Journal of Machine Learning Research 24 (2023) 1-60 Submitted 4/21; Revised 4/23; Published 6/23

Dropout Training is Distributionally Robust Optimal

José Blanchet JOSE.BLANCHET@STANFORD.EDU
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, USA

Yang Kang YANGKANG@STAT.COLUMBIA.EDU
Department of Statistics
Columbia University
New York, NY 10027, USA

José Luis Montiel Olea JLO67@CORNELL.EDU
Department of Economics
Cornell University
Ithaca, NY 14850, USA

Viet Anh Nguyen NGUYEN@SE.CUHK.EDU.HK
Department of Systems Engineering and Engineering Management
Chinese University of Hong Kong
Hong Kong SAR

Xuhui Zhang XUHUI.ZHANG@STANFORD.EDU
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, USA

Editor: Samory Kpotufe

Abstract

This paper shows that dropout training in generalized linear models is the minimax solution of a two-player, zero-sum game where an adversarial nature corrupts a statistician's covariates using a multiplicative nonparametric errors-in-variables model. In this game, nature's *least favorable distribution* is *dropout noise*, where nature independently deletes entries of the covariate vector with some fixed probability δ. This result implies that dropout training indeed provides out-of-sample expected loss guarantees for distributions that arise from multiplicative perturbations of in-sample data. The paper makes a concrete recommendation on how to select the tuning parameter δ. The paper also provides a novel, parallelizable, unbiased multi-level Monte Carlo algorithm to speed-up the implementation of dropout training. Our algorithm has a much smaller computational cost compared to the naive implementation of dropout, provided the number of data points is much smaller than the dimension of the covariate vector.

Keywords: generalized linear models, generalization error, distributionally robust optimization, machine learning, multi-level Monte Carlo

1. Introduction

Dropout training is an increasingly popular estimation method in machine learning.[1] The general idea consists in ignoring some dimensions of the covariate vector at random while estimating the parameters of a statistical model. A common motivation for dropout training is that the random feature selection implicitly performs *model averaging*, potentially improving prediction error.[2]

This paper contributes to the growing literature explaining how dropout training can improve a predictor's *generalization error*, e.g., Wager et al. (2013), Helmbold and Long (2015), Wei et al. (2020). Broadly speaking, a predictor's generalization error refers to its ability to *"perform well on new, previously unseen inputs—not just those on which our model was trained."* (Goodfellow et al., 2016, Section 5.2). As we explain below, our main result shows that dropout training improves generalization error over distributions that arise from multiplicative perturbations of in-sample covariates.

To make this point, this paper studies dropout training in the context of generalized linear models. A generalized linear model for the scalar outcome variable $Y \in \mathcal{Y} \subseteq \mathbb{R}$, given a d-dimensional vector of covariates $X = (X_1, \ldots, X_d)^\top \in \mathbb{R}^d$, is defined by the conditional density

$$f(Y|X,\theta) \equiv h(Y,\phi) \exp\left(\left(Y\beta^\top X - \Psi(\beta^\top X)\right)/a(\phi)\right), \tag{1}$$

where the model's parameters are $\theta \equiv (\beta^\top, \phi)^\top$; see McCullagh and Nelder (1989, Equation 2.4). In our notation, $h(\cdot,\phi)$ is a real-valued function parameterized by ϕ and defined on the domain $\mathcal{Y}$, $a(\cdot)$ is a positive function of ϕ, and $\Psi(\cdot)$ is the log-partition function which we assume to be defined on all the real line. Normal, logistic, and Poisson regression all have conditional densities of the form in Equation (1).

Prediction in generalized linear models is typically based on the conditional expectation of Y, given X, implied by the likelihood in (1), evaluated at an estimated parameter vector $\widehat{\theta}$ (for example, one common choice for $\widehat{\theta}$ is the vector containing the maximum likelihood estimators for β and ϕ based on the available data).[3] We define the generalization error of a predictor for Y as:

$$\mathbb{E}_{\mathbb{Q}}\left[-\ln f(Y|X,\widehat{\theta}\,)\right], \text{ where } (X,Y) \sim \mathbb{Q}. \tag{2}$$

[1] Section 7.12 of Goodfellow et al. (2016) provides a textbook treatment on dropout training. Bishop (1995) and Srivastava et al. (2014) are seminal references on this topic.

[2] See Hinton et al. (2012) for a discussion about this point in the context of neural networks. See also Draper (1994) and Raftery et al. (1997) for classical results on the optimality of model averaging for prediction purposes.

[3] Throughout the paper, we will typically use $\widehat{\theta}, \widehat{\beta}, \widehat{\phi}$ to denote the estimators of parameters θ, β, ϕ, without necessarily making explicit reference to the data or the sample size.

The above expectation is computed by fixing the estimated parameter vector, $\widehat{\theta}$, and then drawing new covariates and outcomes according to the joint distribution $\mathbb{Q}$. When $\mathbb{Q}$ differs from the data's empirical distribution, we interpret (2) as a measure of *out-of-sample* performance (as the draws from $\mathbb{Q}$ can be thought of as new, and previously unseen, inputs).

Our main result (Theorem 1) shows that, when $\mathbb{Q}$ corresponds to *multiplicative* perturbation of the covariates' empirical distribution (in a sense we make precise), the generalization error of dropout training is no larger than the in-sample loss of dropout training averaged over *dropout noise* (Corollary 1). Therefore, our main result implies that dropout training generalizes well to previously unseen data distributions arising from in-sample covariates' multiplicative perturbations.

We establish our main result by showing that dropping out input features when training generalized linear models can be viewed as a Distributionally Robust Optimization (DRO) problem (Shapiro et al., 2014). A DRO problem is a two-player, zero-sum game between a decision maker (a statistician) and an adversary (nature). The statistician wishes to choose an action to minimize a given expected loss (e.g., squared loss in a typical linear regression setting or, more generally, the negative of the log-likelihood function). At the same time, nature intends this loss to be maximal. We consider a framework in which nature can harm the statistician by corrupting the available covariates using a multiplicative nonparametric errors-in-variables model, as in the classic work of Hwang (1986).[4] Under mild assumptions, nature's *least favorable distribution* in this game turns out to be dropout noise, where nature independently deletes entries of the covariate vector with some fixed probability δ. Because dropout noise is least favorable, we can use the in-sample loss of dropout training—averaged over the dropout noise—to upper bound the generalization error of predictors computed via dropout training, under any probability distribution that arises via multiplicative perturbation of the covariates in the sample.

One might argue that the significance and usefulness of Theorem 1 is limited, as the distributions over which dropout training generalizes well only cover perturbations around the data's empirical distribution. Indeed, in the quintessential definition of generalization error, the new examples are typically drawn from the true data-generating process, which we denote as P^*. Do our results have anything to say about the case in which the generalization error in (2) is evaluated at P^*? We answer this question in the positive, provided that the dropout probability, δ, is chosen to be $c/\sqrt{n}$ where n denotes the number of training examples. In particular, we show that the generalization error of dropout training based on examples drawn from the true data-generating process will be bounded above by the loss of dropout in the training sample (averaged over the dropout noise) with a probability

[4] As far as we know, this simple action space for nature is a novel model for conceptually and quantitatively understanding dropout training.

that depends on c (Theorem 2). Consequently, by choosing a target probability, say 95%, it is possible to provide a concrete recommendation for selecting c and, therefore, for δ. As we will explain later, our concrete recommendation for choosing the dropout probability in generalized linear models stands in contrast to ad hoc suggestions available for neural networks, in which an input unit is usually included with probability 0.8, and a hidden unit is included with probability 0.5 (Goodfellow et al., 2016, Chapter 7, p. 257).

Finally, and to analyze the implications of our results numerically, we make an additional contribution by suggesting a new stochastic optimization implementation of dropout training. The generalization error of the standard stochastic gradient descent implementation of dropout training here profits from the implicit regularization imposed by the stochastic gradient descent routine (Wei et al., 2020). Since our theoretical analysis makes no use of this implicit regularization, it is important to have an algorithm that does not introduce any further bias to the solution of dropout training. We borrow ideas from the multi-level Monte Carlo literature—in particular from the work of Blanchet et al. (2019a)—to suggest an unbiased (in a sense we will make precise) dropout training routine. Our algorithm is easily parallelizable and has a much smaller computational cost than naive dropout training methods when the number of features is large (Theorem 3). Our algorithm thus complements the recent literature suggesting approaches to speed-up dropout training by either using a parallelized implementation of stochastic gradient descent (Zinkevich et al., 2010) or a fast dropout training based on Gaussian approximations (Wang and Manning, 2013).

The rest of the paper is organized as follows. Section 2 explains dropout training in the context of generalized linear models. Section 3 presents a general description of the DRO framework used in this paper. Section 4 zeros in on the DRO problem by using the negative log-likelihood of generalized linear models to define a loss function for the statistician, and by allowing nature to harm the statistician via a multiplicative errors-in-variables model for the covariates. This section also presents our main theorem. Section 5 presents our approach to select the dropout probability, δ. Section 6 discusses different computational methods available for implementing dropout training (full integration, stochastic gradient descent, naïve Monte Carlo integration) and suggests our *unbiased multi-level Monte Carlo* algorithm. Section 7 presents some simulations comparing our recommended selection of δ to cross-validation and our preferred implementation of dropout training over stochastic gradient descent. Finally, Section 8 discusses extensions of our results to a particular class of feed-forward neural networks with a single hidden layer. We show that dropout training of the hidden units in the hidden layer is distributionally robust and optimal. All of the proofs are collected in the Appendix.

2. Dropout Training in Generalized Linear Models

This section describes dropout training in the context of generalized linear models. As with some other recent papers in the literature, we view generalized linear models as a convenient, transparent, and relevant framework to better understand the theoretical and algorithmic properties of dropout training.

For a given covariate vector x_i, and a user-selected constant, $\delta \in [0, 1)$, define the d-dimensional random vector

$$\xi_i = (\xi_{i,1}, \ldots, \xi_{i,d})^\top \in \{0, 1/(1-\delta)\}^d,$$

where each of the d entries of ξ_i is an independent draw from a scaled Bernoulli distribution with parameter $1 - \delta$. This is, for $j = 1, \ldots, d$:

$$\xi_{i,j} = \begin{cases} 0 & \text{with probability } \delta, \\ (1-\delta)^{-1} & \text{with probability } (1-\delta). \end{cases} \tag{3}$$

The distribution of $\xi_{i,j}$ collapses to $\xi_{i,j} = 1$ with probability 1 when $\delta = 0$.

Let $\odot$ denote the binary operator defining element-wise multiplication between two vectors of the same dimension. Consider the covariate vector

$$x_i \odot \xi_i \equiv (x_{i,1}\xi_{i,1}, \ldots, x_{i,d}\xi_{i,d})^\top. \tag{4}$$

Some entries of the new covariate vector are 0 (those for which $\xi_{i,j} = 0$), and the rest are equal to $x_{i,j}/(1-\delta)$.

In a slight abuse of notation, let $\mathbb{E}_\delta$ denote the distribution of the random vector ξ_i, whose distribution is parameterized by δ. The estimators of (β, ϕ) obtained by *dropout training* correspond to any parameters $(\widehat{\beta}(\delta), \widehat{\phi}(\delta))$ that solve the problem

$$\inf_{\beta,\phi} \; \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\delta \left[-\ln f(y_i | x_i \odot \xi_i, \beta, \phi) \right]. \tag{5}$$

Note that the maximum likelihood estimator of (β, ϕ), denoted as $(\widehat{\beta}_{\mathrm{ML}}, \widehat{\phi}_{\mathrm{ML}})$, equals $(\widehat{\beta}(0), \widehat{\phi}(0))$.

There is some empirical evidence that using intentionally corrupted features for training has the potential to improve the performance of machine learning algorithms (Maaten et al., 2013). Even if one is willing to accept that feature corruption is desirable for estimation, the choice of dropout noise in (3) seems arbitrary.

To explain the empirical success of the use of dropout noise. Wager et al. (2013) and Helmbold and Long (2015) theoretically analyzed the dropout training criterion (5) as the composition of the original loss (e.g., Equation 5 with $\delta = 0$) and a penalty term that they view as a regularizer. They then analyzed this dropout regularizer and compared its regularization effect with other regularizers (such as L1 and L2 penalties). Specifically, Wager et al. (2013) proposed a convex quadratic approximation for the regularization penalty using first-order Taylor approximation in generalized linear models. While Helmbold and Long (2015) conducted a more explicit analysis in the context of logistic regression and suggested several non-standard properties of the dropout regularizer.

Relative to these papers, our work focuses on providing a novel decision-theoretic interpretation of dropout training (in the population and the sample). We will argue there is a natural two-player, zero-sum game between a decision maker (statistician) and an adversary (nature) in which dropout training emerges naturally as a minimax solution. In this game, dropout noise is nature's least favorable distribution, and dropout training becomes the statistician's optimal action. The framework we use, from the stochastic optimization literature, is distributionally robust optimization.

3. Distributionally Robust Optimization

Consider a general prediction problem where there is a multivariate predictor $X \in \mathbb{R}^d$ and a scalar outcome variable $Y \in \mathbb{R}$. A distributionally robust optimization (DRO) problem is a simultaneous two-player zero-sum game between a decision maker (in our context, the statistician) and an adversary (nature).[5] In this section, we describe the action space for each player, their strategies, and the payoff function.

3.1 Actions and Payoff

The statistician's action space consists of vectors $\theta \in \Theta$. The ranking of the statistician's actions is contingent on the realization of (X, Y), and this is captured by a real-valued loss function $\ell(X, Y, \theta)$. If the statistician knew the distribution of (X, Y)—which we denote by $\mathbb{Q}$—the statistician's preferred choice of θ would be the solution to

$$\inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}} \left[\ell(X, Y, \theta) \right]. \tag{6}$$

[5] A seminal reference is the robust inventory control problem of Scarf (1958). Recent references describing the use of distributionally robust stochastic programs (like those considered in this paper) include Delage and Ye (2010) and Shapiro (2017). Christensen and Connault (2019) used distributionally robust optimization to characterize the sensitivity of counterfactual analysis with respect to distributional assumptions in a class of structural econometric models.

Instead of assuming that the distribution $\mathbb{Q}$ is determined exogenously, we think of the distribution $\mathbb{Q}$ as being chosen by nature. Thus, nature's action space consists of a set of probability distributions denoted as $\mathcal{U}$. We refer to this set as the *distributional uncertainty set*. If nature knew the action selected by the statistician, nature's preferred action would be

$$\sup_{\mathbb{Q}\in\mathcal{U}} \mathbb{E}_{\mathbb{Q}}[\ell(X,Y,\theta)]. \tag{7}$$

3.2 Strategies and Solution

The choices of θ and $\mathbb{Q}$ are assumed to happen simultaneously. A statistician's strategy for this game consists of a choice of θ; nature's strategy for this game consists of a choice of $\mathbb{Q}$.

A *Nash equilibrium* for this game is a pair $(\theta^\star, \mathbb{Q}^\star)$ such that: a) given $\mathbb{Q}^\star$, the parameter $\theta^\star$ solves (6) and b) given $\theta^\star$, the distribution $\mathbb{Q}^\star$ solves (7).

The *minimax solution* for this game is a pair $(\theta, \mathbb{Q})$ that solves

$$\inf_{\theta\in\Theta} \sup_{\mathbb{Q}\in\mathcal{U}} \ \mathbb{E}_{\mathbb{Q}}[\ell(X,Y,\theta)], \tag{8a}$$

while the *maximin* solution is based on the mathematical program

$$\sup_{\mathbb{Q}\in\mathcal{U}} \inf_{\theta\in\Theta} \ \mathbb{E}_{\mathbb{Q}}[\ell(X,Y,\theta)]. \tag{8b}$$

If $\mathbb{Q}$ solves (8b), we say that $\mathbb{Q}$ is *nature's least favorable distribution*. The mathematical program in (8a) is typically referred to as a DRO problem.

4. Dropout Training is Distributionally Robust Optimal

We now specialize the general DRO framework of Section 3 by imposing two restrictions. First, we use the negative log-likelihood of generalized linear models (McCullagh and Nelder, 1989) as a loss function for the statistician. Second, we define nature's uncertainty set (i.e., the possible data distributions that nature can take) using the multiplicative errors-in-variables model of Hwang (1986).

4.1 Statistician's Payoff

We define the loss function for the statistician to be the negative of the logarithm of the likelihood in (1), that is,

$$\ell(X,Y,\theta) = -\ln h(Y,\phi) + (\Psi(\beta^\top X) - Y(\beta^\top X))/a(\phi), \tag{9}$$

where $\theta \equiv (\beta^\top, \phi)^\top \in \Theta \subseteq \mathbb{R}^d \times \mathbb{R}_+$. Equation (9) defines the statistician's objective and its set of actions.

4.2 Nature's Distributional Uncertainty Set

We now define the possible distributions that nature can choose. We start out by letting $\mathbb{Q}_0$ denote some benchmark or *reference* distribution over (X, Y). This distribution will typically be the empirical distribution of the data, but we present our main result allowing for a general $\mathbb{Q}_0$. This distribution need not correspond to that induced by a generalized linear model. In other words, our framework allows for the statistician's model to be misspecified.

Next, we define nature's action space by considering perturbations of $\mathbb{Q}_0$. Although there are different ways of doing this—for example, by using either f-divergences (such as the Kullback–Leibler divergence, as used by Nguyen et al. 2020 and the χ^2 divergence, as done by Duchi and Namkoong 2019) or the optimal transport distance (such as the Wasserstein distance, as in Blanchet et al. 2019b) to define a neighborhood—we herein use a nonparametric multiplicative errors-in-variables model as in Hwang (1986).[6]

The idea is to allow nature to independently introduce measurement error to the covariates using multiplicative noise. Let $\xi \equiv (\xi_1, \ldots, \xi_d)^\top$ be defined as a d-dimensional vector of random variables that are independent of (X, Y). We perturb the distribution $\mathbb{Q}_0$ by considering the transformation

$$(X, Y) \mapsto (X_1\xi_1, \ldots, X_d\xi_d, Y)^\top.$$

As a result, each covariate X_j is distorted, in a multiplicative fashion, by ξ_j. We often abbreviate $(X_1\xi_1, \ldots, X_d\xi_d)^\top$ by $X \odot \xi$, where $\odot$ signifies element-wise multiplication.

We restrict the distribution of ξ in the following way: First, for a parameter $\delta \in [0, 1)$, we define $\mathcal{Q}_j(\delta)$ to be the set of distributions for ξ_j that are supported on the interval $[0, 1/(1-\delta)]$ and that have mean equal to 1. More specifically,

$$\mathcal{Q}_j(\delta) \equiv \left\{ \mathbb{Q}_j : \mathbb{Q}_j \text{ is a probability distribution on } \mathbb{R},\ \mathbb{Q}_j([0, (1-\delta)^{-1}]) = 1,\ \mathbb{E}_{\mathbb{Q}_j}[\xi_j] = 1 \right\}. \tag{10}$$

This set of distributions, prescribed using support and first-order moment information, is popular in the DRO literature thanks to its simplicity and tractability (Wiesemann et al., 2014). From the perspective of an errors-in-variables model, these distributions are also

[6] The different choices of action space in the DRO literature have been shown to enjoy regularization benefits. For example, Duchi and Namkoong (2019) showed that a DRO formulation with χ^2 divergence leads to convex variance regularization. Blanchet et al. (2019b) showed that DRO with Wasserstein distance amounts to square-root LASSO in linear regressions. In contrast, the action space we introduce induces dropout regularization.

attractive because they preserve the expected value of the covariates, assuming that X_j and ξ_j are drawn independently.

