

# Cross-Learning from Scarce Data via Multi-Task Constrained Optimization

Leopoldo Agorio, Juan Cerviño, Miguel Calvo-Fullana, Alejandro Ribeiro, and Juan Andrés Bazerque.

**Abstract**—A learning task, understood as the problem of fitting a parametric model from supervised data, fundamentally requires the dataset to be large enough to be representative of the underlying distribution of the source. When data is limited, the learned models fail generalize to cases not seen during training. This paper introduces a multi-task *cross-learning* framework to overcome data scarcity by jointly estimating *deterministic* parameters across multiple, related tasks. We formulate this joint estimation as a constrained optimization problem, where the constraints dictate the resulting similarity between the parameters of the different models, allowing the estimated parameters to differ across tasks while still combining information from multiple data sources. This framework enables knowledge transfer from tasks with abundant data to those with scarce data, leading to more accurate and reliable parameter estimates, providing a solution for scenarios where parameter inference from limited data is critical. We provide theoretical guarantees in a controlled framework with Gaussian data, and show the efficiency of our cross-learning method in applications with real data including image classification and propagation of infectious diseases.

**Index Terms**—Supervised learning, multi-task, optimization.

## I. INTRODUCTION

The machine learning problem, in general, involves extracting information from a dataset, which is typically achieved by fitting the parameters of a model [1], whether it be a neural network or a more specific parametric function that incorporates additional knowledge about the data source. Once fitted, this parametric model can be used for classification, prediction, or estimation, serving various purposes. For example, one might want to classify input signals, as in the case of a robot or autonomous vehicle that captures an image and needs to detect whether there is an obstacle to avoid [2]. One might also want to predict, for instance, the solar energy generation capacity of a power system, in order to set prices in a day-ahead market [3]. In some cases, estimating the parameters themselves constitutes the learning objective. For example, an infection propagation model, such as for influenza or COVID, can be used to predict the number of infected individuals, but the parameters themselves contain information about how the population behaves and which policies are most effective [4].

This work was supported in part by Spain’s Agencia Estatal de Investigación under grant RYC2021-033549-I.

L. Agorio and J. A. Bazerque are with the Department of Electrical Engineering, School of Engineering, Universidad de la República, Montevideo, 11300, Uruguay (e-mail: lagorio@fing.edu.uy, jbazerque@fing.edu.uy)

J. Cerviño is with the Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: jcervino@mit.edu).

M. Calvo-Fullana is with the Department of Engineering, Universitat Pompeu Fabra, Barcelona, 08018, Spain (e-mail: miguel.calvo@upf.edu).

A. Ribeiro is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA (e-mail: aribeiro@seas.upenn.edu).

Appropriate parameters are typically found by minimizing a loss or fit function, which measures how well the model explains the data for a given parameter set. Beyond this, a fundamental design decision is whether to assume the parameters are deterministic or random, a choice that opens two distinct methodological paths [5]. In the Bayesian model, parameters are treated as random variables governed by a prior distribution, which may in turn be modeled by a second layer of hyperparameters. In the contrasting deterministic paradigm, the parameters are assumed to be fixed, unknown quantities that define the data’s underlying distribution. Both the probabilistic and deterministic approaches are valid. The choice between them often depends on one’s confidence in a potential prior distribution. For instance, if parameters are known to lie within a certain interval, imposing a uniform prior distribution over that interval can improve estimation, especially in a low-data regime. However, if this prior information is incorrect, it will introduce bias, degrading the estimation quality. In this paper, we adopt the deterministic parameter paradigm. We do not assume a prior distribution but instead seek to incorporate additional information to improve performance, particularly when working in a *low-data regime*, as it the norm in diverse fields such as medical data [6], speech recognition [7], and anomaly detection [8].

Our approach sits at the intersection of multi-task learning [9], multi-agent systems, and meta-learning [10], [11]. Multi-task learning has the potential of augmenting the data by addressing different tasks together, incorporating data from multiple sources or domains and combining it into a joint learning problem. If these sources are related, then they cross-fertilize to improve performance. Such data augmentation is particularly useful when data from all sources are scarce, or in cases in which there is a particular incipient source which has not produced enough data yet. This brings us to the concept of meta-learning, which seeks to determine whether models trained on past data are useful for explaining future data—especially when that future data comes from a previously unseen source. Through meta-learning, we can use past experience to learn quickly in a new situation, combining information from similar past cases with the limited data available in the new context. The intuition from multi-task and meta-learning leads us to propose *cross-learning* as a methodology for combining the training from multiple sources.

The central challenge lies in *what* to share between sources and *how* to share it [12], [13]. Various approaches have been proposed. In the deep learning context, some studies explore sharing common data representations, which often translates to sharing the early layers of a neural network [14], [15]. Other

works have developed deep architectures with specific blocks shared across tasks [16], [17], often empirically determining which shared blocks yield the most significant impact. In these approaches, part of the architecture is common, while another part remains private to each task. Measuring task-relatedness is also an active research area in the field of multi-task learning. Some work focuses on grouping tasks to learn them jointly within each group [18]. Other methods evaluate task differences by measuring discrepancies between their learned models [19], or use a convex surrogate of the learning objective to model task relationships [20]. In contrast to Bayesian formulations that model task-relatedness via shared priors [21], [22], our approach does not assume any prior structure on the learned functions.

In this paper, we formalize and extend cross-learning for supervised learning, a methodology based on a constrained optimization framework we originally applied in the reinforcement learning paradigm [23], [24]. The core idea is to manage the bias-variance trade-off in multi-task settings explicitly. Our preliminary work [25] drew intuition from the two extremes of multi-task learning: (i) training separate, task-specific estimators (using only local data, which can suffer from high variance) and (ii) training a single, consensus model on all data (which can introduce high bias if tasks are dissimilar). Cross-learning finds a balance between these two extremes by allowing task-specific models while imposing constraints that keep them close to each other, thereby controlling the bias-variance trade-off. This work builds on that foundation with several key contributions. First, we provide a theoretical proof that the cross-learning formulation successfully exploits this trade-off, outperforming both the naive separate and consensus approaches. Second, we generalize the method to support arbitrary functional constraints, lifting intuition from [26]. This is a critical extension, particularly for neural networks, where models with close parameters can still produce highly distant output distributions. By imposing constraints directly on the model outputs rather than the parameter space, we can more effectively regularize the model's behavior. Third, we develop new algorithms to solve these optimization problems in the dual domain. Finally, we validate our theoretical findings and algorithmic performance through experiments on two distinct problems. The first involves fitting an infection model using data from the COVID-19 pandemic, treating data from different countries as separate but related tasks. The second is an image classification task to distinguish objects using distinct representations of the same object. The experimental results confirm that cross-learning consistently outperforms the alternative approaches of training completely separate models or merging all data into a single consolidated model.

## II. MULTI-TASK LEARNING

We consider the problem of jointly optimizing a set of parametric functions, each solving a regression or classification task over a separate dataset. These datasets are collected from different but *related* sources; therefore, we seek a method to fit the *deterministic* parameters of the associated functions jointly. Formally, consider a finite set of tasks  $t \in [1, \dots, T]$ , and let

the parametric function  $f : \mathcal{X} \times \Theta \rightarrow \mathcal{Y}$  be the map between the input space  $\mathcal{X} \subset \mathbb{R}^P$ , and output space  $\mathcal{Y} \subset \mathbb{R}^Q$  parameterized by  $\theta \in \mathbb{R}^S$ . Our goal is to find the parameterizations that minimize the expected loss  $\ell : \mathbb{R}^Q \times \mathbb{R}^Q \rightarrow \mathbb{R}^+$  over the probability distribution  $p_t(x, y)$ ,

$$\{\theta_t^*\}_{t=1}^T = \arg \min_{\{\theta_t\}_{t=1}^T} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{p_t(x, y)} [\ell(y, f(x, \theta_t))]. \quad (\text{P}_{\text{ML}})$$

We do not assume, however, to have access to the distributions  $p_t(x, y)$ . Instead, we align with the multi-task learning literature, assuming that each task  $t$  is equipped with a dataset  $\mathcal{D}_t$ , containing  $N_t$  samples  $(x_i, y_i) \in (\mathcal{X}, \mathcal{Y})$ ,  $i = 1, \dots, N_t$ . The datasets  $\mathcal{D}_t$  are assumed to be drawn according to the unknown joint probabilities  $p_t(x, y)$ . The empirical version of the multi-task learning problem (P<sub>ML</sub>) can thus be written as,

$$\{\hat{\theta}_t\}_{t=1}^T = \arg \min_{\{\theta_t\}_{t=1}^T} \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(y_i, f(x_i, \theta_t)). \quad (\text{P}_S)$$

The sole difference between the empirical multi-task learning problem (P<sub>S</sub>) and its statistical counterpart (P<sub>ML</sub>) is the fact that the expectations over  $p_t(x, y)$  have been replaced by an empirical sum over  $\mathcal{D}_t$ . Under a deterministic model for the parameters  $\theta_t$ , this empirical problem (P<sub>S</sub>) is *separable* across tasks. However, given that the number of samples  $N_t$  is finite, solving for the parameters  $\hat{\theta}_t$  separately can be detrimental. If there is correlation across tasks, solving them separately can lead to a loss of information, as data from one task could be beneficial for learning others. Such separate estimation defeats the primary purpose of multi-task learning, which is to learn tasks jointly. To leverage the joint information from different sources, a simple approach is to merge all datasets and learn a common solution for all tasks, or equivalently, imposing a consensus constraint  $\theta_t = \theta$  for all  $t$ , forcing all tasks to be modeled by the same parameter  $\theta_c$ , solution to

$$\hat{\theta}_c = \arg \min_{\theta} \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(y_i, f(x_i, \theta)). \quad (\text{P}_C)$$

This *consensus* approach (P<sub>C</sub>) does provide a joint solution to our multi-task learning problem, merging data from all sources. However, forcing a strict consensus might be too restrictive in practice, as a solution that is good for all tasks might not be the best for any individual task. Hence, we explore a more balanced approach that trades off between learning individual parameters  $\hat{\theta}_t$ , as in the *separable* problem (P<sub>S</sub>), and a common parameter  $\theta_c$  as in the *consensus* multi-task learning problem (P<sub>C</sub>).

### A. Multi-Task Learning Bias-Variance Trade-Off

To motivate our proposed approach, let us present a reductive yet informative example to illustrate how bias and variance affect the solutions to (P<sub>S</sub>) and (P<sub>C</sub>).

