g \in \mathcal{G}_0$. By T6, $\delta S_1 + [1-\delta]S_2 \in \mathcal{S}$. Hence, $\delta S_1 + [1-\delta]S_2 \in \mathcal{S}_e := \{S \in \mathcal{S} : g \cdot S = S \text{ for all } g \in \mathcal{G}_0\}$. ■

**Lemma 20 (Minimax theorem)** *Under the conditions of Theorem 3,*

$$\min_{S \in \mathcal{S}_e} \max_{\Pi \in \Gamma_1} r_1(S, \Pi) = \max_{\Pi \in \Gamma_1} \min_{S \in \mathcal{S}_e} r_1(S, \Pi). \tag{13}$$

**Proof of Lemma 20** We will show that the conditions of Theorem 1 in Fan (1953) are satisfied. By Lemma 7, $C(\mathcal{Z}, \mathbb{R})$ is metrizable by some metric $\rho_0$. By Lemma 17, $(\mathcal{S}_e, \rho_0)$ is a compact metric space. Moreover, by Lemma 15, $(\Gamma_1, \xi)$ is a compact metric space. As all metric spaces are Hausdorff, $(\mathcal{S}_e, \rho_0)$ and $(\Gamma_1, \xi)$ are Hausdorff. By Lemma 16, for each for each $S \in \mathcal{S}_e$, $r_1(S, \cdot)$ is upper semicontinuous on $(\Gamma_1, \xi)$. By Lemma 18, for each $\Pi \in \Gamma_1$, $r_1(\cdot, \Pi)$ is lower semicontinuous on $(\mathcal{S}_e, \rho_0)$. It remains to show that $r_1$ is concavelike on $\Gamma_1$ (called "concave on" $\Gamma_1$ by Fan) and that $r_1$ is convexlike on $\mathcal{S}_e$ (called "convex on" $\mathcal{S}_e$ by Fan). To see that $r_1$ is concavelike on $\Gamma_1$, note that $\Gamma_1$ is convex (P7), and also that, for all $S \in \mathcal{S}_e$, $r_1(S, \cdot)$ is linear, and therefore concave, on $\Gamma_1$. Hence, $r_1$ is concavelike on $\Gamma_1$ (page 409 of Terkelsen, 1973). To see that $r_1$ is convexlike on $\mathcal{S}_e$, note that $\mathcal{S}_e$ is convex (Lemma 19), and also that, for all $\Pi \in \Gamma_1$, $r_1(\cdot, \Pi)$ is convex on $\mathcal{S}_e$. Hence, $r_1$ is convexlike on $\mathcal{S}_e$ (ibid.). Thus, by Theorem 1 in Fan (1953), (13) holds. ■

We conclude by proving Theorem 3.

**Proof of Theorem 3** We follow arguments given on page 93 of Chang (2006) to show that, under the conditions of this theorem, (13) implies that there exists an $S^\star \in \mathcal{S}_e$ and a $\Pi^\star \in \overline{\Gamma}_1$ such that

$$\max_{\Pi \in \Gamma_1} r_1(S^\star, \Pi) = r_1(S^\star, \Pi^\star) = \min_{S \in \mathcal{S}_e} r_1(S, \Pi^\star). \tag{14}$$

Noting that pointwise maxima of lower semicontinuous functions are themselves lower semicontinuous, Lemma 18 implies that $\max_{\Pi \in \Gamma_1} r_1(\cdot, \Pi)$ is lower semicontinuous. Because $\mathcal{S}_e$ is compact (Lemma 17), there exists an $S^\star \in \mathcal{S}_e$ such that

$$\max_{\Pi \in \Gamma_1} r_1(S^\star, \Pi) = \min_{S \in \mathcal{S}_e} \max_{\Pi \in \Gamma_1} r_1(S, \Pi).$$

Similarly, Lemma 16 implies that $\min_{S \in \mathcal{S}_e} r_1(S, \cdot)$ is upper semicontinuous on $(\Gamma_1, \xi)$. Because $(\Gamma_1, \xi)$ is compact (Lemma 15), there exists a $\Pi^\star \in \Gamma_1$ such that

$$\min_{S \in \mathcal{S}_e} r_1(S, \Pi^\star) = \max_{\Pi \in \Gamma_1} \min_{S \in \mathcal{S}_e} r_1(S, \Pi).$$

By Lemma 20, the above two displays show that $\max_{\Pi \in \Gamma_1} r_1(S^\star, \Pi) = \min_{S \in \mathcal{S}_e} r_1(S, \Pi^\star)$. Combining this result with the elementary fact that $\min_{S \in \mathcal{S}_e} r_1(S, \Pi^\star) \leq r_1(S^\star, \Pi^\star) \leq \max_{\Pi \in \Gamma_1} r_1(S^\star, \Pi)$ shows that (14) holds.

Recall from below (12) that $r_1(S_T, \Pi) = r(T, \Pi)$ for all $\Pi \in \Gamma_1$ and $T \in \mathcal{T}_e$. Moreover, since $\mathcal{S}_e = \{S_T : T \in \mathcal{T}_e\}$ (Lemma 12), there exists a $T^\star \in \mathcal{T}_e$ such that $S = S_{T^\star}$. Combining these observations shows that (i) $\max_{\Pi \in \Gamma_1} r_1(S^\star, \Pi) = \max_{\Pi \in \Gamma_1} r_1(S_{T^\star}, \Pi) = \max_{\Pi \in \Gamma_1} r(T^\star, \Pi)$; (ii) $r_1(S^\star, \Pi^\star) = r_1(S_{T^\star}, \Pi^\star) = r(T^\star, \Pi^\star)$; and (iii) $\min_{S \in \mathcal{S}_e} r_1(S, \Pi^\star) = \min_{T \in \mathcal{T}_e} r_1(S_T, \Pi^\star) = \min_{T \in \mathcal{T}_e} r(T, \Pi^\star)$. Hence, by (14), $\max_{\Pi \in \Pi^\star} r(T^\star, \Pi) = r_1(T^\star, \Pi^\star) = \min_{T \in \mathcal{T}_e} r_1(T, \Pi^\star)$. Equivalently, for all $T \in \mathcal{T}_e$ and $\Pi \in \Gamma_1$, $r(T^\star, \Pi) \leq r(T^\star, \Pi^\star) \leq r(T, \Pi^\star)$. ■

### 7.2.5 Proof of Theorem 4

**Proof of Theorem 4** Fix $T \in \mathcal{M}$, and let $(m_1, m_2, m_3, m_4) \in \prod_{k=1}^4 \mathcal{M}_k$ be the corresponding modules. Recall from Algorithm 2 that, for a given $(\boldsymbol{d}, x_0)$, $x_0^0 := \frac{x_0 - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}$ and $\boldsymbol{d}^0 \in \mathbb{R}^{n \times p \times 2}$

is defined so that $\boldsymbol{d}^0_{i*1} = \frac{x_i - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}$ for all $i = 1, \ldots, n$ and $\boldsymbol{d}^0_{*j2} = \frac{\boldsymbol{y} - \bar{\boldsymbol{y}}}{s(\boldsymbol{y})}$ for all $j = 1, \ldots, p$. Now, for any $(\boldsymbol{d}, x_0) \in \mathcal{D}_0$,

$$T(\boldsymbol{d})(x_0) = \bar{\boldsymbol{y}} + s(\boldsymbol{y}) m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, x_0^0 \right] \right)_{j*} \right),$$

and so $S_T$ takes the form

$$S_T(z(\boldsymbol{d}, x_0)) = m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, x_0^0 \right] \right)_{j*} \right).$$

Because $S_T$ does not depend on the last four arguments of $z(\boldsymbol{d}, x_0)$, we know that $T$ satisfies (5), that is, is invariant to shifts and rescalings of the features and is equivariant to shifts and rescalings of the outcome. It remains to show permutation invariance , namely (4). By the permutation invariance of the sample mean and sample standard deviation, it suffices to establish the analogue of this property for $S_T$, namely that $S_T(z(A\boldsymbol{d}B, Bx_0)) = S_T(z(\boldsymbol{d}, x_0))$ for all $(\boldsymbol{d}, x_0) \in \mathcal{D}_0$, $A \in \mathcal{A}$, and $B \in \mathcal{B}$. For an array $M$ of size $\mathbb{R}^{n \times p \times o}$, we will write $AMB$ to mean the $\mathbb{R}^{n \times p \times o}$ array for which $(AMB)_{**\ell} = AM_{**\ell}B$ for all $\ell = 1, 2, \ldots, o$. Note that

$$\begin{aligned}
S_T(z(A\boldsymbol{d}B, Bx_0)) &= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(A\boldsymbol{d}^0 B)_{i**} \right) \,\middle|\, B^\top x_0^0 \right] \right)_{j*} \right) \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} (A m_1(\boldsymbol{d}^0) B)_{i**} \right) \,\middle|\, B^\top x_0^0 \right] \right)_{j*} \right) \qquad \text{(by M1)} \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( B^\top \frac{1}{n} \sum_{i=1}^{n} (A m_1(\boldsymbol{d}^0))_{i**} \right) \,\middle|\, B^\top x_0^0 \right] \right)_{j*} \right) \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( B^\top \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, B^\top x_0^0 \right] \right)_{j*} \right) \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ B^\top m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, B^\top x_0^0 \right] \right)_{j*} \right) \qquad \text{(by M2)} \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( B^\top \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, x_0^0 \right] \right)_{j*} \right) \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} \left( B^\top m_3 \left( \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, x_0^0 \right] \right) \right)_{j*} \right) \qquad \text{(by M3)} \\
&= m_4 \left( \frac{1}{p} \sum_{j=1}^{p} m_3 \left( \left[ m_2 \left( \frac{1}{n} \sum_{i=1}^{n} m_1(\boldsymbol{d}^0)_{i**} \right) \,\middle|\, x_0^0 \right] \right)_{j*} \right)
\end{aligned}$$

$$= S_T(z(\boldsymbol{d}, x_0)).$$

Hence, $T$ satisfies (4). ■

## 8. Extensions and Discussion

We have focused on a particular set of invariance properties on the collection of priors $\Gamma$, namely P1-P3. Our arguments can be generalized to handle other properties. As a simple example, suppose P3 is strengthened so that $\Gamma$ is invariant to nonzero (rather than only nonnegative) rescalings $\tilde{b}$ of the outcome – this property is in fact satisfied in all of our experiments. Under this new condition, the results in Section 2 remain valid with the definition of the class of equivariant estimators $\mathcal{T}_e$ defined in (4) and (5) modified so that $\tilde{b}$ may range over $\mathbb{R}\backslash\{0\}$. Moreover, for any $T$, Jensen's inequality shows that the $\Gamma$-maximal risk of the symmetrized estimator that averages $T(\boldsymbol{x}, \boldsymbol{y})(x_0)$ and negative $T(\boldsymbol{x}, -\boldsymbol{y})(x_0)$ is no worse than that of $T$. To assess the practical utility of this observation, we numerically evaluated the performance of symmetrizations of the estimators learned in our experiments. Symmetrizing improved performance across most settings (see Appendix F). We, therefore, recommend carefully characterizing the invariance properties of a given problem when setting out to meta-learn an estimator.

Much of this work has focused on developing and studying a framework for meta-learning a $\Gamma$-minimax estimator for a single, prespecified collection of priors $\Gamma$. In some settings, it may be difficult to *a priori* specify a single such collection that is both small enough so that the $\Gamma$-minimax estimator is not too conservative while also being rich enough so that the priors in this collection actually place mass in a neighborhood of the true data-generating distribution. Two approaches for overcoming this challenge seem to warrant further consideration. The first would be to employ an empirical Bayes approach (Efron and Morris, 1972), wherein a large dataset from a parallel situation can be used to inform about the possible forms that the prior might take; this, in turn, would also inform about the form that the collection $\Gamma$ should take. Recent advances in the development of empirical Bayes priors for prediction problems can be used if this approach is taken (e.g., Nabi et al., 2020). The second approach involves using AMC to approximate $\Gamma$-minimax estimators over various choices of $\Gamma$, and then to use a stacked ensemble to combine the predictions from these various base estimators. In our data experiments, we saw that a simple version of this ensemble that combined four base AMC estimators consistently performed at least as well as the best of these base estimators.

In this work, we have focused on the case where the problem of interest is a supervised learning problem and the objective is to predict a continuous outcome based on iid data. While the AMC algorithm generalizes naturally to a variety of other sampling schemes and loss functions (see Luedtke et al., 2020), our characterization of the equivariance properties of an optimal estimator was specific to the iid regression setting that we considered. In future work, it would be interesting to characterize these properties in greater generality, including in classification settings and inverse reinforcement learning settings (e.g., Russell, 1998; Geng et al., 2020).

# Appendices

## Appendix A. Review of amenability

In this appendix, we review the definition of an amenable group, an important implication of amenability, and also some sufficient conditions for establishing that a group is amenable. This material will prove useful in our proof of Theorem 1 (see Section 7.2.2). We refer the reader to Pier (1984) for a thorough coverage of amenability.

**Definition S1 (Amenability)** *Let $\mathcal{G}$ be a locally compact, Hausdorff group and let $L^\infty(\mathcal{G})$ be the space of Borel measurable functions that are essentially bounded with respect to the Haar measure. A mean on $L^\infty(\mathcal{G})$ is defined as a linear functional $M \in L^\infty(\mathcal{G})^*$ such that $M(\lambda) \geq 0$ whenever $\lambda \geq 0$ and $M(1_\mathcal{G}) = 1$. A mean $M$ is said to be left invariant for a group $\mathcal{G}$ if and only if $M(\delta_g * \lambda) = M(\lambda)$ for all $\lambda \in L^\infty(\mathcal{G})$, where $(\delta_g * \lambda)(h) = \lambda(g^{-1}h)$. The group $\mathcal{G}$ is said to be amenable if and only if there is a left invariant mean on $L^\infty(\mathcal{G})$.*

We now introduce the fixed point property, and subsequently present a result showing its close connection to the definition given above. Throughout this work, we equip all group actions $\mathcal{G} \times \mathcal{W} \to \mathcal{W}$ with the product topology.