Consider now the joint random vector $(X, Y, \xi) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d$. For a constant $\delta \in [0,1)$ consider the joint distributions over (X, Y, ξ) defined by

$$\mathcal{U}(\mathbb{Q}_0, \delta) = \{\mathbb{Q}_0 \otimes \mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d : \mathbb{Q}_j \in \mathcal{Q}_j(\delta) \ \forall j = 1, \ldots, d\}, \tag{11}$$

where $\otimes$ denotes the product measure (meaning that the joint distribution is the product of the independent marginals $\mathbb{Q}_j$, $j = 0, \ldots, d$). Thus, in the game, we consider $\mathcal{U}(\mathbb{Q}_0, \delta)$ as nature's action space or *nature's distributionally uncertainty set.*

We will make only one assumption about the reference distribution: $\mathbb{Q}_0$:

Assumption 1 *The distribution $\mathbb{Q}_0$ satisfies $\mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)] < \infty$ for any $\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)$, any $\theta \in \Theta$, and any scalar $\delta \in [0,1)$.*

This assumption implies a minimal regularity condition to guarantee that the expected loss is well defined for both the statistician and nature. Assumption 1 holds trivially when $\mathbb{Q}_0$ is the empirical distribution of the data, which is one of the main cases of interest in the paper.

4.3 Dropout Training is Distributionally Robust Optimal

Theorem 1 *Consider the two-player zero-sum game where the statistician has the loss function in* (9) *and nature has the action space in* (11) *for some reference distribution $\mathbb{Q}_0$ and a scalar $\delta \in [0,1)$. If Assumption 1 holds, then for any $\theta \in \Theta$,*

$$\sup_{\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)} \mathbb{E}_{\mathbb{Q}}\left[\ell(X \odot \xi, Y, \theta)\right] \tag{12}$$

is equivalent to

$$\mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta)], \tag{13}$$

where $\mathbb{Q}^\star = \mathbb{Q}_0 \otimes \mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$, and $\mathbb{Q}_j^\star = (1-\delta)^{-1} \times$ Bernoulli$(1-\delta)$ is a scaled Bernoulli distribution for any $j = 1, \ldots, d$, i.e., under $\mathbb{Q}_j^\star$

$$\xi_j = \begin{cases} 0 & \text{with probability } \delta, \\ (1-\delta)^{-1} & \text{with probability } (1-\delta). \end{cases} \tag{14}$$

In addition, let $\theta^\star \in \Theta$ be a minimizer of (13)*. Then $(\theta^\star, \mathbb{Q}^\star)$ constitutes a Nash equilibrium of the two-player zero-sum game defined by* (9) *and* (11)*, and $\mathbb{Q}^\star$ is nature's least favorable distribution.*

Proof See Appendix A.2. ■

The first part of the theorem characterizes nature's worst-case perturbation of $\mathbb{Q}_0$. From the statistician's perspective, nature's worst-case perturbation of $\mathbb{Q}_0$ is given by the *dropout noise* $\mathbb{Q}^\star$ in (14). Under this *worst-case* distribution, nature independently corrupts each entry of $X = (X_1, \dots, X_d)^\top$, either by dropping the j-th component (if $\xi_j = 0$), or by replacing it by $X_j/(1-\delta)$. Dropout training—which here refers to estimating the parameter θ after adding dropout noise to X—thus becomes the statistician's preferred way of estimating the parameter θ when facing an adversarial nature. This gives a decision-theoretic foundation for the use of dropout training.

Note that, in order to recover the objective function introduced in (5) (the sample average of the contaminated log-likelihood), it suffices to set the reference measure—$\mathbb{Q}_0$—as the empirical distribution $\widehat{\mathbb{P}}_n$ of $\{(x_i, y_i)\}_{i=1}^n$, which satisfies Assumption 1.

We provide now some intuition about how dropout noise becomes nature's worst-case distribution. Algebra shows that, in light of Assumption 1, the expected loss under an arbitrary distribution $\mathbb{Q}$ is finite and can be written as

$$\begin{aligned}\mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)] = &-\mathbb{E}_{\mathbb{Q}_0}\left[\ln h(Y,\phi)\right] \\ &+ \mathbb{E}_{\mathbb{Q}_0}\Big[\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[(\Psi((\beta \odot X)^\top \xi) - Y((\beta \odot X)^\top \xi))/a(\phi)]\Big],\end{aligned}$$

where the first expectation is taken with respect to the reference distribution, and the second with respect to ξ. For fixed values of (X, Y, θ), we can define the function

$$A_{(X,Y,\theta)}((\beta \odot X)^\top \xi) \equiv (\Psi((\beta \odot X)^\top \xi) - Y((\beta \odot X)^\top \xi))/a(\phi).$$

The key step to establishing Theorem 1 is to show that

$$\sup\Big\{\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[A_{(X,Y,\theta)}((\beta \odot X)^\top \xi)] : \mathbb{Q}_j \in \mathcal{Q}_j(\delta)\Big\} = \mathbb{E}_{\mathbb{Q}_1^\star \otimes \dots \otimes \mathbb{Q}_d^\star}[A_{(X,Y,\theta)}((\beta \odot X)^\top \xi)] \tag{15}$$

for any θ. The proof of this equality crucially exploits the convexity of the function $A_{(X,Y,\theta)} : \mathbb{R} \to \mathbb{R}$, which we show to be a consequence of the convexity of the log-partition function $\Psi(\cdot)$.[7] We make four important remarks about the equality in (15), and about the role that convexity plays in the optimality of dropout training.

Remark 1. First, a natural question to ask is whether the convexity of the log-partition function Ψ imposes a significant restriction on the class of generalized linear models that

[7] To derive our result, we first characterize the worst-case distribution for the expectation of a real-valued convex function (Lemma 1) and then we generalize this result to functions that depend on ξ only through linear combinations, as $A_{(X,Y,\theta)}(\cdot)$ (Lemma 2).

we are considering. It is known that the log-partition function of any generalized linear model with an open parameter space is convex; see Proposition 3.1 in Wainwright and Jordan (2008). This immediately implies that the convexity of Ψ should not be viewed as a significant restriction.

Remark 2. Second, it is reasonable to inquire whether our results concerning the optimality of dropout training could be extended beyond generalized linear models. For instance, one could be interested in considering a model in which linear predictors are obtained using an objective function of the form

$$\mathbb{E}_{\mathbb{Q}_0}[f(Y, \beta^\top X)],$$

where $f(y, \cdot) : \mathbb{R} \to \mathbb{R}$, and $(X, Y) \sim \mathbb{Q}_0$. This formulation includes certain types of extremum estimators as well as single-index models estimated by nonlinear least squares. Theorem 4 in Appendix A.6.1 shows that, if dropout noise is nature's worst-case distribution among multiplicative perturbations of covariates; that is, if

$$\sup_{\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)} \mathbb{E}_{\mathbb{Q}}[f(Y, \beta^\top X \odot \xi)] = \mathbb{E}_{\mathbb{Q}^\star}[f(Y, \beta^\top X \odot \xi)] \tag{16}$$

for some $\delta \in (0, 1]$, all $d \geq 1$, all $\beta \in \mathbb{R}^d$ and all reference distributions $\mathbb{Q}_0$, then $f(y, \cdot)$ *must* be a convex function on the real line for any given y. This means that convexity indeed plays a crucial role in the optimality of dropout training and, while it is possible to consider extensions outside the class of generalized linear models, such extensions will still have to impose some form of convexity (which may turn out to be very strong).[8] In Appendix A.6.2, we also argue that, beyond generalized linear models, the convexity of the objective function and the definition of $\mathcal{U}$ are, in general, not sufficient to make dropout training minimax-optimal.

Remark 3. Third, to further understand the extent to which convexity-like restrictions imply the optimality of dropout noise, we consider a generalization of the multiplicative errors-in-variables model that allows for correlated noise. We show that, in this case, there is a simple generalization of dropout (that we term "block dropout") that drops out blocks of variables at the same time. See Appendix A.6.3.

Remark 4. Fourth, Theorem 1 does not imply that $\mathcal{U}(\mathbb{Q}_0, \delta)$ is the largest class of distributions for which dropout training is distributionally robust optimal (or for which dropout noise is the worst-case distribution). In appendix A.6.4 we show that—in the context of the linear regression model—the dropout estimator can also be obtained by solving a distributionally robust optimization problem over "additive" contamination models (provided

[8] For instance, in Appendix A.6.1, we show that the only single-index model with an objective function of the form $f(y, \beta^\top x) = (y - g(\beta^\top x))^2$ for which dropout training is optimal is the linear regression model; that is, $g(\beta^\top x) = \beta^\top x$.

the additive perturbations are independent of the data, have a mean of zero, and satisfy a certain bound on their second moments). Thus, the union of additive and multiplicative perturbations still gives dropout noise as the worst-case distribution, and makes the dropout training estimator distributionally robust optimal.

How about the Nash Equilibrium of the two-player zero-sum game defined by Equations (9) and (11)? The equality in Equation (15) clearly shows that $\mathbb{Q}^{\bigstar}$ is nature's best response for any $\theta \in \Theta$. If there is a vector $\theta^{\bigstar}$ that solves the dropout training problem in Equation (13), then this vector is the statistician best's response to nature's choice of $\mathbb{Q}^{\bigstar}$. Consequently, $(\theta^{\bigstar}, \mathbb{Q}^{\bigstar})$ is a Nash equilibrium.

Finally, we discuss the extent to which $\mathbb{Q}^{\bigstar}$ can be referred to as nature's least favorable distribution, which has been defined as nature's solution to the maximin problem. It is well known that the maximin value of a game is always smaller than its minimax value:[9]

$$\sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)] \leq \inf_{\theta\in\Theta} \sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)].$$

We have shown that the right-hand side of the display above equals the infimum of (13). Therefore, if there is a $\theta^{\bigstar} \in \Theta$ that solves such program, then $\mathbb{Q}^{\bigstar}$ achieves the upper bound to the maximin value of the game. To see this, note that by definition of the left-hand side of the display above

$$\inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}^{\bigstar}}[\ell(X\odot\xi, Y, \theta)] \leq \sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)],$$

while we also have

$$\inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}^{\bigstar}}[\ell(X\odot\xi, Y, \theta)] = \inf_{\theta\in\Theta} \sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)],$$

by Equation (13). This makes dropout noise nature's least favorable distribution.

Now that we have established that dropout training gives the minimax solution of the DRO game, we discuss the implications of this result regarding the out-of-sample performance of dropout training. Suppose $\mathbb{Q}_0$ is the empirical measure $\widehat{\mathbb{P}}_n$ supported on n training samples $\{(x_i, y_i)\}_{i=1}^n$. Let $\widehat{\theta}(\delta)$ denote the estimators of (β, ϕ) based on dropout training with dropout probability δ. The *in-sample* loss of dropout training, averaged over dropout

[9] This follows from the fact that for any $\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)$:

$$\inf_{\theta\in\Theta} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)] \leq \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)] \leq \sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}}[\ell(X\odot\xi, Y, \theta)].$$

See also the discussion of the minimax theorem in Ferguson 1967, p. 81.

noise, is

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\mathbb{Q}^\bigstar}[\ell(X\odot\xi, Y, \widehat{\theta}(\delta))|X=x_i, Y=y_i], \tag{17}$$

which is equivalent to the objective function of dropout noise that we presented in equation (5), evaluated at the parameters estimated using dropout training; that is

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\delta}\left[-\ln f\left(y_i|x_i\odot\xi_i, \widehat{\beta}(\delta), \widehat{\phi}(\delta)\right)\right].$$

A typical concern about estimation procedures is whether their performance in a specific sample translates to good performance out of sample. In our context, the out-of-sample performance of dropout training can be thought of as the expected loss that would arise for some other data distribution $\tilde{\mathbb{Q}}$ over (X, Y) at the parameter estimated via dropout training:

$$\mathbb{E}_{\tilde{\mathbb{Q}}}[\ell(X, Y, \widehat{\theta}(\delta))].$$

The following is a direct corollary of Theorem 1:

Corollary 1 *Let $\widehat{\theta}(\delta) = (\widehat{\beta}(\delta)^\top, \widehat{\phi}(\delta))^\top$ denote the dropout estimators of β and ϕ given dropout noise δ. Consider any distribution $\tilde{\mathbb{Q}}$ over (X, Y) that can be obtained from $\widehat{\mathbb{P}}_n$ by perturbing covariates with mean-one, independent multiplicative error $\xi_j \in [0,(1-\delta)^{-1}]$. That is, if $\mathbb{Q}$ corresponds to the distribution of a vector of the form*

$$(\tilde{X}_1\xi_1, \dots, \tilde{X}_d\xi_d, Y)^\top,$$

where $(\tilde{X}_1, \dots, \tilde{X}_d, Y) \sim \widehat{\mathbb{P}}_n$ and $(\xi_1, \dots \xi_d)$ is a vector of i.i.d. random variables (independent of the empirical distribution of the data), supported on $[0,(1-\delta)^{-1}]$. Then,

$$\mathbb{E}_{\tilde{\mathbb{Q}}}[\ell(X, Y, \widehat{\theta}(\delta))] \leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\delta}\left[-\ln f\left(y_i|x_i\odot\xi_i, \widehat{\beta}(\delta), \widehat{\phi}(\delta)\right)\right]. \tag{18}$$

This means that the objective function used for dropout training will provide an upper bound for the out-of-sample loss associated with multiplicative perturbations of the training data.

Finally, we note that Theorem 1 was stated for a scalar δ that is homogeneous across the multiplicative noise ξ_j. To model non-identical dropout noise, we can substitute the sets in Equations (10) and (11) by $\mathcal{Q}_j(\delta_j)$ for a collection of parameters $(\delta_1, \dots, \delta_d) \in [0,1)^d$. In this case, the results of Theorem 1 hold with $\mathbb{Q}_j^\bigstar = (1-\delta_j)^{-1} \times \text{Bernoulli}(1-\delta_j)$ for $j = 1, \dots, d$.

5. Statistical Guidance on Choosing δ

Theorem 1 above showed that dropout training is distributionally robust optimal and that nature's least favorable distribution is dropout noise with probability $\delta \in [0,1)$. This section suggests a strategy to pick this parameter to control the generalization error of dropout training.

5.1 Additional Notation

Let $\widehat{\phi}_n$ denote an arbitrary $\sqrt{n}$-asymptotically normal estimator for the scale parameter ϕ. Such an estimator can be the maximum likelihood estimator, which is known to be consistent and asymptotically normal under mild regularity conditions on the joint distribution of (X_i, Y_i) (Fahrmeir and Kaufmann, 1985). We also allow $\widehat{\phi}_n$ to be the dropout training estimator of ϕ but with a dropout probability of $c/\sqrt{n}$. Finally, as we did before, we let $\widehat{\beta}(\delta)$ denote the dropout estimator of β under dropout probability δ.

In this section, we assume that the observed data was generated by a generalized linear model. Let (β^*, ϕ^*) denote the parameters that were used to generate the training data, and let P^* denote the corresponding joint distribution of covariates and outcomes under the true data generating process (hence, the training sample consists of n i.i.d. draws from P^*).

A key quantity in this section is

$$\mathcal{L}_n(\beta, \phi, \delta) \equiv \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\delta \left[-\ln f(y_i | x_i \odot \xi_i, \beta, \phi) \right], \tag{19}$$

which equals the in-sample (or empirical) loss at parameters β and ϕ, but averaged over the dropout noise with dropout probability δ.

To speak about the generalization error of dropout training, we define the "population" loss at (β, ϕ) as

$$\mathcal{L}(\beta, \phi) \equiv \mathbb{E}_{P^*} \left[-\ln f(Y | X, \beta, \phi) \right]. \tag{20}$$

Then,

$$\mathcal{L}(\widehat{\beta}(\delta_n), \widehat{\phi}_n) \tag{21}$$

is the loss that will arise from evaluating the negative log-likelihood in Equation (1) at $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$ and then averaging over values of (X, Y) drawn according to P^*. This corresponds to the definition of generalization error used in Equation (2), evaluated at P^*. If the negative log-likelihood is used as a measure of performance of the estimators $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$, then an upper bound on (21) can be thought of as an upper bound on the generalization error of $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$.

The main result in this section shows that the event

$$\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) \geq \mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n)), \tag{22}$$

can be guaranteed to hold with probability at least $1-\alpha$, for an appropriate choice of the sequence δ_n. This means that we can upper bound the generalization error of the estimators $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$ with probability at least $1-\alpha$, as we explain next.

5.2 Additional Assumptions and Main Result of this Section

Assumption 2 *The log-partition function $\Psi(\cdot)$ has a bounded second derivative.*

Assumption 3 *The second moment matrix $\mathbb{E}_{P^*}[XX^\top]$ is finite, positive definite.*

Assumption 4 *For some σ^2:*

$$\sqrt{n}\left(\mathcal{L}_n(\beta^*, \widehat{\phi}_n, 0) - \mathcal{L}(\beta^*, \phi^*)\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2).$$

We can now state this section's main result. Define the random variable

$$\mu_n(\beta, \phi, \delta) \equiv \mathcal{L}_n(\beta, \phi, \delta) - \mathcal{L}_n(\beta, \phi, 0), \tag{23}$$

which is nonnegative for any $\delta \in [0,1)$ by Theorem 1.

Theorem 2 *Suppose that Assumptions 2, 3, 4 hold. Then for any sequence $\delta_n = c/\sqrt{n}$,*

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) - \mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n)\right) \xrightarrow{d} \mathcal{N}(\mu_\infty(\beta^*, \phi^*, c), \sigma^2),$$

where $\mu_\infty(\beta^*, \phi^*, c) \geq 0$ is a linear function of c, strictly increasing for almost every (β^*, ϕ^*), and equal to the probability limit of

$$\sqrt{n}\mu_n\left(\beta^*, \widehat{\phi}_n, \delta_n\right),$$

and $\mu_n(\cdot)$ is defined as in Equation (23).

Proof See Appendix A.3. ■

The main message of Theorem 2 is that the probability of the event

$$\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) \geq \mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n)$$

can be approximated, as the sample size goes large, by the probability—under a normal random variable with positive mean—of the positive half of the real line. That is:

$$\mathbb{P}^* \left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) \geq \mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n) \right) = P\left(\mathcal{N}(\mu_\infty(\beta^*, \phi^*, c), \sigma^2) \geq 0 \right) + o(1).$$

Since μ_∞ is nonnegative, it is a strictly increasing function of c for almost every (β^*, ϕ^*), then c can be chosen to guarantee that the probability of the first term in the right-hand side of the equality is as close to one as desired. This means that we can choose the dropout probability to guarantee that the generalization error associated with the estimator $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$—which we have denoted as $\mathcal{L}(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$—admits an upper bound (estimable based on information in the sample) with high probability. See Equation (29) for our explicit recommendation on how to choose $\delta = c/\sqrt{n}$ based on Theorem 2.

