Regularizing the Forward Pass

Patrick Power Shomik Ghosh

October 31, 2025

Abstract

In certain applied microeconomic settings, it's typical to view one's dataset as the realization of a stratified cluster randomized control trial: treatment is assigned at the cluster level and controls vary at both the individual and cluster level. *Locally*, this makes it more likely that observation will be from the same cluster which changes the finite sample challenge of generalization. We introduce an estimation method which takes accounts for this by partialing out non-parametric cluster effects via regularized bi-level gradient descent. We provide a python library based on JAX: https://github.com/pharringtonp19/rfp.

Keywords: Causal Inference, Deep Learning

1 Introduction

In many economic contexts, treatment varies at a level above the unit of interest. For instance, we may be interested in how some policy affects individuals, but across each zip code in the state, the policy is either in effect or not in effect.

These settings are attractive in practice because they often balance the desire for identification[1] with the aim of providing insight into the general equilibrium effects of the policy. They give up "within-cluster variation" of the treatment for variation in density of the treatment.

To compensate for the fact that there is no within-cluster variation, researchers will often condition on a cluster level feature. This can have two impacts. First, as figure 1 shows, this can change the local neighborhood structure of the observations: it can make it more likely that one's neighbors are from the same cluster. Second, this can increase the variance of one's estimator.

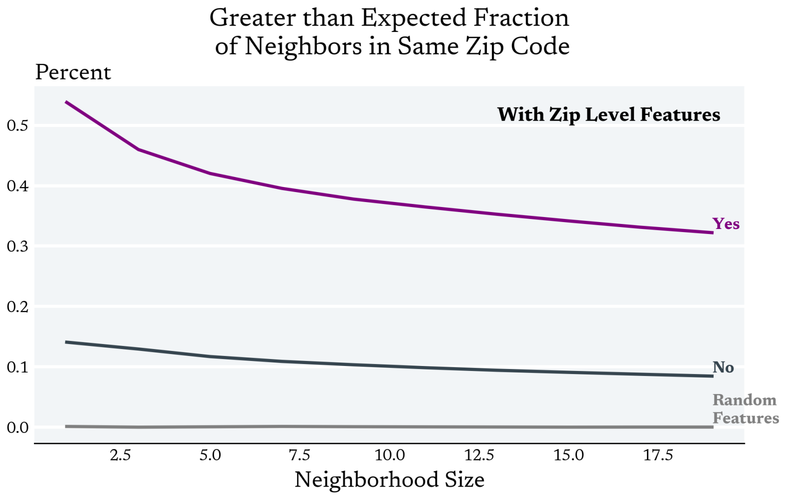


Figure 1: Within the control group, adding cluster level features increases the likelihood that the local neighborhood is overpopulated with observations from the same cluster.

$$Y = f(x) + g_c(x) + \varepsilon_i$$

What is the central challenge?

- How do you make local corrections in high dimensions?
- "In high dimensions, it is crucial to distribute probability mass where it matters rather than uniformly in all directions around each training point." Bengio et al. [2000]

How do we approach this central challenge?

[1] We define this more precisely later

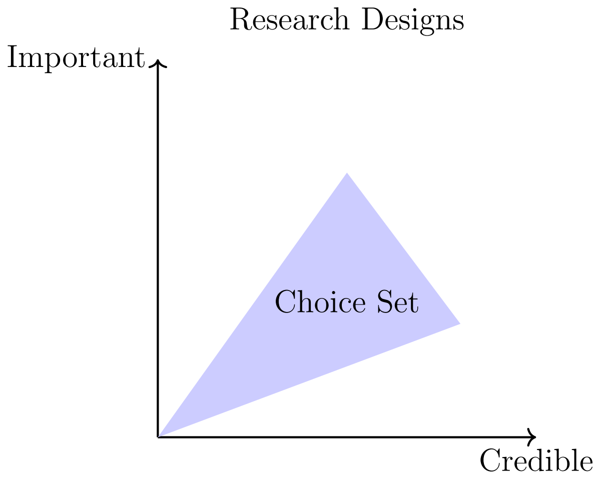


Figure 2

The typical approach to causal inference is to select control variables such that locally, treatment is as good as randomly assigned: $\tilde{Y}_i \perp D_i | X_i$. Meaning that conditional on the controls, treatment is not related to the potential outcomes in expectation. In many policy settings, including ours, the clusters which adapt the policy are not randomly selected which necessitates the inclusion of controls that vary at the cluster level. When studying the impacts of the Right to Counsel, we include aggregate eviction counts prior to the implementation as this was a key variable in the Connecticut Bar's Foundations consideration of where to first make legal aid lawyers available.