**Example 1.** Consider the task of recovering a parameter  $\theta_t^* \in \mathbb{R}^d$  from samples corrupted by additive, zero-mean, i.i.d. noise

$$y_{tn} = \theta_t^* + \eta_{tn}, \quad n = 1, \dots, N_t \quad (1)$$

with  $\eta_{tn} \sim \mathcal{N}(0, \sigma^2)$ . To estimate  $\theta_t^*$  from the samples  $y_{tn}$ , we minimize the mean square error as follows,

$$\hat{\theta}_t = \arg \min_{\theta} \frac{1}{N_t} \sum_{n=1}^{N_t} \|y_{tn} - \theta\|^2 = \frac{1}{N_t} \sum_{n=1}^{N_t} y_{tn}. \quad (\bar{P}_S)$$

The estimator  $\hat{\theta}_t$  in  $(\bar{P}_S)$ , is a particular case of the separable estimator  $(P_S)$  with a constant regressor  $f(x, \theta) = \theta$  and square loss  $\ell(y, f(x, \theta)) = \|y - \theta\|^2$ . If the true parameters  $\{\theta_t^*\}$  are close to one another, it might be beneficial to consider a single estimator based on all available samples,

$$\hat{\theta}_c = \arg \min_{\theta} \sum_{t=1}^T \frac{1}{N_t} \sum_{n=1}^{N_t} \|y_{tn} - \theta\|^2, \quad (\bar{P}_C)$$

which is the simplified version of  $(P_C)$  for the model in (1). Notice that in this reduced case, the consensus estimator admits a closed-form solution given by the average of all samples, which can be expressed as the average  $\hat{\theta}_c = \frac{1}{T} \sum_{t=1}^T \hat{\theta}_t$  of the separate estimates  $\hat{\theta}_t$  in  $(\bar{P}_S)$ . Since the noise is zero-mean, the estimators in  $(\bar{P}_S)$  are unbiased, and by virtue of the i.i.d. model, their variance reduces with the number of samples according to

$$\text{var}(\hat{\theta}_t) = \mathbb{E} \left[ \left\| \hat{\theta}_t - \mathbb{E}[\hat{\theta}_t] \right\|^2 \right] = \frac{d}{N_t} \sigma^2. \quad (2)$$

On the other hand, we show in Appendix A, that the consensus estimator has variance  $\text{var}(\hat{\theta}_c) = \frac{1}{T^2} \sum_{t=1}^T \frac{d}{N_t} \sigma^2$ , which simplifies if all datasets have the same size  $N_t = N$ ,

$$\text{var}(\hat{\theta}_c) = \frac{1}{T} \frac{d}{N} \sigma^2. \quad (3)$$

Intuitively, pooling data from all tasks increases the effective dataset size from  $N_t = N$  in (2) to  $TN$  in (3), thereby reducing the uncertainty in the average estimate. However, this reduction in variance comes at the cost of introducing bias:

$$\mathbb{E} [\hat{\theta}_c - \theta_t^*] = \frac{1}{T} \sum_{\tau=1}^T \mathbb{E} [\hat{\theta}_\tau] - \theta_t^* = \frac{1}{T} \sum_{\tau=1}^T (\theta_\tau^* - \theta_t^*) \neq 0.$$

In the light of this basic example, we propose the following *cross-learning* estimator that balances bias and variance by lying in between the fully separable solution  $(P_S)$  and the consensus counterpart  $(P_C)$ . This approach allows the estimated parameters to differ across tasks while still combining information from multiple data sources,

$$\begin{aligned} \{\theta_t^\dagger\}, \theta_g^\dagger &= \arg \min_{\{\theta_t\}, \theta_g} \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(y_i, f(x_i, \theta_t)) \quad (P_{CL}) \\ \text{subject to: } &\|\theta_t - \theta_g\| \leq \epsilon, \quad t = 1, \dots, T. \end{aligned}$$

The *cross-learning* problem  $(P_{CL})$  introduces a task-specific parameter  $\theta_t^\dagger$  for each task and a global parameter  $\theta_g^\dagger$  shared among all tasks. This formulation connects the separable multi-task learning problem  $(P_S)$  and the consensus approach  $(P_C)$  by imposing a constraint on the closeness of the task-specific parameters to the global parameter. This constraint relaxes the strict equality of the consensus problem, allowing the solutions for different tasks to vary.

The centrality parameter  $\epsilon$  controls the degree of similarity between the task-specific solutions. A larger  $\epsilon$  allows solutions to be more task-specific, while a smaller  $\epsilon$  encourages them to be closer to the global solution. Note that, by enforcing  $\epsilon = 0$ , all solutions are forced to be equal and  $(P_{CL})$  becomes equivalent to enforce consensus in  $(P_C)$ . Conversely for a sufficiently large  $\epsilon$ , the constraint becomes inactive, and  $(P_{CL})$  is equivalent to the separable problem  $(P_S)$ .

The advantages of solving problem  $(P_{CL})$  are twofold. First, unlike the separable problem it combines information across tasks through the constraint, thereby exploiting data from related tasks. Second, unlike the consensus approach, it allows for task-specific solutions, leading to better performance on individual tasks. As we will show, cross-learning controls the bias-variance trade-off by enforcing the task-specific solutions to be close. We will formally prove that for the simplified model in Example 1, there exists a value of  $\epsilon$  for which our cross-learning estimator achieves a guaranteed improvement in mean squared error compared to both the separable and consensus estimators. And we will demonstrate that these improvements generalize to more complex scenarios in experiments with real data.

### III. PERFORMANCE ANALYSIS

Throughout this section, we will consider the simplified model given by (1), with the associated regressor  $f(x, \theta) = \theta$  and the quadratic loss  $\ell(y, f(x, \theta)) = \|y - \theta\|^2$ . Under these assumptions,  $\hat{\theta}_t = (1/N_t) \sum_{n=1}^{N_t} y_{nt}$  is a sufficient statistic and the cross-learning estimator in  $(P_{CL})$  simplifies to

$$\begin{aligned} \{\theta_t^\dagger\}, \theta_g^\dagger &= \arg \min_{\theta_t, \theta_g} \sum_{t=1}^T \|\hat{\theta}_t - \theta_t\|^2 \quad (\bar{P}_{CL}) \\ \text{subject to } &\|\theta_t - \theta_g\| \leq \epsilon, \quad t = 1, \dots, T. \end{aligned}$$

We will compare this estimator  $(\bar{P}_{CL})$  to its fully separable  $(\bar{P}_S)$  and strict consensus  $(\bar{P}_C)$  alternatives using the mean squared error as a performance metric, for which we define

$$\mathcal{E}_S = \frac{1}{T} \sum_{t=1}^T \|\hat{\theta}_t - \theta_t^*\|^2, \quad (4)$$

$$\mathcal{E}_C = \frac{1}{T} \sum_{t=1}^T \|\hat{\theta}_c - \theta_t^*\|^2, \quad (5)$$

$$\mathcal{E}_{CL}(\epsilon) = \frac{1}{T} \sum_{t=1}^T \|\theta_t^\dagger(\epsilon) - \theta_t^*\|^2, \quad (6)$$

for the errors resulting from the separable  $(\bar{P}_S)$ , consensus  $(\bar{P}_C)$ , and cross-learning  $(P_{CL})$  estimators, respectively. The error is measured with respect to the true data-generating parameters  $\theta_t^*$ . In what follows, we show that by appropriately tuning the centrality parameter  $\epsilon$ , the error of our cross-learning estimator  $(P_{CL})$  can always be made smaller than the error produced by both the separate  $(P_S)$  and consensus estimators  $(P_C)$ .

To begin with, we compare the cross-learning estimator to the consensus one. We note that  $\mathcal{E}_{CL}(\epsilon) = \mathcal{E}_C$  when  $\epsilon = 0$  since in this case the constraint in  $(P_{CL})$  imposes consensus. If we define the difference in error as  $\Delta \mathcal{E}(\epsilon) := \mathcal{E}_{CL}(\epsilon) -$



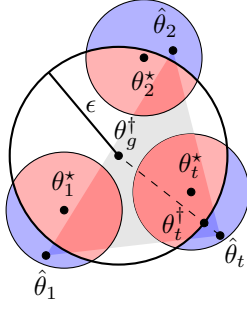


Fig. 2: Geometric argument for the proof of Proposition 2. The generators  $\theta_t^*$  are inside the ball  $\mathcal{B}(\theta_g^\dagger, \epsilon)$ . The separate estimates  $\hat{\theta}_t$  are projected into  $\mathcal{B}(\theta_g^\dagger, \epsilon)$  resulting in the cross-learning estimator  $\theta_t^\dagger$ . Points  $\hat{\theta}_t$  in the red region will not be changed, so that  $\hat{\theta}_t = \theta_t^\dagger$ , and points in the blue region will be projected to the surface where the error to  $\theta_t^*$  will be lower.

If  $\epsilon$  is chosen large enough to satisfy

$$\epsilon \geq \max_t \max_\tau \|\theta_t^* - \hat{\theta}_\tau\|, \quad (15)$$

then the deterministic error of the cross-learning estimator is smaller than the error of the separable estimator, i.e.,

$$\mathcal{E}_{CL}(\epsilon) - \mathcal{E}_S \leq 0. \quad (16)$$

*Proof.* We will start by showing that (15) implies that the generators  $\theta_t^*$  belong to the ball of center  $\theta_g^\dagger$  and radius  $\epsilon$ , which we denote by  $\mathcal{B}(\theta_g^\dagger, \epsilon)$ , i.e.,

$$\|\theta_t^* - \theta_g^\dagger\| \leq \epsilon, \text{ for all } t. \quad (17)$$

For that purpose, we consider Lemma 1 in Appendix B, which establishes that  $\theta_g^\dagger$  can be written as a convex combination  $\theta_g^\dagger = \sum_{t=1}^T \gamma_t \hat{\theta}_t$  of the separable estimators, with coefficients  $\gamma_t \geq 0$  such that  $\sum_{t=1}^T \gamma_t = 1$ . It follows  $\theta_t^* - \theta_g^\dagger = \sum_{\tau=1}^T \gamma_\tau (\theta_t^* - \hat{\theta}_\tau)$ , and thus

$$\|\theta_t^* - \theta_g^\dagger\| \leq \sum_{\tau=1}^T \gamma_\tau \|\theta_t^* - \hat{\theta}_\tau\| \leq \max_{\tau=1, \dots, T} \|\theta_t^* - \hat{\theta}_\tau\| \leq \epsilon, \quad (18)$$

so that (17) holds for all  $t = 1, \dots, T$ .