Some elementary algebra can be used to illustrate the main argument behind the proof. It is convenient to start analyzing the difference

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) - \mathcal{L}(\beta^*, \phi^*) \right)$$

instead of

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) - \mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n) \right).$$

Algebra shows that

$$\begin{aligned}
\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) - \mathcal{L}(\beta^*, \phi^*) \right) &= \sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) - \mathcal{L}_n(\beta^*, \widehat{\phi}_n, \delta_n) \right) && (24)\\
&+ \sqrt{n}\mu_n(\beta^*, \widehat{\phi}_n, \delta_n) && (25)\\
&+ \sqrt{n}\left(\mathcal{L}_n(\beta^*, \widehat{\phi}_n, 0) - \mathcal{L}(\beta^*, \phi^*) \right). && (26)
\end{aligned}$$

We show that the sum of these terms converges in distribution to a normal. The proof has three main steps that we explain below.

Step 1. We start by showing that term (24) converges in probability to zero. To do this, we show that

$$\sqrt{n}(\widehat{\beta}(c/\sqrt{n}) - \beta^*) \tag{27}$$

is asymptotically normal and that the derivative of $\mathcal{L}_n(\cdot)$ with respect to β (evaluated at β^*) converges in probability to zero. One important observation is that mean of the asymptotic distribution of Equation (27) is nonzero, and depends linearly on c. Consequently, even though the dropout training estimator is asymptotically normal, its limiting distribution exhibits bias (and the norm of such bias increases linearly as a function of c^2). This first step thus shows that choosing a larger dropout probability comes at the cost of a decrease in accuracy in the estimation of β^*. Our recommended procedure for choosing c will guarantee that the generalization error of dropout training will be bounded with some prespecified probability, despite the decrease in estimation accuracy.

Step 2. We then show that term (25) has a finite probability limit. This is the key step in the proof. Importantly, we can characterize this limit explicitly and show that

$$\sqrt{n}\mu_n(\beta^*, \widehat{\phi}_n, \delta_n) \xrightarrow{p} c \cdot \mu,$$

where

$$\mu \equiv \left(\left(\sum_{\xi \in \mathcal{A}} \mathbb{E}_{P^*}[\Psi((X \odot \xi)^\top \beta^*)] \right) - d\mathbb{E}_{P^*}[\Psi(X^\top \beta^*)] + \mathbb{E}_{P^*}[YX^\top]\beta^* \right) \Big/ a(\phi^*),$$

and $\mathcal{A}$ is the collection of all vectors in $\{0,1\}^d$ for which there is only one zero. In Section 7, we provide an expression for this term in the linear regression model.

Step 3. The term (26) above has, by Assumption 4, an asymptotically normal distribution with mean zero and variance σ^2.

Step 4. Finally, in order to establish Theorem 1 we show that

$$\sqrt{n}\left(\mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n)) - \mathcal{L}(\beta^*, \phi^*) \right) = o_{P^*}(1), \tag{28}$$

which means that the expected loss evaluated at $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$ converges in probability to $\mathcal{L}(\beta^*, \phi^*)$ at a rate of at least $n^{-1/2}$.

It is important to mention that the DRO interpretation of dropout training can be leveraged to select the dropout parameter δ. For example, a possible approach is choosing δ so that the true data-generating process belongs to nature's choice set with some prespecified probability. This approach, which is often advocated in the literature on machine learning and robustness (Hansen and Sargent, 2008), often leads to a very pessimistic selection of δ simply because this criterion is not informed at all by the loss function defining the decision problem. Further, in our problem, it is impossible to apply this approach, given that the set of multiplicative perturbations of the empirical distribution will generally not cover the true data-generating process.

Another approach involves using generalization bounds leading to finite sample guarantees; see, for instance, a summary of this discussion in Section 6.2 of Rahimian and Mehrotra (2019). This method, while appealing, often requires either distributions with compact support or strong control on the tails of the underlying distributions. Also, often, the bounds depend on constants that may be too pessimistic or too difficult to compute.

Finally, Blanchet et al. (2019b) recently introduced a method for the case in which nature's choice set is defined in terms of the Wasserstein distance around the empirical distribution. The idea therein is that—for a fixed δ—every distribution that belongs to nature's choice set corresponds to an optimal parameter choice for the statistician. Thus, one can collect each and every of the statistician's optimal choices associated with each distribution in nature's uncertainty set, and treat the resulting region as a confidence set for the true parameter. This confidence set grows bigger (in the sense of nested confidence regions) as δ increases. The goal is then to minimize δ subject to a desired level of coverage in the underlying parameter to estimate. This leads to a data-driven choice of δ explicitly linked to the statistician's decision problem. However, this approach is not feasible for our problem because, once again, regardless of the value of δ, the parameter choices for each of the multiplicative perturbations of the empirical distribution will generally fail to cover the true parameter.

5.3 Our recommendation for choosing δ.

We advocate choosing the parameter δ to control how often the in-sample loss obtained from dropout training exceeds the population loss. The proof of Theorem 2 shows that $\mu_\infty(\beta^*, \phi^*, c)$ is of the form $c \cdot \mu$, where μ depends on (β^*, ϕ^*). Consequently, as long as $\mu > 0$, it is straightforward to pick c to guarantee a pre-specified coverage of the population loss: for any $\alpha \in (0, 1)$, if we pick c to be

$$z_{1-\alpha} \cdot \sigma/\mu, \tag{29}$$

where $z_{1-\alpha}$ is the 1-α quantile of a standard normal, and μ is evaluated at consistent estimators of β^* and σ^* then the probability of the event (22) asymptotically approaches $1 - \alpha$.

6. An Algorithm for Dropout Training

The goal of this section is to suggest an algorithm for solving the dropout training problem

$$\inf_{\theta \in \Theta} \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta)],$$

where $\mathbb{Q}^\star = \widehat{\mathbb{P}}_n \otimes \mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$ and $\mathbb{Q}_j^\star$, $j = 1, \ldots, d$ is the dropout noise distribution defined in (14). Notice that we here consider the specific case in which $\mathbb{Q}_0$ is set to the empirical measure $\widehat{\mathbb{P}}_n$ supported on n training samples $\{(x_i, y_i)\}_{i=1}^n$. We will use $\theta_n^\star$ to denote the solution of the dropout training problem above. It will sometimes be convenient to rewrite this dropout training problem as

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta) \mid X = x_i, Y = y_i], \tag{30}$$

which coincides with expression (5). Conditioning on the values of (x_i, y_i) makes it clear that the expectation is computed over the d-dimensional vector ξ. We now briefly describe three common approaches to implement dropout training and discuss some of its limitations.

6.1 Naive Dropout Training

Because $\mathbb{Q}_j^\star$ places mass on only two points, namely 0 and $(1-\delta)^{-1}$, the support of the joint distribution $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$ has cardinality 2^d. Thus, a naïve approach to solving the dropout training problem specified by Equation (30) is to expand the objective function as a sum with $n \cdot 2^d$ terms, then apply a tailored gradient descent algorithm to the resulting optimization problem. Computationally, however, this approach is too demanding because the number of individual terms in the objective function grows exponentially with d.

6.2 Dropout Training via Stochastic Gradient Descent

Another method to solve the dropout training problem in Equation (13) is to use stochastic gradient descent (Robbins and Monro, 1951). This is tantamount to i) taking a draw of (x_i, y_i) according to its empirical distribution, ii) independently taking a draw of ξ_i using the distribution in Equation (3), and iii) computing the stochastic gradient descent update using

$$\nabla \ln f(y_i | x_i \odot \xi_i, \beta, \phi).$$

Given a current estimate $\widehat{\theta}$, we compute an unbiased estimate of the gradient to the objective function of (13), and move in the direction of the negative gradient by a step of suitable size. Since $\mathbb{Q}^\star$ is discrete, the expectation under $\mathbb{Q}^\star$ can be written as a finite sum, and, by differentiating under the expectation, we have

$$\nabla_\theta \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \widehat{\theta})] = \mathbb{E}_{\mathbb{Q}^\star}\left[\nabla_\theta \ell(X \odot \xi, Y, \widehat{\theta})\right]. \tag{31}$$

The standard SGD algorithm uses a naïve Monte Carlo estimator as an estimate of the gradient (31), that is, at iteration $k \in \mathbb{N}$ with incumbent solution $\widehat{\theta}^k$,

$$\nabla_\theta \mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \widehat{\theta}^k)] \approx \nabla_\theta \ell(x_k \odot \xi_k, y_k, \widehat{\theta}^k),$$

where (x_k, y_k, ξ_k) is an independent draw from $\mathbb{Q}^\star$.

One drawback of using SGD for our problem is that it is not easily parallelizable, and thus its implementation can be quite slow. Moreover, under the strong convexity assumption of the loss function ℓ, SGD exhibits only linear convergence (Nemirovski et al., 2009, Section 2.1). In contrast, gradient descent (GD) offers exponential convergence (Boyd and Vandenberghe, 2004, Section 9.3.1).

6.3 Naïve Monte Carlo Approximation for Dropout Training

Consider solving dropout training problem (30) using a naïve Monte Carlo approximation. Instead of using 2^d terms to compute

$$\mathbb{E}_{\mathbb{Q}^\star}[\ell(X \odot \xi, Y, \theta) \mid X = x_i, Y = y_i],$$

we approximate this expectation by taking a large number of K i.i.d. draws $\{\xi_i^k\}_{k=1}^K$, $\xi_i^k \in \mathbb{R}^d$, according to the distribution $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$. When d is large, this approximation is computationally cheaper than the naïve dropout training procedure described above, provided that $K \ll 2^d$.

Thus, the naïve Monte Carlo approximation of the dropout training problem is

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \left[\frac{1}{K} \sum_{k=1}^{K} \ell(x_i \odot \xi_i^k, y_i, \theta) \right], \tag{32}$$

where the random vectors ξ_i^k are sampled independently—over both k and i—using the distribution $\mathbb{Q}_1^\star \otimes \ldots \otimes \mathbb{Q}_d^\star$.

Relative to the solution of the dropout training problem—which we denoted as $\theta_n^\star$—the minimizer of approximation (32) is consistent and asymptotically normal as $K \to \infty$. This follows by standard arguments; for example, those in Shapiro et al. (2014, Section 5.1). There are, however, two problems that arise when using problem (32) as a surrogate for the dropout training problem. First, the solution to approximation (32) is a biased estimator for $\theta_n^\star$. This means that, if we average the solution over the $K \cdot n$ different values of ξ_i^k, the average solution need not equal $\theta_n^\star$. Second, implementing approximation (32) requires a choice of K and, to the best of our knowledge, there is no off-the-shelf procedure for picking it.

6.4 Unbiased Multi-level Monte Carlo Approximation for Dropout Training

To address these two issues, we apply the recent techniques suggested by Blanchet et al. (2019a) that we refer to as *unbiased multi-level Monte Carlo approximations.* Multi-level Monte Carlo methods (Giles, 2008, 2015) are a set of techniques for approximating the expectation of random variables. The adjective "multi-level" emphasizes the fact that random samples of different *levels* of accuracy are used in the approximation. Before presenting the detailed algorithm, we provide a heuristic description. To this end, let $\widehat{\theta}_n^{\bigstar}(K)$ denote the *level* K solution of the problem in (32); that is, the solution based on K draws. Define the random variable

$$\Delta_K \equiv \widehat{\theta}_n^{\bigstar}(K) - \widehat{\theta}_n^{\bigstar}(K-1)$$

and, for simplicity, assume that $\widehat{\theta}_n^{\bigstar}(0)$ is a vector of zeros. Under suitable regularity conditions:

$$\sum_{K=1}^{\infty} \mathbb{E}[\Delta_K] = \lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^{\bigstar}(K)] = \theta_n^{\bigstar}.$$

Consider now picking K^* randomly from some discrete distribution supported on the natural numbers. Let $p(\cdot)$ denote the probability mass function of such distribution and consider a Monte Carlo approximation scheme in which—after drawing K^*—we sample $K^* \cdot n$ different random vectors $\xi_i^k \in \mathbb{R}^d$ according to $\mathbb{Q}_1^{\bigstar} \otimes \ldots \otimes \mathbb{Q}_d^{\bigstar}$. The estimator

$$Z(K^*) \equiv \frac{\Delta_{K^*}}{p(K^*)}$$

has two sources of randomness. These are, firstly, the random choice of K^* and, secondly, the random draws ξ_i^k. Averaging over both yields

$$\mathbb{E}[Z(K^*)] = \sum_{K=1}^{\infty} \mathbb{E}[Z(K^*)|K^* = K] \cdot p(K) = \sum_{K=1}^{\infty} (\mathbb{E}[\Delta_K]/p(K)) \cdot p(K) = \theta_n^{\bigstar}.$$

Thus, by taking into account the randomness in the selection of K, we have managed to provide a rule for deciding the number of draws (specifically, our recommendation is to pick K^* at random), and, at the same time, we have removed the bias of naïve Monte Carlo approximations.

One possible concern with our suggested implementation is that the expected computational cost of $Z(K^*)$ could be infinitely large. Fortunately, this issue can be easily resolved by an appropriate choice of the distribution $p(\cdot)$. To see this, define the computational cost simply as the number of random draws that are required to obtain $Z(K^*)$. In the construction we have described above, we need $K^* \cdot n$ draws for the construction of the estimator. Thus, the average cost is

$$\mathbb{E}[K^* \cdot n] = n \sum_{K=1}^{\infty} K \cdot p(K)$$

which, under mild integrability conditions on $p(\cdot)$, will be finite.[10]

We now present the algorithm that will be used to solve the dropout training problem. To ensure that the estimator $Z(K^*)$ has a finite variance, instead of defining Δ_K as the difference between the level K and $K-1$ solutions to the approximation problem (32) in the above heuristic arguments, we use solutions for a sample of size 2^{K+1} and with its *odd* and *even* sub-samples of size 2^K.

6.5 Algorithm for the Unbiased Multilevel Monte Carlo

We present a parallelized version using L processors, which works even when $L = 1$. Parallel computing reduces the variance of the estimator, so we suggest using as many processors as are available per run.

Fix an integer $m_0 \in \mathbb{N}$ such that $2^{m_0+1} \ll 2^d$. For each processor $l = 1, \ldots, L$ we consider the following steps.

i) Take a random (integer) draw, m_l^*, from a geometric distribution with parameter $r > 1/2$.[11]

ii) Given m_l^*, take $2^{K_l^*+1}$ i.i.d. draws from the d-dimensional vector $\xi_i \sim \mathbb{Q}_1^{\bigstar} \otimes \ldots \otimes \mathbb{Q}_d^{\bigstar}$, where

$$K_l^* \equiv m_0 + m_l^*.$$

Repeat this step independently for each $i = 1, \ldots, n$.

iii) Solve problem (32) using the first 2^{m_0} i.i.d. draws of ξ_i for each i. Let θ_{l,m_0} denote a minimizer.

iv) Denote by $\widehat{\theta}_n^{\bigstar}(2^{K_l^*+1})$, $\widehat{\theta}_n^{O}(2^{K_l^*})$, and $\widehat{\theta}_n^{E}(2^{K_l^*})$ any solution to the following optimization problems (all of which are based on sample average approximations as the Monte

[10] For example, if $p(\cdot)$ is selected as a geometric distribution with parameter r, the expected computational cost will be $n(1-r)/r$.

[11] To see why we require that $r > 1/2$, notice that, if the computational cost of evaluating $Z(K^*)$ (as in the heuristic description above) increases exponentially in K and takes the form $C \cdot 2^K$, the expected computational cost will be

$$\sum_{K=1}^{\infty} Cr(2(1-r))^K = Cr(1/2(1-r)),$$

provided $2(1-r) < 1$, or equivalently, $r > 1/2$. As we show in the proof of Theorem 3, constraining the variance requires then imposing $r < 3/4$. Ultimately, optimizing the product of computational cost and variance leads to the optimal selection $r = 1 - 2^{-3/2}$.

Carlo approximation 32):

$$\widehat{\theta}_n^{\bigstar}(2^{K_l^*+1}) \in \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2^{K_l^*+1}}\sum_{k=1}^{2^{K_l^*+1}} \ell(x_i \odot \xi_i^k, y_i, \theta)\right),$$

$$\widehat{\theta}_n^{O}(2^{K_l^*}) \in \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2^{K_l^*}}\sum_{k=1}^{2^{K_l^*}} \ell(x_i \odot \xi_i^{2k-1}, y_i, \theta)\right),$$

$$\widehat{\theta}_n^{E}(2^{K_l^*}) \in \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2^{K_l^*}}\sum_{k=1}^{2^{K_l^*}} \ell(x_i \odot \xi_i^{2k}, y_i, \theta)\right).$$

Intuitively, $\widehat{\theta}_n^O$ and $\widehat{\theta}_n^E$ denote the solutions to problem (32) but using a sample of size $2^{K_l^*}$ with only *odd* indices and *even* indices, respectively.

v) Define

$$\bar{\Delta}_{K_l^*} \equiv \widehat{\theta}_n^{\bigstar}(2^{K_l^*+1}) - \frac{1}{2}(\widehat{\theta}_n^{O}(2^{K_l^*}) + \widehat{\theta}_n^{E}(2^{K_l^*}))$$

and let

$$Z(K_l^*) = \frac{\bar{\Delta}_{K_l^*}}{r(1-r)^{K_l^*-m_0}} + \theta_{l,m_0}.$$

Our recommended estimator is

$$\frac{1}{L}\sum_{l=1}^{L} Z(K_l^*).$$

We now show that the suggested algorithm gives an estimator with desirable properties. We do so under the following regularity assumptions.

Assumption 5 *Suppose that the parameter space Θ is compact. Suppose in addition that the optimal solution $\theta_n^{\bigstar}$ to the dropout training problem in* (30) *is (globally) unique.*

Assumption 6 *Let $\widehat{\theta}_n^{\bigstar}(K)$ denote the solution of the problem in* (32) *based on K draws. Suppose that as $K \to \infty$,*

$$\mathbb{E}[\|K^{\frac{1}{2}}(\widehat{\theta}_n^{\bigstar}(K) - \theta_n^{\bigstar})\|_2^4) = O(1),$$

where the expectation is taken over the i.i.d. dropout noise distribution used to generate ξ_i^k.

Assumption 7 *Assume that, for each (X, Y, ξ), $\ell(X \odot \xi, Y, \cdot)$ is thrice continuously differentiable over Θ and that the Hessian matrix $\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}[\ell(X \odot \xi, Y, \theta_n^{\bigstar})]$ is non-singular.*

Theorem 3 *Under Assumption 5,* $\mathbb{E}[Z(K_l^*)] = \theta_n^\bigstar$. *The number of random draws required to compute* $Z(K_l^*)$ *is* $n \cdot 2^{K_l^*+1}$ *and thus the expected computational complexity for producing* $Z(K_l^*)$ *equals*

$$\frac{n(2^{m_0+1})r}{2r-1} < n(2^{m_0+1}) \ll n2^d.$$

Suppose, in addition, that $\widehat{\theta}_n^\bigstar(K)$ *is almost surely in the interior of* Θ *for large enough* K. *If Assumptions 6 and 7 hold, and* $r < 3/4$, *then* $\mathrm{Var}(Z(K_l^*)) < \infty$.

Proof See Appendix A.4. ■

Our suggested algorithm has finite expected computational complexity that does not grow exponentially with the dimensionality d, thus, every time we need to obtain $\widehat{\theta}_n^\bigstar(2^{K_l^*+1})$, we can do so by gradient descent. Combined with parallelization, the unbiased multi-level Monte Carlo approach produces an unbiased estimator with a variance that can be made arbitrarily small if L is large enough, provided that the regularity assumptions that give $\mathrm{Var}(Z(K_l^*)) < \infty$ hold.