**Definition S2 (Fixed point property)** *We say that a locally compact, Hausdorff group $\mathcal{G}$ has the fixed point property if, whenever $\mathcal{G}$ acts affinely on a compact convex set $\mathcal{K}$ in a locally convex topological vector space $E$ with the map $\mathcal{G} \times \mathcal{K} \to \mathcal{K}$ continuous, there is a point in $x_0 \in \mathcal{K}$ fixed under the action of $\mathcal{G}$.*

**Theorem S3 (Day's Fixed Point Theorem)** *A locally compact, Hausdorff group $\mathcal{G}$ has the fixed point property if and only if $\mathcal{G}$ is amenable.*

**Proof** See the proof of Theorem 5.4 in Pier (1984). ■

The following results are useful for establishing amenability.

**Lemma S4** *Any compact group is amenable.*

**Proof** Take the normalized Haar measure as an invariant mean. ■

**Lemma S5** *Any locally compact Abelian group is amenable.*

**Proof** See the proof of Proposition 12.2 in Pier (1984). ■

**Lemma S6** *Let $\mathcal{G}$ be a locally compact group and $\mathcal{N}$ a closed normal subgroup of $\mathcal{G}$. If $\mathcal{N}$ and $\mathcal{G}/\mathcal{N}$ are amenable, then $\mathcal{G}$ is amenable.*

**Proof** Assume that a continuous affine action of $\mathcal{G}$ on a nonempty compact convex set $\mathcal{K}$ is given. Let $\mathcal{K}^{\mathcal{N}}$ be the set of all fixed points of $\mathcal{N}$ in $\mathcal{K}$. Since $\mathcal{N}$ is amenable, Theorem S3 implies that $\mathcal{K}^{\mathcal{N}}$ is nonempty. Since the group action is continuous, $\mathcal{K}^{\mathcal{N}}$ is a closed subset of $\mathcal{K}$ and hence is compact. Since the action is affine, $\mathcal{K}^{\mathcal{N}}$ is convex. Now, note that, for all $x \in \mathcal{K}^{\mathcal{N}}$, $g \in \mathcal{G}$, and $n \in \mathcal{N}$, the fact that $g^{-1}ng \in \mathcal{N}$ implies that $g^{-1}ngx = x$ which implies $ngx = gx$. Hence, $\mathcal{K}^{\mathcal{N}}$ is preserved by the action of $\mathcal{G}$. The action of $\mathcal{G}$ on $\mathcal{K}^{\mathcal{N}}$ factors to an action of $\mathcal{G}/\mathcal{N}$ on $\mathcal{K}^{\mathcal{N}}$, which has a fixed point $x_0$ since $\mathcal{G}/\mathcal{N}$ is amenable. But then $x_0$ is fixed by each $g \in \mathcal{G}$. Hence, $\mathcal{G}$ is amenable. ■

## Appendix B. Examples of collections $\mathcal{S}$ where T1-T6 hold

### B.1 Infinite-dimensional class

We start by presenting an infinite-dimensional class $\mathcal{S}$ that satisfies T1-T6, and then we subsequently present a finite-dimensional class. To define this class, we fix $c, \alpha > 0$ and a function $F : \mathcal{Z} \to \mathbb{R}^+$ some function that is invariant to permutations, shifts, and rescalings, in the sense that both of the following hold:

F1. *Permutations:* For all $((\boldsymbol{x}, \boldsymbol{y}), x_0) \in \mathcal{D}_0$, $A \in \mathcal{A}$ and $B \in \mathcal{B}$, it holds that

$$F(z((A\boldsymbol{x}B, A\boldsymbol{y}), B^\top x_0)) = F(z((\boldsymbol{x}, \boldsymbol{y}), x_0).$$

F2. *Shifts and rescalings:* For all $((\boldsymbol{x}, \boldsymbol{y}), x_0) \in \mathcal{D}_0$, $a \in \mathbb{R}^p$, $b \in (\mathbb{R}^+)^p$, $\tilde{a} \in \mathbb{R}$, and $\tilde{b} > 0$, it holds that $F(z((\boldsymbol{x}^{a,b}, \tilde{a} + \tilde{b}\boldsymbol{y}), a + b \odot x_0)) = F(z((\boldsymbol{x}, \boldsymbol{y}), x_0)$, where $\boldsymbol{x}^{a,b}$ is the $n \times p$ matrix with row $i$ equal to $a + b \odot \boldsymbol{x}_{i*}$.

These conditions bear some resemblance to T4 and T5. One example of a function $F$ satisfies the above conditions is a constant function.

The infinite-dimensional class of $\mathcal{Z} \to \mathbb{R}$ functions that we consider is defined as

$$\mathcal{S}_{F,\alpha,c} := \left\{ S : \forall \boldsymbol{z} \in \mathcal{Z}, |S(\boldsymbol{z})| \leq F(\boldsymbol{z}), \sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq c \right\}.$$

We will now show that this class satisfies T1-T6. Conditions T1 and T2 follow immediately from the definition of $\mathcal{S}_{F,\alpha,c}$. We now show that T3 holds. Because $C(\mathcal{Z}, \mathbb{R})$ is complete, it suffices to show that, if $S_n \to S$ converges compactly and $S_n \in \mathcal{S}_{F,\alpha,c}$, then $S \in \mathcal{S}_{F,\alpha,c}$. Let $S_n \to S$ compactly. To see that $|S(\boldsymbol{z})| \leq F(\boldsymbol{z})$, note that

$$|S(\boldsymbol{z})| \leq |S_n(\boldsymbol{z}) - S(\boldsymbol{z})| + |S_n(\boldsymbol{z})| \leq F(\boldsymbol{z}) + |S_n(\boldsymbol{z}) - S(\boldsymbol{z})|$$

and then take the limit as $n \to \infty$. To see that $S$ satisfies the Hölder condition, note that, for any $\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}$,

$$\frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq \frac{|S(\boldsymbol{z}) - S_n(\boldsymbol{z})|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} + \frac{|S_n(\boldsymbol{z}) - S_n(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} + \frac{|S_n(\boldsymbol{z}') - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha}$$

and again take the limit as $n \to \infty$. Hence, $\frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq c$ for each $\boldsymbol{z} \neq \boldsymbol{z}'$, and so $\sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq c$. Hence $S \in \mathcal{S}_{F,\alpha,c}$, and thus T3 holds. We now show that T4

and T5 hold. To do this, we will use the group theoretic notation defined in Section 7.1. As noted in that section, T4 and T5 are equivalent to the condition that $g \cdot S \in \mathcal{S}_{F,\alpha,c}$ for all $g \in \mathcal{G}_0$ and $S \in \mathcal{S}_{F,\alpha,c}$. We will therefore fix $S \in \mathcal{S}_{F,\alpha,c}$ and $g \in \mathcal{G}_0$ and show that $g \cdot S \in \mathcal{S}_{F,\alpha,c}$. For $\boldsymbol{z} \in \mathcal{Z}$, we have that

$$|(g \cdot S)(\boldsymbol{z})| = |S(g \cdot \boldsymbol{z})| \leq F(g \cdot \boldsymbol{z}) = F(\boldsymbol{z}),$$

where the inequality holds since $S \in \mathcal{S}_{F,\alpha,c}$. Note that for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$, $\|g \cdot \boldsymbol{z} - g \cdot \boldsymbol{z}'\| = \|\boldsymbol{z} - \boldsymbol{z}'\|$. Hence,

$$\begin{aligned}\sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|(g \cdot S)(\boldsymbol{z}) - (g \cdot S)(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} &= \sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|(g \cdot S)(\boldsymbol{z}) - (g \cdot S)(\boldsymbol{z}')|}{\|g \cdot \boldsymbol{z} - g \cdot \boldsymbol{z}'\|^\alpha} \\ &= \sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq c,\end{aligned}$$

where the inequality holds since $S \in \mathcal{S}_{F,\alpha,c}$. Hence, $g \cdot S \in \mathcal{S}_{F,\alpha,c}$, and so T4 and T5 hold. It remains to show T6. To see that this holds, fix $S_1, S_2 \in \mathcal{S}_{F,\alpha,c}$ and $\delta \in (0,1)$ and let $S = \delta S_1 + (1-\delta)S_2$. By the triangle inequality and the fact that $S_1, S_2 \in \mathcal{S}_{F,\alpha,c}$, we have the following two displays for any $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$:

$$\begin{aligned}&|S(\boldsymbol{z})| = |\delta S_1(\boldsymbol{z}) + (1-\delta)S_2(\boldsymbol{z})| \leq \delta|S_1(\boldsymbol{z})| + (1-\delta)|S_2(\boldsymbol{z})| \leq F(\boldsymbol{z}), \\ &\sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq \delta \sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S_1(\boldsymbol{z}) - S_1(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} + (1-\delta) \sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S_2(\boldsymbol{z}) - S_2(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq c.\end{aligned}$$

Hence, $S \in \mathcal{S}_{F,\alpha,c}$, and so T6 holds.

### B.2 Finite-dimensional class

#### B.2.1 Overview

For an explicit representation of $\mathcal{Z}$, we have

$$\mathcal{Z} = \mathcal{O}_n^p \times \mathcal{O}_n \times \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \times \mathbb{R},$$

where $\mathcal{O}_n = \{a \in \mathbb{R}^n \mid \bar{a} = 0, s(a) = 1\}$. For ease of communication, we will abbreviate

$$\begin{aligned}z_a &= \left(\frac{\boldsymbol{x} - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})}, \frac{\boldsymbol{y} - \bar{\boldsymbol{y}}}{s(\boldsymbol{y})}\right) \in \mathcal{O}_n^p \times \mathcal{O}_n, \\ z_t &= \frac{x_0 - \bar{\boldsymbol{x}}}{s(\boldsymbol{x})} \in \mathbb{R}^p \\ z_m &= \left(\frac{\bar{\boldsymbol{x}}}{s(\boldsymbol{x})}, \frac{\bar{\boldsymbol{y}}}{s(\boldsymbol{y})}\right) \in \mathbb{R}^p \times \mathbb{R} \\ z_s &= (\log s(\boldsymbol{x}), \log s(\boldsymbol{y})) \in \mathbb{R}^p \times \mathbb{R},\end{aligned}$$

so that $z = (z_a, z_t, z_m, z_s)$. Here, $z_a$ stands for the angular component, $z_t$ stands for the test point, $z_m$ stands for the mean, $z_s$ stands for the standard deviation.

To define our parametric example for $\mathcal{S}$, we can use separation of variables to consider the coordinates of $\boldsymbol{z}$ separately. We will consider estimators belonging to the class $\mathcal{S}$ of all

$S$ for which there exist a $B \in \mathbb{N}$, $(C_b)_{b=1}^B$ on the $(B-1)$-simplex, $S_{a,b} \in \mathcal{S}_a$, $S_{t,b} \in \mathcal{S}_t$, and $S_{g,b} \in \mathcal{S}_g$ such that

$$S(\boldsymbol{z}) = \sum_{b=1}^{B} C_b S_{a,b}(z_a) S_{t,b}(z_t) S_{g,b}(z_m, z_s). \tag{S1}$$

We refer to $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ as the angular part, test point part, and group part of $\mathcal{S}$, respectively. In what follows, we will describe conditions on $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ that make it so that T1-T6 hold. We will then describe interesting collections $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ that satisfy these conditions.

First note that we have the following inequality:

$$\begin{aligned}
|S(\boldsymbol{z}) - S(\boldsymbol{z}')| &\leq \sum_{b=1}^{B} C_b |S_{a,b}(z_a) S_{t,b}(z_t) S_{g,b}(z_m, z_s) - S_{a,b}(z'_a) S_{t,b}(z'_t) S_{g,b}(z'_m, z'_s)| \\
&\leq \quad \sum_{b=1}^{B} C_b |S_{t,b}(z_t)||S_{g,b}(z_m, z_s)||S_{a,b}(z_a) - S_{a,b}(z'_a)| \\
&\quad + \sum_{b=1}^{B} C_b |S_{a,b}(z'_a)||S_{g,b}(z_m, z_s)||S_{t,b}(z_t) - S_{t,b}(z'_t)| \\
&\quad + \sum_{b=1}^{B} c_b |S_{a,b}(z'_a)||S_{t,b}(z'_t)||S_{g,b}(z_m, z_s) - S_{g,b}(z'_m, z'_s)|.
\end{aligned}$$

Thus if for all $b$, $S_{a,b}$, $S_{t,b}$, and $S_{g,b}$ were uniformly bounded by $M^{1/3}$ and each of their global Hölder constant was less than or equal to $\frac{c}{3M^{2/3}}$, then $\sup_{\boldsymbol{z} \in \mathcal{Z}} |S(\boldsymbol{z})| \leq M$ and $\sup_{\boldsymbol{z} \neq \boldsymbol{z}' \in \mathcal{Z}} \frac{|S(\boldsymbol{z}) - S(\boldsymbol{z}')|}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \leq c$. Hence, if $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ are such that functions in these collections are uniformly bounded by $M^{1/3}$ and are $\frac{c}{3M^{2/3}}$-Hölder, then $\mathcal{S} \subseteq \mathcal{S}_{M,\alpha,c}$. In that case, conditions T1 and T2 hold. Since every compact subset of $\mathcal{Z}$ can be written as a subset of a product of compact sets $K = K_1 \times K_2 \times K_3$, $K_1 \subseteq \mathcal{O}_n^{p+1}$, $K_2 \subset \mathbb{R}^p$, $K_3 \subset \mathbb{R}^{2p+2}$, for condition T3 to hold, it suffices to show $\mathcal{S}_a, \mathcal{S}_t$, and $\mathcal{S}_g$ are closed. Condition T4 holds if $\mathcal{S}_a$ is closed under rotations with respect to the $n$ observations and if $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ are closed under permutations with respect to the $p$ features. The latter can be done by letting $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ be $p$-fold tensor products of an identical space of functions. Condition T5 is satisfied when $\mathcal{S}_g$ is closed under shifts. Finally, condition T6 always holds by equation (S1), but in our construction, we enforce that $\mathcal{S}_a$, $\mathcal{S}_t$, and $\mathcal{S}_g$ are convex so that if any two were singletons, $\mathcal{S}$ would equal the remaining one. For example, if $\mathcal{S}_t = \mathcal{S}_g = \{1\}$, $\mathcal{S} = \mathcal{S}_a$.