$$x_i = \left[\ \underbrace{\text{Month, Plaintiff, Gender, Rent,}}_{\text{Individual level}}\ \underbrace{\text{Zip Code Eviction Count}}_{\text{Cluster level}}\ \right]$$

The motivating observation on this paper is that cluter level controls together with cluster treatment assignment change the neighborhood structure of the observations. As figure 1 illustrates, they make it more likely that *local* observations within the treatment/control group will be from the same cluster. Allowing for potentially nonparametric cluster effects, this introduces additional variance into the estimation process. In this paper, we introduce an estimation framework that handles this issue in a conceptually similar way to typical deep learning estimation strategies with i.i.d. data.

2 The Essence of Causal Inference

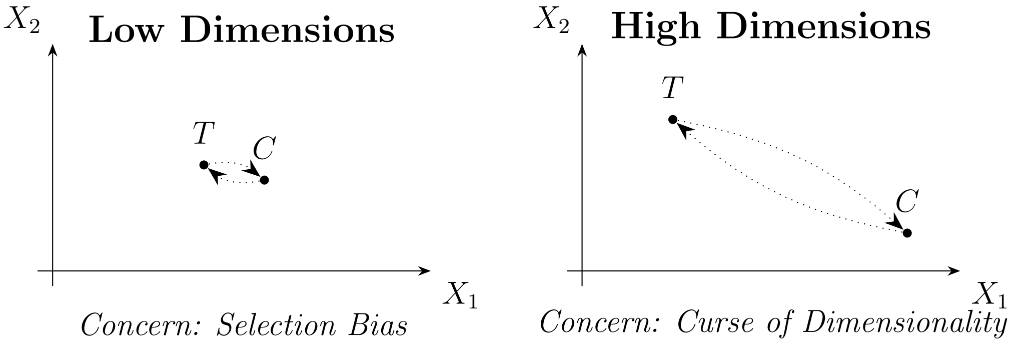
At the individual level, causal inference is a missing data problem. For each unit of interest, we observe at most one potential outcome.

$$Y_i = \tilde{Y}_i(1)D_i + \tilde{Y}_i(0)(1 - D_i)$$

At the population level, the problem becomes one of selection bias. Treatment groups often differ in ways that affect the outcome. The challenge is to find a suitable set of controls such that treatment can be thought to be indepdent of the potential outcome of interest .

$$\tilde{Y}_i(1) \perp D_i | X_i, \quad \Delta\tilde{Y}_i(1) \perp D_i | X_i$$

In the finite sample, the problem is one of generalization. With few control variables, the data points between the treatment and control group are more similar with respect to observed features, but yet the prediction problem is difficult because we are concerned that treatment and control observations might differ unobserved ways (Selection Bias). Adding more control variables makes the observations less likely to differ in latent ways, but now the prediction problem can be challenging because now observations differ in observed ways (Curse of Dimensionality).



add part about how this changes when we have cluster level controls

3 Motivation

3.1 Context

3.2 Identification

The typical definition in the literature of identification is surprisingly unhelpful when working with clustered data. Lewbel [2019] writes, "Econometric identification really means just one thing: model parameters or features being uniquely determined from the **observable population** that generates the data." With clustered data, though, the **observable population** is not well defined.[2] So we have to frame things in terms of the underlying data generating process.[3].