7. Numerical Experiments

We conduct numerical experiments in this section to compare our preferred implementation of dropout training to stochastic gradient descent, as well as our recommended selection of δ to cross-validation. The benefits of our suggested unbiased multi-level Monte Carlo algorithm are analyzed using high-dimensional regression, whereas our selection of δ is analyzed using a low-dimensional regression model.

7.1 Advantage of the Unbiased Multi-level Monte Carlo Estimator

We present a simple numerical experiment to illustrate the advantage of using the unbiased multi-level Monte Carlo estimator suggested in Section 6.4. We consider the linear regression problem with known variance and we focus on solving the dropout training problem with our recommended δ chosen according to Proposition 2.

Our simulation setting considers a linear regression model with a covariate vector having dimensionality $d = 100$ and sample size $n = 50$. We pick a known regression coefficient $\beta_0 \in \mathbb{R}^d$ being a vector with all entries equal to 1. With fixed coefficients, we assume the covariate vector follows an independent Gaussian, as well as for the regression noise. More specifically, we can get our 50 observations (x_i, y_i) via

- sampling $x_i \sim \mathcal{N}(0, I_d)$, $i = 1, \ldots, n$,

- sampling $y_i \in \mathbb{R}$ conditional on x_i, where y_i is given by the linear assumption and ε_i are i.i.d. random noise following $\mathcal{N}(0, 10^2)$, for $i = 1, \ldots, n$.

Our simulation first considers a high-dimension setting (relatively low ratio between sample size per dimension $n/d = 0.5$) with high noise-to-signal ratio (variability on residual noise is high compared to the variability on x_i).

If we set $\mathbb{Q}_0$ to be the empirical distribution of $\{(x_i, y_i)_{i=1}^n\}$, the dropout training problem in the linear regression model is

$$\min_{\beta \in \mathbb{R}^d} \mathbb{E}_{\mathbb{Q}^\star} \left[\left(\beta^\top (X \odot \xi) - Y \right)^2 \right].$$

Corollary 2 in Appendix A.5 shows that, in the linear regression model, the dropout training problem can be written as

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \left[(\mathbf{Y} - \mathbf{X}\beta)^\top (\mathbf{Y} - \mathbf{X}\beta) + \frac{\delta}{1-\delta} \beta^\top \mathbf{\Lambda} \beta \right],$$

where $\mathbf{Y} = [y_1, y_2, \ldots, y_n]^\top$, $\mathbf{X} = [x_1, x_2, \ldots, x_n]^\top$ and $\mathbf{\Lambda}$ is the diagonal matrix with its diagonal elements given by the diagonals of $\mathbf{X}^\top \mathbf{X}$. Moreover, there is a closed-form solution for the dropout training problem and it is given by the ridge regression formula:

$$\beta_n^* = \left(\mathbf{X}^\top \mathbf{X} + \frac{\delta}{1-\delta} \mathbf{\Lambda} \right)^{-1} \mathbf{X}^\top \mathbf{Y}.$$

We choose the dropout probability δ following Proposition 2. More specifically, Proposition 2 suggests the choice $\delta = c/\sqrt{n}$ where $c = z_{1-\alpha} \cdot \sigma/\mu$. For linear regression with known variance, it is straightforward to compute

$$\mu = \frac{1}{2\phi^*} \sum_{j=1}^d \mathbb{E}_{P^*}[X_j^2](\beta_j^*)^2,$$

and

$$\sigma^2 = \mathbb{V}\mathrm{ar}_{P^*} \left[\frac{1}{2} \log(2\pi\phi^*) + \frac{(Y - (\beta^*)^\top X)^2}{2\phi^*} \right].$$

Choosing $\alpha = 0.1$ and note that $\beta^* = \beta_0, \phi^* = 10^2$, we have $\delta \approx 0.26$.

Since neither our suggested multi-level Monte Carlo algorithm nor standard SGD (as defined in Section 6.2) uses closed-form formulae for their implementation, we analyze the extent to which these procedures can approximate the parameter β_n^*. The two algorithms we compare are:

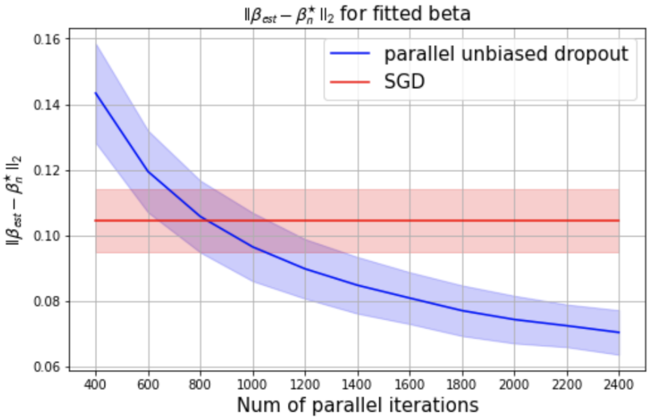

Figure 1: l_2 difference

- Standard SGD algorithm with a learning rate 0.0001 and initialization at the origin. Note that we take batched SGD instead of the single-sample SGD introduced in Section 6.2.

- Multi-level Monte Carlo algorithm with a geometric rate $r = 0.6$ and a burn-in period $m_0 = 5$. Note that in each parallel run, we use gradient descent (GD) with 0.01 learning rate and initialization at origin for steps iii) and iv) in Section 6.4.

We run our simulation on a cluster with two Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40 GHz processors (with 10 cores each), and a total memory of 128 GB. We fix 60 seconds as a "wall-clock time", so that we terminate the two algorithms after 60 seconds.[12] We run 1000 independent experiments. For each run, we calculate and report the average parameter estimation divergence to β_n^* and 1-standard deviation error bar for the divergence. We consider different number of parallelizations (i.e., L in Section 6.4) from 400 to 2400. We cap the run at 2400 due to the saturation of divergence after $\sim$ 2000 parallelizations.

Figure 1 shows the l_2 divergence from the true β_n^* of the two algorithms for varying L, while Figure 2 and Figure 3 show the l_∞ and l_1 divergence, respectively. We observed that our unbiased estimator outperforms the standard SGD algorithm once the number of parallel iterations exceeds some moderate threshold ($\sim$ 1000 here). We provide supporting evidence in Appendix A.7 to argue our choice of the learning rate, initialization, and wall-clock time, where our proposed algorithm is robust to any reasonable choices.

[12] The parameters for the SGD algorithm are appropriately tuned to achieve good convergence within 60 s (see Appendix A.7 for the tuning procedure). We do not claim that this choice of parameters is optimal.

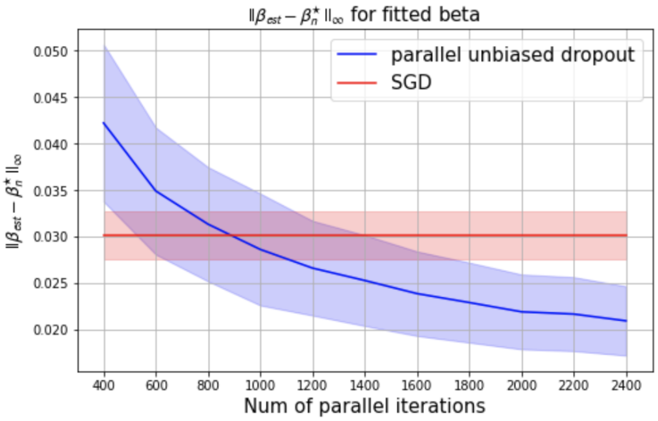

Figure 2: l_∞ difference

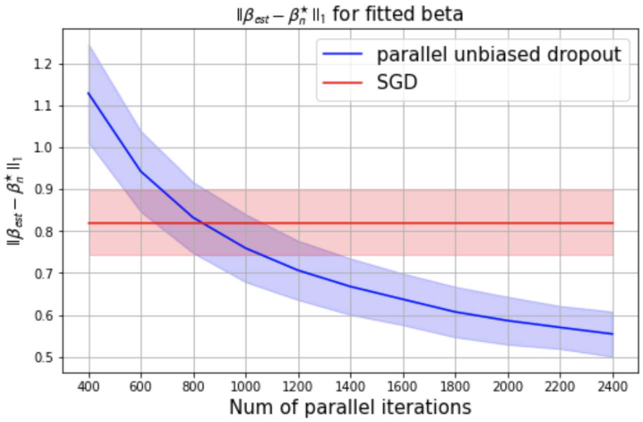

Figure 3: l_1 difference

7.2 Coverage of the True Loss of Dropout Training

Here, we validate that our recommended selection of δ guarantees that the in-sample loss of dropout training is covering the true loss with arbitrarily high probability as prescribed by Proposition 2.

We use the same linear regression model with dimensionality $d = 10$ and $n \in \{10^3, 10^4\}$ training samples. We choose different quantiles of the normal as in Proposition 2. We also include 10-fold cross-validation and ordinary least squares for comparison; see Table 1, where we estimate the frequency of coverage over 1000 independent runs. The main message is that our suggested choice of δ guarantees that the in-sample loss of dropout training exceeds the true, unknown, population loss with probability $1 - \alpha$. When using standard OLS or choosing δ by cross-validation, the in-sample loss is smaller than the population loss with probability close to 1/2, which implies that these methods are unsatisfactory in terms of frequency of coverage.

	$\alpha = 0.2$	$\alpha = 0.1$	$\alpha = 0.05$	10-fold CV	plain OLS
$n = 10^3$	0.77 ± 0.01	0.88 ± 0.01	0.94 ± 0.01	0.52 ± 0.02	0.40 ± 0.01
$n = 10^4$	0.79 ± 0.01	0.90 ± 0.01	0.94 ± 0.01	0.49 ± 0.02	0.47 ± 0.02

Table 1: Frequency of in-sample loss covering the true population loss. Our recommended selection of $\delta = c/\sqrt{n}$ with $c = z_{1-\alpha}\sigma/\mu$ has a theoretical $1-\alpha$ coverage probability.

8. Extensions

In this section, we discuss the extent to which the decision-theoretic support for dropout training carries over to neural networks. The main idea is to use a GLM model where the natural parameter is no longer a linear function of the covariates, but instead a neural network. We note that dropout training in neural networks typically drops neuron activations in both the input layer and the hidden layer. Dropout only at the input layer (data augmentation) is more closely related to our analysis in the context of GLM models and is also considered in the literature, for example by Devries and Taylor (2017) and Park et al. (2019) for vision and speech recognition tasks. In the remaining part of this section, we consider feed-forward neural network with one hidden layer, and consider dropout noise in both the input and hidden layers.

We also note that there are already concrete recommendations for how to select the dropout probability in neural networks. For example, Goodfellow et al. (2016, p. 257) says, “The probability of sampling a mask value of one (causing a unit to be included) is a hyperparameter fixed before training begins. It is not a function of the current value of the model parameters or the input example. Typically, an input unit is included with a probability 0.8, and a hidden unit is included with a probability 0.5. We then run forward propagation, back-propagation, and the learning update as usual." It is not entirely clear to us what the theoretical support is for this recommendation. We think that our results for choosing δ in GLM models (which depends on the sample size, and on the estimated values of the parameters of the model) provide, at the very least, a principled alternative to select the dropout probability of features in the hidden layer that precedes the output layer. Our simulations for GLM models (Table 1) suggest that this strategy indeed gives good bounds on generalization error.

8.1 One-hidden-layer Feed-Forward Neural Networks

Suppose the scalar response variable Y is generated by the conditional density

$$f(Y|X,\theta,\phi) \equiv h(Y,\phi)\exp\left(\left(Y\Omega_\theta(X)\right)-\Psi(\Omega_\theta(X))\right)/a(\phi)\right), \tag{33}$$

where $\Omega_\theta(X)$ is a neural network with parameters θ and $X \in \mathbb{R}^d$. This is a simple extension of the regression model that has been used recently to study deep neural networks—see Schmidt-Hieber (2020) in which the conditional density is Gaussian.

In this section, we will assume that $\Omega_\theta(X)$ is a neural network with a *single hidden layer*, a *differentiable activation (squashing) function*, and a *linear output function*. A function $h : \mathbb{R} \to [0,1]$ is a squashing function if it is non-decreasing and if

$$\lim_{r\to\infty} h(r) = 1, \quad \lim_{r\to-\infty} h(r) = 0.$$

For further detail, see Hornik et al. (1989, Definition 2.3).

Although these types of networks—which will be formally described below—are restrictive compared to the modern deep learning architectures, they can approximate any Borel measurable function from a finite-dimensional space to another, provided the hidden units in the hidden layer are large (Hornik et al., 1989).

Consider a neural network with K units in the hidden layer, each using input weights $w_k \in \mathbb{R}^d$, $k = 1,\ldots,K$. Denote the activation function in the hidden layer as $h(\cdot)$. Assume a linear output function with a vector of weights $\beta \in \mathbb{R}^K$. Thus, the network under consideration is defined by the function:

$$\Omega_\theta(X) \equiv \beta_1 h(w_1^\top X) + \ldots + \beta_K h(w_K^\top X) = \beta^\top H(X),$$

where $H(X) = (h(w_1^\top X), \ldots, h(w_K^\top X))^\top$. The neural network is parameterized by $\theta \equiv (\beta^\top, w_1^\top, \ldots, w_k^\top)^\top$. Under this model, the distribution of $Y|X$ is a GLM model with covariates $H(X)$.

8.1.1 Statistician's Objective Function

We will endow the statistician with the loss function given by the negative of the conditional log-likelihood for the model in Equation (33).

8.1.2 Nature's Uncertainty Set

We allow nature to introduce additional noise to the statistician's model. We do this in two steps. First, we allow nature to distort the distribution of X using a multiplicative noise denoted as $\xi(1) \in \mathbb{R}^d$. This is exactly analogous to what we did in the GLM model, where nature was allowed to pick a distribution for the covariates of the form $(X \odot \xi(1))$. We allow nature to contaminate the *input layer* with independent and multiplicative noise. Second, we also allow nature to contaminate *each of the hidden units* with multiplicative noise $\xi(2) \in \mathbb{R}^K$. That is, nature is also allowed to pick a vector $\xi(2) = (\xi(2)_1, \ldots, \xi(2)_K)^\top$, independently of $\xi(1) \in \mathbb{R}^d$, to distort the each of the K units in the hidden layer as

$$H(X) \odot \xi(2) \equiv (h(w_1^\top X)\xi(2)_1, \ldots, h(w_K^\top X)\xi(2)_K)^\top.$$

Our choice of a one-hidden neural network was simply for expositional simplicity, but the analysis would be the same with a feed-forward neural network with L hidden layers.

8.1.3 Minimax Solution

The minimax solution of the DRO game is given by

$$\inf_\theta \sup_{\mathbb{Q}} \mathbb{E}_{\mathbb{Q}} \left[-\ln f(Y|H(X \odot \xi_1) \odot \xi_2, \beta, \phi) \right], \tag{34}$$

where $\mathbb{Q}$ now refers to the joint distribution of $(X, Y, \xi(1), \xi(2))$ and $f(Y|X, \beta, \phi)$ is the GLM density defined in (1). We continue working with the assumption that $\xi \equiv (\xi(1)^\top, \xi(2)^\top)^\top$ has independent marginals and that it is independent of (X, Y).

We would like to solve for the worst-case distributions of the random vectors $\xi(1)$ and $\xi(2)$, assuming that both of these satisfy the restrictions analogous to Equation (11). The solution for the distribution of $\xi(2)$ can be obtained as a corollary to Theorem 1, as it suffices to define

$$\tilde{X} \equiv H(X \odot \xi(1))$$

and to view (34) as the DRO problem in a linear regression model, in which the data is $(\tilde{X}, Y)$ and $\xi(2) \in \mathbb{R}^K$ is simply the multiplicative noise that transforms the covariates into $(\tilde{X} \odot \xi(2))$.

The worst-case choice of $\xi(1)$, the multiplicative error for the inputs, is more difficult to characterize and we were not able to find general results for it. Below, we provide a heuristic argument suggesting that dropout noise might approximate the worst-case choice when the output layer is a Gaussian linear model. Let $\xi(1)_j$ denote the j-th coordinate of $\xi(1)$. Suppose that the distribution of this random variable places most of its mass on the interval $[1-\epsilon, 1+\epsilon]$.[13] This allows us to 'linearize' the output of each of the hidden units around the output corresponding to unperturbed inputs as

$$\begin{aligned} h(w_k^\top(X \odot \xi(1))) &= h(w_k^\top(X \odot (\xi(1) - \mathbf{1})) + w_k^\top X) \\ &\approx h(w_k^\top X) + \Big(\dot{h}(w_k^\top X) \cdot (w_k \odot X)^\top (\xi(1) - \mathbf{1})\Big). \end{aligned}$$

In the notation above, $\mathbf{1}$ denotes the d-dimensional vector of ones. For the sake of exposition, ignore the approximation error in the linearization above. If we fix $(X, Y, \xi(2))$, then the worst-case choice for the distribution of $\xi(1)$, denoted by $\mathbb{Q}(1)$, maximizes

$$\mathbb{E}_{\mathbb{Q}(1)}\left[\left(\sum_{k=1}^{K} \beta_k \cdot \xi(2)_k \cdot \left[h(w_k^\top X) + \left(\dot{h}(w_k^\top X) \cdot \sum_{j=1}^{d} w_{k,j} \cdot X_j \cdot (\xi(1)_j - 1)\right)\right]\right)^2\right]$$

among all distributions with independent marginals for which $\mathbb{E}_{\mathbb{Q}(1)}[\xi(1)_j] = 1$ for all $j = 1, \dots, d$. It can then be shown that such a maximization problem is equivalent to maximizing

$$\mathbb{E}_{\mathbb{Q}(1)}\left[\left(\sum_{k=1}^{K} \beta_k \cdot \xi(2)_k \cdot \dot{h}(w_k^\top X) \cdot \left[\sum_{j=1}^{d} w_{k,j} \cdot X_j \cdot (\xi(1)_j - 1)\right]\right)^2\right], \tag{35}$$

which, in turn, can be written as

$$\mathbb{E}_{\mathbb{Q}(1)}\left[\Big(a^\top (\xi(1) - \mathbf{1})\Big)^2\right]$$

for an appropriate choice of a vector $a \in \mathbb{R}^d$ that depends only on $(\beta, \xi(2), h, \dot{h}, w, X)$. Lemma 2 in Appendix A.2 shows that the solution to this problem is dropout noise.

[13] This is compatible with dropout noise for which δ is very close to zero.

9. Concluding Remarks

This paper examines *dropout training*, an increasingly popular estimation method in machine learning. Dropout training is a fundamental part of modern machine learning techniques for training very deep networks (Goodfellow et al., 2016).

Our main result (Theorem 1) established a novel decision-theoretic foundation for the use of dropout training. We showed that this method, when applied to generalized linear models, can be viewed as the minimax solution to an adversarial two-player, zero-sum game between a statistician and nature. The framework used in this paper is known in the stochastic optimization literature (Shapiro et al., 2014) as a distributionally robust optimization (DRO) problem.

Our minimax result showed, by construction, that dropout training indeed provides out-of-sample performance guarantees for distributions that arise from multiplicative perturbations of the in-sample data. Our result thus justified, explicitly, the ability of dropout training to enhance a predictor's out-of-sample performance, which is one of the reasons often invoked to promote the dropout method.