Next, notice that by virtue of Lemma 2 in Appendix B, the cross-learning estimator  $\theta_t^\dagger$  is the projection  $\theta_t^\dagger = P_{\mathcal{B}}(\hat{\theta}_t)$  of  $\hat{\theta}_t$  to the ball  $\mathcal{B} := \mathcal{B}(\theta_g^\dagger, \epsilon)$  of radius  $\epsilon$  and center  $\theta_g^\dagger$  (see Figure 2). We have shown in (18) that all  $\theta_t^*$  belong to  $\mathcal{B}$ . Thus, we have  $P_{\mathcal{B}}(\theta_t^*) = \theta_t^*$ , and since the projection is non-expansive we can bound

$$\|\theta_t^* - \theta_t^\dagger\| = \|P_{\mathcal{B}}(\theta_t^*) - P_{\mathcal{B}}(\hat{\theta}_t)\| \leq \|\theta_t^* - \hat{\theta}_t\|, \quad (19)$$

for all  $t = 1, \dots, T$ . The desired result (16) follows from averaging both sides of (19) across  $t$ . ■

**Example 2** To showcase Propositions 1 and 2, we present a controlled example in which data come from a multivariate Gaussian distribution in  $\mathbb{R}^2$ . For each task, we consider the centers at  $[\mu, 0]$ ,  $[-\mu, 0]$ ,  $[0, \mu]$ , and  $[0, -\mu]$ . In this case, the mean square error is  $\mu^2 + \sigma^2/2$  for the consensus estimator and  $2\sigma^2$  for the separable case. Figure 3 shows the error  $\mathcal{E}_{CL}(\epsilon)$

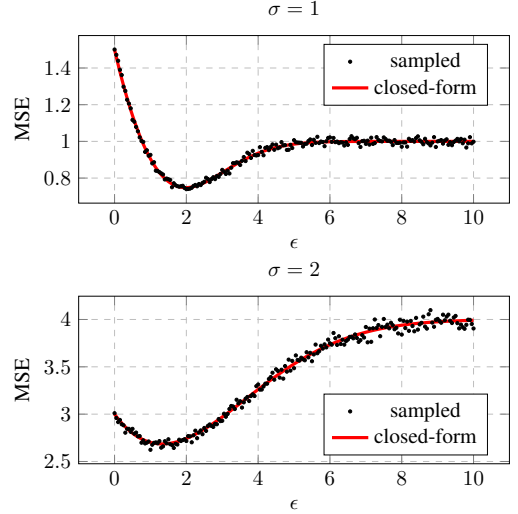


Fig. 3: Mean square error of the cross-learning estimate as a function of  $\epsilon$ . The value of  $\epsilon$  interpolates between consensus ( $\epsilon = 0$ ), and separate estimation ( $\epsilon = \infty$ ). We present the case with small variance ( $\sigma = 1$ ) wherein the separable estimator outperforms the consensus one, and the case with large variance ( $\sigma = 2$ ) in which consensus is better than separate estimation. In either case, the plots exhibit a value of  $\epsilon \in (0, \infty)$  for which cross-learning outperforms both the consensus and separable estimators.

found in simulations for  $\epsilon \in (0, 10)$ . There is a value of  $\epsilon$  in Figure 3 such that the cross-learning estimator outperforms both the consensus and separable estimators. This corresponds with Proposition 1, which proves that there exists a value  $\epsilon$  such that the cross-learning estimator outperforms its consensus counterpart. It also lines up with Proposition 2, which shows that if  $\epsilon \geq \epsilon_0 = \max_t \max_\tau \|\theta_t^* - \hat{\theta}_\tau\| > 0$ , the mean square error of the cross-learning estimator is bounded above by the mean square error of the separate counterpart.

The comparison in Figure 3, however, looks better than our theoretical result in Proposition 2, which only established a lower-than-or-equal type of bound. Thus, Proposition 2 did not guarantee the performance of the cross-learning estimator to be strictly better than the separate counterpart. To show that indeed the cross-learning estimator outperforms the separate one, we need yet another proposition. In this direction, consider the set

$$\mathcal{C}(\epsilon) = \{(\hat{\theta}_1, \dots, \hat{\theta}_T) : \exists t \in \{1, \dots, T\} \text{ s.t. } \|\hat{\theta}_t - \theta_g^\dagger\| \geq 3\epsilon\}. \quad (20)$$

with  $\theta_g^\dagger$  being the centroid in  $(\bar{\mathcal{P}}_{CL})$ . Proposition 3, below, establishes that the cross-learning presents a strictly lower error than the separable estimator when  $(\hat{\theta}_1, \dots, \hat{\theta}_T)$  is chosen from the set  $\mathcal{C}(\epsilon)$ . Furthermore, the probability of  $\mathcal{C}(\epsilon)$  is strictly positive, which together with the uniform bound in Proposition 2 is sufficient to prove that the mean squared error averaged across all datasets inside and outside  $\mathcal{C}(\epsilon)$  is strictly lower when using cross-learning.

**Proposition 3.** *There exists a value  $\epsilon_0 > 0$  such that for all  $\epsilon > \epsilon_0$ , if  $\epsilon$  and  $(\hat{\theta}_1, \dots, \hat{\theta}_T) \in \mathcal{C}(\epsilon)$  are substituted in  $(\bar{\mathcal{P}}_{CL})$ ,*

the error  $\mathcal{E}_{CL}(\epsilon)$  of the resulting estimator  $\theta_t^\dagger$  is bounded by

$$\mathcal{E}_{CL}(\epsilon) \leq \mathcal{E}_S - \frac{1}{T}\epsilon^2. \quad (21)$$

*Proof.* For each dataset  $(\hat{\theta}_1, \dots, \hat{\theta}_T)$  and  $\epsilon > 0$  compute  $\theta_g^\dagger$  via cross-learning, and define

$$\epsilon_0 = \inf\{\epsilon > 0 : \|\theta_t^\star - \theta_g^\dagger\| \leq \epsilon \text{ for all } t = 1, \dots, T\}. \quad (22)$$

We have shown in (17) that for any  $\epsilon \geq \max_t \max_\tau \|\theta_t^\star - \hat{\theta}_\tau\|$  as in (15), the condition  $\|\theta_t^\star - \theta_g^\dagger\| \leq \epsilon$  in (22) is satisfied. Thus, the set in (22) is nonempty and the infimum in (22) is well defined. Furthermore,  $\epsilon_0$  is strictly positive, since it depends on the pairwise distances between the ground truth parameters  $\theta_t^\star$  which are assumed to be different one each other. Next, we consider the set  $\mathcal{C}(\epsilon)$ , defined in (20) and depicted in Figure 2, which collects (sufficient statistics of) datasets  $(\hat{\theta}_1, \dots, \hat{\theta}_T)$  such that at least one of its points  $\hat{\theta}_t$  belongs to a (blue) zone outside of the ball  $\mathcal{B}(\theta_g^\dagger, \epsilon)$ . Projecting  $\hat{\theta}_t$  into  $\mathcal{B}(\theta_g^\dagger, \epsilon)$  results in a  $\theta_t^\dagger$  that is closer to  $\theta_t^\star$ . Following this intuition, we will show that (21) is satisfied for all  $(\hat{\theta}_1, \dots, \hat{\theta}_T) \in \mathcal{C}(\epsilon)$ . In this direction, we use the triangle inequality to write

$$3\epsilon \leq \|\hat{\theta}_t - \theta_g^\dagger\| = \|\hat{\theta}_t - \theta_t^\star + \theta_t^\star - \theta_g^\dagger\| \quad (23)$$

$$\leq \|\hat{\theta}_t - \theta_t^\star\| + \|\theta_t^\star - \theta_g^\dagger\| \leq \|\hat{\theta}_t - \theta_t^\star\| + \epsilon, \quad (24)$$

where we used (22) and (20), valid for one  $\hat{\theta}_t$ . It follows that

$$\|\hat{\theta}_t - \theta_t^\star\| \geq 2\epsilon. \quad (25)$$

On the other hand, since  $\theta_t^\dagger$  is the projection of  $\hat{\theta}_t$  onto the ball  $\mathcal{B}(\theta_g^\dagger, \epsilon)$ , then  $\hat{\theta}_t - \theta_t^\dagger$  must be co-linear with  $\hat{\theta}_t - \theta_g^\dagger$  such that  $\hat{\theta}_t - \theta_t^\dagger = \mu_t(\hat{\theta}_t - \theta_g^\dagger)$ . Hence, we can write  $\hat{\theta}_t - \theta_g^\dagger = \hat{\theta}_t - \theta_t^\dagger + \theta_t^\dagger - \theta_g^\dagger = \mu_t(\hat{\theta}_t - \theta_g^\dagger) + \theta_t^\dagger - \theta_g^\dagger$  or equivalently  $(1 - \mu_t)(\hat{\theta}_t - \theta_g^\dagger) = \theta_t^\dagger - \theta_g^\dagger$ . Hence, we can bound

$$\begin{aligned} \|\theta_t^\dagger - \theta_t^\star\| &= \|\theta_t^\dagger - \theta_g^\dagger + \theta_g^\dagger - \theta_t^\star\| \\ &= \|(1 - \mu_t)(\hat{\theta}_t - \theta_g^\dagger) + \theta_g^\dagger - \theta_t^\star\| \\ &\leq \|\hat{\theta}_t - \theta_t^\star\| + \mu_r(\|\theta_g^\dagger - \theta_t^\star\| - \|\hat{\theta}_t - \theta_t^\star\|) \\ &\leq \|\hat{\theta}_t - \theta_t^\star\| + \mu_r(\epsilon - 2\epsilon) \end{aligned}$$

where we used (22) and (25). Moreover, since according to (20)  $\hat{\theta}_t \notin \mathcal{B}(\theta_g^\dagger, \epsilon)$ , the projecting constraint must be active, yielding  $\epsilon = \|\theta_t^\dagger - \theta_g^\dagger\| = (1 - \mu_t)\|\hat{\theta}_t - \theta_g^\dagger\| \geq 3\epsilon(1 - \mu_t)$  according to (20), which implies  $\mu_t \geq 2/3$ . It follows,

$$\|\theta_t^\dagger - \theta_t^\star\| \leq \|\hat{\theta}_t - \theta_t^\star\| - \mu_r\epsilon \leq \|\hat{\theta}_t - \theta_t^\star\| - \frac{2}{3}\epsilon. \quad (26)$$

To conclude the proof, we square both sides of (26) and substitute (25) to obtain

$$\|\theta_t^\dagger - \theta_t^\star\|^2 \leq \left(\|\hat{\theta}_t - \theta_t^\star\| - \frac{2\epsilon}{3}\right)^2 \quad (27)$$

$$\leq \|\hat{\theta}_t - \theta_t^\star\|^2 - 2\frac{2\epsilon}{3}\|\hat{\theta}_t - \theta_t^\star\| + \left(\frac{2\epsilon}{3}\right)^2 \quad (28)$$

$$\leq \|\hat{\theta}_t - \theta_t^\star\|^2 - 2\frac{2\epsilon}{3}(2\epsilon) + \left(\frac{2\epsilon}{3}\right)^2 \quad (29)$$

$$\leq \|\hat{\theta}_t - \theta_t^\star\|^2 - \epsilon^2. \quad (30)$$

The inequality (30) is true for at least one  $t \in \{1, \dots, T\}$  such that  $\|\hat{\theta}_t - \theta_g^\dagger\| \geq 3\epsilon$  as defined in  $\mathcal{C}(\epsilon)$ . For all other  $t \in \{1, \dots, T\}$  we still have (19). Recall that (19) relies on  $\|\theta_t^\star - \theta_g^\dagger\| \leq \epsilon$  in (17), which holds for all  $\epsilon > \epsilon_0$ . Thus, we can average across  $t$  and the slack  $\epsilon^2$  in (30) will reduce by a factor  $T$  which results in (21). ■

If we can show that  $\mathcal{C}(\epsilon)$  has positive probability, then the cross-learning estimator will have a strictly lower mean-square error than the separable one. We will follow that path in establishing that the cross-learning estimator outperforms the separate and consensus ones in our main result.