[2] We bold the text to make the expression jump out to the reader

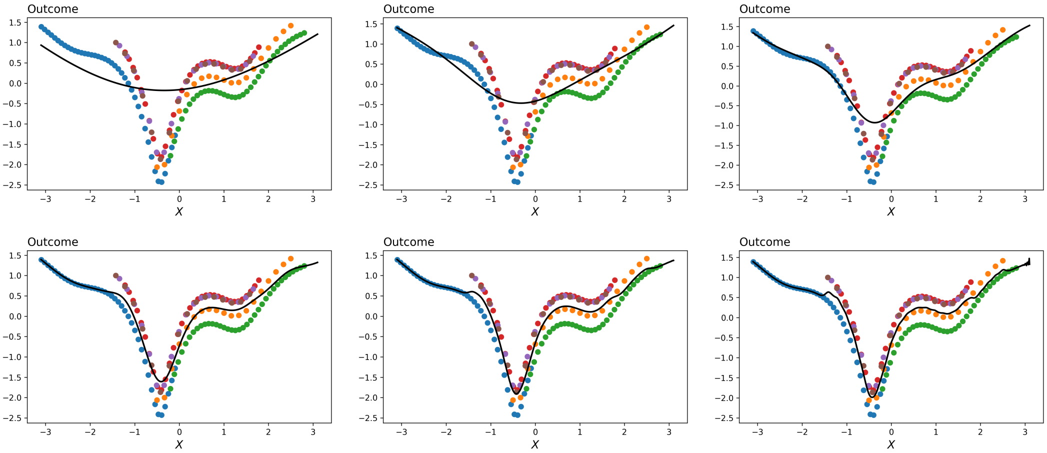
[3] We believe that Lewbel [2019] would agree with this

3.3 Challenge

Our approach to causal inference follows the potential outcome framework as articulated by *Mostly Harmless Econometrics*. Under a selection on observable assumption, that is treatment is as good as randomly assigned, the corresponding conditional expectation function has a causal interpretation - $\mathbb{E}[Y|X,D]$[4] .

Estimating the condition expectation function can be challenging when we (1) observe only a subset of the clusters in both the treated (control) group, (2) the distribution of covariates differ across clusters, and (3) the distribution of outcomes conditional on covariates differ across clusters. Figure 3 captures the behavior of a typical smoothing estimator in this context. A standard nonparametric model (black line) with a "smoothing" hyperparameter struggles: in order to fit the 'v'-shaped component of the data where there is general consensus across the clusters, the bandwidth of the estimator must be small. In doing so, though, it over fits the tails. Intuitively what's needed in this context is an estimator that is "locally" aware of the cluster structure of the data.

Figure 3: The Tragic Triad of Clustered Data



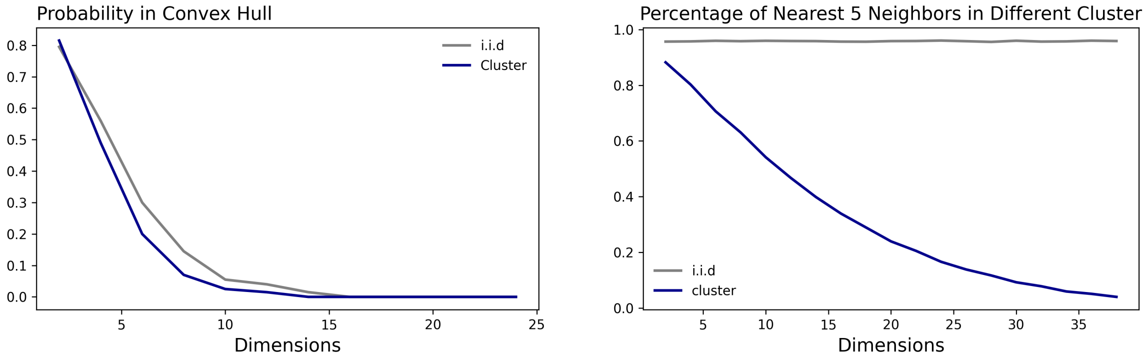
Reproduced Here: In this figure, we assume away within-cluster variation. Each dot corresponds to an observation. The different colors highlight the various clusters.

It's interesting to note that these issues are perhaps only magnified as we increase the dimensionality of the data. Extending the work of Balestriero et al. [2021], we illustrate in figure 5a that clustered sampling doesn't change the fundamental issue of learning in high dimensions (extrapolation) so much as it motivates us to reconsider how we go about it, as highlighted in figure 5b.