In addition to our theoretical result, we also suggested a new strategy to select the dropout probability and a new stochastic optimization implementation of dropout training. For the latter, we borrowed ideas from the multi-level Monte Carlo literature—in particular from Blanchet et al. (2019a)—to suggest an unbiased dropout training routine that is easily parallelizable and that has a smaller computational cost than naïve dropout training methods when the number of features is large (Theorem 3). Crucially, we showed that under some regularity conditions, our estimator has finite variance (which means that there are also theoretical, and not just practical, gains from parallelization).

The connection between dropout training and the multiplicative errors-in-variables model established in this paper is novel. We think this connection is potentially interesting because multiplicative errors have found a range of applications in empirical work across various disciplines, from economics to epidemiology. For example, Alan et al. (2009) uses it to account for measurement error in consumption data when estimating the elasticity of intertemporal substitution via Euler equations. Pierce et al. (1992) and Lyles and Kupper (1997) use it to relate health outcomes to the exposure of a chemical toxicant that is observed with error. Moreover, due to privacy considerations, statistical agencies such as the U.S. Census Bureau sometimes mask data using multiplicative noise, as discussed by Kim and Winkler (2003) and Nayak et al. (2011). Examples of data sets that contain variables masked with multiplicative noise include the Commodity Flow Survey Data (2017) and the Survey of Business Owners (2012)—both from the U.S. Census Bureau—and the U.S. Energy Information Administration Residential Energy Consumption survey. Applications of dropout training in these contexts could be an interesting area for future work.

We also discussed the extent to which our theoretical results extended to Neural Networks (in particular, to the universal approximators in Hornik et al. 1989 consisting of a single-hidden layer and a squashing activation function). Our results showed that Theorem 1 can be used to establish the optimality of dropout training to estimate the parameters of the last hidden layer in general feed-forward neural networks, where the output layer takes the form of a generalized linear model. We hope that our analysis serves as a foundation to understand the benefits of dropout training in neural networks.

Acknowledgments

We would like to thank Matias Cattaneo, Max Farrell, Michael Leung, Ulrich Müller, Mark Peletier, Hashem Pesaran, Ashesh Rambachan, Roger Moon, Frank Schorfheide, Stefan Wager, and participants at the Statistics Seminar series at Columbia University for helpful comments and suggestions. José Blanchet acknowledges support from NSF grants 1820942, 1838576, 1915967, 2118199, 2229012, 2312204 and the Chinese Merchant Bank. Material in this paper is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-20-1-0397. Viet Anh Nguyen acknowledges the support from the CUHK's Improvement on Competitiveness in Hiring New Faculties Funding Scheme.

Appendix A.

This appendix collections all the proofs of main results, as well as additional theoretical and empirical results.

A.1 Probability Limit of the Dropout Training Estimator for β

Proposition 1 *Suppose that Assumptions 2 and 3 hold. Let the data $\{(x_i, y_i)\}_{i=1}^n$ consist of n i.i.d. draws from a distribution $\mathbb{P}^*$. Then for any sequence $\delta_n \to \delta \in [0, 1)$ as $n \to \infty$, $\widehat{\beta}(\delta_n)$ converges in probability to*

$$\beta^*(\delta) \equiv \arg\min_\beta \, \mathbb{E}_{P^*} \left[\mathbb{E}_\delta \left[-\ln f(Y|X \odot \xi, \beta, \phi) \right] \right], \tag{36}$$

where the minimizer in (36) is unique and does not depend on ϕ.

Proof The dropout estimator of β maximizes

$$Q_n(\beta) \equiv \frac{1}{n} \sum_{i=1}^n y_i(\beta^\top x_i) - \mathbb{E}_{\delta_n}[\Psi(\beta^\top (x_i \odot \xi))].$$

In a slight abuse of notation, let ξ_δ denote a realization of dropout noise parameterized by δ. Then it is possible to re-write the objective function as a weighted average of the functions

$$Q_{n,\xi_{\delta_n}}(\beta) \equiv \frac{1}{n} \sum_{i=1}^n y_i(\beta^\top x_i) - \Psi(\beta^\top (x_i \odot \xi_{\delta_n})).$$

It will be convenient to define the limiting objective function to be

$$Q(\beta) \equiv \mathbb{E}_{P^*}[Y(\beta^\top X)] - \mathbb{E}_{P^*}[\mathbb{E}_\delta[\Psi(\beta^\top (X \odot \xi))]],$$

which, by Assumptions 1 and 2, is finite and strictly concave. The population objective function is then the average (over dropout noise) of

$$Q_{\xi_\delta}(\beta) \equiv \mathbb{E}_{P^*}[Y(\beta^\top X)] - \mathbb{E}_{P^*}[\Psi(\beta^\top (X \odot \xi_\delta))].$$

It is straightforward to show that $\beta^*(\delta)$ in Equation (36) denotes the unique maximizer of $Q(\beta)$.

The proof of the proposition follows from standard arguments in the theory of extremum estimators. In particular, it suffices to verify the conditions of Theorem 2.7 in Newey and McFadden (1994).

Condition i) in Newey and McFadden (1994) requires $Q(\beta)$ to be uniquely maximized at $\beta^*(\delta)$. This holds because Assumptions 1 and 2 imply that $Q(\beta)$ is strictly concave.

Condition ii) in Newey and McFadden (1994) requires $\beta^*(\delta)$ to be an element in the interior of a strictly convex set, which holds because, in the GLM models under consideration, the parameter space is $\mathbb{R}^d$. Furthermore, $Q_n(\beta)$ is trivially concave by Assumption 1.

Condition iii) requires $Q_n(\beta)$ to converge in probability to $Q(\beta)$ for every β. For this purpose, it suffices to show that $Q_{n,\xi_{\delta_n}}(\beta)$ converges in probability to $Q_{\xi_\delta}(\beta)$ for each fixed β, and for a sequence ξ_{δ_n} and ξ_δ that have zeros and non-zeros in exactly the same entries. Assumptions 1 and 2 imply $\mathbb{E}_{P^*}[Y(\beta^\top X)] < \infty$ for all β. Thus, using the Law of Large Numbers for i.i.d. sequences,

$$\frac{1}{n}\sum_{i=1}^{n} Y_i(\beta^\top X_i) \xrightarrow{p} \mathbb{E}_{P^*}[Y(\beta^\top X)].$$

Finally, Assumptions 1 and 2 imply that the triangular array

$$Z_{n,i} = \Psi(\beta^\top (X_i \odot \xi_{\delta_n})), \quad 1 \leq i \leq n,$$

satisfies the conditions for the Law of Large Numbers for triangular arrays (Durrett, 2019, Theorem 2.2.11), and, consequently,

$$\frac{1}{n}\sum_{i=1}^{n} \Psi(\beta^\top (X_i \odot \xi_{\delta_n})) \xrightarrow{p} \mathbb{E}_{P^*}\left[\Psi(\beta^\top (X \odot \xi_\delta))\right].$$

This completes the proof. ■

A.2 Proof of Theorem 1

The proof of Theorem 1 relies on the following two preparatory results.

Lemma 1 (Extremal expectation of a univariate convex function) *For any $-\infty < a < b < +\infty$, let ζ be a random variable in $[a, b]$ with mean $\mu \in [a, b]$. For any convex, continous function $f : [a, b] \to \mathbb{R}$, the distribution of ζ that maximizes $\mathbb{E}[f(\zeta)]$ among all distributions over $[a, b]$ with a given mean $\mu \in [a, b]$ is a scaled and shifted Bernoulli distribution, i.e.,*

$$\zeta = \begin{cases} a & \textit{with probability } (b-\mu)/(b-a), \\ b & \textit{with probability } (\mu-a)/(b-a). \end{cases} \tag{37}$$

Proof Let Q^* denote the probability measure induced by the random variable in (37). By definition

$$\mathbb{E}_{Q^*}[f(\zeta)] = \frac{b-\mu}{b-a}f(a) + \frac{\mu-a}{b-a}f(b).$$

Suppose first that $\mu = a$. In this case, Jensen's inequality implies that for any other probability measure Q over $[a,b]$ with mean $\mu = a$,

$$\mathbb{E}_Q[f(\zeta)] \leq f(\mathbb{E}_Q[\zeta]) = f(a) = \mathbb{E}_{Q^*}[f(\zeta)].$$

An analogous result holds if $\mu = b$.

Consider then the case in which $\mu \in (a,b)$. For an arbitrary probability measure Q over $[a,b]$ with mean $\mu \in (a,b)$, we have

$$\int_{[a,b]} f(\zeta)\mathrm{d}Q = \int_{[a,b]} f\big(a\frac{b-\zeta}{b-a} + b\frac{\zeta-a}{b-a}\big)\mathrm{d}Q \leq \int_{[a,b]} \big(\frac{b-\zeta}{b-a}f(a) + \frac{\zeta-a}{b-a}f(b)\big)\mathrm{d}Q,$$

where the inequality follows from the convexity of f. By the linearity of the integral operator and the fact that $\int_{[a,b]} \zeta \mathrm{d}Q = \mu$, we find

$$\int_{[a,b]} f(\zeta)\mathrm{d}Q \leq \frac{b-\mu}{b-a}f(a) + \frac{\mu-a}{b-a}f(b).$$

Because the probability measure Q was chosen arbitrarily, this implies that the distribution of ζ in Equation (37) maximizes the expectation of $f(\zeta)$. ■

Lemma 2 *Fix a vector of tuning parameters $\delta \in (0,1)^d$. Let $\mathcal{Q}_j(\delta_j)$ be defined as in* (10). *Suppose that A is a convex and continuous function on $\mathbb{R}$. For any $\theta \in \mathbb{R}^d$, we have*

$$\sup\Big\{\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A(\theta^\top\xi)] : \mathbb{Q}_j \in \mathcal{Q}_j(\delta_j)\Big\} = \mathbb{E}_{\mathbb{Q}_1^\bigstar\otimes\ldots\otimes\mathbb{Q}_d^\bigstar}[A(\theta^\top\xi)],$$

where $\mathbb{Q}_j^\bigstar$ is a scaled Bernoulli distribution of the form $\mathbb{Q}_j^\bigstar = (1-\delta_j)^{-1} \times Bernoulli((1-\delta_j))$ for each $j = 1,\ldots,d$.

Proof First, note that $\mathbb{Q}_j^\bigstar \in \mathcal{Q}_j(\delta_j)$ for each j, and thus $\mathbb{Q}_1^\bigstar \otimes \ldots \otimes \mathbb{Q}_d^\bigstar$ is a feasible solution to the maximization problem. It suffices to show that, for any set of feasible measures $\mathbb{Q}_j \in \mathcal{Q}_j(\delta_j), j = 1,\ldots,d$, we have

$$\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A(\theta^\top\xi)] \leq \mathbb{E}_{\mathbb{Q}_1^\bigstar\otimes\ldots\otimes\mathbb{Q}_d^\bigstar}[A(\theta^\top\xi)].$$

Towards this end, pick any $k \in \{1, \dots, d\}$. By Fubini's theorem, we can write

$$\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)] = \mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_{k-1} \otimes \mathbb{Q}_{k+1} \otimes \dots \otimes \mathbb{Q}_d} \mathbb{E}_{\mathbb{Q}_k}[A(\theta^\top \xi)].$$

For any fixed value $(\xi_1, \dots, \xi_{k-1}, \xi_{k+1}, \dots, \xi_d)$ the function $\xi_k \mapsto A(\sum_{j \neq k} \theta_j \xi_j + \theta_k \xi_k)$ is convex in the variable ξ_k over the interval $[0, (1 - \delta_k)^{-1}]$. Thus by Lemma 1,

$$\mathbb{E}_{\mathbb{Q}_k}[A(\sum_{j \neq k} \theta_j \xi_j + \theta_k \xi_k)] \leq \mathbb{E}_{\mathbb{Q}_k^\star}[A(\sum_{j \neq k} \theta_j \xi_j + \theta_k \xi_k)] \quad \text{for any fixed } (\xi_1, \dots, \xi_{k-1}, \xi_{k+1}, \dots, \xi_d).$$

Thus by the monotonicity of the expectation operator,

$$\begin{aligned} \mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)] &\leq \mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_{k-1} \otimes \mathbb{Q}_{k+1} \otimes \dots \otimes \mathbb{Q}_d} \mathbb{E}_{\mathbb{Q}_k^\star}[A(\theta^\top \xi)] \\ &= \mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_{k-1} \otimes \mathbb{Q}_k^\star \otimes \mathbb{Q}_{k+1} \otimes \dots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)]. \end{aligned}$$

By cycling through all possible values of $k \in \{1, \dots, d\}$ we conclude that

$$\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[A(\theta^\top \xi)] \leq \mathbb{E}_{\mathbb{Q}_1^\star \otimes \dots \otimes \mathbb{Q}_d^\star}[A(\theta^\top \xi)].$$

Therefore, the postulated claim holds. ■

We are now ready to prove Theorem 1.

Proof Note that for $\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)$, Assumption **??** implies $\mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)]$ is finite for any $\theta \in \Theta$ and any scalar $\delta \in [0, 1)$. Therefore, from Fubini's theorem and the definition of the loss function:

$$\begin{aligned} \mathbb{E}_{\mathbb{Q}}[\ell(X \odot \xi, Y, \theta)] &= \mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[\ell(X \odot \xi, Y, \theta)]\right] \\ &= \mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[-\ln h(Y, \phi) + (\Psi(\beta^\top (X \odot \xi)) - Y(\beta^\top (X \odot \xi)))/a(\phi)]\right] \\ &= -\mathbb{E}_{\mathbb{Q}_0}\left[\ln h(Y, \phi)\right] \\ &\quad + \mathbb{E}_{\mathbb{Q}_0}\left[\mathbb{E}_{\mathbb{Q}_1 \otimes \dots \otimes \mathbb{Q}_d}[(\Psi(\beta^\top (X \odot \xi)) - Y(\beta^\top (X \odot \xi)))/a(\phi)]\right]. \end{aligned}$$

It can then be shown that, for any β, X and ξ:

$$\beta^\top (X \odot \xi) = (\beta \odot X)^\top \xi.$$

Thus, we can fix the values of (X, Y, θ) and define the function

$$A_{(X,Y,\theta)}((\beta \odot X)^\top \xi) \equiv (\Psi(\beta^\top (X \odot \xi)) - Y \beta^\top (X \odot \xi))/a(\phi).$$

Note that $A_{(X,Y,\theta)}$ satisfies the condition of Lemma 2. Therefore

$$\sup\left\{\mathbb{E}_{\mathbb{Q}_1\otimes\ldots\otimes\mathbb{Q}_d}[A_{(X,Y,\theta)}((\beta\odot X)^\top\xi)] : \mathbb{Q}_j\in\mathcal{Q}_j(\delta_j)\right\} = \mathbb{E}_{\mathbb{Q}_1^\star\otimes\ldots\otimes\mathbb{Q}_d^\star}[A_{(X,Y,\theta)}((\beta\odot X)^\top\xi)],$$

for any (X,Y,θ), which completes the proof. ■

A.3 Proof of Theorem 2

Proof We write $\sqrt{n}(\mathcal{L}_n(\widehat{\beta}(\delta_n),\widehat{\phi}_n,\delta_n)-\mathcal{L}(\beta^*,\phi^*))$ as the sum of the following three terms

$$\sqrt{n}\left(\mathcal{L}_n(\widehat{\beta}(\delta_n),\widehat{\phi}_n,\delta_n)-\mathcal{L}_n(\beta^*,\widehat{\phi}_n,\delta_n)\right), \tag{38a}$$

$$\sqrt{n}\left(\mathcal{L}_n(\beta^*,\widehat{\phi}_n,\delta_n)-\mathcal{L}_n(\beta^*,\widehat{\phi}_n,0)\right), \tag{38b}$$

$$\sqrt{n}\left(\mathcal{L}_n(\beta^*,\widehat{\phi}_n,0)-\mathcal{L}(\beta^*,\phi^*,0)\right). \tag{38c}$$

By Assumption 4, the last term converges in distribution to a normal random variable, so we only need to analyze terms (38a) and (38b). We establish the proof following the four steps outlined in the main body of the paper.

Step 1: The same arguments as in Theorem 3.1 in Newey and McFadden (1994) we can show that for any sequence $\delta_n = c/\sqrt{n}$

$$\sqrt{n}(\widehat{\beta}(\delta_n)-\beta^*)\xrightarrow{d}\Sigma(\beta^*)^{-1}\mathcal{N}_d(-c\tilde{\mu}, a(\phi^*)\Sigma(\beta^*)),$$

where

$$\Sigma(\beta)\equiv\mathbb{E}_{P^*}[\ddot{\Psi}(X^\top\beta)XX^\top],$$

and

$$\tilde{\mu}\equiv\left(\sum_{\xi\in\mathcal{A}}\mathbb{E}_{P^*}[\dot{\Psi}((X\odot\xi)^\top\beta^*)(X\odot\xi)]\right)-(d-1)\mathbb{E}_{P^*}[YX]+\Sigma(\beta^*)\beta^*.$$

The set $\mathcal{A}$ above is defined as $\{\xi\in\{0,1\}^d : \text{exactly one entry of } \xi \text{ is zero}\}$. The argument is essentially the same as in every proof of asymptotic normality for extremum (or M-estimators), with the only difference being that, because of the dropout noise, the score term is asymptotically normal with a nonzero mean. In fact,

$$\nabla_\beta\mathcal{L}_n(\beta,\widehat{\phi}_n,\delta_n)\equiv\nabla_\beta\mathcal{L}_n(\beta,\widehat{\phi}_n,0)+\frac{1}{a(\widehat{\phi}_n)}\left(\frac{1}{n}\sum_{i=1}^n\mathbb{E}_{\delta_n}[(X_i\odot\xi)\dot{\Psi}(\beta^\top(X_i\odot\xi))]-X_i\dot{\Psi}(X_i^\top\beta)\right),$$

where

$$\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}_n, 0) \equiv -\frac{1}{a(\widehat{\phi}_n)} \frac{1}{n} \sum_{i=1}^{n} X_i(Y_i - \dot{\Psi}(X_i^\top \beta)).$$

Recognizing the term $\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}_n, 0)$ as the negative of the score function in the GLM model and doing some algebra, it is possible to show that $\nabla_\beta \mathcal{L}_n(\beta, \widehat{\phi}_n, \delta_n)$ is $o_p(1)$. Therefore (38a) is $o_p(1)$.