**Theorem 1.** Consider a set of deterministic parameters  $\theta_t^\star$ ,  $t = 1, \dots, T$  and their associated datasets collecting data corrupted by zero-mean Gaussian noise  $\nu_{tn} \sim \mathcal{N}(0, \sigma)$  according to the additive model  $y_{tn} = \theta_t^\star + \nu_{tn}$ ,  $t = 1, \dots, T$ ,  $n = 1, \dots, N_t$ . Define the square errors  $\mathcal{E}_S$ ,  $\mathcal{E}_C$ , and  $\mathcal{E}_{CL}(\epsilon)$ , as in (4), (5), and (6) by quantifying the error between a single ground truth parameter  $\theta_t^\star$  and the solutions of the separable ( $\bar{\mathbf{P}}_S$ ), consensus ( $\bar{\mathbf{P}}_C$ ), and cross-learning ( $\bar{\mathbf{P}}_{CL}$ ) estimators, respectively. Under these definitions, we have

$$\mathbb{E} \left[ \inf_{\epsilon > 0} \mathcal{E}_{CL}(\epsilon) - \mathcal{E}_C \right] < 0, \quad (31)$$

$$\mathbb{E} \left[ \inf_{\epsilon > 0} \mathcal{E}_{CL}(\epsilon) - \mathcal{E}_S \right] < 0. \quad (32)$$

*Proof.* We start by comparing the errors yield by consensus and by the separate estimator, dividing the proof in two complementary cases  $\mathbb{E}[\mathcal{E}_C] \leq \mathbb{E}[\mathcal{E}_S]$  or  $\mathbb{E}[\mathcal{E}_C] > \mathbb{E}[\mathcal{E}_S]$ . Starting with the case  $\mathbb{E}[\mathcal{E}_C] < \mathbb{E}[\mathcal{E}_S]$ , we established in Proposition 1 that there exists an  $\epsilon > 0$  such that (31) holds. Since we assumed  $\mathbb{E}[\mathcal{E}_C] < \mathbb{E}[\mathcal{E}_S]$ , then (32) must also hold. In the case  $\mathbb{E}[\mathcal{E}_S] < \mathbb{E}[\mathcal{E}_C]$ , we will first argue that for any dataset, we can find a value of  $\epsilon$  such that  $\mathcal{E}_{CL}(\epsilon) - \mathcal{E}_S$  is lower than or equal to zero. But, this is a direct consequence of Proposition 2, since the right-hand side of (15) only depends on the dataset. Therefore, (32) holds, and since we are considering the case  $\mathbb{E}[\mathcal{E}_S] < \mathbb{E}[\mathcal{E}_C]$ , then (31) must hold. To ensure that the inequality in (32) is strict we argue that the set  $\mathcal{C}(\epsilon)$  in (20) is nonempty because we can separate the points  $\hat{\theta}_t$  away in Fig. 2 keeping  $\theta_g^\dagger$  unchanged, so that  $\|\hat{\theta}_t - \theta_g^\dagger\|$  surpasses  $3\epsilon$  in (20). Furthermore, if we allow  $\theta_g^\dagger$  to move infinitesimally, we can move the points  $\hat{\theta}_t$  in a set  $\mathcal{C}(\epsilon)$  with positive volume, so that the probability of  $\mathcal{C}(\epsilon)$  is positive under our Gaussian data model. This positive probability together with (21) yields (16), concluding the proof. ■

The strict inequalities in Theorem 1 mean that by selecting the value of  $\epsilon$  properly (possibly depending on the dataset samples  $\{y_{tn}\}$ ) the cross-learning estimator outperforms its separable and consensus counterparts. Through the supporting Propositions 1 and 2, which are described by Figures 1 and 2 respectively, we recognize that the cross-learning constraint can simultaneously reduce the bias affecting the consensus estimator, and mitigate the higher variance of a separate result, attaining an optimal balance in between. Although these theoretical findings were formally established within a controlled framework with Gaussian data, this intuition about

how the cross-learning method achieves this optimal balance in the variance-bias tradeoff generalizes to more complex scenarios. Specifically, this balance carries out to the experiments of Section VI-A which are performed on real data. Before presenting these experiments, we propose a variant of cross-learning that is preferable when the model outputs, as opposed to their parameters, are assumed to be close to one another.

#### IV. CROSS-LEARNING WITH COUPLED OUTPUTS

Our cross-learning formulation ( $P_{CL}$ ) so far has put emphasis on the model parameters. While there are many problems where obtaining the parameters of a model is the primary goal, in most cases, they are just a vehicle to obtain a regression or classification output. Particularly when neural networks are involved, the optimal values of the filter taps provide little to no intuition into the phenomenon being learned. Moreover, close distances between these parameters may not reflect similar outputs to the same inputs. In these cases, it may be reasonable to leverage similarities in the model outputs themselves, as opposed to their parameters, for which we introduce an alternative cross-learning formulation with functional constraints

$$\begin{aligned} \{\theta_t^\dagger\}, \theta_g^\dagger &= \arg \min_{\{\theta_t\}, \theta_g} \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(y_i, f(x_i, \theta_t)) \quad (P_{CLF}) \\ \text{subject to} \quad & \frac{1}{N_t} \sum_{i=1}^{N_t} |f(x_i, \theta_t) - f(x_i, \theta_g)| \leq \epsilon, \end{aligned}$$

with  $\ell : \mathbb{R}^Q \times \mathbb{R}^Q \rightarrow \mathbb{R}_+$ ,  $\ell(y_1, y_2) = 0$  iff  $y_1 = y_2$ , being a non-negative loss function that measures the quality of learning. As before, we fit the parameters  $\theta_t$  of the models  $f(x, \theta_t)$  jointly across tasks, but now the model outputs are coupled via the functional constraints.

While we will not provide a formal analysis of this case, we claim that this cross-learning formulation ( $P_{CLF}$ ) with functional constraints comes to also exploit the bias-variance trade-off inherent to the problem. We support this claim with the experiments of Section VI-B, and by relating ( $P_{CLF}$ ) to the parametric case ( $P_{CL}$ ). Specifically, we notice that ( $P_{CLF}$ ) is a relaxation of ( $P_{CL}$ ), as stated by the following proposition.

**Proposition 4.** *If the model  $f(x, \theta)$  is  $L$ -Lipschitz in the parametrization, i.e.,  $|f(x, \theta_t) - f(x, \theta_g)| \leq L|\theta_t - \theta_g|$ , then the feasible sets  $\mathcal{C}_{CL}(\epsilon)$  and  $\mathcal{C}_{CLF}(\epsilon)$  for the cross-learning estimators ( $P_{CL}$ ) and ( $P_{CLF}$ ) with coupled parameters and constraints, respectively, satisfy  $\mathcal{C}_{CL}(\epsilon) \subseteq \mathcal{C}_{CLF}(\epsilon)$ .*

*Proof.* Let  $(\theta_t, \theta_g) \in \mathcal{C}_{CL}(\epsilon)$  such that

$$\mathcal{C}_{CL}(\epsilon) = \{(\theta_t, \theta_g) : \|\theta_t - \theta_g\| \leq \epsilon\}. \quad (33)$$

hence,  $\|\theta_t - \theta_g\| \leq \epsilon$  holds. Furthermore, by application of the Lipschitz property of the model function  $f$ , we have that

$$\|f(x_i, \theta_t) - f(x_i, \theta_g)\| \leq \|\theta_t - \theta_g\| \leq L\epsilon \quad (34)$$

Averaging over the samples  $i = 1, \dots, N_t$  we obtain the bound  $\frac{1}{N_t} \sum_{i=1}^{N_t} \|f(x_i, \theta_t) - f(x_i, \theta_g)\| \leq \frac{1}{N_t} \sum_{i=1}^{N_t} L\epsilon = L\epsilon$ . This implies  $(\theta_t, \theta_g) \in \mathcal{C}_{CLF}(L\epsilon)$  since according to ( $P_{CLF}$ ),  $\mathcal{C}_{CLF}(\epsilon) = \{(\theta_t, \theta_g) : \frac{1}{N_t} \sum_{i=1}^{N_t} |f(x_i, \theta_t) - f(x_i, \theta_g)| \leq \epsilon\}$ . ■

The inclusion  $\mathcal{C}_{CL}(\epsilon) \subseteq \mathcal{C}_{CLF}(L\epsilon)$  means that ( $P_{CLF}$ ) imposes a *weaker* coupling between task-specific parameters and the shared representation compared to the parametric problem. Consequently, while both formulations promote cross-task information sharing, the output constraint does so in a more flexible manner. In other words, ( $P_{CLF}$ ) allows for greater variability among task-specific parameters as long as the corresponding function outputs remain sufficiently close. In this case, the solution of ( $P_{CLF}$ ) with  $\epsilon = 0$  does not imply strict model consensus, since the parameters could differ across tasks and the outputs could also differ for inputs not seen during training. As  $N_t$  grows and the training set becomes more representative, the models reach consensus, which introduces bias as described in previous sections. If  $N_t$  reduces, we still expect a residual bias under this weaker broad-sense consensus enforced by ( $P_{CLF}$ ) with  $\epsilon = 0$ . On the other hand, when  $\epsilon \rightarrow \infty$  the conclusion is more direct. That is, we recover the fully separable case and its higher variance as we did when coupling the parameters. As in the parametric case, the cross-learning estimator, in this new form with coupled outputs, sets itself in between these consensus and separable counterparts.

#### V. DUAL ALGORITHMS

Upon presenting the cross-learning estimators ( $P_{CL}$ ) and ( $P_{CLF}$ ) as formulations to better balance the variance-bias trade-offs, in this section we propose two methods to solve their corresponding optimization problems.

##### A. Dual Algorithm for Parametric Constraints

Our first algorithm is built using the Alternating Direction Method of Multipliers (ADMM) [27]. This formulation is suitable to solve the optimization problem ( $P_{CL}$ ) that defines the cross-learning estimator with parametric constraints. In the case in which an algorithm is available to solve the separable problem ( $P_S$ ), this ADMM formulation allows us to solve the cross-learning problem ( $P_{CL}$ ) using the separable solution as a building block. We want to obtain the optimal cross-learning parameters  $\theta_t^\dagger$  for each of the tasks  $t = 1, \dots, T$ . To reduce notation, let us define the  $\theta = (\theta_1, \dots, \theta_T, \theta_g)$  containing all optimization variables and write the loss in terms of  $\theta$  and the data  $(x_t, y_t)$  for task  $t$  as

$$\ell(\theta) = \sum_t \ell(y_t, f(x_t, \theta_t)) = \sum_t \|y_t - f(x_t, \theta_t)\|^2 \quad (35)$$

Under this notation, the cross-learning problem to solve is

$$\min_{\theta} \ell(\theta), \text{ subject to } \theta \in \mathcal{C} \quad (36)$$

with  $\mathcal{C} = \mathcal{C}_{CL}(\epsilon)$  being the feasible set defined in (33). By introducing the barrier function

$$C(\theta) = \begin{cases} 0 & \theta \in \mathcal{C} \\ \infty & \theta \notin \mathcal{C} \end{cases} \quad (37)$$

together with an auxiliary variable  $z$ , we can write (36) as

$$\min_{\theta, z} \ell(\theta) + C(z), \text{ subject to } \theta = z \quad (38)$$



**Algorithm 1** Cross-Learning with Parametric Constraints

---

```

1: Initialize:  $\theta^0, z^0 = 0$ , and  $u^0 = 0$ 
2: for  $k = 0, 1, 2, \dots, k_{max}$  do
3:   Compute  $\lambda_k = \lambda_0 \Gamma^k \bmod K$ 
4:   for each task  $t = 1, \dots, T$  do
5:      $\theta_t^{k+1} = \arg \min_{\theta} \ell_{\lambda_k}(y_t, f(x_t, \theta), z_t, u_t)$ 
6:   end for
7:    $\theta_g^{k+1} = z_g^k - u_g^k$ 
8:    $z^{k+1} = P_C(\theta^{k+1} + u^k)$ 
9:    $u^{k+1} = u^k + \theta^{k+1} - z^{k+1}$ 
10: end for

```

---

Written as (38), we can solve cross-learning via the ADMM

$$\theta^{k+1} = \text{Prox}_{\lambda_k \ell}(z^k - u^k) \quad (39)$$

$$z^{k+1} = P_C(\theta^{k+1} + u^k) \quad (40)$$

$$u^{k+1} = u^k + v^{k+1} - z^{k+1} \quad (41)$$

with the proximal operator of the loss in (39) defined by

$$\theta^{k+1} = \arg \min_{\theta} \sum_t \ell(y_t, f(x_t, \theta_t)) + \frac{1}{2\lambda_k} \|\theta - z^k + u^k\|^2 \quad (42)$$

where  $\lambda_k = \lambda_0 \Gamma^k$  is a weighting parameter that progressively accentuates the effect of the loss over the quadratic penalty.