### B.2.2 Angular Part ($\mathcal{S}_a$)

We define $\mathcal{S}_a$ by truncating an orthonormal basis for the tensor product space $L^2(\mathcal{O}_n)^{\otimes(p+1)}$ to a specified finite number of terms and then taking the subset of the span of those basis vectors that are contained in $\mathcal{S}_{M^{1/3},\alpha,c/(3M^{2/3})}$ for some $c$ and $M$. Note that $\mathcal{O}_n \cong \mathbb{S}^{n-2}$, where "$\cong$" denotes an isomorphic relation and $\mathbb{S}^{n-2}$ is the $(n-2)$-dimensional unit sphere. Let $\mathbf{1}$ be the $n$-dimensional vector of 1's, and note that $\mathcal{O}_n$ can be expressed in the following form:

$$\mathcal{O}_n = \{w \in \mathbb{R}^n \mid n^{-1/2} w^T \mathbf{1} = 0, n^{-1} w^T w = 1\}.$$

Let $U \in O(n)$, the orthogonal group, be such that $n^{-1/2}U\mathbf{1} = e_n$, the $n$th elementary basis vector. Such a $U$ exists because $\|n^{-1/2}\mathbf{1}\| = 1$. Then,

$$\mathcal{O}_n = \{\sqrt{n}U^T v \mid v \in \mathbb{R}^n, v_n = 0, \|v\|^2 = 1\}$$

We have the isomorphism $\zeta : L^2(\mathcal{O}_n) \to L^2(\mathbb{S}^{n-2})$, $\zeta(f)(v) = f(\sqrt{n}U^T v)$. Thus, if we have an orthonormal basis for $L^2(\mathbb{S}^{n-2})$, we may use the operator $\zeta^{-1}$ to obtain an orthonormal basis for $L^2(\mathcal{O}_n)$. Let $\mathbf{H}_\ell$ be the space of harmonic polynomials of degree $\ell$ in $(n-1)$-dimensions. By the Stone-Weierstrass theorem, the direct sum $\bigoplus_{\ell=0}^{\infty} \mathbf{H}_\ell$ is dense in $L^2(\mathbb{S}^{n-2})$. We can truncate the series and stop at a prespecified point $q_a$, so that

$$\mathcal{S}_a = \left( (\bigoplus_{\ell=0}^{q_a} \mathbf{H}_\ell) \circ (\sqrt{n}U^T) \right)^{\otimes(p+1)} \bigcap \mathcal{S}_{M^{1/3},\alpha,c/(3M^{2/3})}(\mathcal{O}_n^{p+1}). \tag{S2}$$

We use the orthonormal basis $\{\mathscr{Y}_{l_1,l_2,\cdots,l_{n-2}} : |l_1| \leq l_2 \leq \cdots \leq l_{n-2}\}$ for the spherical harmonics introduced in Higuchi (1987) (replacing "$Y$" in their notation by "$\mathscr{Y}$" to avoid notational overload), where an explicit expression for this basis is provided in that work. Let $N(n,p,q) = (\sum_{\ell=0}^{q} \dim \mathbf{H}_\ell)^{p+1}$ and

$$\mathcal{C}_a = \left\{ A \in \mathbb{R}^{N(n,p,q_a)} : \prod_{j=0}^{p} \sum_{|l_1| \leq l_2 \leq \cdots \leq l_{n-2} \leq q_a} A_{l_1,\cdots,l_{n-2}} \mathscr{Y}_{l_1,\cdots,l_{n-2}}(\sqrt{n}U^T z^{x,\cdot,j}) \in \mathcal{S}_a \right\},$$

where $z^{x,i,0} = z^{y,i}$. The set $\mathcal{C}_a$ is the coefficient space of the basis expansion in $\mathcal{S}_a$ and is convex and compact if and only if $\mathcal{S}_a$ is convex and compact. The set $\mathcal{S}_a$ is closed under rotations in the $n$ observations since the spherical harmonics for any given degree is closed under rotations. It is also closed under permutations due to the $(p+1)$-fold tensor product form. As an intersection of closed convex sets, it is closed and convex.

### B.2.3 Test Point Part ($\mathcal{S}_t$)

Similarly to $\mathcal{S}_a$, $\mathcal{S}_t$ is defined by truncating an orthonormal basis for $L^2(\mathbb{R}^p)$. Let $\{\psi_k\}_{k=0}^{\infty}$ be the normalized Hermite functions. They form an orthonormal basis of $L^2(\mathbb{R})$ and so their $p$-fold tensor product is an orthonormal basis of $L^2(\mathbb{R}^p)$. We can take

$$\mathcal{S}_t = (\text{span}\{\psi_k \mid k \in \{0,1,\cdots,q_t\}\})^{\otimes p} \cap \mathcal{S}_{M^{1/3},\alpha,c/(3M^{2/3})}(\mathbb{R}^p).$$

We can similarly define the coefficient space $\mathcal{C}_t$:

$$\mathcal{C}_t = \left\{ A \in \mathbb{R}^{q_t p} : \left( z_t \mapsto \prod_{j=1}^{p} \sum_{k=0}^{q_t} A_{jk}\psi_k(z_{t,j}) \right) \in \mathcal{S}_t \right\}.$$

Similarly to $\mathcal{S}_a$, the $p$-fold tensor product form and it being an intersection of closed and convex sets show all of the necessary conditions are satisfied.

### B.2.4 Group Part ($\mathcal{S}_g$)

The $\mathcal{S}_g$ that we will define imposes that the functions are periodic in each dimension, in the sense that, if $S_g \in \mathcal{S}_g$ and $z_g - z_g' = \pm e_i$ for some elementary basis vector $e_i$, then $S_g(z_g) = S_g(z_g')$. In other words, we will be dealing with functions on the $(2p+2)$-dimensional torus, $\mathbb{T}^{2p+2} = (S^1)^{2p+2}$. Since the torus is a product of 1-spheres, we can use the same process as described when defining the angular part ($\mathcal{S}_a$), namely letting

$$\mathcal{S}_g = \left( \bigoplus_{\ell=0}^{q_g} H_\ell \right)^{\otimes(2p+2)} \bigcap \mathcal{S}_{M^{1/3},\alpha,c/(3M^{2/3})}(\mathbb{T}^{2p+2}) \tag{S3}$$

In this case, $H_\ell = \text{span}\{\cos(2\pi\ell x), \sin(2\pi\ell x)\}$ and translations can be dealt with by the sum and difference formulas for sine and cosine. Translations under periodicity are the same as rotation, and since it is known that spherical harmonics are rotationally invariant, $\mathcal{S}_g$ is closed under translations. Similarly, the tensor product form of $\mathcal{S}_a$ and its being an intersection of closed and convex sets implies that the rest of the sufficient conditions described at the end of Section B.2.1 are satisfied.

## Appendix C. Examples of collections $\Gamma$ where P5 holds

We now describe settings where P5 is often applicable. We will specify $\mathcal{P}_1$ in each of these settings, and the model $\mathcal{P}$ is then defined by expanding $\mathcal{P}_1$ to contain the distributions of all possible shifts and rescalings of a random variate drawn from some $P_1 \in \mathcal{P}_1$. The first class of models for which P5 is often satisfied is parametric in nature, with each distribution $P_\theta \in \mathcal{P}_1$ indexed smoothly by a finite dimensional parameter $\theta$ belonging to a subset $\Theta$ of $\mathbb{R}^k$. We note here that, because the sample size $n$ is fixed in our setting, we can obtain an essentialy unrestricted model by allowing $k$ to be large relative to $n$. In parametric settings, $\rho$ can often be defined as $\rho(P_\theta, P_{\theta'}) = \|\theta - \theta'\|_2$, where we recall that $\|\cdot\|_2$ denotes the Euclidean norm. If $\Gamma_1$ is uniformly tight, which certainly holds if $\Theta$ is bounded, then P5 holds provided $\theta \mapsto R(T, P_\theta)$ is upper-semicontinuous for all $T \in \mathcal{T}_e$. For a concrete example where the conditions of P5 are satisfied, consider the case that $\Theta = \{\theta : \|\theta\|_0 \leq \mathfrak{s}_0, \|\theta\|_1 \leq \mathfrak{s}_1\}$ for sparsity parameters $\mathfrak{s}_0$ and $\mathfrak{s}_1$ on $\|\theta\|_0 := \#\{j : \theta_j \neq 0\}$ and $\|\theta\|_1 := \sum_j |\theta_j|$, and $P_\theta$ is the distribution for which $X \sim N(\mathbf{0}_p, \text{Id}_p)$, and $Y|X \sim N(\theta^\top X, 1)$. This setting is closely related to the sparse linear regression example that we study numerically in Section 5.3.2.

Condition P5 also allows for nonparametric regression functions. Define $\phi^p$ to be the $p$-dimensional standard Gaussian measure. Define $L_0^2(\phi^p) = \{f \in L^2(\phi^p) \mid \int f(x) d\phi^p(x) = 0\}$. Let $\mathcal{F} \subset L_0^2(\phi^p)$ satisfy the following conditions:

(i) $\mathcal{F}$ is bounded. $\sup_{f\in\mathcal{F}} \|f\|_{L^2(\phi^p)} < \infty$.

(ii) $\mathcal{F}$ is uniformly equivanishing. $\lim_{N\to\infty} \sup_{f\in\mathcal{F}} \|f 1_{B(0,N)^c}\|_{L^2(\phi^p)} = 0$.

(iii) $\mathcal{F}$ is uniformly equicontinuous. $\lim_{r\searrow 0} \sup_{f\in\mathcal{F}} \sup_{y\in B(0,r)} \|\tau_y f - f\|_{L^2(\phi^p)} = 0$ where $\tau_y$ is the translation by $y$ operator.

(iv) $\mathcal{F}$ is closed in $L^2(\phi^p)$.

(v) There exists $q' > 2$ such that $\mathcal{F} \subset L^{q'}(\phi^p)$.

By a generalization of the Riesz-Kolmogorov theorem as seen in Guo and Zhao (2019), $\mathcal{F}$ is compact under assumptions (i) through (iv). Let $c > 0$, $\alpha \in (0, 1]$. We suppose that $\mathcal{S} = \mathcal{S}^0$ where $\mathcal{S}^0$ is the set of all functions $S : \mathcal{Z} \to \mathbb{R}$ such that $|S(\boldsymbol{z})| \leq F(\boldsymbol{z})$, $|S(\boldsymbol{z}) - S(\boldsymbol{z}')| \leq c\|\boldsymbol{z} - \boldsymbol{z}'\|_2^\alpha$ for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$. Assume further that $F$ is bounded, i.e.

$$\sup_{z \in \mathcal{Z}} |F(z)| = B_{\mathcal{S}^0} < \infty, \tag{S4}$$

and also that $F$ is constant in the orbits induced by the group action on $\mathcal{Z}$ defined in Section 7.1.

For each $f \in \mathcal{F}$, let $P_f$ denote the distribution of $X \sim N(0, \mathrm{Id}_p)$, $Y \mid X \sim N(f(X), 1)$. Suppose that $\mathcal{P}_1 = \{P_f \mid f \in \mathcal{F}\}$. With the metric $\rho(f, g) = \|f - g\|_{L^2(\phi^p)}$, $(\mathcal{P}_1, \rho)$ is a complete separable compact metric space. We also see that $P \mapsto R(T, P)$ is continuous.

**Lemma S7** *For all $T \in \mathcal{T}_e$, $P \mapsto R(T, P)$ is continuous in this example.*

**Proof** To ease presentation, we introduce some notation. For $f \in \mathcal{F}$, let $f(\boldsymbol{x}) := (f(x_i))_{i=1}^n$, $\bar{f}(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^n f(x_i)$, $s_f(\boldsymbol{d}) := s(\boldsymbol{y} + f(\boldsymbol{x}))$, and $\bar{y}(\boldsymbol{y}) := \bar{\boldsymbol{y}}$. Let $S_{T,f}$ denote the map $(\boldsymbol{d}, x_0) \mapsto S_T(z_f(\boldsymbol{d}, x_0))$, where $z_f(\boldsymbol{d}, x_0)$ takes the same value as $z_f(\boldsymbol{d}, x_0)$ except that the entry $\frac{\boldsymbol{y} - \bar{\boldsymbol{y}}}{s(y)}$ is replaced with $\frac{\boldsymbol{y} + f(\boldsymbol{x}) - \bar{\boldsymbol{y}} - \bar{f}(\boldsymbol{x})}{s_f}$. Also let $\phi^\star := \phi^{p(n+1)+n}$. For $q \in [1, \infty)$ and a function $f : \mathcal{D} \times \mathcal{X}$, we let $\|f\|_{L^q(\phi^\star)} := [\int |f(\boldsymbol{x}, \boldsymbol{y}, x_0)|^q \phi^\star(d\boldsymbol{x}, d\boldsymbol{y}, dx_0)]^{1/q}$. We let $\|f\|_{L^\infty(\phi^\star)} := \inf\{c \geq 0 : f(\boldsymbol{x}, \boldsymbol{y}, x_0) \leq c \;\; \phi^\star\text{−a.s.}\}$. For $f : \mathcal{D} \to \mathbb{R}$, we write $\|f\|_{L^q(\phi^\star)}$ to mean $\|(\boldsymbol{d}, x_0) \mapsto f(\boldsymbol{d})\|_{L^q(\phi^\star)}$, and follow a similar convention for functions that only take as input $\boldsymbol{x}$, $x_i$, $\boldsymbol{y}$, or $x_0$. We will write $\lesssim$ to mean inequality up to a positive multiplicative constant that may only depend on $\mathcal{S}$ or $\mathcal{F}$.