[4] We abuse terminology here by writing "the" conditional expectation function.

$$\omega \longmapsto \mathbb{E}[Y|X,D](\omega)$$

$$\int Y d\mathbb{P}_A = \int \mathbb{E}[Y|X,D] d\mathbb{P}_A, \quad \forall A \in \sigma(X \times D)$$



(a)

(b)

Figure 5: (a) Probability that a observation in the test set is in the convex hull formed by the training set (b) Fraction of the five nearest neighbors in a different cluster. Data consists of 25 cluster and 25 observations per cluster. Clusters differ only in the mean which is drawn from an isotropic gaussian distribution: Reproduced Here and Here

These three issues, which together we term the*Tragic Triad of Clustered Data*[5] can all occur when treatment is assigned at the cluster level and the conditional expectation function involves controls which vary at the cluster level. In this case, cluster fixed effects are collinear with treatment which means we cannot fit within clusters, but must partial out the potentially nonparametric cluster effects.

We highlight in figure 6 that subject to the typical caveats of hyperparameter tuning, our model fits the 'v'-shaped nature of the data where there is consensus across the clusters without overfitting in the tails." That is, our model, which we formally define in the next section, is implicitly locally aware of the cluster structure of the data.

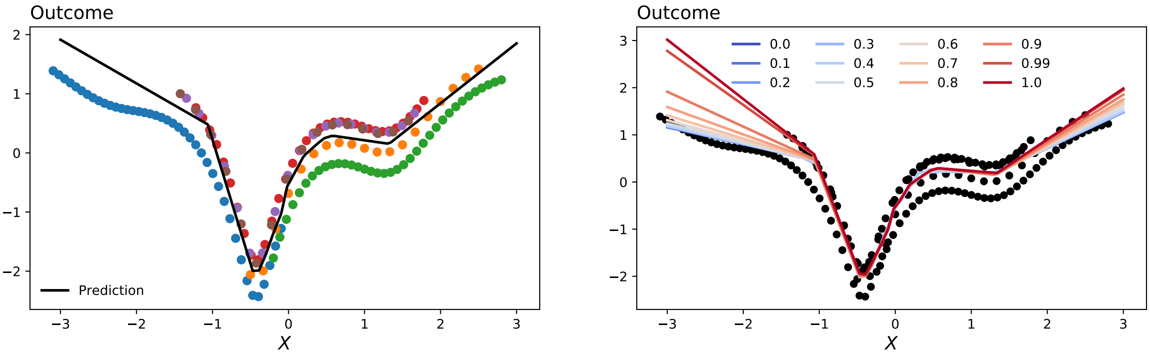


Figure 6

4 Method

As we alluded to previously, intuitively we would like the model to locally partial out the cluster effects. The challenge is that local methods don't work in high dimensions. The

[5] Borrowing the "Tragic Triad" component from Yu et al. [2020]

"success" of neural networks in high dimensions is there ability to learn "feature representations". So in some sense, our aim is to partial out the cluster effects from this learnt feature representation.

Our approach is motivated in part by Domingos [2020] which illustrates how neural networks can be understood within the framework of kernel machines. Specifically, Domingos [2020] shows that in the gradient flow regime, neural networks can be understood as kernel machines where the kernel is formed by integrating the inner product of gradient of the neural network evaluated at the corresponding points of the domain. **Suggestive that we should partial out the cluster effects along the path of learning because it's the entire path that matters.**

$$K_\theta(x, x') = \int_0^1 \Big\langle \nabla_\theta f_x\big(\theta(t)\big), \nabla_\theta f_{x'}\big(\theta(t)\big) \Big\rangle dt$$

The Search Space

$$\theta \longmapsto \big(x \longmapsto f_\theta(x, z)\big)$$

The Search Method

$$\theta_t = \theta_{t-1} - \alpha \nabla \mathcal{L}(\theta_{t-1})$$

Figure 7: Caption

4.1 Description

We can express our objective function as the weighted average of a loss function evaluated at two different parameter values.

$$R_k(\theta) := \frac{1}{N} \sum_i \alpha L_i(\theta) + (1 - \alpha) L_i(\theta_c^k(\theta)) \tag{1}$$

The first term captures the typical squared prediction loss. Here we assume that the outcome is a real-valued, but the same approach works for discrete outcomes as well.

$$L_i(\theta) := (y_i - f(\theta, x_i))^2 \tag{2}$$

The second term contains the same loss function, but the parameter corresponds to the learnt parameter based on k steps of gradient descent on observations from the same cluster.

$$\theta_c^k(\theta) := \theta^{k-1} - \alpha \nabla_\theta \frac{1}{n_c} \sum_{i \in c} L_i(\theta^{k-1}), \quad \theta^0 = \theta \tag{3}$$

Together, this can be understood as a regularized version of the popular metal-learning algorithm MAML. We highlight the importance of the regularization parameter in the results section.