To take full advantage of the ADMM, we expand the penalty  $\|\theta - z^k + u^k\|^2 = \sum_{t=1}^T \|\theta_t - z_t^k + u_t^k\|^2 + \|\theta_g - z_g^k + u_g^k\|^2$  with  $z_t^k$  and  $u_t^k$  denoting the elements of  $z^k$  and  $u^k$ , respectively, corresponding to task  $t$ , and  $z_g^k$  and  $u_g^k$  being the ones corresponding to the centroid. Such an expansion facilitates a distributed solution of (42) in which the parameters  $\theta_t$  for task  $t$  are obtained by solving

$$\theta_t^{k+1} = \arg \min_{\theta} \ell_{reg}(y_t, f(x_t, \theta_t), z_t, u_t) \quad (43)$$

with the regularized loss defined by  $\ell_{\lambda_k}(y_t, f(x_t, \theta_t), z_t, u_t) = \ell(y_t, f(x_t, \theta_t)) + \frac{1}{2\lambda_k} \|\theta_t - z_t^k + u_t^k\|^2$ .

The centroid, in turns, is only involved in the penalty in (42). Hence, it admits the closed-form  $\theta_g = z_g - u_g$ . This describes the entire procedure, which is summarized in Algorithm 1.

### B. Dual Algorithm for Coupled Outputs

Next, we construct a primal-dual algorithm to solve (P<sub>CLF</sub>). With  $\lambda_t > 0$  denoting the dual variable associated with domain  $t$ , and  $\lambda = [\lambda_1, \dots, \lambda_T]^T$ , the Lagrangian associated with (P<sub>CLF</sub>) takes the form

$$L(\theta_t, \theta_g, \lambda) = \frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(y_i, f(x_i, \theta_t)) + \lambda_t \left( \frac{1}{N_t} \sum_{i=1}^{N_t} |f(x_i, \theta_t) - f(x_i, \theta_g)| - \epsilon \right). \quad (44)$$

We define the dual function for problem (P<sub>CLF</sub>) as  $d(\lambda) := \min_{\theta_t, \theta_g} L(\theta_t, \theta_g, \lambda)$ , and the corresponding convex dual problem as the maximization of  $d(\lambda)$  in the positive orthant, that is,  $D_{CLF}^* := \max_{\{\lambda_t \geq 0\}} d(\lambda)$  [28].

Problem  $D_{CLF}^*$  can be solved [29], [30] by alternating gradient steps on the Lagrangian and the dual function. Upon

**Algorithm 2** Cross-Learning Functional Algorithm

---

```

1: Initialize models  $\{\theta_t^0\}$ ,  $\theta_g^0$ , and dual variables  $\lambda = 0$ 
2: for epochs  $e = 1, 2, \dots$  do
3:   for batch  $i$  in epoch  $e$  do
4:     Update params.  $\theta_t^{k+1} = \theta_t^k - \eta_P \hat{\nabla}_{\theta_t} L(\theta_t, \theta_g, \lambda) \forall t$ 
5:     Update  $g$  params.  $\theta_g^{k+1} = \theta_g^k - \eta_P \hat{\nabla}_{\theta_g} L(\theta_t, \theta_g, \lambda)$ 
6:   end for
7:   Update dual variable for all  $t \in [1, \dots, T]$ 
8:    $\lambda_t^{k+1} = \left[ \lambda_t^k + \eta_D \left( \frac{1}{N_t} \sum_{i=1}^{N_t} |f(x_i, \theta_t) - f(x_i, \theta_g)| - \epsilon \right) \right]_+$ 
9: end for

```

---

selecting step-sizes  $\eta_P > 0$  and  $\eta_D > 0$ , and initializing the parameters  $\{\theta_t^0\}$ ,  $\theta_g^0$ , and the dual variables  $\lambda_t^0$ , we update the parameters by taking a gradient descent step on  $L(\theta_t, \theta_g, \lambda)$ ,

$$\theta_t^{k+1} = \theta_t^k - \eta_P \nabla_{\theta_t} L(\theta_t, \theta_g, \lambda) \text{ for all } t \in [1, \dots, T], \quad (45)$$

$$\theta_g^{k+1} = \theta_g^k - \eta_P \nabla_{\theta_g} L(\theta_t, \theta_g, \lambda), \quad (46)$$

and then the multipliers with a gradient ascent step over  $d(\lambda)$

$$\lambda_t^{k+1} = \left[ \lambda_t^k + \eta_D \left( \frac{1}{N_t} \sum_{i=1}^{N_t} |f(x_i, \theta_t) - f(x_i, \theta_g)| - \epsilon \right) \right]_+$$

for  $t = 1, \dots, T$ , projecting the result into the nonnegative orthant via  $[\cdot]_+ = \max\{0, \cdot\}$ . The overall is detailed in Algorithm 2. We refer the reader to [31] for a proof of convergence in a stochastic setup.

## VI. NUMERICAL EXAMPLES

In this section, we test our cross-learning estimators on real-world data in two scenarios: a time-series prediction task using COVID-19 epidemiological data [32] and an image classification task using the Office-Home dataset [33].

### A. COVID-19 SIR Model Fitting

In this section, we formulate an SIR fitting problem for the COVID-19 pandemic data obtained from the OWID database [32]. Although complex models are often required to capture the full dynamics of an epidemic, we use a simple, interpretable, bi-parametric SIR model. This choice will allow for a clear interpretation of our cross-learning method. The SIR model is described by a system of differential equations

$$\frac{dS}{d\tau} = -\frac{\beta}{N} SI, \quad \frac{dI}{d\tau} = \frac{\beta}{N} SI - \gamma I, \quad \frac{dR}{d\tau} = \gamma I, \quad (47)$$

where variables  $S$ ,  $I$ , and  $R$  stand for the number of susceptible, infected, and removed individuals, respectively, which evolve with time  $\tau$ , and the constant  $N = S + I + R$  stands for the initial total population.

We consider the multiple tasks of predicting the infection time series  $(S_t, I_t, R_t)$  for  $T$  different countries. The loss  $\ell$  is defined as the mean squared error between the predicted and observed infection data for each country. By following Algorithm 1, which optimizes the regularized losses  $\ell_{\lambda_k}$ , we obtain specific parameters  $\theta_t^\dagger = (\beta_t, \gamma_t)$  for each country, together with a centroid  $\theta_g^\dagger$ . The central question we investigate



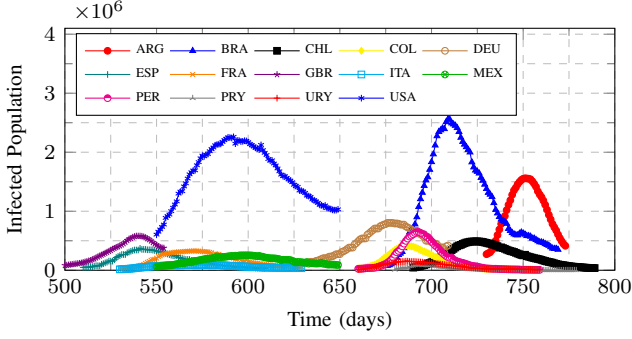


Fig. 4: Infected population over time for  $T = 14$  different countries in the OWID dataset. Countries exhibit different dynamics for the evolution of their infected populations.

is: Would it have been possible to predict the peak and timing of Argentina’s (ARG) COVID-19 wave using only early data from their own incipient records, supplemented with data from previous waves in other countries? To answer this, we use the epidemic waves from  $T - 1 = 13$  countries, shown in Figure 4. Our goal is to predict the latter part of the ARG wave using only its initial data points. We evaluate performance using two metrics: the lag error, which measures the difference in days between the predicted and actual peak of the infection wave, and the peak error, which measures the difference in the number of infected people at the peak. Figure 5 illustrates the prediction results when using only the first 10 days of ARG wave data and all previously available data from Fig. 4. The separate estimator, which uses only ARG data, fails to capture the peak. With a fitted recovery rate  $\gamma = 0$ , it incorrectly predicts unbounded exponential growth. On the other hand, enforcing consensus, yields a single common pair of  $(\beta, \gamma)$  parameters for all  $T = 14$  countries, averaging out the unique dynamics of the ARG wave, resulting in a prediction that severely underestimates the severity of the outbreak. In contrast, the cross-learning approach (with  $\epsilon = 0.1$ ) effectively leverages the data from all 15 countries. It produces a model that accurately predicts both the timing (lag) and the magnitude (peak) of the ARG infection wave.

Under cross-learning, the distance of the final parameters in parameter space depends on the decision of the value  $\epsilon$ . Particularly  $\epsilon = \infty$  forces no closeness between parameters, being equivalent to a separate approach, while  $\epsilon = 0$  forces all parameters to be equal, resulting in strict consensus. Figure 6 compares the separate SIR estimators to the cross-learning ones for  $\epsilon = 0.1$ . A separate estimator for ARG yields  $\gamma = 0$ , which is expected since the parameter  $\gamma$ , specifically, is tied to the downward slope, and there are not enough points in the ARG dataset to fit it properly. By contrast, the parameters  $\theta_t^\dagger = (\beta_t, \gamma_t)$  are closer to the centroid  $\theta_g^\dagger$ . Thus, the cross-learning estimator forces both ARG parameters to be nonzero. Close inspection of the SIR model (47) reveals that the peak of infections occurs when  $\beta_t S_t / N_t = \gamma_t$ , since that implies  $dI_t/d\tau = 0$ . Furthermore,  $\beta_t$  and  $\gamma_t$  must be similar to each other if the peak occurs at an early stage of the epidemic, in which  $S_t/N_t \simeq 1$ . As Figure 6 shows, this similarity is captured by the cross-learning estimator and is a key enabler

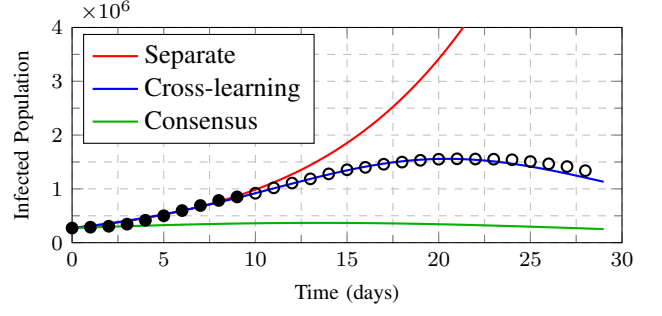


Fig. 5: SIR model predictions for ARG using the separate ( $\beta = 0.1481, \gamma = 0.0$ ), cross-learning ( $\beta = 0.6608, \gamma = 0.4388$ ), and consensus ( $\beta = 0.3497, \gamma = 0.2802$ ) estimators. Filled and hollow black dots represent the training and test datasets, respectively. Cross-learning achieves a more accurate prediction of the peak of infections with an error of 0.07% in the number of cases and finding the exact day when the peak occurs, as compared to errors of 2474% and 76.38% and time lags of 108 and 8 days for the separate and consensus estimators, respectively.