Fix $\varepsilon \in (0, 1)$ and $T \in \mathcal{T}_e$. Now, for any $f \in \mathcal{F}$, a change of variables shows that

$$\begin{aligned}
R(T, P_f) &= E_{P_f}\left[\int [T(\boldsymbol{X}, \boldsymbol{Y})(x_0) - f(x_0)]^2 \, d\phi^p(x_0)\right] \\
&= \int [T(\boldsymbol{x}, \boldsymbol{y})(x_0) - f(x_0)]^2 (2\pi)^{-\frac{n}{2}} \exp\left[-\frac{1}{2}\sum_{i=1}^n \{y_i - f(x_{i\cdot})\}^2\right] \phi^{p(n+1)}(d\boldsymbol{x}, dx_0) d\boldsymbol{y} \\
&= \int [T(\boldsymbol{x}, \boldsymbol{y} + f(\boldsymbol{x}))(x_0) - f(x_0)]^2 \, \phi^\star(d\boldsymbol{x}, dx_0, d\boldsymbol{y}) \\
&= \int \left[\bar{\boldsymbol{y}} + s(\boldsymbol{y} + f(\boldsymbol{x})) S_{T,f}(\boldsymbol{d}, x_0) + \bar{f}(\boldsymbol{x}) - f(x_0)\right]^2 \phi^\star(d\boldsymbol{x}, dx_0, d\boldsymbol{y}).
\end{aligned}$$

Hereafter we write $d\phi^\star$ to denote $\phi^\star(d\boldsymbol{x}, dx_0, d\boldsymbol{y})$.

Fix $f, g \in \mathcal{F}$. Most of the remainder of this proof will involve establishing that $R(T, P_f) - R(T, P_g) \lesssim \varepsilon^{-2}\|f - g\|_{L^2(\phi^p)} + \varepsilon$. By symmetry, it will follow that $|R(T, P_f) - R(T, P_g)| \leq \varepsilon^{-2}\|f - g\|_{L^2(\phi^p)} + \varepsilon$.

In what follows we will use the notation $(g - f)(x_0)$ to mean $g(x_0) - f(x_0)$, $(\bar{g} - \bar{f})(\boldsymbol{x})$ to mean $\bar{g}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})$, etc. The above yields that

$$\begin{aligned}
&R(T, P_f) - R(T, P_g) \\
&\quad = \int \left[(\bar{f}(\boldsymbol{x}) - f(x_0))^2 - (\bar{g}(\boldsymbol{x}) - g(x_0))^2\right] d\phi^\star
\end{aligned} \tag{S5}$$

$$+ 2 \int \bar{\boldsymbol{y}} \left[ (g - f)(x_0) - (\bar{g} - \bar{f})(\boldsymbol{x}) \right] d\phi^\star \tag{S6}$$

$$+ 2 \int \bar{\boldsymbol{y}} \left[ s_f(\boldsymbol{d}) S_{T,f}(\boldsymbol{d}, x_0) - s_g(\boldsymbol{d}) S_{T,g}(\boldsymbol{d}, x_0) \right] d\phi^\star \tag{S7}$$

$$+ \int \left[ s_f^2(\boldsymbol{d}) S_{T,f}(\boldsymbol{d}, x_0)^2 - s_g^2(\boldsymbol{d}) S_{T,g}(\boldsymbol{d}, x_0)^2 \right] d\phi^\star \tag{S8}$$

$$+ 2 \int \left[ (\bar{f}(\boldsymbol{x}) - f(x_0)) s_f(\boldsymbol{d}) S_{T,f}(\boldsymbol{d}, x_0) - (\bar{g}(\boldsymbol{x}) - g(x_0)) s_g(\boldsymbol{d}) S_{T,g}(\boldsymbol{d}, x_0) \right] d\phi^\star. \tag{S9}$$

We bound the labeled terms on the right-hand side separately. After some calculations, it can be seen that (S5) and (S6) are bounded by a constant multiplied by $\|f - g\|_{L^2(\phi^p)}$. These calculations, which are omitted, involve several applications of the triangle inequality, the Cauchy-Schwarz inequality, and condition (i).

The integral in (S7) bounds as follows:

$$\begin{aligned}
&\int \bar{\boldsymbol{y}} \left[ s_f(\boldsymbol{d}) S_{T,f}(\boldsymbol{d}, x_0) - s_g(\boldsymbol{d}) S_{T,g}(\boldsymbol{d}, x_0) \right] d\phi^\star \\
&\quad = \int \bar{\boldsymbol{y}} S_{T,f}(\boldsymbol{d}, x_0) [s_f(\boldsymbol{d}) - s_g(\boldsymbol{d})] d\phi^\star + \int \bar{\boldsymbol{y}} s_g(\boldsymbol{d}) [S_{T,f}(\boldsymbol{d}, x_0) - S_{T,g}(\boldsymbol{d}, x_0)] d\phi^\star \\
&\quad \leq \|\bar{y} S_{T,f}[s_f - s_g]\|_{L^1(\phi^\star)} + \|\bar{y} s_g [S_{T,f} - S_{T,g}]\|_{L^1(\phi^\star)}.
\end{aligned} \tag{S10}$$

We start by studying first term of the right-hand side above. Note that, by (S4) and the assumption that $|S(\boldsymbol{z})| \leq F(\boldsymbol{z})$ for all $\boldsymbol{z} \in \mathcal{Z}$ and $S \in \mathcal{S}$, we have that $|S_{T,f}(\boldsymbol{d}, x_0)| \leq B_{\mathcal{S}^0}$. Combining this with Cauchy-Schwarz, the first term on the right-hand side above bounds as

$$\|\bar{y} S_T [s_f - s_g]\|_{L^1(\phi^\star)} \leq B_{\mathcal{S}_0} \|\bar{y}\|_{L^2(\phi^\star)} \|s_f - s_g\|_{L^2(\phi^\star)}. \tag{S11}$$

To continue the above bound, we will show that $\|s_f - s_g\|_{L^2(\phi^\star)} \lesssim \|f - g\|^{1/2}_{L^2(\phi^p)}$. Noting that

$$\begin{aligned}
s_f^2(\boldsymbol{d}) - s_g^2(\boldsymbol{d}) = \frac{1}{n} \sum_{i=1}^n \Bigg[ & f(x_i)^2 - g(x_i)^2 + 2(y_i - \bar{\boldsymbol{y}})[f(x_i) - g(x_i) + \bar{g}(\boldsymbol{x}) - \bar{f}(\boldsymbol{x})] \\
& + 2[g(x_i)\bar{g}(\boldsymbol{x}) - f(x_i)\bar{f}(\boldsymbol{x})] + \bar{f}(\boldsymbol{x})^2 - \bar{g}(\boldsymbol{x})^2 \Bigg]
\end{aligned}$$

we see that, by the triangle inequality and the Cauchy-Schwarz inequality,

$$\|s_f^2 - s_g^2\|_{L^1(\phi^\star)} \lesssim \|f - g\|_{L^2(\phi^p)}.$$

For $a > 0, b > 0$, $|\sqrt{a} - \sqrt{b}| \leq \sqrt{|a - b|}$, and so $|s_f(\boldsymbol{d}) - s_g(\boldsymbol{d})| \leq \sqrt{|s_f^2(\boldsymbol{d}) - s_g^2(\boldsymbol{d})|}$, which implies that $|s_f(\boldsymbol{d}) - s_g(\boldsymbol{d})|^2 \leq |s_f^2(\boldsymbol{d}) - s_g^2(\boldsymbol{d})|$, which in turn implies that $\|s_f - s_g\|^2_{L^2(\phi^\star)} \leq \|s_f^2 - s_g^2\|_{L^1(\phi^\star)}$. Combining this with the above and taking square roots of both sides gives the desired bound, namely

$$\|s_f - s_g\|_{L^2(\phi^\star)} \lesssim \|f - g\|^{1/2}_{L^2(\phi^p)}. \tag{S12}$$

Recalling (S11), we then see that the first term on the right-hand side of (S10) satisfies

$$\|\bar{y} S_{T,f}[s_f - s_g]\|_{L^1(\phi^\star)} \lesssim \|f - g\|^{1/2}_{L^2(\phi^p)}.$$

We now study the second term in (S10). Before beginning our analysis, we note that, for all $\boldsymbol{d}$,

$$1 \leq 1_{\{s_g(\boldsymbol{d}) \leq \varepsilon\}} + 1_{\{s_g(\boldsymbol{d}) > \varepsilon\} \cap \{|s_g(\boldsymbol{d}) - s_f(\boldsymbol{d})| < \varepsilon/2\}} + 1_{\{|s_g(\boldsymbol{d}) - s_f(\boldsymbol{d})| \geq \varepsilon/2\}}. \tag{S13}$$

Combining the above with the triangle inequality, the second term in (S10) bounds as:

$$\begin{aligned}\|\bar{y} s_g[S_{T,f} - S_{T,g}]\|_{L^1(\phi^\star)} \leq{}& \|\bar{y} s_g[S_{T,f} - S_{T,g}] 1_{\{s_g \leq \varepsilon\}}\|_{L^1(\phi^\star)} \\ &+ \|\bar{y} s_g[S_{T,f} - S_{T,g}] 1_{\{s_g > \varepsilon\} \cap \{|s_f - s_g| < \varepsilon/2\}}\|_{L^1(\phi^\star)} \\ &+ \|\bar{y} s_g[S_{T,f} - S_{T,g}] 1_{\{|s_g - s_f| \geq \varepsilon/2\}}\|_{L^1(\phi^\star)}. \end{aligned} \tag{S14}$$

In the above normed quantities, expressions like $1_{\{s_g \leq \varepsilon\}}$ should be interpreted as functions, e.g. $1_{\{s_g(\cdot) \leq \varepsilon\}}$. By (S4), the first term on the right-hand side bounds as

$$\|\bar{y} s_g[S_{T,f} - S_{T,g}] 1_{s_g \leq \varepsilon}\|_{L^1(\phi^\star)} \lesssim \varepsilon.$$

For the second term, we start by noting that

$$\begin{aligned}&\|z_f(\boldsymbol{d}) - z_g(\boldsymbol{d})\|_2 \\ &= \left\| \frac{(s_g - s_f)(\boldsymbol{d})}{s_g(\boldsymbol{d}) s_f(\boldsymbol{d})} (\boldsymbol{y} - \bar{\boldsymbol{y}}) + \frac{1}{s_f(\boldsymbol{d}) s_g(\boldsymbol{d})} [s_f(\boldsymbol{d})(f - g + \bar{g} - \bar{f})(\boldsymbol{x}) + (s_g - s_f)(\boldsymbol{d})(f - \bar{f})(\boldsymbol{x})] \right\|_2.\end{aligned}$$

Using that $(a + b + c)^\kappa \leq a^\kappa + b^\kappa + c^\kappa$ whenever $a, b, c > 0$ and $\kappa \in (0, 1]$, this then implies that

$$\begin{aligned}\|z_f(\boldsymbol{d}) - z_g(\boldsymbol{d})\|_2^\alpha \leq{}& \left\| \frac{(s_g - s_f)(\boldsymbol{d})}{s_g(\boldsymbol{d}) s_f(\boldsymbol{d})} (\boldsymbol{y} - \bar{\boldsymbol{y}}) \right\|_2^\alpha + \left\| \frac{(f - g + \bar{g} - \bar{f})(\boldsymbol{x})}{s_g(\boldsymbol{d})} \right\|_2^\alpha \\ &+ \left\| \frac{(s_g - s_f)(\boldsymbol{d})(f - \bar{f})(\boldsymbol{x})}{s_f(\boldsymbol{d}) s_g(\boldsymbol{d})} \right\|_2^\alpha,\end{aligned}$$

where above $\alpha$ is the exponent from the Hölder condition satisfied by $\mathcal{S}^0$. Combining the Hölder condition with the above, we then see that

$$\begin{aligned}|S_{T,f}(\boldsymbol{d}, x_0) - S_{T,g}(\boldsymbol{d}, x_0)| \lesssim{}& \left\| \frac{(s_g - s_f)(\boldsymbol{d})}{s_g(\boldsymbol{d}) s_f(\boldsymbol{d})} (\boldsymbol{y} - \bar{\boldsymbol{y}}) \right\|_2^\alpha + \left\| \frac{(f - g + \bar{g} - \bar{f})(\boldsymbol{x})}{s_g(\boldsymbol{d})} \right\|_2^\alpha \\ &+ \left\| \frac{(s_g - s_f)(\boldsymbol{d})(f - \bar{f})(\boldsymbol{x})}{s_f(\boldsymbol{d}) s_g(\boldsymbol{d})} \right\|_2^\alpha.\end{aligned}$$