4.2 Why Might This Work?

> “Simpler functions after k steps of gradient descent are closer to complex functions than complex functions are to other complex functions after k steps of gradient descent”

$$d(h, g) > d(h, f_c^*(\theta)) +$$

We can think about the optimization problem as minimizing a metric define on a set of functions

$$\begin{aligned} d(\theta_1, \theta_2) &= \int \|f(\theta_1) - h(\theta_2)\|^2 d\hat{\mathbb{P}}_X \\ &= \sum_{c \in C} \int \|f(\theta_1) - h(\theta_2)\|^2 d\hat{\mathbb{P}}_{X|C} \end{aligned}$$

The second term in our objective function though, is captured by the following metric.

$$d(\theta_1, \theta_2) = \sum_{c \in C} \int \|f(\theta_c^*(\theta_1) - f(\theta_c^*(\theta_2)\|^2 d\hat{\mathbb{P}}_X$$

What’s potentially interesting, though, is that this is in fact not necessarily a metric because the triangle inequality is not always satisfied.

$$\begin{aligned} & d(\theta_1, \theta_2) > d(\theta_1, \theta_3) + d(\theta_3, \theta_2) \\ = \sum_{c \in C} \int \|f(\theta_c^*(\theta_1)) - f(\theta_c^*(\theta_2))\| d\hat{\mathbb{P}}_{X|C} \leq & \sum_{c \in C} \int \|f(\theta_c^*(\theta_1)) - f(\theta_c^*(\theta_3))\| d\hat{\mathbb{P}}_{X|C} \\ & + \sum_{c \in C} \int \|f(\theta_c^*(\theta_3)) - f(\theta_c^*(\theta_2))\| d\hat{\mathbb{P}}_{X|C} \end{aligned}$$

What solutions does the model prefer? How does this preference change when the model is overparameterized?

The Model

$$f(\theta, x) = \theta^T \phi(x)$$

The Update Rule

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta \mathcal{L}(\theta_t)$$

Standard Loss Function

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \big(y_i - \theta^T \phi(x_i)\big)^2$$

Standard Update Rule

$$\theta_{t+1} = \theta_t - \frac{\alpha 2}{n}\sum_{i=1}^{n} \big(\theta_t^T \phi(x_i) - y_i\big)\phi(x_i)$$

The RFP Loss Function

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \gamma(y_i - \theta^T \phi(x_i))^2 + (1-\gamma)(y_i - \theta_i^*(\theta)^T \phi(x_i))^2$$

The RFP Update Rule

$$\theta_{t+1} = \theta_t - \frac{\alpha 2}{n}\sum_{i=1}^{n} \Big[\gamma\big(y_i - \theta^T \phi(x_i)\big) + (1-\gamma)\big(y_i - \theta_i^*(\theta)^T \phi(x_i)\big)\partial \theta_i^*(\theta)^T\Big]\phi(x_i)$$

At a high level, it suggests early stopping at the cluster level. Building off the popular meta-learning algorithm MAML (Finn et al. [2017]), we train a neural network via bi-level gradient descent. In the inner level, we allow the parameters of the neural network to adopt to each cluster in parallel. By adding a regularization penalty which differentiates our approach from MAML, the model intuitively learns a representation which has a low prediction loss, but where given a few steps of gradient descent can significantly improve its loss on any cluster. In this way, we partial out the potentially nonparametric effects.