Step 2: For the term in (38b), note first that it is nonnegative. Also: $\mathcal{L}_n(\beta^*, \widehat{\phi}_n, \delta_n) - \mathcal{L}_n(\beta^*, \widehat{\phi}_n, 0)$ equals

$$\frac{1}{a(\widehat{\phi}_n)} \left(\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \mathbb{E}_{\delta_n}[\Psi((X_i \odot \xi)^\top \beta^*)] - \Psi(X_i^\top \beta^*) \right).$$

The term in parenthesis has a finite mean equal to

$$\Delta_n \equiv \mathbb{E}_{P^*}\mathbb{E}_{\delta_n}[\Psi((X \odot \xi)^\top \beta^*)] - \mathbb{E}_{P^*}[\Psi(X^\top \beta^*)]. \tag{39}$$

It can be shown—by verifying the conditions for the Law of Large Numbers for triangular arrays (Theorem 2.2.11 in Durrett 2019)—that

$$\sqrt{n}(\mathcal{L}_n(\beta^*, \delta_n) - \mathcal{L}_n(\beta^*, 0) - a(\widehat{\phi}_n)^{-1}\Delta_n) \xrightarrow{p} 0.$$

Moreover, Assumptions 1 and 2 imply $\sqrt{n}\Delta_n \xrightarrow{p} \Delta$, where

$$\Delta \equiv c \left(\sum_{\xi \in \mathcal{A}} \mathbb{E}_{P^*}[\Psi((X \odot \xi)^\top \beta^*)] - d\mathbb{E}_{P^*}[\Psi(X^\top \beta^*)] + \mathbb{E}_{P^*}[\dot{\Psi}(X^\top \beta^*) X^\top \beta^*] \right).$$

Step 3: Since, by Assumption, the term in (34c) is asymptotically normal, then we have shown that

$$\sqrt{n}(\mathcal{L}_n(\widehat{\beta}(\delta_n), \widehat{\phi}_n, \delta_n) - \mathcal{L}(\beta^*, \phi^*)) \xrightarrow{d} \mathcal{N}(\Delta / a(\phi^*), \sigma^2).$$

Step 4: Finally, we show that

$$n\left(\mathcal{L}((\widehat{\beta}(\delta_n), \widehat{\phi}_n)) - \mathcal{L}(\beta^*, \phi^*)\right) = O_{P^*}(1).$$

We have defined

$$\mathcal{L}_n(\beta, \phi, \delta) \equiv \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_\delta \left[-\ln f(y_i | x_i \odot \xi_i, \beta, \phi) \right],$$

and

$$\mathcal{L}(\beta, \phi) \equiv \mathbb{E}_{P^\star} \left[-\ln f(Y | X, \beta, \phi) \right].$$

Let $(\beta^\star, \phi^\star)$ be a stationary point of $\mathcal{L}$ in the interior of the parameter space, so that

$$\frac{\partial}{\partial \beta}\mathcal{L}(\beta^\star, \phi^\star) = 0, \quad \frac{\partial}{\partial \phi}\mathcal{L}(\beta^\star, \phi^\star) = 0.$$

Further, let $\widehat{\phi}_n$ denote an arbitrary $\sqrt{n}$-consistent, asymptotically normal estimator of $\phi^\star$. Under our assumptions, $\mathcal{L}$ continuously differentiable. Then, by Taylor expansion,

$$\mathcal{L}(\widehat{\beta}(\delta_n), \widehat{\phi}_n) - \mathcal{L}(\beta^\star, \phi^\star) = \frac{\partial}{\partial \beta}\mathcal{L}(\tilde{\beta}, \tilde{\phi})(\widehat{\beta}(\delta_n) - \beta^\star) + \frac{\partial}{\partial \phi}\mathcal{L}(\tilde{\beta}, \tilde{\phi})(\widehat{\phi}_n - \phi^\star),$$

where $(\tilde{\beta}, \tilde{\phi})$ lies on the line between $(\widehat{\beta}(\delta_n), \widehat{\phi}_n)$ and $(\beta^\star, \phi^\star)$. We have shown that

$$\sqrt{n}(\widehat{\beta}(\delta_n) - \beta^\star) \xrightarrow{d} \Sigma(\beta^*)^{-1}\mathcal{N}_d(-c\tilde{\mu}, a(\phi^*)\Sigma(\beta^*)).$$

Also we have that

$$\frac{\partial}{\partial \beta}\mathcal{L}(\tilde{\beta}, \tilde{\phi}) \to 0, \quad \frac{\partial}{\partial \phi}\mathcal{L}(\tilde{\beta}, \tilde{\phi}) \to 0$$

in probability, since $(\tilde{\beta}, \tilde{\phi}) \to (\beta^\star, \phi^\star)$ in probability and $(\beta^\star, \phi^\star)$ is a stationary point of $\mathcal{L}$. Thus

$$\sqrt{n}(\mathcal{L}(\widehat{\beta}(\delta_n), \widehat{\phi}_n) - \mathcal{L}(\beta^\star, \phi^\star)) \to 0,$$

in probability by Slutsky's theorem. Moreover, because $\mathcal{L}$ is twice continuously differentiable, then

$$n(\mathcal{L}(\widehat{\beta}(\delta_n), \widehat{\phi}) - \mathcal{L}(\beta^\star, \phi^\star)) = O_p(1).$$

■

A.4 Proof of Theorem 3

Proof By definition

$$Z(K_l^*) = \frac{\bar{\Delta}_{K_l^*}}{r(1-r)^{m_l^*}} + \theta_{l,m_0},$$

where K_l^* is a discrete random variable with probability mass function:

$$p(K_l^*) = r(1-r)^{K_l^* - m_0},$$

and supported on the integers larger than m_0.

We first show that the estimator $Z(K_l^*)$ is unbiased (as we average over both K_l^* and ξ_i^k). Algebra shows that

$$\begin{aligned}
\mathbb{E}[Z(K_l^*)] &= \sum_{K=m_0}^{\infty} \mathbb{E}[Z(K_l^*)|K_l^* = K]p(K) \\
&= \sum_{K=m_0}^{\infty} \mathbb{E}\left[\frac{\bar{\Delta}_{K_l^*}}{p(K_l^*)} + \theta_{l,m_0}\middle| K_l^* = K\right]p(K) \\
&= \sum_{K=m_0}^{\infty} \mathbb{E}\left[\frac{\bar{\Delta}_{K}}{p(K)} + \theta_{l,m_0}\middle| K_l^* = K\right]p(K) \\
&= \left(\sum_{K=m_0}^{\infty} \mathbb{E}\left[\widehat{\theta}_n^{\bigstar}(2^{K+1}) - \frac{1}{2}(\widehat{\theta}_n^{O}(2^{K}) + \widehat{\theta}_n^{E}(2^{K}))\right]\right) + \mathbb{E}[\theta_{l,m_0}] \\
&= -\frac{1}{2}\left(\mathbb{E}[\widehat{\theta}_n^{O}(2^{m_0})] + \mathbb{E}[\widehat{\theta}_n^{E}(2^{m_0})]\right) + \mathbb{E}[\theta_{l,m_0}] + \lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^{\bigstar}(2^{K+1})].
\end{aligned}$$

The expectations in the last line are all finite because Θ is compact. In addition, since the draws are i.i.d. and θ_{l,m_0} is the solution to the problem (32) when 2^{m_0} draws are used we have

$$-\frac{1}{2}\left(\mathbb{E}[\widehat{\theta}_n^{O}(2^{m}_0)] + \mathbb{E}[\widehat{\theta}_n^{E}(2^{m}_0)]\right) + \mathbb{E}[\theta_{l,m_0}] = 0.$$

Moreover, the sequence of random variables $\{\widehat{\theta}_n^{\bigstar}(2^{K+1})\}$ is uniformly integrable, because Θ is a compact subset of a finite-dimensional Euclidean space. Finally, we know that

$$\widehat{\theta}_n^{\bigstar}(2^{K+1}) \xrightarrow{p} \theta_n^{\bigstar}$$

as $K \to \infty$. The uniform integrability of the sequence of estimators then implies

$$\lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^{\bigstar}(2^{K+1})] = \mathbb{E}\left[\lim_{K\to\infty} \widehat{\theta}_n^{\bigstar}(2^{K+1})\right] = \theta_n^{\bigstar},$$

see Theorem 6.2 in DasGupta (2008). We conclude that

$$\mathbb{E}[Z(K_l^*)] = \lim_{K\to\infty} \mathbb{E}[\widehat{\theta}_n^{\bigstar}(2^{K+1})] = \theta_n^{\bigstar}.$$

Now we show that the expected computational cost of $Z(K_l^*)$ is finite. In order to compute $Z(K)$ for a given K we need $n{\cdot}2^{K+1}$ random draws. Thus, the expected computational cost of $Z(K_l^*)$ is

$$\begin{aligned}\sum_{K=m_0}^{\infty} n2^{K+1}r(1-r)^{K-m_0} &= n\cdot(2^{m_0+1})\cdot r\sum_{K=m_0}^{\infty}2^{K-m_0}(1-r)^{K-m_0}\\ &= n\cdot(2^{m_0+1})\cdot r\sum_{K=m_0}^{\infty}(2(1-r))^{K-m_0}.\end{aligned}$$

The term above converges to

$$\frac{n\cdot(2^{m_0+1})\cdot r}{1-2(1-r)}=\frac{n\cdot(2^{m_0+1})\cdot r}{2r-1},$$

provided that $2(1-r)<1$, which holds because we have chosen $r>1/2$.

For the proof of finite variance, we intend to show that

$$\mathbb{E}\left[\bar{\Delta}_K^\top\bar{\Delta}_K\right]=O(2^{-2K}) \tag{40}$$

as $K\to\infty$. Equation (40) guarantees that every processor generates an estimator $Z(K_l^*)$ with finite variance. Since K_l^* is a discrete random variable with probability mass function

$$p(K_l^*)=r(1-r)^{K^*-m_0},$$

and

$$\begin{aligned}\mathbb{E}[Z(K_l^*)^\top Z(K_l^*)] &= \sum_{K=m_0}^{\infty}\mathbb{E}\left[Z(K_l^*)^\top Z(K_l^*)|K_l^*=K\right]p(K)\\ &= \sum_{K=m_0}^{\infty}\mathbb{E}\left[\left(\frac{\bar{\Delta}_K}{p(K)}+\theta_{l,m_0}\right)^\top\left(\frac{\bar{\Delta}_K}{p(K)}+\theta_{l,m_0}\right)\right]p(K)\\ &\leq 2\left(\sum_{K=m_0}^{\infty}\mathbb{E}\left[\frac{\bar{\Delta}_K^\top\bar{\Delta}_K}{p(K)^2}\right]p(K)+\sum_{K=m_0}^{\infty}\mathbb{E}\left[\theta_{l,m_0}^\top\theta_{l,m_0}\right]p(K)\right)\\ &\leq C\left(\sum_{K=m_0}^{\infty}\frac{2^{-2K}}{p(K)}+\sup_{\theta\in\Theta}\|\theta\|_2^2p(K)\right)\\ &\leq C\left(\sum_{K=m_0}^{\infty}\frac{1}{2^{2m_0}2^{2(K-m_0)}p(K)}+\sup_{\theta\in\Theta}\|\theta\|_2^2p(K)\right)\\ &\leq C_1\left(\sum_{K=m_0}^{\infty}\frac{1}{r4^{m_0}}\frac{1}{(4(1-r))^{K-m_0}}\right)+C_2.\end{aligned}$$

The geometric sum in the last expression is finite because we have assumed that $r < \frac{3}{4}$.

To show Equation (40), we do a Taylor expansion of the first-order conditions of the problem (32) around $\theta_n^\bigstar$. The Karush–Kuhn–Tucker optimality condition for the level-2^K solution $\widehat{\theta}_n^\bigstar(2^K)$ of the problem in expression (32) implies

$$0 = \sum_{i=1}^{n} \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_\theta \ell(x_i \odot \xi_i^k, y_i, \widehat{\theta}_n^\bigstar(2^K)) \right].$$

It follows by Taylor expansion and Assumption 4 that

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_\theta \ell(x_i \odot \xi_i^k, y_i, \theta_n^\bigstar) \right] + \sum_{i=1}^{n} \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_{\theta\theta} \ell(x_i \odot \xi_i^k, y_i, \theta_n^\bigstar) \right] \left(\widehat{\theta}_n^\bigstar(2^K) - \theta_n^\bigstar \right) \\ &\quad + R_{K,\theta} \\ &= \sum_{i=1}^{n} \left[\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_\theta \ell(x_i \odot \xi_i^k, y_i, \theta_n^\bigstar) \right] \\ &\quad + \sum_{i=1}^{n} \nabla_{\theta\theta} \mathbb{E}_{\mathbb{Q}^\bigstar} \left[\ell(X \odot \xi, Y, \theta_n^\bigstar) | X = x_i, Y = y_i \right]] \left(\widehat{\theta}_n^\bigstar(2^K) - \theta_n^\bigstar \right) \\ &\quad + R_K + R_{K,\theta}, \end{aligned} \tag{41}$$

where

$$\begin{aligned} R_K \equiv & \left(\sum_{i=1}^{n} \left(\frac{1}{2^K} \sum_{k=1}^{2^K} \nabla_{\theta\theta} \ell(x_i \odot \xi_i^k, y_i, \theta_n^\bigstar) - \nabla_{\theta\theta} \mathbb{E}_{\mathbb{Q}^\bigstar} \left[\ell(X \odot \xi, Y, \theta_n^\bigstar) | X = x_i, Y = y_i \right] \right) \right) \\ & \cdot \left(\widehat{\theta}_n^\bigstar(2^K) - \theta_n^\bigstar \right), \end{aligned}$$

and

$$\| R_{K,\theta} \|_2 \leq \sum_{i=1}^{n} \sup_{\theta \in \Theta, \xi} \| \nabla_{\theta\theta\theta} \ell(x_i \odot \xi, y_i, \theta) \|_2 \left\| \widehat{\theta}_n^\bigstar(2^K) - \theta_n^\bigstar \right\|_2^2 \leq C_3 \left\| \widehat{\theta}_n^\bigstar(2^K) - \theta_n^\bigstar \right\|_2^2$$

by Assumption 4. Thus, by Assumption 3, we have

$$\mathbb{E}[R_{K,\theta}^\top R_{K,\theta}] = O(2^{-2K})$$

as $K \to \infty$. Moreover, by the multivariate version of Theorem 2 in Bahr (1965) which follows from the Cramér–Wold theorem, we have that

$$\mathbb{E}\left[\left\|\sum_{i=1}^{n}\left(\frac{1}{2^K}\sum_{k=1}^{2^K}\nabla_{\theta\theta}\ell(x_i\odot\xi_i^k, y_i, \theta_n^\bigstar) - \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}\left[\ell(X\odot\xi, Y, \theta_n^\bigstar)|X=x_i, Y=y_i\right]\right)\right\|_2^4\right]$$

is $O(2^{-2K})$.

We can express $R_K^\top R_K$ as $\|R_K\|^2$. The Cauchy–Schwarz inequality implies

$$\begin{aligned}
&\mathbb{E}[R_K^\top R_K]\\
\leq&\mathbb{E}\left[\left\|\sum_{i=1}^{n}\left(\frac{1}{2^K}\sum_{k=1}^{2^K}\nabla_{\theta\theta}\ell(x_i\odot\xi_i^k, y_i, \theta_n^\bigstar) - \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}\left[\ell(X\odot\xi, Y, \theta_n^\bigstar)|X=x_i, Y=y_i\right]\right)\right\|_2^2\times\right.\\
&\left.\left\|\widehat{\theta}_n^\bigstar(2^K)-\theta_n^\bigstar\right\|_2^2\right].
\end{aligned}$$

By Hölder's inequality we have

$$\begin{aligned}
&\mathbb{E}[R_K^\top R_K]\\
\leq&\mathbb{E}\left[\left\|\sum_{i=1}^{n}\left(\frac{1}{2^K}\sum_{k=1}^{2^K}\nabla_{\theta\theta}\ell(x_i\odot\xi_i^k, y_i, \theta_n^\bigstar) - \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}\left[\ell(X\odot\xi, Y, \theta_n^\bigstar)|X=x_i, Y=y_i\right]\right)\right\|_2^4\right]^{\frac{1}{2}}\times\\
&\qquad\mathbb{E}\left[\left\|\widehat{\theta}_n^\bigstar(2^K)-\theta_n^\bigstar\right\|_2^4\right]^{\frac{1}{2}}\\
\leq&O(2^{-2K}).
\end{aligned}$$

Finally, consider the solutions $\widehat{\theta}_n^\bigstar(2^{K_l^*+1}), \widehat{\theta}_n^O(2^{K_l^*}), \widehat{\theta}_n^E(2^{K_l^*})$ conditional on $K_l^* = K$. Denote the remainder terms in Equation (41) corresponding to the level 2^{K+1} solution $\widehat{\theta}_n^\bigstar(2^{K+1})$ as $R_{K+1}^\bigstar, R_{K+1,\theta}^\bigstar$. Similarly, denote the remainder terms in Equation (41) corresponding to the level-2^K solution $\widehat{\theta}_n^O(2^K)$ (and, respectively, $\widehat{\theta}_n^E(2^K)$) as $R_K^O, R_{K,\theta}^O$ ($R_K^E, R_{K,\theta}^E$). By the construction of $\widehat{\theta}_n^O(2^K), \widehat{\theta}_n^E(2^K)$ using odd and even indices, we have, from Equation (41)

$$\begin{aligned}
&-\sum_{i=1}^{n}\nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}[\ell(X\odot\xi, Y, \theta_n^\bigstar)|X=x_i, Y=y_i]\left(\widehat{\theta}_n^\bigstar(2^{K+1})-\frac{1}{2}(\widehat{\theta}_n^O(2^K)+\widehat{\theta}_n^E(2^K))\right)\\
=&R_{K+1}^\bigstar-\frac{1}{2}(R_K^O+R_K^E)+R_{K+1,\theta}^\bigstar-\frac{1}{2}(R_{K,\theta}^O+R_{K,\theta}^E).
\end{aligned}$$

By Assumption 4,

$$\sum_{i=1}^{n} \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}[\ell(X \odot \xi, Y, \theta_n^\bigstar)|X = x_i, Y = y_i] = n \cdot \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}[\ell(X \odot \xi, Y, \theta_n^\bigstar)]$$

is invertible. Thus, we have shown that

$$\begin{aligned}\bar{\Delta}_K &\equiv \widehat{\theta}_n^\bigstar(2^{K+1}) - \frac{1}{2}(\widehat{\theta}_n^O(2^K) + \widehat{\theta}_n^E(2^K)) \\ &= \Big(n \cdot \nabla_{\theta\theta}\mathbb{E}_{\mathbb{Q}^\bigstar}[\ell(X \odot \xi, Y, \theta_n^\bigstar)]\Big)^{-1} \Big(R_{K+1}^\bigstar - \frac{1}{2}(R_K^O + R_K^E) + R_{K+1,\theta}^\bigstar - \frac{1}{2}(R_{K,\theta}^O + R_{K,\theta}^E)\Big).\end{aligned}$$

Since each of the terms on the right-hand side has been shown to be $O(2^{-2K})$, we conclude that $\mathbb{E}[\bar{\Delta}_K^\top \bar{\Delta}_K] = O(2^{-2K})$. ■

A.5 Dropout Training in Linear Regression

Here, diag(M) denotes the diagonal matrix formed by the diagonal elements of M.