Multiplying both sides by $|\bar{\boldsymbol{y}} s_g(\boldsymbol{d}) 1_{\{s_g(\boldsymbol{d}) > \varepsilon, |(s_f - s_g)(\boldsymbol{d})| < \varepsilon/2\}}|$, we then see that

$$\begin{aligned}&\left| \bar{y} s_g(\boldsymbol{d}) [S_{T,f}(\boldsymbol{d}, x_0) - S_{T,g}(\boldsymbol{d}, x_0)] 1_{\{s_g(\boldsymbol{d}) > \varepsilon, |(s_f - s_g)(\boldsymbol{d})| < \varepsilon/2\}} \right| \\ &\lesssim |\bar{\boldsymbol{y}}| s_g(\boldsymbol{d}) \left\| \frac{(s_g - s_f)(\boldsymbol{d})}{s_g(\boldsymbol{d}) s_f(\boldsymbol{d})} (\boldsymbol{y} - \bar{\boldsymbol{y}}) \right\|_2^\alpha 1_{\{s_g(\boldsymbol{d}) > \varepsilon, |(s_f - s_g)(\boldsymbol{d})| < \varepsilon/2\}}\end{aligned}$$

$$
\begin{aligned}
&+ |\bar{\boldsymbol{y}}| s_g(\boldsymbol{d}) \left\| \frac{(f - g + \bar{g} - \bar{f})(\boldsymbol{x})}{s_g(\boldsymbol{d})} \right\|_2^{\alpha} 1_{\{s_g(\boldsymbol{d}) > \varepsilon, |(s_f - s_g)(\boldsymbol{d})| < \varepsilon/2\}} \\
&+ |\bar{\boldsymbol{y}}| s_g(\boldsymbol{d}) \left\| \frac{(s_g - s_f)(\boldsymbol{d})(f - \bar{f})(\boldsymbol{x})}{s_f(\boldsymbol{d}) s_g(\boldsymbol{d})} \right\|_2^{\alpha} 1_{\{s_g(\boldsymbol{d}) > \varepsilon, |(s_f - s_g)(\boldsymbol{d})| < \varepsilon/2\}} \\
\lesssim\ & \varepsilon^{-\alpha} |\bar{\boldsymbol{y}}| s_g(\boldsymbol{d})^{1-\alpha} \left\| \boldsymbol{y} - \bar{\boldsymbol{y}} \right\|_2^{\alpha} |(s_g - s_f)(\boldsymbol{d})|^{\alpha} \\
&+ |\bar{\boldsymbol{y}}| s_g(\boldsymbol{d})^{1-\alpha} \left\| (f - g + \bar{g} - \bar{f})(\boldsymbol{x}) \right\|_2^{\alpha} \\
&+ \varepsilon^{-\alpha} |\bar{\boldsymbol{y}}| s_g^{1-\alpha} \left\| (f - \bar{f})(\boldsymbol{x}) \right\|_2^{\alpha} |(s_g - s_f)(\boldsymbol{d})|^{\alpha}.
\end{aligned}
$$

The inequality above remains true if we integrate both sides against $\phi^\star$. The resulting three terms on the right-hand side can be bounded using Hölder's inequality. In particular, we have that

$$
\begin{aligned}
\varepsilon^{-\alpha} \left\| |\bar{y}|^{\alpha} \|y - \bar{y}\|_2^{\alpha} |s_g - s_f|^{\alpha} |\bar{y}|^{1-\alpha} s_g^{1-\alpha} \right\|_{L^1(\phi^\star)} &\leq \varepsilon^{-\alpha} \left\| \bar{y} \|y - \bar{y}\|_2 (s_g - s_f) \right\|_{L^1(\phi^\star)}^{\alpha} \left\| \bar{y} s_g \right\|_{L^1(\phi^\star)}^{1-\alpha} \\
&\lesssim \varepsilon^{-\alpha} \|f - g\|_{L^2(\phi^p)}^{\alpha/2}, \\
\left\| \bar{y} s_g^{1-\alpha} \|(f - g + \bar{g} - \bar{f})(x)\|_2^{\alpha} \right\|_{L^1(\phi^\star)} &\leq \|\bar{y} s_g\|_{L^1(\phi^\star)}^{1-\alpha} \left\| \bar{y} \|(f - g + \bar{g} - \bar{f})(x)\|_2 \right\|_{L^1(\phi^\star)}^{\alpha} \\
&\lesssim \|f - g\|_{L^2(\phi^p)}^{\alpha/2}, \\
\varepsilon^{-\alpha} \left\| \bar{y} s_g^{1-\alpha} \|(f - \bar{f})(x)\|_2^{\alpha} \|s_g - s_f\|^{\alpha} \right\|_{L^1(\phi^\star)} &\leq \varepsilon^{-\alpha} \|\bar{y} s_g\|_{L^1(\phi^\star)}^{1-\alpha} \left\| \|(f - \bar{f})(x)\|_2 |s_g - s_f| \right\|_{L^1(\phi^\star)}^{\alpha} \\
&\lesssim \varepsilon^{-\alpha} \|f - g\|_{L^2(\phi^p)}^{\alpha/2}.
\end{aligned}
$$

Hence, we have shown that the second term on the right-hand side of (S14) satisfies

$$
\left\| \bar{y} s_g [S_{T,f} - S_{T,g}] 1_{s_g > \varepsilon, |s_g - s_f| < \varepsilon/2} \right\|_{L^1(\phi^\star)} \lesssim \varepsilon^{-\alpha} \|f - g\|_{L^2(\phi^p)}^{\alpha/2}.
$$

We now study the third term on the right-hand side of (S14). We start by noting that, by Markov's inequality and (S12),

$$
\begin{aligned}
P_{\phi^\star} \left( |s_g(\boldsymbol{D}) - s_f(\boldsymbol{D})| \geq \frac{\varepsilon}{2} \right) &= P \left( |s_g(\boldsymbol{D}) - s_f(\boldsymbol{D})|^2 \geq \frac{\varepsilon^2}{4} \right) \\
&\leq \frac{4}{\varepsilon^2} \|s_f - s_g\|_{L^2(\phi^\star)}^2 \lesssim \varepsilon^{-2} \|f - g\|_{L^2(\phi^p)}.
\end{aligned}
$$

Moreover, by the generalized Hölder's inequality with parameters $(4, 2, \infty, 4)$, we see that

$$
\begin{aligned}
&\left\| \bar{y} s_g [S_{T,f} - S_{T,g}] 1_{\{|s_g - s_f| \geq \varepsilon/2\}} \right\|_{L^1(\phi^\star)} \\
&\leq \|\bar{y}\|_{L^4(\phi^\star)} \|s_g\|_{L^2(\phi^\star)} \|S_{T,f} - S_{T,g}\|_{L^\infty(\phi^\star)} \left\| 1_{\{|s_g - s_f| \geq \varepsilon/2\}} \right\|_{L^4(\phi^\star)} \\
&\leq 2 \|\bar{y}\|_{L^4(\phi^\star)} \|s_g\|_{L^2(\phi^\star)} B_{\mathcal{S}_0} P(|s_g - s_f| \geq \varepsilon/2)^{1/4} \\
&\lesssim \varepsilon^{-1/2} \|f - g\|_{L^2(\phi^p)}^{1/4}.
\end{aligned}
$$

Combining our bounds for the three terms on the right-hand side of (S14), we have shown that

$$\|\bar{y} s_g [S_{T,f} - S_{T,g}]\|_{L^1(\phi^\star)} \lesssim \varepsilon + \varepsilon^{-\alpha} \|f - g\|_{L^2(\phi^p)}^{\alpha/2} + \varepsilon^{-1/2} \|f - g\|_{L^2(\phi^p)}^{1/4}. \tag{S15}$$

The above provides our bound for the (S7) term from the main expression.

We now study the (S8) term from the main expression. We start by decomposing this term as

$$\int [s_f^2 S_{T,f}^2 - s_g^2 S_{T,g}^2] d\phi^\star = \int S_{T,f}^2 (s_f^2 - s_g^2) d\phi^\star + \int s_g^2 [S_{T,f}^2 - S_{T,g}^2] d\phi^\star,$$

where for brevity, we have suppressed the dependence on $s_f$, $s_g$, $S_{T,f}$, and $S_{T,g}$ on their arguments. By (S12), the first term is bounded by a constant times $\|f - g\|_{L^2(\phi^p)}$. For the second term, we note that the uniform bound on $S_{T,f}$ and $S_{T,g}$ shows that

$$\|s_g^2 [S_{T,f}^2 - S_{T,g}^2]\|_{L^1(\phi^\star)} \lesssim \|s_g^2 [S_{T,f} - S_{T,g}]\|_{L^1(\phi^\star)}$$

Similarly to as we did when studying (S7), we can use (S13) and the triangle inequality to write

$$\begin{aligned} \|s_g^2 [S_{T,f} - S_{T,g}]\|_{L^1(\phi^\star)} \leq & \left\| s_g^2 [S_{T,f} - S_{T,g}] 1_{\{s_g \leq \varepsilon\}} \right\|_{L^1(\phi^\star)} \\ & + \left\| s_g^2 [S_{T,f} - S_{T,g}] 1_{\{s_g > \varepsilon, |s_f - s_g| < \varepsilon/2\}} \right\|_{L^1(\phi^\star)} \\ & + \left\| s_g^2 [S_{T,f} - S_{T,g}] 1_{\{|s_g - s_f| \geq \varepsilon/2\}} \right\|_{L^1(\phi^\star)}. \end{aligned}$$

The first term on the right upper bounds by a constant times $\varepsilon^2$. The analyses of the second and third terms are similar to the analysis of the analogous terms from (S7). A minor difference between the study of these terms and that of (S7) is that, when applying Hölder's inequality to separate the terms in each normed expression, we use (v) to ensure that $\|s_g\|_{L^{q'}(\phi^\star)} < \infty$ for some $q' > 2$. This helps us deal with the fact that $s_g^2$, rather than $s_g$, appears in the normed expressions above. Due to the similarity of the arguments to those given for (S7), the calculations for controlling the second and third terms are omitted. After the relevant calculations, we end up showing that, like (S7), (S8) is bounded by a constant times the right-hand side of (S15).

To study (S9) from the main expression, we rewrite the integral as

$$\begin{aligned} & \int \left[ (\bar{f}(\boldsymbol{x}) - f(x_0)) s_f(\boldsymbol{d}) S_{T,f}(\boldsymbol{d}, x_0) - (\bar{g}(\boldsymbol{x}) - g(x_0)) s_g(\boldsymbol{d}) S_{T,g}(\boldsymbol{d}, x_0) \right] d\phi^\star \\ & = \int s_f(\boldsymbol{d}) S_{T,f}(\boldsymbol{d}, x_0) [\bar{f}(\boldsymbol{x}) - \bar{g}(\boldsymbol{x}) + f(x_0) - g(x_0)] d\phi^\star \\ & \quad + \int S_{T,f}(\boldsymbol{d}, x_0) (\bar{g}(\boldsymbol{x}) + g(x_0)) (s_f - s_g)(\boldsymbol{d}) d\phi^\star \\ & \quad + \int s_g(\boldsymbol{d}) (\bar{g}(\boldsymbol{x}) + g(x_0)) [S_{T,f}(\boldsymbol{d}, x_0) - S_{T,g}(\boldsymbol{d}, x_0)] d\phi^\star. \end{aligned}$$

Each of the terms in the expansion can be bounded using similar techniques to those used earlier in this proof. Combining our bounds on (S5) through (S9), we see that

$$|R(T, P_f) - R(T, P_g)| \lesssim \varepsilon^{-2} \|f - g\|_{L^2(\phi^p)} + \varepsilon.$$

As $f, g$ were arbitrary, we see that, for any sequence $\{f_k\}$ in $\mathcal{F}$ such that $f_k \to f$ in $L^2(\phi^p)$ as $k \to \infty$, it holds that $\limsup_k |R(T, P_{f_k}) - R(T, P_f)| \lesssim \varepsilon$. As $\varepsilon \in (0, 1)$ was arbitrary, this shows that $R(T, P_{f_k}) \to R(T, P_f)$ as $k \to \infty$. Hence, $P \mapsto R(T, P)$ is continuous in this example. ■

## Appendix D. Further details on numerical experiments

### D.1 Meta-Learning Benchmarks

We implemented MAML via the learn2learn python package (Arnold et al., 2020), which in turn makes use of the Torchmeta package (Deleu et al., 2019) when generating the sinusoid functions. We trained MAML on a total of $10^6$ datasets with a batch size of 25 datasets and used the same learning rates and number of adaptation steps as were used in `learn2learn/examples/maml_sine.py`. We tried two network architectures, namely the same two-hidden layer perceptron architecture that was used in the sinusoid experiments in Finn et al. (2017) and a larger network whose hidden layers contained the same number of nodes (40) but that used a total of five hidden layers. For each of the three regression settings considered (sinusoid, Gaussian process with a 1-dimensional feature, and Gaussian process with a 5-dimensional feature), we reported results for the architecture that performed best across the sample sizes considered. This ended up corresponding to reporting results for the smaller network architecture across all three settings.