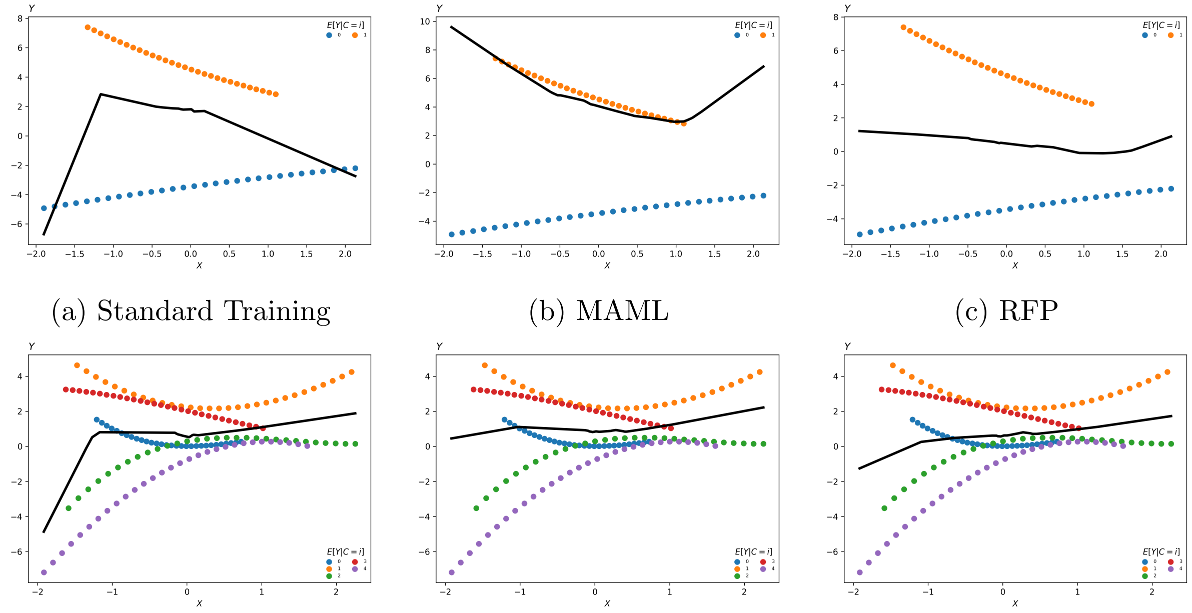
4.3 Regularization

- "In contrast with classical convex empirical risk minimization, where explicit regularization is necessary to rule out trivial solutions, we found that regularization plays a rather different role in deep learning. It appears to be more of a tuning parameter that often helps improve the final test error of a model, but the absence of all regularization does not necessarily imply poor generalization error. As reported by Krizhevsky et al. (2012), ‘2-regularization (weight decay) sometimes even helps optimization, illustrating its poorly understood nature in deep learning.”Zhang et al. [2021]

- “such uniform convergence bounds would require the sample size to be polynomially large in the dimension of the input and exponential in the depth of the network, posing a clearly unrealistic requirement in practice.”Zhang et al. [2021]

5 Results

We illustrate the relative importance of our design choices in figure 8. First, standard methods ignore the clustered nature of the data and therefore tend to overfit in the tails where the local observations are from the same cluster. Second, MAML - which is bi-level gradient descent with out the regularization terms - learns parameters values from which it can “quickly” minimize the cluster specific loss values. This learnt initialization though has no guarantee of in sample performance. As figure 8e illustrates, MAML essentially ignores one of the clusters. Importantly though, this issue isn’t apparent when there are multiple clusters with significant overlap. Therefore, the regularization parameter appears important when the clusters don’t overlap significantly, which we showed previously can occur with clustered data.



(a) Standard Training

(b) MAML

(c) RFP

(d) Standard Training

(e) MAML

(f) RFP

Figure 8: The grey and black dots represent data from separate clusters. Each figure corresponds to fitting a neural network to this data under different training algorithms

6 Conclusion

Why do we have estimators? It's because we cannot draw in high dimensional spaces.

In this paper we show that clustered treatment assignment together with cluster level controls can increase the variance of standard smoothing estimators. We illustrate that a regularized version of the popular meta-learning algorithm (MAML) is a potentially attractive estimation method in this context in that it partials out the potentially nonparametric cluster effects via bi-level gradient descent. Intuitively, this is akin to early stopping at the cluster level.

References

Randall Balestriero, Jerome Pesenti, and Yann LeCun. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*, 2021.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

Pedro Domingos. Every model learned by gradient descent is approximately a kernel machine. *arXiv preprint arXiv:2012.00152*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Arthur Lewbel. The identification zoo: Meanings of identification in econometrics. *Journal of Economic Literature*, 57(4):835–903, 2019.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.