Corollary 2 (Linear regression with $\phi = 1$) *For linear regression with $\ell(x, y, \beta) = (\beta^\top x - y)^2$, we have*

$$\min_{\beta\in\mathbb{R}^d} \max_{\mathbb{Q}\in\mathcal{U}(\widehat{\mathbb{P}}_n,\delta)} \mathbb{E}_{\mathbb{Q}}\Big[\big(\beta^\top(X \odot \xi) - Y\big)^2\Big] = \min_{\beta\in\mathbb{R}^d} \mathbb{E}_{\mathbb{Q}^\bigstar}\Big[\big(\beta^\top(X \odot \xi) - Y\big)^2\Big],$$

where $\mathbb{Q}^\bigstar = \widehat{\mathbb{P}}_n \otimes \mathbb{Q}_1^\bigstar \otimes \ldots \otimes \mathbb{Q}_d^\bigstar$ and $\mathbb{Q}_j^\bigstar = (1-\delta)^{-1} \times$ Bernoulli$(1-\delta)$ for each $j = 1, \ldots, d$. Moreover,

$$\min_{\beta\in\mathbb{R}^d} \mathbb{E}_{\mathbb{Q}^\bigstar}\Big[\big(\beta^\top(X \odot \xi) - Y\big)^2\Big] = \min_{\beta\in\mathbb{R}^d} \frac{1}{n}\Big[(\mathbf{Y} - \mathbf{X}\beta)^\top(\mathbf{Y} - \mathbf{X}\beta) + \frac{\delta}{1-\delta}\beta^\top \mathbf{\Lambda}\beta\Big], \tag{42}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\mathbf{X}^\top\mathbf{X})$ which implies that the dropout training estimator equals

$$\widehat{\beta}(\delta) = \Big(\mathbf{X}^\top\mathbf{X} + \frac{\delta}{1-\delta}\,\mathrm{diag}(\mathbf{X}^\top\mathbf{X})\Big)^{-1}\mathbf{X}^\top\mathbf{Y}.$$

Finally, if $\mathbb{E}_{P^*}[XX^\top]$ is a diagonal matrix with strictly positive entries then

$$\widehat{\beta}(\delta) \xrightarrow{p} (1-\delta)\mathbb{E}_{P^*}[XX^\top]^{-1}\mathbb{E}_{P^*}[XY].$$

Proof The first part of the corollary follows directly from (12) and (13) in our main theorem. The second part of the corollary follows from Proposition 1. According to this proposition, the limit of $\widehat{\beta}(\delta)$ is

$$\beta^*(\delta) = \Big(\mathbb{E}_{P^*}[XX^\top] + (\delta/1-\delta)\,\mathrm{diag}(\mathbb{E}_{P^*}[XX^\top])\Big)^{-1}\mathbb{E}_{P^*}[YX]. \tag{43}$$

Thus, if $\mathbb{E}_{P^*}[XX^\top]$ is a diagonal matrix, we obtained the desired limit.

One interesting implication of Equation (43) is that $\beta^*(0)$—which gives the probability limit of the maximum likelihood estimator—generally differs from $\beta^*(\delta)$ when $\delta \neq 0$, which is the probability limit of the dropout estimator.[14] In particular, the estimator in Equation (43) differs from the best linear predictor of y using x as long as $E_{P^*}[YX] \neq 0$ and $\delta \neq 0$. This estimator differs from Ridge regression in that $\mathrm{diag}(\mathbb{E}_{P^*}[XX^\top])$ replaces the identity matrix (and this simple adjustment makes the estimator scale equivariant). ■

A.6 Additional Theoretical Results

A.6.1 Convexity of f is necessary for the optimality of dropout noise

Theorem 4 *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a function such that $f(y,\cdot) : \mathbb{R} \to \mathbb{R}$ admits bounded and continuous second-order derivatives for any given $y \in \mathbb{R}$. If, for some $\delta \in (0,1]$, all $d \geq 1$, all $\beta \in \mathbb{R}^d$ and all reference distributions $\mathbb{Q}_0$ (where $(X,Y) \sim \mathbb{Q}_0$), it holds that*

$$\sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}}[f(Y,\beta^\top X \odot \xi)] = \mathbb{E}_{\mathbb{Q}^\star}[f(Y,\beta^\top X \odot \xi)], \tag{44}$$

then $f(y,\cdot)$ is a convex function on the real line for any given y.

Proof Since (44) holds for any reference distribution $\mathbb{Q}_0$, we consider the case where $\mathbb{Q}_0$ is given by the Dirac measure on the point (x,y), where $x \in \mathbb{R}^d, y \in \mathbb{R}$. Since dropout is the solution to problem (44), we have that the dropout noise on the last coordinate, viz., $\mathbb{Q}_d^\star = (1-\delta)^{-1} \times Bernoulli(1-\delta)$, solves

$$\max_{\mathbb{Q}_d\in\mathcal{Q}_d(\delta)} \mathbb{E}_{\mathbb{Q}_d}\left[\mathbb{E}_{\mathbb{Q}_1^\star\otimes\cdots\mathbb{Q}_{d-1}^\star}\left[f(y,\beta^\top x\odot\xi)\right]\right].$$

Define

$$\phi(\xi_d) = \mathbb{E}_{\mathbb{Q}_1^\star\otimes\cdots\mathbb{Q}_{d-1}^\star}\left[f(y,\sum_{j=1}^{d-1}\beta_j x_j\xi_j + \beta_d x_d\xi_d)\right],$$

[14] Relatedly, Farrell et al. (2021)—who study deep neural networks and their use in semiparametric inference—report that their numerical exploration of dropout increased bias and interval length compared to nonregularized models.

where ϕ is implicitly indexed by β, x and y. By duality in semi-infinite linear programming (see Theorem 1 and Equation 4 in Isii 1962), there exists an affine function $L(\cdot)$ such that

$$\phi(\xi_d) \leq L(\xi_d) \; \forall \xi_d \in [0, \frac{1}{1-\delta}] \text{ with equality at } \xi_d = 0 \text{ or } \frac{1}{1-\delta}.$$

Let $\xi_d = (1-\alpha)\frac{1}{1-\delta}$ for $\alpha \in [0,1]$, we have that

$$\begin{aligned}\phi((1-\alpha)\frac{1}{1-\delta}) &\leq L((1-\alpha)\frac{1}{1-\delta}) \\ &= L(\alpha \cdot 0 + (1-\alpha)\frac{1}{1-\delta}) \\ &= \alpha L(0) + (1-\alpha)L(\frac{1}{1-\delta}) \\ &= \alpha\phi(0) + (1-\alpha)\phi(\frac{1}{1-\delta}).\end{aligned}$$

Since β, x, y are arbitrary, this is equivalent to saying that, for all $(z_1, \ldots, z_{d-1}), z, y$ and $\alpha \in (0,1)$,

$$\begin{aligned}&\sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) f(y, \sum_{j=1}^{d-1} z_j i_j + (1-\alpha)z) \\ &\leq \alpha \cdot \sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) f(y, \sum_{j=1}^{d-1} z_j i_j) \\ &\quad + (1-\alpha) \cdot \sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) f(y, \sum_{j=1}^{d-1} z_j i_j + z),\end{aligned}$$

where $p(i_1, \ldots, i_{d-1})$ is the probability mass function for independent Bernoulli trials $i_1, \ldots, i_{d-1}$. Rearranging the inequality, we have

$$\begin{aligned}&\sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) \left(f(y, \sum_{j=1}^{d-1} z_j i_j + (1-\alpha)z) - f(y, \sum_{j=1}^{d-1} z_j i_j) \right) \\ &\leq (1-\alpha) \left(\sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) \left(f(y, \sum_{j=1}^{d-1} z_j i_j + z) - f(y, \sum_{j=1}^{d-1} z_j i_j) \right) \right).\end{aligned}$$

By second-order Taylor expansion around $z = 0$, we have

$$f(y, \sum_{j=1}^{d-1} z_j i_j + (1-\alpha)z) - f(y, \sum_{j=1}^{d-1} z_j i_j) = f_x(y, \sum_{j=1}^{d-1} z_j i_j)(1-\alpha)z + O(1)(1-\alpha)^2 z^2,$$

and

$$f(y,\sum_{j=1}^{d-1} z_j i_j + z) - f(y,\sum_{j=1}^{d-1} z_j i_j) = f_x(y,\sum_{j=1}^{d-1} z_j i_j) z + \frac{1}{2}\left(f_{xx}(y,\sum_{j=1}^{d-1} z_j i_j) + o(1)\right) z^2,$$

where f_x, f_{xx} denote partial derivatives of f with respect to the second argument. Therefore, we have the former inequality becomes (by the cancellation of terms)

$$\begin{aligned}&\sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) O(1)(1-\alpha)\\ \leq &\sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1})\left(f_{xx}(y,\sum_{j=1}^{d-1} z_j i_j) + o(1)\right).\end{aligned}$$

Thus letting $\alpha \to 1$, and then letting $z \to 0$, we obtain

$$\sum_{(i_1,\ldots,i_{d-1})\in\{0,1\}^{d-1}} p(i_1,\ldots,i_{d-1}) f_{xx}(y,\sum_{j=1}^{d-1} z_j i_j) \geq 0,$$

for all $z_j \in \mathbb{R}, \forall j$.

Now we consider $z_1 = \cdots = z_{d-1} = c \in \mathbb{R}$ and denote $S_d = \sum_{j=1}^{d-1} i_j$. Thus S_d follows the binomial distribution $B(d-1,p)$ where $p = (1-\delta)$. We rewrite the previous inequality as

$$\mathbb{E}_{S_d \sim B(d-1,p)}[f_{xx}(y, cS_d)] \geq 0.$$

Let us consider $c = \frac{\tilde{c}}{(d-1)p}$, and rewrite

$$cS_d = \frac{\tilde{c}}{(d-1)p}(S_d - p(d-1)) + \tilde{c}.$$

By the standard Central Limit Theorem,

$$\frac{S_d - p(d-1)}{\sqrt{d-1}} \Rightarrow \mathcal{N}(0, p(1-p)),$$

where $\Rightarrow$ denotes convergence in law. Therefore, we have that cS_d converges to the Dirac measure at $\tilde{c}$ as $d \to \infty$. Since $f_{xx}(y,\cdot)$ is bounded and continuous, we conclude that $f_{xx}(y,\tilde{c}) \geq 0$ for arbitrary $\tilde{c} \in \mathbb{R}$. The convexity of $f(y,\cdot)$ then follows. ■

Corollary 3 *Consider the single index models with $f(y, \beta^\top x) = (y - g(\beta^\top x))^2$. Then*

$$\sup_{\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)} \mathbb{E}_{\mathbb{Q}}[(Y - g(\beta^\top X \odot \xi))^2] = \mathbb{E}_{\mathbb{Q}^\star}[(Y - g(\beta^\top X \odot \xi))^2]$$

for all $\delta \in (0, 1], d \geq 1$, all $\beta \in \mathbb{R}^d$ and all reference distribution $\mathbb{Q}_0$ (where $(X, Y) \sim \mathbb{Q}_0$) if and only if g is a linear function.

Proof The "if" part is given by Theorem 1 of the paper, as $f(y, u) = (y - g(u))^2$ is convex if g is linear. For the "only if" part, we know that

$$f_{xx}(y, u) = -2yg^{''}(u) + 2g(u)g^{''}(u) + 2(g^{'}(u))^2 \geq 0$$

Suppose that $g^{''}(u_0) \neq 0$ for some $u_0 \in \mathbb{R}$, then, by choosing a sufficiently large (in absolute value) y, we have $f_{xx}(y, u_0) < 0$, a contradiction. Thus $g^{''}(u) = 0$ for all u, which implies that g must be linear. ■

A.6.2 Convexity is not sufficient for the optimality of dropout noise

Consider a slight generalization of the set $\mathcal{U}$ used in Theorem 1 by defining

$$\mathcal{Q}_j(\delta) = \{\mathbb{Q}_j : \mathbb{Q}_j([0, (1-\delta)^{-1}]) = 1, \mathbb{E}_{\mathbb{Q}_j}[\xi_j] = 1, \mathbb{Q}_j([0, \epsilon]) = \delta\}$$

and

$$\mathcal{U}(\mathbb{Q}_0, \delta) = \{\mathbb{Q}_0 \otimes \mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d : \mathbb{Q}_j \in \mathcal{Q}_j(\delta)\}.$$

The only difference between $\mathcal{U}(\mathbb{Q}_0, \delta)$ and our previous construction is that we "force" the random variables $\mathbb{Q}_j$ to have positive mass on the interval $[0, \epsilon]$. Consider the problem

$$\sup_{\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)} \mathbb{E}_{\mathbb{Q}}[f\left(Y, \beta^\top (X \odot \xi)\right)]$$

where $f(y, \cdot) : \mathbb{R} \to \mathbb{R}$ is a convex function for every y. Just as we did in the proof of Theorem 1, we fix $\xi_2, \ldots, \xi_d$, and try to solve

$$\sup_{\mathbb{Q}_1 \in \mathcal{Q}_1(\delta)} \mathbb{E}_{\mathbb{Q}_0 \otimes \mathbb{Q}_1}\left[f\left(Y, \beta_1 X_1 \xi_1 + \sum_{j=2}^{d} \beta_j X_j \xi_j\right)\right].$$

We argue that the optimal solution $\mathbb{Q}_1$ in general depends on the values of $\xi_2, \ldots, \xi_d$ and, more importantly, $\mathbb{Q}_1$ may fail to include zero in its support. This last point implies that

$\mathbb{Q}_1$ cannot be interpreted as dropout noise (as there is no sense in which the first variable is ever dropped out). We make this point by showing a simple example.

Consider $\mathbb{Q}_0$ is the Dirac measure on (x, y) with $y > 0$ and consider $f(y, z) = (y - z_+)^2$ (i.e., square loss with the so-called "ReLu" activation function). Now, $f(y, \beta_1 x_1 \xi_1 + \sum_{j=2}^d \beta_j x_j \xi_j) = ((\beta_1 x_1 \xi_1 + \sum_{j=2}^d \beta_j x_j \xi_j)_+ - y)^2 = \beta_1^2 x_1^2 ((\xi_1 + \tilde{z})_+ - \tilde{y})^2$ where we assume $\beta_1 x_1 > 0$. Thus, the problem is equivalent to

$$\sup_{\mathbb{Q}_1 \in \mathcal{Q}_1(\delta)} [((\xi_1 + \tilde{z})_+ - \tilde{y})^2].$$

If $\tilde{z} \in (-0.99, -\epsilon)$, then by Isii (1962, Theorem 1), the problem has the dual

$$\inf_{a,b,c} a + b + c\delta \text{ s.t. } ((\xi_1 + \tilde{z})_+ - \tilde{y})^2 \leq a + b\xi_1 + c 1_{\{\xi_1 \in [0,\epsilon]\}} \ \forall \xi_1 \in \left[0, \frac{1}{1-\delta}\right]$$

and moreover the optimal solution $\mathbb{Q}_1^*$ is supported in the set

$$\left\{ \xi_1 \in \left[0, \frac{1}{1-\delta}\right] : ((\xi_1 + \tilde{z})_+ - \tilde{y})^2 = a^* + b^* \xi_1 + c^* 1_{\{\xi_1 \in [0,\epsilon]\}} \right\},$$

where a^*, b^*, c^* constitute the solution to the dual problem. We claim that 0 is not in the support of $\mathbb{Q}_1^*$. Otherwise, $\xi_1 = 0$ need to satisfy

$$((\xi_1 + \tilde{z})_+ - \tilde{y})^2 = a^* + b^* \xi_1 + c^* 1_{\{\xi_1 \in [0,\epsilon]\}}.$$

Since the left-hand side is constant for ξ_1 in a neighborhood of 0, we have that $b^* \geq 0$. Since $-0.99 < \tilde{z} < -\epsilon$, we have that, for $\xi_1 > 0.99$, $((\xi_1 + \tilde{z})_+ - \tilde{y})^2 < \tilde{y}^2$ while $a^* + b^* \xi_1 + c^* 1_{\{\xi_1 \in [0,\epsilon]\}} > \tilde{y}^2$. Hence, the support of $\mathbb{Q}_1^*$ is a subset of $[0, 0.99]$, contradicting the requirement that $\mathbb{Q}_1^*$ has mean 1.

We conclude that, while convexity is somewhat necessary for dropout to be optimal, it need not be sufficient.

A.6.3 Block Dropout

Finally, we generalize the notion of dropout so that the noise distributions $\mathbb{Q}_1, \ldots, \mathbb{Q}_d$ are arbitrarily correlated. Thus, we define

$$\mathcal{U}(\mathbb{Q}_0, \delta) = \{\mathbb{Q}_0 \otimes \mathbb{Q}_{1:d} : \mathbb{Q}_{1:d}([0, (1-\delta)^{-1}]^d) = 1, \mathbb{E}_{\mathbb{Q}_{1:d}}[\xi] = (1, \ldots, 1)^\top\}$$

and consider the problem

$$\sup_{\mathbb{Q} \in \mathcal{U}(\mathbb{Q}_0, \delta)} \mathbb{E}_{\mathbb{Q}}[f(Y, \beta^\top X \odot \xi)].$$

We give an example where this is a meaningful generalization of the notion of dropout. Consider $f(y,z) = (y-z)^2$ (i.e., square loss), also that $\mathbb{Q}_0$ is a Dirac measure on (x,y). Then the problem

$$\sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}_{1:d}}\left[\left(y-(\sum_{i=1}^{d} x_i\beta_i\xi_i)\right)^2\right]$$

is equivalent to

$$\sup_{\mathbb{Q}\in\mathcal{U}(\mathbb{Q}_0,\delta)} \mathbb{E}_{\mathbb{Q}_{1:d}}\left[\left(\sum_{i=1}^{d} x_i\beta_i\xi_i\right)^2\right].$$

By Theorem 1 in Isii (1962), the problem has dual

$$\inf_{a_0,\ldots,a_d}\sum_{i=0}^{d} a_i \quad \text{s.t.} \quad \left(\sum_{i=1}^{d} x_i\beta_i\xi_i\right)^2 \leq a_0+\sum_{i=1}^{d} a_i\xi_i \;\forall \xi\in[0,(1-\delta)^{-1}]^d$$

and, moreover, the optimal solution $\mathbb{Q}_{1:d}^*$ is supported in the set

$$\left\{\xi\in[0,(1-\delta)^{-1}]^d : \left(\sum_{i=1}^{d} x_i\beta_i\xi_i\right)^2 = a_0^*+\sum_{i=1}^{d} a_i^*\xi_i\right\}, \tag{45}$$

where $a_0^*,\ldots,a_d^*$ are the optimal dual variables. Suppose that $x_i\beta_i \neq 0$ for all i. Then, since $\left(\sum_{i=1}^{d} x_i\beta_i\xi_i\right)^2$ is convex quadratic, any ξ in the support points set (45) must be in the boundary of the cube $[0,(1-\delta)^{-1}]^d$. Moreover, since we have the mean constraint $\mathbb{E}_{\mathbb{Q}_{1:d}^*}[\xi] = (1,\ldots,1)^\top$, there exists ξ in (45) with blocks of components comprising zeros.

The above arguments extend to any $\mathbb{Q}_0$ and f as long as the function $\mathbb{E}_{\mathbb{Q}_0}[f(Y,\beta^\top X\odot\xi)]$ is a strictly convex function of ξ. Solving precisely the support of the noise requires solving for the dual variables a_i^*, which is a semi-infinite linear programming (finite-dimensional linear objective with infinitely many constraints). The numerical methods in the literature approximate the semi-infinite problem with a sequence of finite programming problems, which are then solved by applying appropriate linear or nonlinear programming algorithms Hettich and Kortanek (1993). Since the computation of the noise relies on the value of β, we suspect the overall computational burden will be large (as one first needs to solve for the worst-case block dropout noise, and then solve for the optimal β).

A.6.4 Additive Perturbations in the Linear Regression Model

Let $\xi \equiv (\xi_1,\ldots,\xi_d)^\top$ be defined as a d-dimensional vector of random variables that are independent of (X,Y). We now perturb the distribution $\mathbb{Q}_0$ by considering the transformation

$$(X, Y) \mapsto (X_1 + \xi_1, \ldots, X_d + \xi_d, Y)^\top.$$

As a result, each covariate X_j is distorted in an additive fashion by ξ_j. We abbreviate $(X_1 + \xi_1, \ldots, X_d + \xi_d)^\top$ by $X + \xi$.