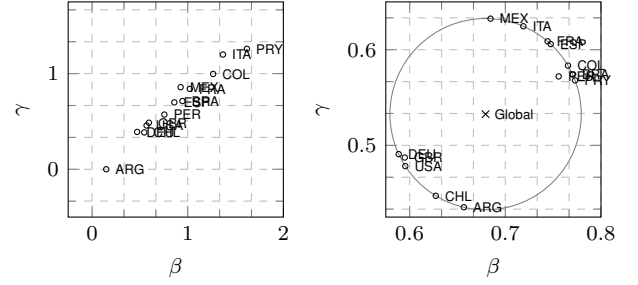


Fig. 6: Learned SIR parameters  $(\beta, \gamma)$  per country by separate estimators (left) and cross-learning (right).

for predicting the peak in Figure 5, where the separate estimator fails. Notice also that, while the centralized estimator also forces  $\beta_c \simeq \gamma_c$ , it reduces the influence of ARG data on the estimation. This causes distortion on the parameter  $\beta$ , which models the initial upright slope. As seen in Figure 5, such a distortion results in a worse prediction of the peak compared to the one obtained by our cross-learning approach. Remarkably, cross-learning does not require specific knowledge of these model insights but captures them by just setting a rather simple similarity constraint for the parameters across tasks. Furthermore, we performed an ablation study whose results are shown in Figure 7. Choosing  $\epsilon = 0.1$  is optimal, in the sense that it yields both the minimum peak and lag errors. Still, there is a wide range of  $\epsilon$  values yielding errors that are orders of magnitude lower than those corresponding to the separate or consensus estimators, indicating that the approach is robust and does not require extensive hyperparameter tuning to provide significant benefits.

Finally, we analyze the prediction error as a function of the number of available data points for ARG. In Figure 8, we show the error using separate, consensus, and cross-learning approaches. The independent approach requires a substantial amount of data (nearly 20 days) to correctly identify the

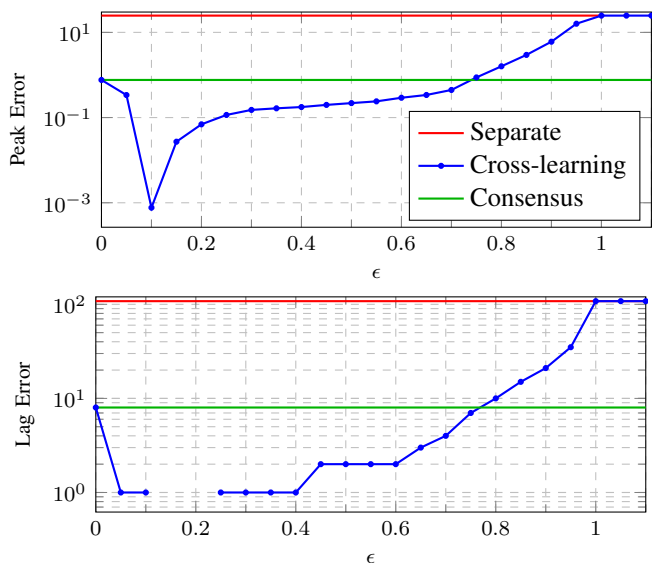


Fig. 7: Study on the effect of the centrality parameter  $\epsilon$  on the peak error (top) and lag error (bottom) in logarithmic scale. The cross-learning estimator coincides with the consensus one at  $\epsilon = 0$ , reduces to a minimum error (with no lag for  $\epsilon \in \{0.15, 0.20\}$ .) and then reaches its asymptotic value, the error corresponding to the separate estimator at  $\epsilon = 1.0$ .

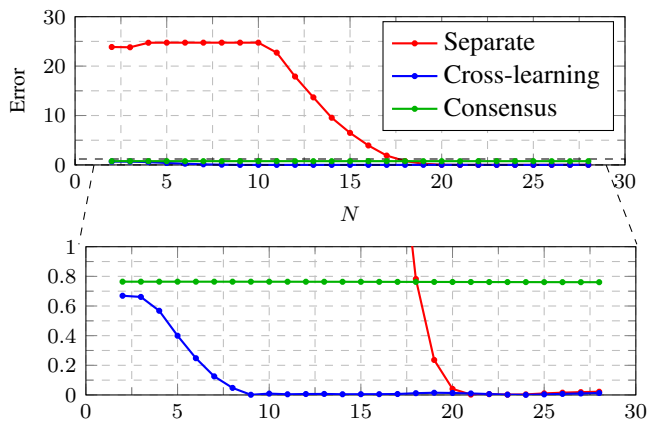


Fig. 8: Prediction error as a function of the number of available ARG data points ( $N$ ). The bottom panel provides a zoomed-in view of the low-error region.

trend and make an accurate prediction. In contrast, our cross-learning method achieves a low error with as few as 10 days of data. The performance of the consensus approach remains poor regardless of the amount of ARG data, as this specific information is diluted within the global dataset.

### B. Office-Home Dataset Classification

In this subsection, we benchmark our cross-learning method with coupled outputs on an image classification problem with real data coming from the dataset that we introduced in Figure 9. We consider the problem of classifying images belonging to  $P = 65$  different categories (which in Figure 9 are Alarm, Bike, Glasses, Pen, and Speaker), and  $T = 4$  different

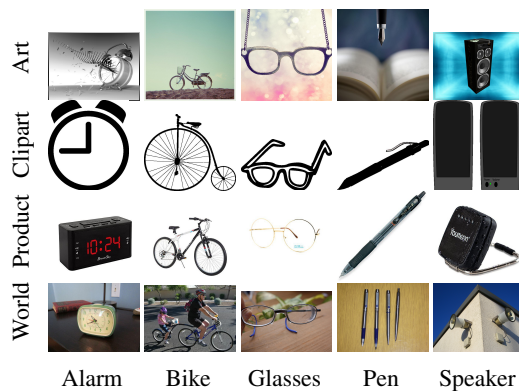


Fig. 9: Example images from 5 of the 65 categories from the 4 domains composing the Office-Home dataset [33]. The 4 domains are Art, Clipart, Product and Real World. In total, the dataset contains 15,500 images of different sizes.

domains. The Office-Home dataset [33] consists of 15,500 RGB images in total coming from the domains (i.e. tasks) (i) Art: an artistic representation of the object, (ii) Clipart: a clip art reproduction, (iii) Product: an image of a product for sale, and (iv) Real World: pictures of the object captured with a camera. Intuitively, by looking at the images, we can conclude that the domains are related. The minimum number of images per domain and category is 15 and the image size varies from the smallest image size of  $18 \times 18$  to the largest being  $6500 \times 4900$  pixels. We pre-processed the images by normalizing them and fitting their size to  $224 \times 224$  pixels.

We classify these images with neural networks  $f(x, \theta_t)$ , with their architecture based on AlexNet [34], reducing the size of the last fully connected layer to 256 neurons. We do not pre-train these networks, but optimize their weights from data via cross-learning using the cross-entropy loss [1], after splitting the dataset in 4/5 of the images for training and 1/5 for testing. Since neural networks are involved, we opt for the version of cross-learning with coupled outputs. Specifically, we use Algorithm 2 to train  $T+1 = 5$  neural networks for the  $T = 4$  domains plus the centroid, with step-sizes  $\eta_P = 0.003$  and  $\eta_D = 10$ , respectively, for different values of  $\epsilon$ . We also used the same split dataset to test the consensus classifier, which is equivalent to merging the images from all domains and training a single neural network on the whole dataset, and the  $T = 4$  classifiers which train their neural networks separately from images of their own specific domains. For comparison purposes, we also run Algorithm 1 on this dataset, which implements cross-learning with coupled parameters.

We show our results in Figure 10, where we use the classification accuracy of the trained networks on the test set as figure of merit. To begin with, we empirically corroborate that the dataset share mutual information across domains, since the consensus classifier achieves an accuracy of 35.59%, outperforming the separate training whose accuracy is 31.94%. This means that utilizing samples from other domains improves the performance of the learned classifier. This is the setting in which cross-learning is most helpful as it will help reduce bias. As it can be seen in Figure 10, a salient fact about

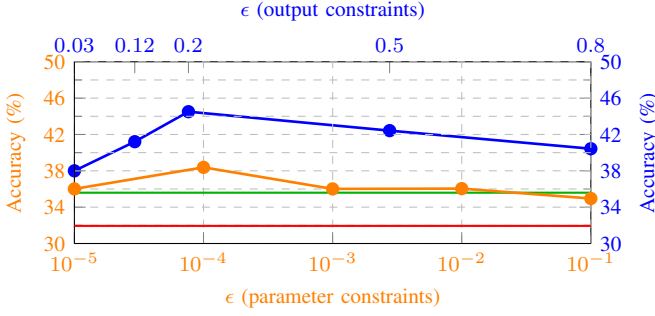


Fig. 10: Overall accuracy in percentage of correctly classified images measured on unseen data (test set). The consensus estimator (green) coincides with the parametric one (orange) when  $\epsilon = 0$  and the tends to the separate one (red) as  $\epsilon \rightarrow \infty$ . The output-constrained cross-learning estimator (blue) obtains the highest accuracy of 44.5%. at  $\epsilon = 0.2$ .

the experiments is that the cross-learning estimator in both the parameter, as well as the output constraint, consistently outperforms both consensus and separate estimators, reaching a maximum accuracy of 44.5% at  $\epsilon = 0.2$  when coupling the outputs. Indeed, in this scenario, the case with output constraints shows an improvement over parameter constraints. However, in both cases the estimator has a positive slope coming from  $\epsilon = 0$ , and a negative one when  $\epsilon$  is big enough, attaining a maximum inbetween, which reproduces the theoretical findings of Section III.

As a byproduct, Algorithm 2 generates dual variables  $\lambda_t$  that contain valuable information of the problem at hand. Figure 11 shows the dual variables  $\lambda$  as a function of the epoch. A larger dual variable indicates that the constraint is harder to satisfy, and a dual-variable equal to 0 indicates that the constraint is inactive. As seen in 11, all  $\lambda_t$  are non-negative, which means that we are effectively in a regime where there the classifiers are utilizing data from other tasks. If we look at the relative values of the dual variables, we see that the domain Art has the smallest value, whereas the domain Real World has the largest one. Given that Art has the least amount of samples, it is the domain that mostly benefits from images coming from other tasks. On the other hand, the largest dual variable is associated with the real-world dataset, have images with more details, textures, and shapes, and are therefore more difficult to classify (cf. Figure 9).