For the Gaussian process example with a 1-dimensional feature, we used the implementation of CNPs provided by Jiang (2021), which corresponds to a Pytorch implementation of the code from Garnelo et al. (2018). We also modified this code so that it could apply to the sinusoidal regression example and the Gaussian process example where the feature is 5-dimensional. The CNPs were updated over the same number of iterations and using the same batch size as AMC, namely $10^6$ and 25, respectively. We tried two network architectures for the CNPs, namely the same architecture as was used in Garnelo et al. (2018), with the input size modified in one of the Gaussian process settings to account for the 5-dimensional feature, and also a deeper architecture that has a similar number of hidden layers as does the architecture used for AMC. In particular, the encoder and decoder in this larger architecture each had nine hidden layers consisting of 100 nodes. Similar to as we did for MAML, for each of the three regression settings considered, we reported results for the architecture that performed best across the sample sizes considered. This corresponded to reporting CNP results for the smaller architecture for the Gaussian process with a 5-dimensional feature, and the larger architecture for the Gaussian process with a 1-dimensional feature and the sinusoidal regression.

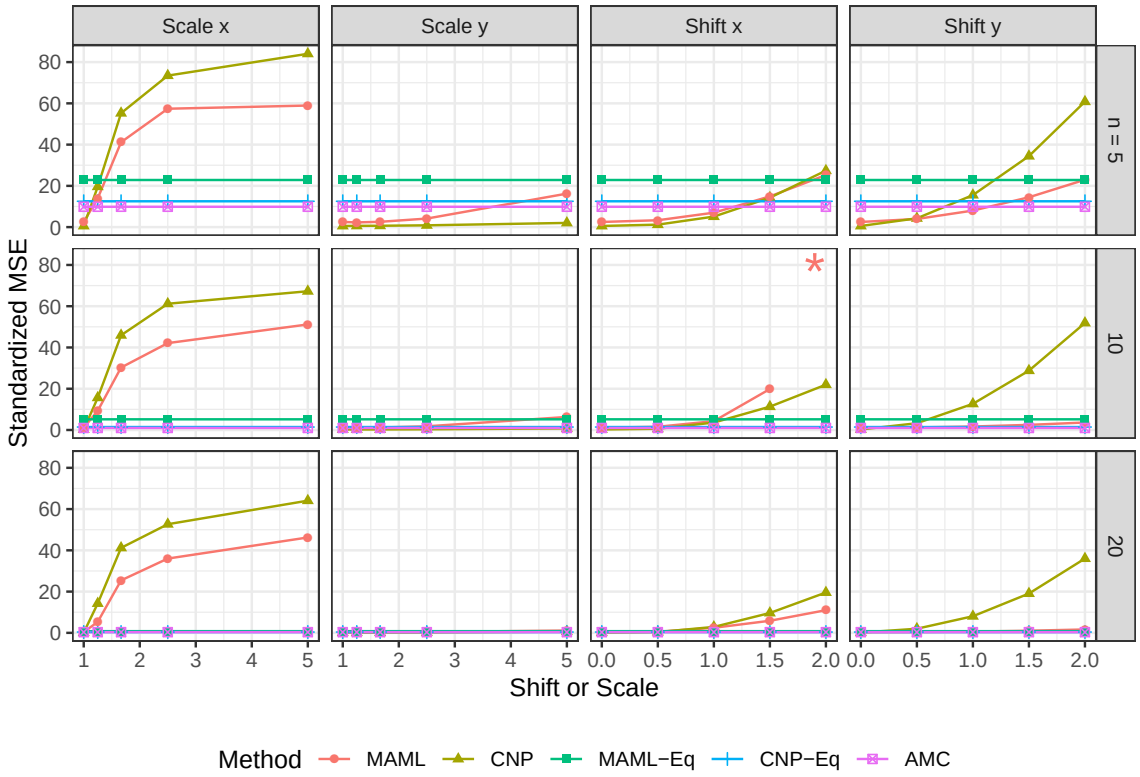



Figure S5: Bayesian standardized MSE ($E_\Pi[R(T,P)]$, where $R$ is defined in Eq. 1) of the five meta-learning algorithms considered in the sinusoidal regression example when the feature $x$ or the outcome $y$ is scaled down by a multiplicative factor (left two columns) or when $x$ or $y$ is shifted by an additive factor (right two columns). For reference, the numbers reported in Table 1 in the main text are equal to the standardized MSE reported on the far-left side of each facet times the variance of the error (0.09). The three equivariant procedures (MAML-Eq, CNP-Eq, and AMC) have constant standardized MSE under the shifts and rescalings considered. The non-equivariant procedures, namely MAML and CNPs, are sensitive even to small shifts or rescalings of $x$, and CNPs are also sensitive to small shifts in $y$.

## D.2 Comparing to Analytically-Derived Estimators with Known Theoretical Performance Guarantees

### D.2.1 Preliminaries

We now introduce notation that will be useful for defining $\Gamma_1$ in the two examples. In both examples, all priors in $\Gamma_1$ imply the same prior $\Pi_X$ over the distribution $P_X$ of the features. This prior $\Pi_X$ imposes that the $\Sigma$ indexing $P_X$ is equal in distribution to $\mathrm{diag}(W^{-1})^{-1/2}W^{-1}\mathrm{diag}(W^{-1})^{-1/2}$, where $W$ is a $p \times p$ matrix drawn from a Wishart distribution with scale matrix $2\,\mathrm{Id}_p$ and 20 degrees of freedom, and $\mathrm{diag}(W^{-1})$ denotes a

matrix with the same diagonal as $W^{-1}$ and zero in all other entries. The expression for $\Sigma$ normalizes by $\text{diag}(W^{-1})^{-1/2}$ to ensure that the diagonal of $\Sigma$ is equal to $\mathbf{1}_p$, which we require of distributions in $\mathcal{P}_X$. We let $\Gamma_\mu$ be a collection of Markov kernels $\kappa : \mathcal{P}_X \to \mathcal{R}$, so that, for each $\kappa$ and $P_X \in \mathcal{P}_X$, $\kappa(\cdot, P_X)$ is a distribution on $\mathcal{R}$. The collections $\Gamma_\mu$ differ in the two examples, and will be presented in the coming subsections. Let $\text{Unif}(\mathcal{B})$ denote a uniform distribution over the permutations in $\mathcal{B}$. For each $\kappa \in \Gamma_\mu$, we let $\Pi_\kappa$ represent a prior on $\mathcal{P}_1$ from which a draw $P$ can be generated by sampling $P_X \sim \Pi_X$, $\mu | P_X \sim \kappa(\cdot, P_X)$, and $B | P_X, \mu \sim \text{Unif}(\mathcal{B})$, and subsequently returning the distribution of $(X, \mu(BX) + \epsilon_P)$, where $X \sim P_X$ and $\epsilon_P \sim N(0,1)$ are independent. We let $\Gamma_1 := \{\Pi_\kappa : \kappa \in \Gamma_\mu\}$. For a general class of estimators $\mathcal{T}$, enforcing that each draw $P$ has a regression function $\mu_P$ of the form $x \mapsto \mu(Bx)$ for some permutation $B$ is useful because it allows us to restrict the class $\Gamma_\mu$ so that each function in this class only depends on the first $\mathfrak{s}$ coordinates of the input, while yielding a regression function $\mu_P$ that may depend on any arbitrary collection of $\mathfrak{s}$ out of the $p$ total coordinates. For the equivariant class that we consider (Algorithm 2), enforcing this turns out to be unnecessary – the invariance of functions in $\mathcal{T}$ to permutations of the features implies that the Bayes risk of each $T \in \mathcal{T}$ remains unchanged if the random variable $B$ defining $\Pi_\kappa \in \Gamma_1$ is replaced by a degenerate random variable that is always equal to the identity matrix. Nonetheless, allowing $B$ to be a random draw from $\text{Unif}(\mathcal{B})$ allows us to ensure that our implied collection of priors $\Gamma$ satisfies P1, P2, and P3, thereby making the implied $\Gamma$ compatible with the preservation conditions imposed in Section 2.

We now use the notation of Kingma and Ba (2014) to detail the hyperparameters that we used. In all settings, we set $(\beta_2, \epsilon) = (0.999, 10^{-8})$. Whenever we were updating the prior network, we set the momentum parameter $\beta_1$ to 0, and whenever we were updating the estimator network, we set the momentum parameter to 0.25. The parameter $\alpha$ differed across settings. In the sparse linear regression setting with $\mathfrak{s} = 1$, we found that choosing $\alpha$ small helped to improve stability. Specifically, we let $\alpha = 0.0002$ when updating both the estimator and prior networks. In the sparse linear regression setting with $\mathfrak{s} = 5$, we used the more commonly chosen parameter setting of $\alpha = 0.001$ for both networks. In the FLAM example, we chose $\alpha = 0.001$ and $\alpha = 0.005$ for the estimator and prior networks, respectively.

The learning rates were of the estimator and prior networks were decayed at rates $t^{-0.15}$ and $t^{-0.25}$, respectively. Such two-timescale learning rate strategies have proven to be effective in stabilizing the optimization problem pursued by generative adversarial networks (Heusel et al., 2017). As noted in Fiez et al. (2019), using two-timescale strategies can cause the optimization problem to converge to a differential Stackelberg, rather than a differential Nash, equilibrium. Indeed, under some conditions, the two-timescale strategy that we use is expected to converge to a differential Stackelberg equilibrium in the hierarchical two-player game where a prior $\Pi$ is first selected from $\Gamma$, and then an estimator $T$ is selected from $\mathcal{T}$ to perform well against $\Pi$. An optimal prior $\Pi^\star$ in this game is called $\Gamma$-least favorable, in the sense that this prior maximizes $\inf_{T \in \mathcal{T}} r(T, \cdot)$ over $\Gamma$. For a given $\Gamma$-least favorable prior $\Pi^\star$, an optimal estimator $T^\star$ in this game is a Bayes estimator against $\Pi^\star$, that is, an estimator that minimizes $r(\cdot, \Pi^\star)$ over $\mathcal{T}$. This $T^\star$ may not necessarily be a $\Gamma$-minimax strategy, that is, $T^\star$ may not minimize $\sup_{\Pi \in \Gamma} r(\cdot, \Pi)$ over $\mathcal{T}$. Nevertheless, we note that, under appropriate conditions, the two notions of optimality necessarily agree. Though such a theoretical guarantee is not likely to hold in our experiments given the neural network

parameterizations that we use, we elected to use this two-timescale strategy because of the improvements in stability that we saw.

In all settings, the prior and estimator were updated over $10^6$ iterations using batches of 100 datasets. For each dataset, performance is evaluated at 100 values of $x_0$.

### D.2.2 Sparse linear regression

We now introduce notation that will be useful for presenting the collection $\Gamma_\mu$ in the sparse linear regression example. For a function $G : \mathbb{R} \to \mathbb{R}$ and a distribution $P_X \in \mathcal{P}_X$, we let $\kappa_G(\cdot, P_X)$ be equal to the distribution of

$$x \mapsto \left( U_0 \frac{\left(e^{G(U_1)}, \ldots, e^{G(U_{\mathfrak{s}})}, 0, \ldots, 0\right)}{\sum_{j=1}^{\mathfrak{s}} e^{G(U_j)}} \right)^\top x,$$

where $U_0 \sim \text{Unif}(-5, 5)$ and $(U_1, \ldots, U_{\mathfrak{s}}) \sim N(\mathbf{0}_{\mathfrak{s}}, \text{Id}_{\mathfrak{s}})$ are drawn independently. Notably, here $\kappa_G(\cdot, P_X)$ does not depend on $P_X$. We let $\Gamma_\mu := \{\kappa_G : G \in \mathscr{G}\}$, where $\mathscr{G}$ takes different values when $\mathfrak{s} = 1$ and when $\mathfrak{s} = 5$. When $\mathfrak{s} = 1$, $\mathscr{G}$ consists of all four-hidden layer perceptrons with identity output activation, where each hidden layer consists of forty leaky ReLU units. When $\mathfrak{s} = 5$, $\mathcal{G}$ consists of all four-hidden layer neural networks with identity output activation, but in this case each layer is a multi-input-output channel equivariant layer as described in Eq. 22 of Zaheer et al. (2017). Each hidden layer is again equipped with a ReLU activation function. The output of each such network is equivariant to permutations of the $\mathfrak{s} = 5$ inputs.

In each sparse linear regression setting considered, we initialized the estimator network by pretraining for 5,000 iterations against the initial fixed prior network. After these 5,000 iterations, we then began to adversarially update the prior network against the estimator network.

Five thousand Monte Carlo replicates were used to obtain the performance estimates in Table 2.

### D.2.3 Fused lasso additive model

When discussing the FLAM example, we will write $x_j$ to denote the $j^{\text{th}}$ feature, that is, we denote a generic $x \in \mathcal{X}$ by $x = (x_1, x_2, \ldots, x_p)$. We emphasize this to avoid any notational confusion with the fact that, elsewhere in the text, $X_i \in \mathcal{X}$ is used to denote the random variable corresponding to the $i^{\text{th}}$ observation.