We restrict the distribution of ξ in the following way. First, for a parameter $\lambda \in [0, \infty)$, we define $\mathcal{Q}_j(\lambda)$ to be the set of distributions for ξ_j that have mean 0 and variance $\mathbb{E}_{\mathbb{Q}_0}[x_j^2]$. More specifically,

$$\mathcal{Q}_j(\delta) \equiv \left\{\mathbb{Q}_j : \mathbb{Q}_j \text{ is a probability distribution on } \mathbb{R}, \, \mathbb{E}_{\mathbb{Q}_j}[\xi_j] = 0, \mathbb{E}_{\mathbb{Q}_j}[\xi_j^2] \leq \lambda \mathbb{E}_{\mathbb{Q}_0}[x_j^2]\right\}. \tag{46}$$

Consider now the joint random vector $(X, Y, \xi) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d$. For a constant $\lambda \in [0, \infty)$ consider the joint distributions over (X, Y, ξ) defined by

$$\tilde{\mathcal{U}}(\mathbb{Q}_0, \lambda) = \{\mathbb{Q}_0 \otimes \mathbb{Q}_1 \otimes \ldots \otimes \mathbb{Q}_d : \mathbb{Q}_j \in \mathcal{Q}_j(\lambda) \; \forall j = 1, \ldots, d\},$$

where $\otimes$ denotes the product measure (meaning that the joint distribution is the product of the independent marginals $\mathbb{Q}_j$, $j = 0, \ldots, d$).

For the linear regression model, for any $\mathbb{Q} \in \tilde{\mathcal{U}}(\widehat{\mathbb{P}}_n, \lambda)$ we have

$$\begin{aligned} \mathbb{E}_{\mathbb{Q}}\left[(Y - (X+\xi)^\top \beta)^2\right] &= \mathbb{E}_{\widehat{\mathbb{P}}_n}\left[(Y - X\beta)^2\right] + \sum_{j=1}^d \beta_j^2 \mathbb{E}_{\mathbb{Q}_j}[\xi_j^2], \\ &\leq \mathbb{E}_{\widehat{\mathbb{P}}_n}\left[(Y - X\beta)^2\right] + \sum_{j=1}^d \beta_j^2 \lambda \mathbb{E}_{\widehat{\mathbb{P}}_n}[x_j^2], \\ &= \frac{1}{n}\left[(\mathbf{Y} - \mathbf{X}\beta)^\top(\mathbf{Y} - \mathbf{X}\beta) + \lambda \beta^\top \mathbf{\Lambda} \beta\right], \end{aligned}$$

where $\mathbf{\Lambda} = \operatorname{diag}(\mathbf{X}^\top \mathbf{X})$. Therefore,

$$\sup_{\mathbb{Q} \in \tilde{\mathcal{U}}(\widehat{\mathbb{P}}_n, \lambda)} \mathbb{E}_{\mathbb{Q}}\left[(Y - (X+\xi)^\top \beta)^2\right] = \frac{1}{n}\left[(\mathbf{Y} - \mathbf{X}\beta)^\top(\mathbf{Y} - \mathbf{X}\beta) + \lambda \beta^\top \mathbf{\Lambda} \beta\right].$$

Corollary 2 implies that, for the linear regression model, the distributionally robust estimator to additive perturbations equals the dropout estimator with dropout probability $\lambda/(1+\lambda)$. This means that the dropout estimator remains distributionally robust over the set of additive and multiplicative perturbations of the empirical distribution:

$$\mathcal{U}(\widehat{\mathbb{P}}_n, \delta) \cup \tilde{\mathcal{U}}(\widehat{\mathbb{P}}_n, \delta/(1-\delta)).$$

A.7 Additional Numerical Results

Here, we outline an empirical rationale behind our parameter choices.

A.7.1 Learning Rate

We first fix an all-zeros initialization scheme, and vary the learning rate. We summarize the average parameter divergence and 1-standard deviation error for 20 repetitions of the SGD algorithm in Table 2. We can observe that the learning rate 0.0001 shows a clear advantage. We plot the average parameter divergence and 1-standard deviation error for the SGD trajectory up to the 180 s wall-clock time in Figure 4, which demonstrates that the divergence saturates with the selected learning rate (also see Figure 5 for detail between the 30 s and 180 s wall-clock times).

A.7.2 Initialization

Next, we fix the learning rate to be 0.0001, and consider different initialization schemes. We note that the mean value (resp., absolute value) of elements in β_n^* is 0.3947 (resp., 0.6977). Table 3 shows the average parameter divergence and the 1-standard deviation from 20 repetitions of the SGD algorithm. We see that the initialization at origin is a fair choice.

A.7.3 Naive Monte Carlo with a fixed K

We compare to the naïve Monte Carlo implementation with a fixed K, where we do gradient descent on objective (32). The gradient-descent learning rate is searched over the grid $\{10^{-i}, i = 0, 1, \ldots\}$. We summarize the average parameter divergence and 1-standard deviation error for 20 repetitions of the approach (with the best learning rate) in Table 4. Note that, for small K, the objective (32) has a high bias, while for large K, there are fewer gradient descent steps completed within 60 s due to heavy computational burdens.

Learning rate	0.1	0.01	0.001
$\|\widehat{\beta}_{SGD} - \beta_n^*\|_\infty$	1.1026 ± 0.1705	0.2717 ± 0.0403	0.0827 ± 0.0133
Learning rate	**0.0001**	0.00001	0.000001
$\|\widehat{\beta}_{SGD} - \beta_n^*\|_\infty$	$\mathbf{0.0301 \pm 0.0025}$	0.6702 ± 0.1082	1.7202 ± 0.0044

Table 2: Comparison for different learning rates, with fixed zero initializations.

A.7.4 Wall-Clock Time

We document the numerical results for 120 s/180 s wall-clock time; see Figures 6–8 for the case of 120 s, and Figures 9–11 for the case of 180 s. We see that the proposed unbiased

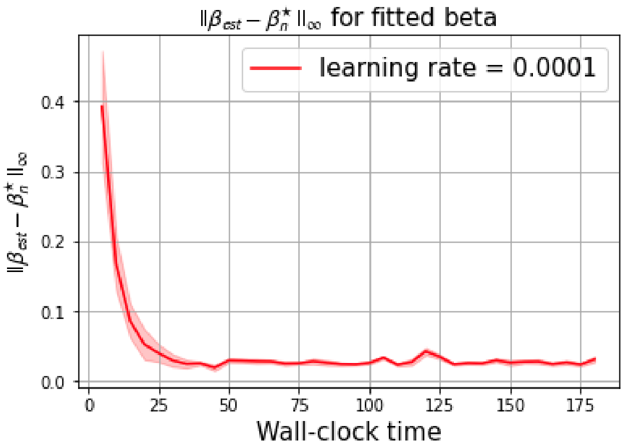

Figure 4: SGD trajectory up to 180 s wall-clock time

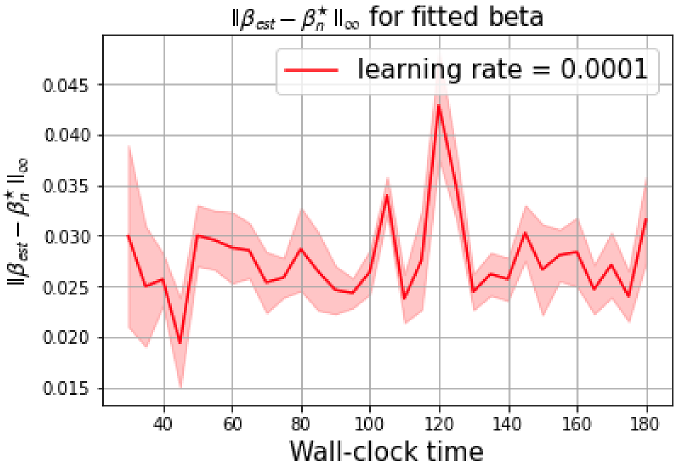

Figure 5: SGD trajectory between 30 s and 180 s wall-clock time

approach outperforms the standard SGD when the number of parallel iterations reaches above some threshold.

Initializations	**all zeros**	all 0.2 s	all 1 s
$\|\widehat{\beta}_{SGD}-\beta_n^*\|_\infty$	$\mathbf{0.0301 \pm 0.0025}$	0.0317 ± 0.0047	0.0614 ± 0.0196
Initializations	i.i.d. $\mathcal{N}(0,1)$	i.i.d. $\mathcal{N}(0,10)$	i.i.d. $\mathcal{N}(0,10^2)$
$\|\widehat{\beta}_{SGD}-\beta_n^*\|_\infty$	0.0376 ± 0.0067	0.1006 ± 0.0469	0.3208 ± 0.1432

Table 3: Comparison for different initialization schemes with fixed learning rate 0.0001.

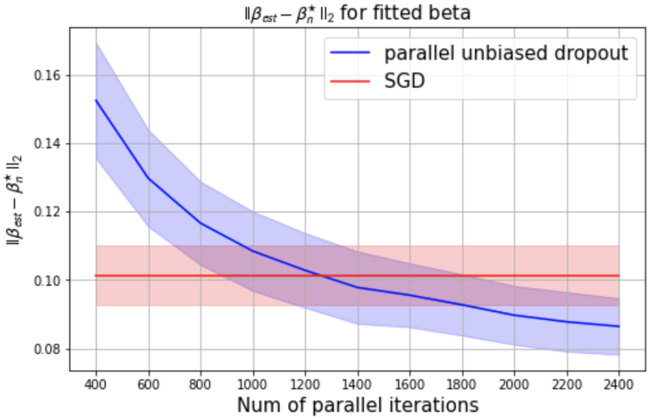

Figure 6: l_2 difference for 120 s wall-clock time

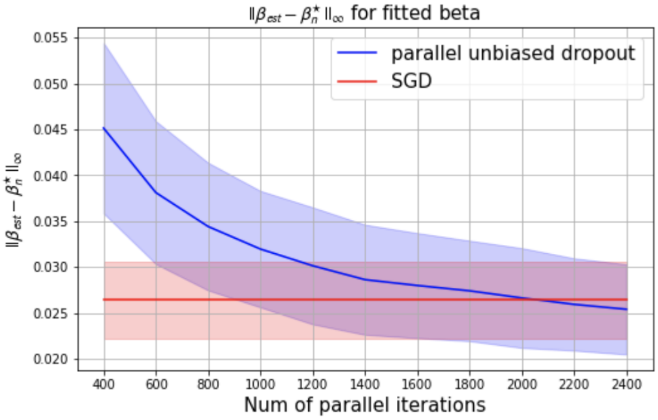

Figure 7: l_∞ difference for 120 s wall-clock time

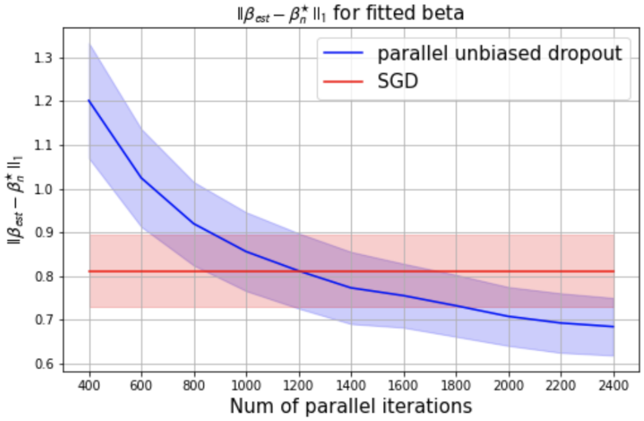

Figure 8: l_1 difference for 120 s wall-clock time

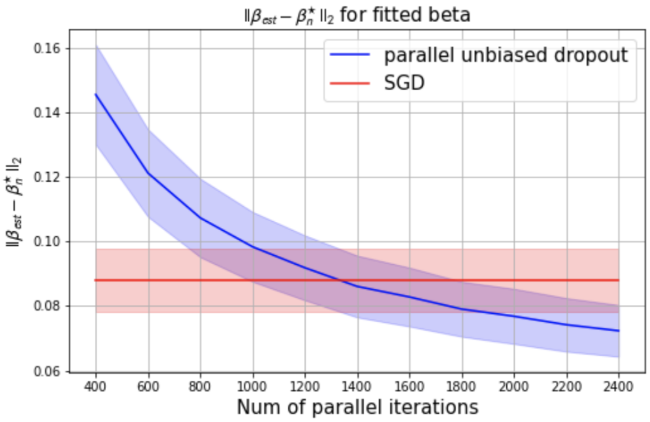

Figure 9: l_2 difference for 180 s wall-clock time

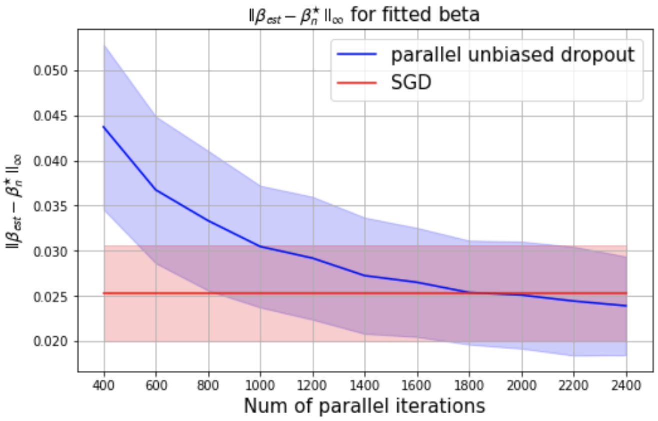

Figure 10: l_∞ difference for 180 s wall-clock time

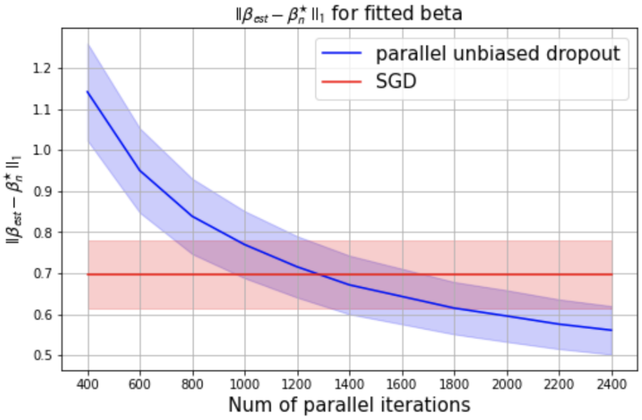

Figure 11: l_1 difference for 180 s wall-clock time

K	2^5	2^{10}	2^{15}
$\|\widehat{\beta}_{naive\,MC} - \beta_n^*\|_\infty$	0.7370 ± 0.1181	0.1281 ± 0.0231	0.7417 ± 0.0046

Table 4: Comparison with naïve Monte Carlo with a fixed K.

References

S Alan, O Attanasio, and M Browning. Estimating Euler equations with noisy data: two exact GMM estimators. *Journal of Applied Econometrics*, 24(2):309–324, 2009.

B Von Bahr. On the convergence of moments in the central limit theorem. *Annals of Mathematical Statistics*, 36(3):808–818, 06 1965.

C M Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

J Blanchet, P Glynn, and Y Pei. Unbiased multilevel Monte Carlo: Stochastic optimization, steady-state simulation, quantiles, and other applications. *arXiv preprint arXiv:1904.09929*, 2019a.

J Blanchet, Y Kang, and K Murthy. Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56:830–857, 2019b.

S Boyd and L Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

T Christensen and B Connault. Counterfactual sensitivity and robustness. *arXiv preprint arXiv:1904.00989*, 2019.

A DasGupta. *Asymptotic Theory of Statistics and Probability.* Springer Verlag, 2008.

E Delage and Y Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

T Devries and G W Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017.

D Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society, Series B*, 56, 1994.

J Duchi and H Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

R Durrett. *Probability: Theory and Examples.* Cambridge University Press, 2019.

L Fahrmeir and H Kaufmann. Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics*, pages 342–368, 1985.

M H Farrell, T Liang, and S Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

T Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*, volume 7. Academic Press New York, 1967.

M B Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

M B Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259, 2015.

I Goodfellow, Y Bengio, and A Courville. *Deep Learning.* MIT Press, 2016.

L P Hansen and T J Sargent. *Robustness.* Princeton University Press, 2008.

D P Helmbold and P M Long. On the inductive bias of dropout. *The Journal of Machine Learning Research*, 16(1):3403–3454, 2015.

R Hettich and K Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

G E Hinton, N Srivastava, A Krizhevsky, I Sutskever, and R R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

K Hornik, M Stinchcombe, and H White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

J T Hwang. Multiplicative errors-in-variables models with applications to recent data released by the US Department of Energy. *Journal of the American Statistical Association*, 81(395):680–688, 1986.

K Isii. On sharpness of Tchebycheff-type inequalities. *Annals of the Institute of Statistical Mathematics*, 14(1):185–197, 1962.

J Kim and W Winkler. Multiplicative noise for masking continuous data. *Statistics*, 1:9, 2003.

R H Lyles and L L Kupper. A detailed evaluation of adjustment methods for multiplicative measurement error in linear regression with applications in occupational epidemiology. *Biometrics*, pages 1008–1025, 1997.

L Maaten, M Chen, S Tyree, and K Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pages 410–418, 2013.

P McCullagh and J A Nelder. *Generalized Linear Models*. Chapman & Hall, 1989.

T K Nayak, B Sinha, and L Zayatz. Statistical properties of multiplicative noise masking for confidentiality protection. *Journal of Official Statistics*, 27(3):527, 2011.

A Nemirovski, A Juditsky, G Lan, and A Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

W K Newey and D McFadden. Large sample estimation and hypothesis testing. *Handbook of Econometrics*, 4:2111–2245, 1994.

V A Nguyen, X Zhang, J Blanchet, and A Georghiou. Distributionally robust parametric maximum likelihood estimation. In *Advances in Neural Information Processing Systems 33*, 2020.

D S Park, W Chan, Y Zhang, C-C Chiu, B Zoph, E D Cubuk, and Q V Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proc. Interspeech 2019*, pages 2613–2617, 2019.

D A Pierce, D O Stram, M Vaeth, and D W Schafer. The errors-in-variables problem: considerations provided by radiation dose-response analyses of the a-bomb survivor data. *Journal of the American Statistical Association*, 87(418):351–359, 1992.

A E Raftery, D Madigan, and J A Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.

H Rahimian and S Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

H Robbins and S Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

H Scarf. A min–max solution of an inventory problem. *Studies in the Mathematical Theory of Inventory and Production*, 10:201–209, 1958.

J Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875 – 1897, 2020.

A Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

A Shapiro, D Dentcheva, and A Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2014.

N Srivastava, G Hinton, A Krizhevsky, I Sutskever, and R Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

S Wager, S Wang, and P S Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems 26*, pages 351–359. 2013.

M J Wainwright and M I Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc, 2008.

S Wang and C Manning. Fast dropout training. In *International Conference on Machine Learning*, pages 118–126, 2013.

C Wei, S Kakade, and T Ma. The implicit and explicit regularization effects of dropout. In *International Conference of Machine Learning*, 2020.

W Wiesemann, D Kuhn, and M Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

M Zinkevich, M Weimer, L Li, and A J Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.