## VII. CONCLUSIONS

We proposed a constrained optimization method to achieve multi-task learning under a deterministic paradigm for the model parameters. Leveraging a controlled Gaussian model, we were able to prove that by forcing the parameter estimators to be close to one another, we obtain a strictly lower mean-squared error than if we use separate or consensus counterparts. Then we tested our cross-learning method on a practical problem with real data, where we fit parameters modeling the propagation of an infectious disease across different populations using a tailored algorithm based on proximal operators. The results obtained in this scenario replicated the theoretical

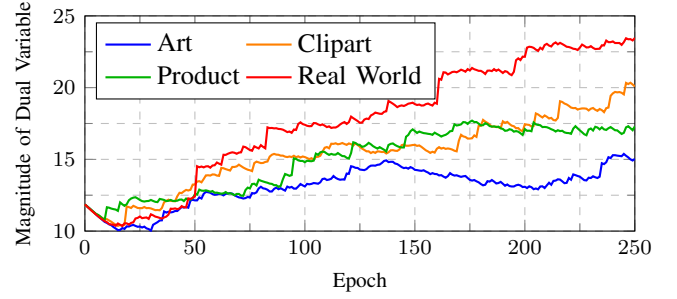


Fig. 11: Dual variable associated with each constraint for the case of  $\epsilon = 0.12$ . Intuitively, a larger dual variable indicates that the constraint is harder to satisfy.

ones, with cross-learning predicting the peak of infections with an error of 0.07%, compared to errors of 76.38% when assuming that all populations follow the same propagation model, and 2474% when estimating separate parameters. In the context of classification with neural networks, we proposed a variant of our cross-learning method that constrains the model outputs, rather than their parameters, to be at a close distance. Such a variant was also tested on real data, comprising images from different categories and domains. In this new test, the cross-learning estimator achieved a higher accuracy of 44.5% when compared to 33.97% and 24.44% for the consensus and separate counterparts, respectively.

## APPENDIX

### A. Bias-Variance

Let us compute the bias and variance of the consensus estimator  $\hat{\theta}_c = (1/T) \sum_{t=1}^T \hat{\theta}_t$  as an estimator for the ground truth parameter  $\theta_t^*$ , under the model  $\hat{\theta}_t = \theta_t^* + \eta_t$  with zero-mean noise  $\eta_t$  of variance  $\sigma^2$ . The bias depends on the pairwise distances between the ground truth estimators, i.e.,

$$\mathbb{E}[\hat{\theta}_c - \theta_t^*] = \mathbb{E}\left[\frac{1}{T} \sum_{\tau=1}^T (\hat{\theta}_\tau - \theta_t^*)\right] = \frac{1}{T} \sum_{\tau=1}^T (\hat{\theta}_\tau - \theta_t^*)$$

so it introduces bias unless all  $\theta_t^*$  coincide. On the other hand

$$\begin{aligned} \text{var}(\hat{\theta}_c) &= \mathbb{E}\left[\|\hat{\theta}_c - \mathbb{E}[\hat{\theta}_c]\|^2\right] = \mathbb{E}\left[\left\|\frac{1}{T} \sum_{t=1}^T (\hat{\theta}_t - \theta_t^*)\right\|^2\right] \\ &= \frac{1}{T} \sum_{t=1}^T \mathbb{E}\left[\|(\hat{\theta}_t - \theta_t^*)\|^2\right] = \frac{1}{T^2} \sum_{t=1}^T \frac{d}{N_t} \sigma^2. \end{aligned}$$

which is smaller, the larger the number of tasks  $T$  is.

### B. Auxiliary Proofs of Proposition 1

We first establish that the cross-learning centroid is a convex combination of the separate estimates.

**Lemma 1.** Consider the centroid  $\theta_g^\dagger$  solution to  $(\bar{\mathbf{P}}_{\text{CL}})$ , and the separate estimates  $\hat{\theta}_t = (1/N_t) \sum_{n=1}^{N_t} y_{nt}$ . There exist a set of coefficients  $\gamma_t \in [0, 1]$  satisfying  $\sum_{t=1}^T \gamma_t = 1$ , such that

$$\theta_g^\dagger = \sum_{t=1}^T \gamma_t \hat{\theta}_t. \quad (48)$$

*Proof.* The Lagrangian of problem  $(\bar{\text{P}}_{\text{CL}})$  takes the form

$$L(\theta_t, \theta_g, \mu_t) = \sum_{t=1}^T \left\| \hat{\theta}_t - \theta_t \right\|^2 + \sum_{t=1}^T \mu_t (\|\theta_t - \theta_g\|^2 - \epsilon^2)$$

Setting its gradient with respect to  $\theta_t$  to zero, we obtain

$$\theta_t^\dagger = \frac{\hat{\theta}_t}{1 + \mu_t} + \frac{\theta_g^\dagger \mu_t}{1 + \mu_t} = \epsilon_t \hat{\theta}_t + (1 - \epsilon_t) \theta_g^\dagger \quad (49)$$

after defining  $\epsilon_t = 1/(1 + \mu_t)$  so that  $(1 - \epsilon_t) = \mu_t/(1 + \mu_t)$ . On the other hand, setting the gradient of the Lagrangian with respect to  $\theta_g$  to zero results in

$$\sum_{t=1}^T (\theta_g^\dagger - \theta_t^\dagger) \mu_t = 0. \quad (50)$$

Hence, we can substitute (49) in (50) to obtain

$$\sum_{t=1}^T \mu_t \theta_g^\dagger = \sum_{t=1}^T \mu_t \theta_t^\dagger = \sum_{t=1}^T \mu_t (\epsilon_t \hat{\theta}_t + (1 - \epsilon_t) \theta_g^\dagger) \quad (51)$$

$$= \sum_{t=1}^T \mu_t \epsilon_t \hat{\theta}_t + \sum_{t=1}^T \mu_t (1 - \epsilon_t) \theta_g^\dagger, \quad (52)$$

which we can solve for  $\theta_g^\dagger$  as in

$$\theta_g^\dagger = \sum_{t=1}^T \frac{\mu_t \epsilon_t \hat{\theta}_t}{\sum_{i=1}^T \mu_i \epsilon_i} = \sum_{t=1}^T \frac{(1 - \epsilon_t) \hat{\theta}_t}{\sum_{i=1}^T (1 - \epsilon_i)}. \quad (53)$$

where we used  $\mu_t \epsilon_t = \mu_t/(1 + \mu_t) = (1 - \epsilon_t)$ , so that (53) takes the form (48) if we identify  $\gamma_t \triangleq (1 - \epsilon_t)/\sum_{i=1}^T (1 - \epsilon_i)$ . ■

**Lemma 2.** Consider the centroid  $\theta_g^\dagger$  solution to  $(\bar{\text{P}}_{\text{CL}})$ , and the separate estimates  $\hat{\theta}_t = (1/N_t) \sum_{n=1}^{N_t} y_{tn}$ . The cross-learning estimate  $\theta_t^\dagger$  in  $(\bar{\text{P}}_{\text{CL}})$  is the projection of  $\hat{\theta}_t$  to the ball  $\mathcal{B}(\theta_g^\dagger, \epsilon)$  of center  $\theta_g^\dagger$  and radius  $\epsilon$ .

*Proof.* By construction  $\theta_t^\dagger \in \mathcal{B}(\theta_g^\dagger, \epsilon)$ . Furthermore, according to (49) in the proof of Lemma 1, the cross-learning estimate  $\theta_t^\dagger$  lies in a convex combination of  $\theta_g^\dagger$  and  $\hat{\theta}_t$ . For those constraints that are inactive, complementary slackness requires  $\mu_t = 0$ . Substituting  $\mu_t = 0$  in (49) we obtain  $\theta_t^\dagger = \hat{\theta}_t$ , which means that  $\hat{\theta}_t$  also lies inside the ball, and hence they are projections of each other. If the constraint is active, then  $\|\theta_t^\dagger - \theta_g^\dagger\| = \epsilon$ , which means that  $\theta_t^\dagger$  lies on the intersection of the border of  $\mathcal{B}(\theta_g^\dagger, \epsilon)$  with the segment between  $\hat{\theta}_t$  and  $\theta_g^\dagger$ , and that is the projection of  $\hat{\theta}_t$  to the ball again. ■

In the next Lemma we show that the consensus estimator  $\hat{\theta}_c$  is close to the centroid  $\theta_g^\dagger$ , which will be needed down the road to assess the performance of cross-learning.

**Lemma 3.** The error between the consensus estimator  $\hat{\theta}_c$  and the cross-learning centroid  $\theta_g^\dagger$  is given by

$$\theta_g^\dagger - \hat{\theta}_c = \frac{1}{T} \sum_{t=1}^T \epsilon_t (\hat{\theta}_c - \hat{\theta}_t) + \mathcal{O}(\epsilon^2) \quad (54)$$

with  $\epsilon_t = \epsilon/\|\hat{\theta}_t - \hat{\theta}_c\| + \mathcal{O}(\epsilon^2) = \mathcal{O}(\epsilon)$  in the limit as  $\epsilon \rightarrow 0$ .

*Proof.* We know from Lemma 1 that  $\theta_g^\dagger = \sum_{t=1}^T \gamma_t \hat{\theta}_t$  with

$$\gamma_t = \frac{(1 - \epsilon_t)}{\sum_{k=1}^T (1 - \epsilon_k)} = \frac{1}{T} \frac{(1 - \epsilon_t)}{(1 - \bar{\epsilon})} \quad (55)$$

where  $\bar{\epsilon} := (1/T) \sum_{k=1}^T \epsilon_k$ . A closer look at (55) reveals that

$$(1 - \bar{\epsilon})(1 + \bar{\epsilon}) = 1 - \bar{\epsilon}^2 \Rightarrow \frac{1}{(1 - \bar{\epsilon})} = 1 + \bar{\epsilon} + \mathcal{O}(\bar{\epsilon}^2). \quad (56)$$

Combining (56) with (55) we obtain

$$\gamma_t = \frac{1}{T} (1 - \epsilon_t) (1 + \bar{\epsilon} + \mathcal{O}(\bar{\epsilon}^2)) = \frac{1}{T} (1 + \bar{\epsilon} - \epsilon_t + \mathcal{O}(\epsilon^2)),$$

which can be substituted in  $\theta_g^\dagger = \sum_{t=1}^T \gamma_t \hat{\theta}_t$ , to obtain

$$\begin{aligned} \theta_g^\dagger - \hat{\theta}_c &= \sum_{t=1}^T \left( \gamma_t - \frac{1}{T} \right) \hat{\theta}_t = \frac{1}{T} \sum_{t=1}^T (\bar{\epsilon} - \epsilon_t + \mathcal{O}(\epsilon^2)) \hat{\theta}_t \\ &= \bar{\epsilon} \hat{\theta}_c - \frac{1}{T} \sum_{t=1}^T \epsilon_t \hat{\theta}_t + \mathcal{O}(\epsilon^2) = \frac{1}{T} \sum_{t=1}^T \epsilon_t (\hat{\theta}_c - \hat{\theta}_t) + \mathcal{O}(\epsilon^2). \end{aligned}$$

To finish the proof, we need to show that  $\epsilon_t$  is of order  $\epsilon$  as  $\epsilon \rightarrow 0$ . In this regime, we can assume without loss of generality that the constraints are active, and write

$$\epsilon = \|\theta_t^\dagger - \theta_g^\dagger\| = \|\epsilon_t \hat{\theta}_t + (1 - \epsilon_t) \theta_g^\dagger - \theta_g^\dagger\| = \epsilon_t \|\hat{\theta}_t - \theta_g^\dagger\|. \quad (57)$$

which implies that  $\epsilon_t = \mathcal{O}(\epsilon)$ . Still, we need one more step to prove  $\epsilon_t = \epsilon/\|\hat{\theta}_t - \hat{\theta}_c\| + \mathcal{O}(\epsilon^2)$ . So we rewrite (57) as

$$\begin{aligned} \epsilon_t &= \epsilon \|\hat{\theta}_t - \theta_g^\dagger\|^{-1} = \epsilon \left( \|\hat{\theta}_t - \hat{\theta}_c + \hat{\theta}_c - \theta_g^\dagger\|^2 \right)^{-1/2} \\ &= \epsilon \left( \|\hat{\theta}_t - \hat{\theta}_c\|^2 + \|\hat{\theta}_c - \theta_g^\dagger\|^2 + 2(\hat{\theta}_t - \hat{\theta}_c)^\top (\hat{\theta}_c - \theta_g^\dagger) \right)^{-1/2} \\ &= \epsilon \|\hat{\theta}_t - \hat{\theta}_c\|^{-1} (1 + 2R_t)^{-1/2} \quad (58) \end{aligned}$$

with  $R_t = \left( \|\hat{\theta}_c - \theta_g^\dagger\|^2 + (\hat{\theta}_c - \theta_g^\dagger)^\top (\hat{\theta}_t - \hat{\theta}_c) \right) \|\hat{\theta}_t - \hat{\theta}_c\|^{-2}$ .