In the FLAM example, each prior $\kappa_G$ in $\Gamma_\mu$ is indexed by a function $G : \mathbb{R}^{\mathfrak{s}+2} \to [0, \infty)^{\mathfrak{s}}$ belonging to the collection of four-hidden layer perceptrons with identity output activation, where each hidden layer consists of forty leaky ReLU units. Specifically, $\kappa_G(\cdot, P_X)$ is a distribution over generalized additive models $x \mapsto \sum_{j=1}^p \mu_j(x_j)$ for which each component $\mu_j$ is piecewise-constant and changes values at most 500 times. To obtain a draw $\mu_P$ from $\kappa_G(\cdot, P_X)$, we can first draw 500 iid observations from $P_X$ and store these observations in the matrix $\tilde{\boldsymbol{X}}$. Each component $\mu_j$ can only have a jump at the 500 points in $\tilde{\boldsymbol{X}}_{*j}$. The magnitude of each jump is defined using the function $G$ and the sign of the jump is defined uniformly at random. More specifically, these increments are defined based on the independent sources of noise $(H_{jk} : j = 1, \ldots, p; k = 1, \ldots, 500)$, which is an

iid collection of Rademacher random variables, and $(U_k : k = 1, \ldots, 500)$, which is an iid collection of $N(\mathbf{0}_{\mathfrak{s}+2}, \mathrm{Id}_{\mathfrak{s}+2})$ random variables. The component $\mu_j$ is chosen to be proportional to the function $f_j(x_j) = \sum_{k=1}^{500} H_{jk} G(U_k)_j I\{x_j \geq \tilde{\boldsymbol{X}}_{kj}\}$. The proportionality constant $c := \sum_{j=1}^{p} \sum_{k=1}^{500} G(U_k)_j$ is defined so that the function $\mu_P(x) = c^{-1} \sum_{j=1}^{p} f_j(x_j)$ saturates the constraint $\|v(\mu)\|_1 \leq M$ that is imposed by $\mathcal{R}$. To recap, the random draw $\mu_P$ from $\kappa_G(\cdot, P_X)$ can be obtained by independently drawing $\tilde{\boldsymbol{X}}$, $(H_{j,k} : j, k)$, and $(U_k : k)$, and subsequently following the steps described above to define the corresponding proportionality constant $c$ and components $f_j$, $j = 1, \ldots, p$.

We evaluated the performance of the learned prediction procedures using a variant of the simulation scenarios 1-4 from the paper that introduced FLAM (Fig. 2 in Petersen et al., 2016). As presented in that work, the four scenarios have $p$ independent Unif$(-2.5, 2.5)$ features, with the components corresponding to $\mathfrak{s}_0 = 4$ of these features being nonzero. These scenarios offer a range of smoothness settings, with scenarios 1-4 enforcing that the components be (1) piecewise constant, (2) smooth, (3) a mix of piecewise constant and smooth functions, and (4) constant in some areas of its domain and highly variable in others. To evaluate our procedures trained with $\|v(\mu_P)\|_0 \leq 5$, we used the R function `sim.data` in the `flam` package (Petersen, 2018) to generate training data from the scenarios in Petersen et al. (2016) with $p = 10$ features. We then generated new outcomes by rescaling the regression function by a positive multiplicative constant so that $\|v(\mu_P)\|_1 = 10$, and subsequently added standard Gaussian noise. To evaluate our procedures trained at sparsity level $\mathfrak{s} = 1$ in a given scenario, we defined a prior over the regression function that first randomly selects one of the four signal components, then rescales this component so that it has total variation equal to 10, and then sets all other components equal to zero. Outcomes were generated by adding Gaussian noise to the sampled regression function. We compared our approach to the FLAM method as implemented in the `flam` package when, in the notation of Petersen et al. (2016), $\alpha = 1$ and $\lambda$ was chosen numerically to enforce that the resulting regression function estimate $\hat{\mu}$ satisfied $\|v(\hat{\mu})\|_1 \approx 10$. Choosing $\lambda$ in this fashion is reasonable in light of the fact that $\|v(\mu_P)\|_1 = 10$ for all settings considered.

Two thousand Monte Carlo replicates were used to obtain the performance estimates in Table 3.

## Appendix E. Additional details and results for data experiments

### E.1 Datasets

We start by describing the six datasets that we considered that are available through the UCI Machine Learning Repository (Dua and Graff, 2017). The first dataset ("abalone") contains information on 4177 abalones. The objective is to predict their age based on 7 features, namely length, diameter, height, whole weight, shucked weight, viscera weight, and shell weight (Nash et al., 1994). The second dataset ("airfoil") is from the National Aeronautics and Space Administration (NASA) that contains information on 1,503 airfoils at various wind tunnel speeds and angles of attack (Brooks et al., 1989). The objective is to estimate the scaled sound level in decibels. Five features are available, namely frequency, angle of attack, chord length, free-stream velocity, and suction side displacement thickness. The third dataset ("fish") was originally used to develop quantitative structure-activity relationship (QSAR) models to predict acute aquatic toxicity towards the fathead minnow. This dataset contains

908 total observations, each of which corresponds to a distinct chemical. The outcome is the $LC_{50}$ for that chemical, which represents the concentration of the chemical that is lethal for 50% of test fish over 96 hours. Six features that describe the molecular characteristics of the chemical are available — see the UCI Machine Learning Repository and Cassotti et al. (2015) for details. The fourth and fifth datasets contain information on 1,599 red wines ("wine-red") and 4,898 white wines ("wine-white") (Cortez et al., 2009). The objective is to predict wine quality score based on 11 available features — see the UCI Machine Learning Repository and Cassotti et al. (2015) for details. The sixth dataset ("yacht") contains information on 308 sailing yachts. The objective is to learn to predict a ship's performance in terms of residuary resistance. Six features describing a ship's dimensions and velocity are available, namely: the longitudinal position of the center of buoyancy, the prismatic coefficient, the length-displacement ratio, the beam-draught ratio, the length-beam ratio, and the Froude number. See Gerritsma et al. (1981) for more information on these features.

The seventh and eighth of our datasets that we considered were used to illustrate regression procedures in James et al. (2013). They are available through the `ISLR` R package (James et al., 2017). One of these datasets ("college") consists of information on 777 colleges in the United States. The objective is to predict out-of-state tuition based on 16 available continuous features. The second of these datasets ("hitters") contains information on 322 baseball players. The objective is to predict salary based on the 16 available continuous features. The ninth dataset ("LAozone") was used to illustrate regression procedures in (Friedman, 2001). It consists of 330 daily meteorological measurements in the Los Angeles basin in 1976. The objective is to predict ozone levels based on 9 available features. The final dataset that we considered ("happiness") was used in the paper that introduced the FLAM to illustrate the performance of the method (Petersen et al., 2016). This dataset consists of information about 109 countries. The objective is to predict the national happiness level via 12 country-level features.

### E.2 Additional results for data experiments

Table S5 displays the cross-validated MSEs across the ten datasets in numerical form. Figure S6 shows the performance of the individual linear algorithms considered at different sparsity levels, and Figure S7 shows the same results but for the stacking algorithms.

## Appendix F. Performance of symmetrized estimators in experiments

We now present the additional experimental results that we alluded to in Section 8. These results were obtained by symmetrizing the meta-learned AMC100 and AMC500 estimators whose performance was reported in Section 5. In particular, we symmetrized a given AMC estimator $T$ as

$$T^{\text{sym}}(\boldsymbol{x}, \boldsymbol{y})(x_0) := \frac{1}{2}\left[T(\boldsymbol{x}, \boldsymbol{y}) - T(\boldsymbol{x}, -\boldsymbol{y})(x_0)\right].$$

When reporting our experimental results, we refer to the symmetrized estimator derived from the meta-learned AMC100 and AMC500 estimators as 'symmetrized AMC100' and 'symmetrized AMC500', respectively. We emphasize that these symmetrized estimators are

| | Features | OLS | Lasso | AMC Linear (ours) | FLAM | AMC FLAM (ours) | Stacked Existing | Stacked AMC (ours) | Stacked Both (ours) |
|---|---|---|---|---|---|---|---|---|---|
| college | 10 | 0.414 | 0.397 | **0.377** | **0.392** | 0.395 | 0.358 | 0.354 | **0.348** |
| happiness | 10 | **0.270** | 0.277 | 0.275 | 0.315 | **0.311** | 0.280 | 0.261 | **0.256** |
| hitters | 10 | 0.667 | **0.660** | 0.662 | 0.626 | **0.619** | 0.602 | 0.615 | **0.585** |
| wine-red | 10 | 0.768 | **0.737** | 0.746 | 0.826 | **0.776** | 0.737 | 0.737 | **0.731** |
| wine-white | 10 | 0.833 | **0.814** | 0.824 | 0.899 | **0.860** | 0.809 | 0.815 | **0.802** |
| LAozone | 9 | 0.341 | **0.335** | 0.337 | **0.335** | 0.367 | 0.310 | 0.320 | **0.309** |
| abalone | 7 | 0.559 | 0.546 | **0.540** | 0.709 | **0.675** | 0.539 | 0.538 | **0.537** |
| fish | 6 | **0.471** | 0.475 | 0.480 | **0.544** | 0.554 | **0.464** | 0.476 | 0.468 |
| yacht | 6 | 0.381 | 0.372 | **0.350** | **0.019** | 0.035 | **0.015** | 0.029 | **0.015** |
| airfoil | 5 | **0.524** | 0.525 | 0.528 | **0.617** | 0.701 | **0.516** | 0.523 | 0.520 |

Table S5: Cross-validated MSEs on the ten datasets. The first 5 datasets had the same number of features the same as were used during meta-training (10), whereas the others had fewer. For each of the three categories (linear estimators, FLAM estimators, and stacked estimators) and each dataset, the algorithm with the lowest Monte Carlo MSE is emphasized in bold. There was no clear ordering between the performance of AMC Linear and the existing estimators (OLS and lasso). AMC FLAM tended to outperform FLAM when the number of features was the same as were used during meta-training, and be slightly outperformed otherwise. When the number of features was the same as were used during meta-training, stacking the existing and AMC estimators consistently outperformed all other approaches. When there were fewer features than were used during meta-training, stacking all available learners performed similarly to stacking only the existing algorithms and still outperformed all individual learners.

derived directly from the AMC100 and AMC500 fits that we reported in Section 5 – we did not rerun our AMC meta-learning algorithm to obtain these estimators.

Table S6 reports the results for the linear regression example. In many settings, the two approaches performed similarly. However, in the sparse setting, the improvements that resulted from symmetrization sometimes resulted in the MSE being cut in half. In one setting (dense, interior, $n = 100$), AMC100 outperformed symmetrized AMC100 slightly – though not deducible from the table, we note here that the difference in MSE in this case was less than 0.003, and it seems likely that this discrepancy is a result of Monte Carlo error. Table S6 reports the results for the fused lasso additive model example. Symmetrization led to a reduction in MSE in most settings. In all other settings, the MSE remained unchanged.

## References

Sébastien MR Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for meta-learning research. *arXiv preprint arXiv:2008.12284*, 2020.

James O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Science & Business Media, 1985.

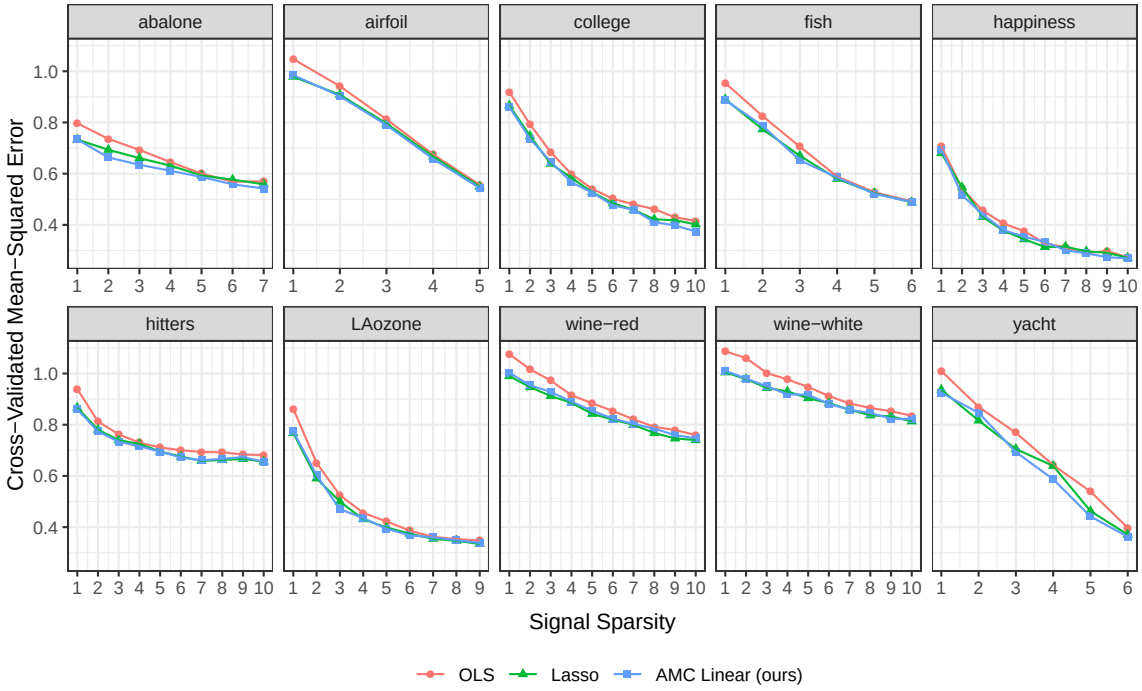



Figure S6: Performance of OLS, lasso, and AMC Linear at different sparsity levels. For each training-validation split of the data, between 1 and $q$ features are selected at random from the original dataset (x-axis), where $q$ is the minimum of 10 and the total number of features in the dataset, and Gaussian noise features are then added so that there are 10 total features. Therefore, the signal is expected to become denser and stronger as the x-axis value increases. AMC Linear consistently outperformed OLS and performed similarly to or better than lasso in most settings (54% of all sparsity-dataset pairs).

Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

Patrick Billingsley. *Convergence of probability measures*. Wiley, 1999.

Tom Bosc. Learning to learn neural networks. *arXiv preprint arXiv:1610.06072*, 2016.

Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Thomas F Brooks, D Stuart Pope, and Michael A Marcolini. *Airfoil self-noise and prediction*, volume 1218. National Aeronautics and Space Administration, Office of Management . . . , 1989.

M Cassotti, D Ballabio, R Todeschini, and V Consonni. A similarity-based qsar model for predicting acute toxicity towards the fathead minnow (pimephales promelas). *SAR and QSAR in Environmental Research*, 26(3):217–243, 2015.

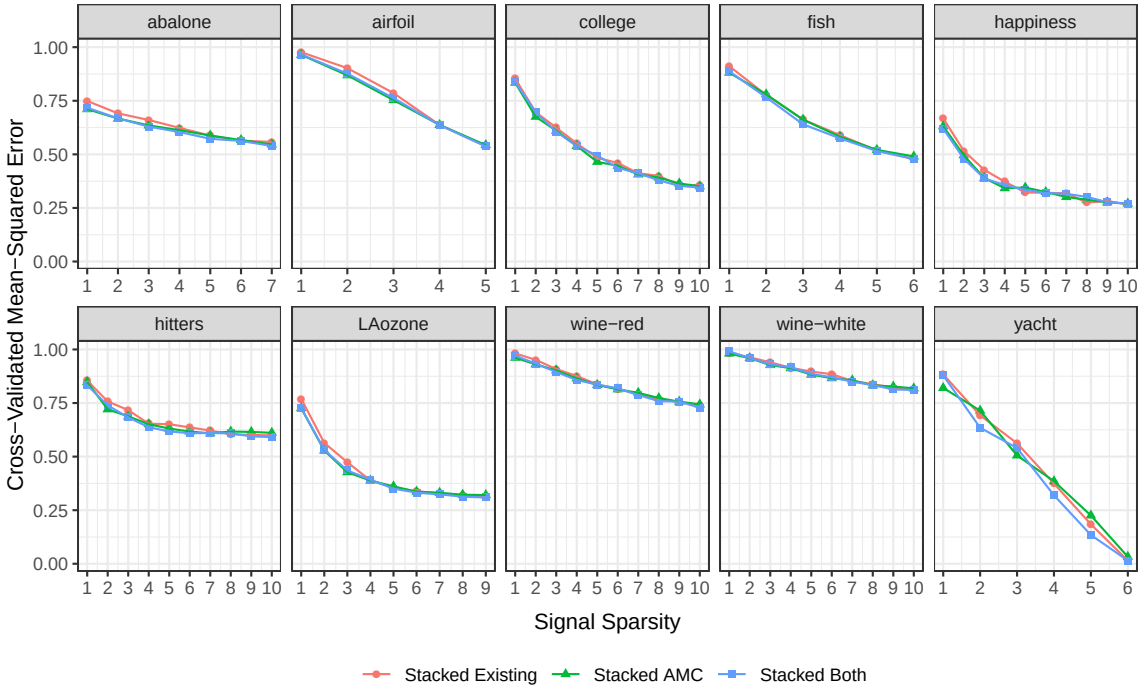



Figure S7: Performance of the three stacking algorithms at different sparsity levels. For each training-validation split of the data, between 1 and $q$ features are selected at random from the original dataset (x-axis), where $q$ is the minimum of 10 and the total number of features in the dataset, and Gaussian noise features are then added so that there are 10 total features. Therefore, the signal is expected to become denser and stronger as the x-axis value increases. Though all algorithms performed similarly, the stacking algorithm that combined all available algorithms (Stacked Both) performed slightly better than the others in a majority of the settings (53% of all sparsity-dataset pairs), and Stacked AMC performed best in most other settings (39% of all sparsity-dataset pairs).

Gary Chamberlain. Econometric applications of maxmin expected utility. *Journal of Applied Econometrics*, 15(6):625–644, 2000.

Kung-Ching Chang. *Methods in nonlinear analysis*. Springer Science & Business Media, 2006.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

Donald L Cohn. *Measure theory*. Springer, 2013.

John B Conway. *A course in functional analysis*, volume 96. Springer, 2010.

(a) Sparse signal

| | Boundary | | Interior | |
|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 |
| OLS | 0.12 | 0.02 | 0.12 | 0.02 |
| Lasso | 0.06 | 0.01 | 0.06 | 0.01 |
| AMC100 (ours) | 0.02 | <0.01 | 0.11 | 0.09 |
| Symmetrized AMC100 (ours) | 0.02 | <0.01 | **0.06** | **0.04** |
| AMC500 (ours) | 0.02 | <0.01 | 0.07 | 0.04 |
| Symmetrized AMC500 (ours) | 0.02 | <0.01 | **0.06** | **0.03** |

(b) Dense signal

| | Boundary | | Interior | |
|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 |
| OLS | 0.13 | 0.02 | 0.13 | 0.02 |
| Lasso | 0.11 | 0.02 | 0.09 | 0.02 |
| AMC100 (ours) | 0.10 | 0.04 | **0.08** | 0.02 |
| Symmetrized AMC100 (ours) | **0.09** | **0.03** | 0.09 | 0.02 |
| AMC500 (ours) | 0.09 | 0.02 | 0.09 | 0.02 |
| Symmetrized AMC500 (ours) | 0.09 | 0.02 | 0.09 | 0.02 |

Table S6: MSEs based on datasets of size $n$ in the linear regression settings. All Monte Carlo standard errors are less than 0.001. Symmetrized AMC100 entries appear in bold when they had lower MSE (rounded to the nearest hundredth) than the corresponding AMC100 entry, and vice versa. Similarly, symmetrized AMC500 entries appear in bold when they had lower MSE than the corresponding AMC500 entry, and vice versa.

Paulo Cortez, Juliana Teixeira, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Using data mining for wine quality assessment. In *International Conference on Discovery Science*, pages 66–79. Springer, 2009.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108, 2004.

Mahlon M Day. Fixed-point theorems for compact convex sets. *Illinois Journal of Mathematics*, 5(4):585–590, 1961.

Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. Torchmeta: A Meta-Learning library for PyTorch, 2019. URL `https://arxiv.org/abs/1909.06576`. Available at: https://github.com/tristandeleu/pytorch-meta.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Bradley Efron and Carl Morris. Limiting the risk of bayes and empirical bayes estimators—part ii: The empirical bayes case. *Journal of the American Statistical Association*, 67(337):130–139, 1972.

Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences of the United States of America*, 39(1):42, 1953.

Tanner Fiez, Benjamin Chasnov, and Lillian J Ratliff. Convergence of learning dynamics in stackelberg games. *arXiv preprint arXiv:1906.01217*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

(a) Sparse signal

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | |
|---|---|---|---|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 | 100 | 500 | 100 | 500 |
| FLAM | 0.44 | 0.12 | 0.47 | 0.17 | 0.38 | 0.11 | 0.51 | 0.19 |
| AMC100 (ours) | 0.34 | 0.20 | 0.18 | 0.08 | 0.27 | 0.14 | 0.17 | 0.08 |
| Symmetrized AMC100 (ours) | **0.32** | **0.18** | 0.18 | 0.08 | **0.26** | **0.13** | **0.16** | 0.08 |
| AMC500 (ours) | 0.48 | 0.12 | 0.19 | 0.06 | 0.35 | 0.10 | 0.23 | 0.08 |
| Symmetrized AM5100 (ours) | **0.43** | 0.12 | **0.17** | **0.05** | **0.32** | **0.09** | **0.21** | **0.07** |

(b) Dense signal

| | Scenario 1 | | Scenario 2 | | Scenario 3 | | Scenario 4 | |
|---|---|---|---|---|---|---|---|---|
| | $n$=100 | 500 | 100 | 500 | 100 | 500 | 100 | 500 |
| FLAM | 0.59 | 0.17 | 0.65 | 0.24 | 0.53 | 0.16 | 0.76 | 0.36 |
| AMC100 (ours) | 1.20 | 0.91 | 0.47 | 0.39 | 0.87 | 0.57 | 0.30 | 0.30 |
| Symmetrized AMC100 (ours) | **1.16** | **0.84** | **0.45** | **0.37** | **0.83** | **0.52** | **0.29** | 0.30 |
| AMC500 (ours) | 0.58 | 0.15 | 0.37 | 0.08 | 0.46 | 0.12 | 0.36 | 0.09 |
| Symmetrized AM5100 (ours) | **0.55** | 0.15 | **0.36** | 0.08 | **0.43** | **0.11** | **0.34** | 0.09 |

Table S7: MSEs based on datasets of size $n$ in the FLAM settings. The Monte Carlo standard errors for the MSEs of FLAM and (symmetrized) AMC are all less than 0.04 and 0.01, respectively. Symmetrized AMC100 entries appear in bold when they had lower MSE (rounded to the nearest hundredth) than the corresponding AMC100 entry, and vice versa. Similarly, symmetrized AMC500 entries appear in bold when they had lower MSE than the corresponding AMC500 entry, and vice versa.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2018.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Sinong Geng, Houssam Nassif, Carlos A Manzanares, A Max Reppen, and Ronnie Sircar. Deep pqr: Solving inverse reinforcement learning using anchor actions. *arXiv e-prints*, pages arXiv–2007, 2020.

J Gerritsma, R Onnink, and A Versluis. Geometry, resistance and stability of the delft systematic yacht hull series. *International shipbuilding progress*, 28(328):276–297, 1981.

Subhashis Ghosal and Aad Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent, and Simon Lacoste-Julien. A variational inequality perspective on generative adversarial networks. *arXiv preprint arXiv:1802.10551*, 2018.

P W Glynn. Likelihood ratio gradient estimation: an overview. In *Proceedings of the 19th conference on Winter simulation*, pages 366–375. ACM, 1987.

Micah Goldblum, Liam Fowl, and Tom Goldstein. Adversarially robust few-shot learning: A meta-learning approach. *arXiv preprint arXiv:1910.00982v2*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Weichao Guo and Guoping Zhao. An improvement on the relatively compactness criteria. *arXiv preprint arXiv:1904.03427*, 2019.

Jason Hartford, Devon R Graham, Kevin Leyton-Brown, and Siamak Ravanbakhsh. Deep models of interactions across sets. *arXiv preprint arXiv:1803.02879*, 2018.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

Atsushi Higuchi. Symmetric tensor spherical harmonics on the n-sphere and their application to the de sitter group so (n, 1). *Journal of mathematical physics*, 28(7):1553–1566, 1987.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

G Hunt and C Stein. Most stringent tests of statistical hypotheses. *Unpublished manuscript*, 1946.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

Gareth James, Daniela Witten, Trevor Hastie, and Rob Tibshirani. *ISLR: Data for an Introduction to Statistical Learning with Applications in R*, 2017. URL `https://CRAN.R-project.org/package=ISLR`. R package version 1.2.

Shali Jiang. Conditional neural process pytorch implementation, 2021. URL `https://github.com/shalijiang/neural-process`.

Peter J Kempthorne. Numerical specification of discrete least favorable prior distributions. *SIAM Journal on Scientific and Statistical Computing*, 8(2):171–184, 1987.

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

Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

Tianyi Lin, Chi Jin, and Michael I Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331v6*, 2019.

A Luedtke, M Carone, N R Simon, and O Sofrygin. Learning to learn from data: using deep adversarial learning to construct optimal statistical procedures. *Science Advances* (in press; available online late Feb or Mar 2020), 2020.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the universality of invariant networks. *arXiv preprint arXiv:1901.09342*, 2019.

J.R. Munkres. *Topology*. Featured Titles for Topology Series. Prentice Hall, Incorporated, 2000. ISBN 9780131816299. URL `https://books.google.com/books?id=XjoZAQAAIAAJ`.

Sareh Nabi, Houssam Nassif, Joseph Hong, Hamed Mamani, and Guido Imbens. Decoupling learning rates using empirical bayes priors. *arXiv preprint arXiv:2002.01129*, 2020.

Warwick J Nash, Tracy L Sellers, Simon R Talbot, Andrew J Cawthorn, and Wes B Ford. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. *Sea Fisheries Division, Technical Report*, 48:p411, 1994.

John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

Wayne Nelson. Minimax solution of statistical decision problems by iteration. *The Annals of Mathematical Statistics*, pages 1643–1657, 1966.

Roger Fandom Noubiap and Wilfried Seidel. An algorithm for calculating $\gamma$-minimax decision rules under generalized moment conditions. *The Annals of Statistics*, 29(4):1094–1116, 2001.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Ashley Petersen. *flam: Fits Piecewise Constant Models with Data-Adaptive Knots*, 2018. URL `https://CRAN.R-project.org/package=flam`. R package version 3.2.

Ashley Petersen, Daniela Witten, and Noah Simon. Fused lasso additive model. *Journal of Computational and Graphical Statistics*, 25(4):1005–1025, 2016.

Jean-Paul Pier. *Amenable locally compact groups*. Wiley-Interscience, 1984.

Eric C Polley and Mark J Van der Laan. Super learner in prediction. Technical report, University of California, Berkeley, 2010.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Deep learning with sets and point clouds. *arXiv preprint arXiv:1611.04500*, 2016.

Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Equivariance through parameter-sharing. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2892–2901. JMLR. org, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Stuart Russell. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103, 1998.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

Chad M Schafer and Philip B Stark. Constructing confidence regions of optimal expected size. *Journal of the American Statistical Association*, 104(487):1080–1089, 2009.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Frode Terkelsen. Some minimax theorems. *Mathematica Scandinavica*, 31(2):405–413, 1973.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.

Aad W Van der Vaart, Sandrine Dudoit, and Mark J van der Laan. Oracle inequalities for multi-fold cross validation. *Statistics and Decisions*, 24(3):351–371, 2006.

Onno van Gaans. Probability measures on metric spaces. Technical report, Technical report, Delft University of Technology, 2003.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, and Daan Wierstra. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Toward multimodal model-agnostic meta-learning. *arXiv preprint arXiv:1812.07172*, 2018.

Abraham Wald. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, pages 265–280, 1945.

Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Adversarial meta-learning. *arXiv preprint arXiv:1806.03316*, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.