Since we already proved that  $\hat{\theta}_c - \theta_g^\dagger = \mathcal{O}(\epsilon)$ , we can write

$$\epsilon_t = \frac{\epsilon(1 + 2\mathcal{O}(\epsilon))^{-1/2}}{\|\hat{\theta}_t - \hat{\theta}_c\|} = \frac{\epsilon(1 - \mathcal{O}(\epsilon))}{\|\hat{\theta}_t - \hat{\theta}_c\|} = \frac{\epsilon}{\|\hat{\theta}_t - \hat{\theta}_c\|} + \mathcal{O}(\epsilon^2),$$

which is the result that we wanted to prove. ■

The next lemma provides an expression for the difference between the cross-learning sample error  $\mathcal{E}_{\text{CL}}$  and its consensus counterpart  $\mathcal{E}_C$  in the regime  $\epsilon \rightarrow 0$ . Remarkably, this expression does not depend on the solution  $\theta_t^\dagger$  but on simpler variables that we can model in closed form.

**Lemma 4.** Consider the sample errors  $\mathcal{E}_C$  and  $\mathcal{E}_{\text{CL}}(\epsilon)$ , resulting from the consensus and cross-learning estimators, as defined in (5) and (6), respectively. As  $\epsilon \rightarrow 0$ ,  $\mathcal{E}_{\text{CL}}(\epsilon)$  converges to  $\mathcal{E}_C$  according to

$$\mathcal{E}_{\text{CL}}(\epsilon) - \mathcal{E}_C = -2\epsilon \frac{1}{T} \sum_{t=1}^T (\theta_t^* - \theta_c^*)^\top \frac{\hat{\theta}_t - \hat{\theta}_c}{\|\hat{\theta}_t - \hat{\theta}_c\|} + \mathcal{O}(\epsilon^2), \quad (59)$$

where  $\theta_c^* = \frac{1}{T} \sum_{t=1}^T \theta_t^*$  averages the ground truth parameters.

*Proof.* Let start by comparing the square errors of the consensus and the cross-learning estimators

$$\|\theta_t^\dagger - \theta_t^*\|^2 = \|\theta_t^\dagger - \hat{\theta}_c + \hat{\theta}_c - \theta_t^*\|^2 \quad (60)$$

$$= \|\hat{\theta}_c - \theta_t^*\|^2 + \|\theta_t^\dagger - \hat{\theta}_c\|^2 - 2(\theta_t^* - \hat{\theta}_c)^\top (\theta_t^\dagger - \hat{\theta}_c)$$

$$= \|\hat{\theta}_c - \theta_t^*\|^2 - 2I_t + \mathcal{O}(\epsilon^2) \quad (61)$$

where we defined  $I_t = (\theta_t^\dagger - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c)$ , and substituted  $\|\theta_t^\dagger - \hat{\theta}_c\|^2 = \mathcal{O}(\epsilon^2)$  since the cross-learning constraint imposes  $\|\theta_t^\dagger - \hat{\theta}_g\| \leq \epsilon$  and Lemma 3 implies  $\|\theta_g^\dagger - \hat{\theta}_c\| = \mathcal{O}(\epsilon)$ .

To express  $I_t$  in terms of  $\hat{\theta}_t$ , we add and subtract  $\theta_g^\dagger$

$$I_t = (\theta_t^\dagger - \theta_g^\dagger + \theta_g^\dagger - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c) \quad (62)$$

$$= (\epsilon_t(\hat{\theta}_t - \theta_g^\dagger) + \theta_g^\dagger - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c) \quad (63)$$

$$= \epsilon_t(\hat{\theta}_t - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c) + J_t + \mathcal{O}(\epsilon^2) \quad (64)$$

where we defined  $J_t = (\theta_g^\dagger - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c)$  and substituted  $\epsilon_t(\hat{\theta}_t - \theta_g^\dagger) = \epsilon_t(\hat{\theta}_t - \hat{\theta}_c) + \mathcal{O}(\epsilon^2)$  using Lemma 3 again.

Next, we average  $J_t$  across tasks to obtain

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T J_t &= \frac{1}{T} \sum_{t=1}^T (\theta_g^\dagger - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c) = (\theta_g^\dagger - \hat{\theta}_c)^\top (\theta_c^* - \hat{\theta}_c) \\ &= \frac{1}{T} \sum_{t=1}^T \epsilon_t(\hat{\theta}_c - \hat{\theta}_t)^\top (\theta_c^* - \hat{\theta}_c) + \mathcal{O}(\epsilon^2) \end{aligned} \quad (65)$$

after substituting (54) for  $(\theta_g^\dagger - \hat{\theta}_c)$ . Then we conclude the proof by averaging  $I_t$ , which results in

$$\frac{1}{T} \sum_{t=1}^T I_t = \frac{1}{T} \sum_{t=1}^T \epsilon_t(\hat{\theta}_t - \hat{\theta}_c)^\top (\theta_t^* - \hat{\theta}_c) \quad (66)$$

$$+ \frac{1}{T} \sum_{t=1}^T \epsilon_t(\hat{\theta}_c - \hat{\theta}_t)^\top (\theta_c^* - \hat{\theta}_c) + \mathcal{O}(\epsilon^2) \quad (67)$$

$$= \frac{1}{T} \sum_{t=1}^T (\theta_t^* - \theta_c^*)^\top \frac{(\hat{\theta}_t - \hat{\theta}_c)}{\|\hat{\theta}_t - \hat{\theta}_c\|} + \mathcal{O}(\epsilon^2) \quad (68)$$

with  $\epsilon_t = \epsilon / \|\hat{\theta}_t - \hat{\theta}_c\| + \mathcal{O}(\epsilon^2)$  as in Lemma 3. ■

## REFERENCES

- [1] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2009, vol. 2.
- [2] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1013–1020.
- [3] H. Lan, C. Zhang, Y. Li, Y. Hong, Y. He, and S. Wen, "Day-ahead spatiotemporal solar irradiation forecasting using frequency-based hybrid principal component analysis and neural network," *Applied Energy*, vol. 247, pp. 389–402, 2019.
- [4] M. Dorr, "Fitting the sir epidemiological model to influenza data," University of Maine, Tech. Rep., 2019.
- [5] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., 1993.
- [6] Y. Zhou, H. Chen, Y. Li, Q. Liu, X. Xu, S. Wang, P. Yap, and D. Shen, "Multi-task learning for segmentation and classification of tumors in 3d automated breast ultrasound images," *Medical Image Analysis*, vol. 70, p. 101918, 2021.
- [7] X. Cai, J. Yuan, R. Zheng, L. Huang, and K. Church, "Speech emotion recognition with multi-task learning," in *Interspeech*, vol. 2021, 2021, pp. 4508–4512.
- [8] M. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "Anomaly detection in video via self-supervised and multi-task learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12742–12752.
- [9] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [10] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. of the 34th Int. Conf. on Machine Learning*, vol. 70, 2017, pp. 1126–1135.
- [11] K. Chua, Q. Lei, and J. D. Lee, "How fine-tuning allows for effective meta-learning," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 8871–8884.
- [12] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.
- [13] P. Vafaeikia, K. Namdar, and F. Khalvati, "A brief review of deep multi-task learning and auxiliary task learning," *arXiv preprint arXiv:2007.01126*, 2020.
- [14] A. Maurer, M. Pontil, and B. Romera-Paredes, "The benefit of multitask representation learning," *Journal of Machine Learning Research*, vol. 17, no. 81, pp. 1–32, 2016.
- [15] Y. Yang and T. Hospedales, "Deep multi-task representation learning: A tensor factorisation approach," *arXiv preprint arXiv:1605.06391*, 2016.
- [16] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.
- [17] M. Wallingford, H. Li, A. Achille, A. Ravichandran, C. Fowlkes, R. Bhotika, and S. Soatto, "Task adaptive parameter sharing for multi-task learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7561–7570.
- [18] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *Int. Conference on Machine Learning*. PMLR, 2020, pp. 9120–9132.
- [19] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1930–1939.
- [20] Y. Zhang and D. Yeung, "A convex formulation for learning task relationships in multi-task learning," *arXiv preprint arXiv:1203.3536*, 2012.
- [21] E. V. Bonilla, K. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Advances in Neural Information Processing Systems*, vol. 20, 2007.
- [22] J. Hensman, N. D. Lawrence, and M. Rattray, "Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters," *BMC Bioinformatics*, vol. 14, no. 1, pp. 1–12, 2013.
- [23] J. Cervino, J. A. Bazerque, M. Calvo-Fullana, and A. Ribeiro, "Meta-learning through coupled optimization in reproducing kernel hilbert spaces," in *American Control Conference (ACC)*, 2019, pp. 4840–4846.
- [24] —, "Multi-task reinforcement learning in reproducing kernel hilbert spaces via cross-learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5947–5962, 2021.
- [25] —, "Multi-task supervised learning via cross-learning," in *29th European Signal Processing Conf. (EUSIPCO)*, 2021, pp. 1381–1385.
- [26] —, "Multi-task bias-variance trade-off through functional constraints," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [27] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [28] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
- [29] L. F. O. Chamon, S. Paternain, M. Calvo-Fullana, and A. Ribeiro, "The empirical duality gap of constrained statistical learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8374–8378.
- [30] L. F. O. Chamon and A. Ribeiro, "Probably approximately correct constrained learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [31] L. F. O. Chamon, S. Paternain, M. Calvo-Fullana, and A. Ribeiro, "Constrained learning with non-convex losses," *IEEE Transactions on Information Theory*, 2022.
- [32] E. Mathieu, H. Ritchie, L. Rodés-Guirao, C. Appel, D. Gavrilov, C. Giattino, J. Hasell, B. Macdonald, S. Dattani, D. Beltekian, E. Ortiz-Ospina, and M. Roser, "Covid-19 pandemic," *Our World in Data*, 2020, https://ourworldindata.org/coronavirus.
- [33] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.
- [34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.