A OLS AND THE PSEUDOINVERSE

Classical statistics (Murphy, 2022) tells us that when X is full rank, the $\hat{w}$ minimizing this expression—known as the OLS regressor—has the following form:

$$\hat{w}_{OLS} = (X^\top X)^{-1} X^\top Y_X \tag{16}$$

Of course, if X is not full rank, the product $X^\top X$ cannot be inverted. In this case, the minimum norm solution can be constructed using the singular value decomposition of X. Specifically, X can be decomposed as $X = U_X D_X V_X^\top$. We can then construct the pseudoinverse of X by using U_X, V_X, and the matrix $D^\dagger$ which is given by taking the transpose of D, and replacing the diagonal singular value elements with their reciprocal. In the case that the singular value is zero, the value of zero is used instead. The pseudoinverse is then constructed as $X^\dagger = V_X D_X^\dagger U_X^\top$. Using these components, the minimum norm solution in the case of a degenerate X matrix is given by the following expression:

$$\hat{w} = X^\dagger Y_X = V_X D_X^\dagger U_X^\top Y_X \tag{17}$$

B DERIVATION OF LOSS OF OLS UNDER COVARIATE SHIFT

We are interested in the following expression for the OOD risk of the OLS regressor:

$$\text{Risk}_{\text{OOD}}(\hat{w}) = \mathbb{E}[\|Y_Z - Z\hat{w}\|_2^2] = \mathbb{E}[\|Zw^* - ZX^\dagger Y\|_2^2] = \mathbb{E}[\|Zw^* - ZX^\dagger(Xw^* + \epsilon)\|_2^2] \tag{18}$$

If we assume that w^* exists within the span of the rows of X, then $X^\dagger X$ acts as an identity on w^*, giving us:

$$= \mathbb{E}[\|ZX^\dagger \epsilon\|_2^2] \tag{19}$$

The euclidean norm is $\|x\|_2 = \sqrt{x^\top x}$, so we can rephrase this expression as a scalar dot product. Scalars can be seen as 1×1 matrices, and are therefore equal to their trace. Therefore we can express this dot product as a trace in order to later use the cyclic property of the trace operator:

$$= \mathbb{E}[\epsilon^\top X^{\dagger\top} Z^\top Z X^\dagger \epsilon] = \mathbb{E}[tr(\epsilon^\top X^{\dagger\top} Z^\top Z X^\dagger \epsilon)] \tag{20}$$

We can cycle the trace and apply the properties of the trace of the product of two $N \times N$ matrices:

$$= \mathbb{E}[tr(\epsilon\epsilon^\top X^{+T} Z^\top Z X^\dagger)] = \mathbb{E}[\sum_{i=1}^{N}\sum_{j=1}^{N}(\epsilon\epsilon^\top)_{i,j}(X^{+T} Z^\top Z X^\dagger)_{i,j}] \tag{21}$$

Since each entry of ϵ is independent from the other entries, and these entries follow the normal distribution $\mathcal{N}(0, \sigma^2)$, by applying the linearity of expectation we know that every term in this sum such that $i \neq j$ will be equal to zero, giving us:

$$= \sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{E}[(\epsilon\epsilon^\top)_{i,j}](X^{+T} Z^\top Z X^\dagger)_{i,j} = \sum_{i=1}^{N}\sigma^2(X^{+T} Z^\top Z X^\dagger)_{i,i} = \sigma^2 tr(X^{+T} Z^\top Z X^\dagger) \tag{22}$$

We will use the singular value decompositions of these two matrices to simplify the expression further after cycling the trace:

$$= \sigma^2 tr(Z^\top Z X^\dagger X^{+T}) = \sigma^2 tr(V_Z D_Z^\top U_Z^\top U_Z D_Z V_Z^\top V_X D_X^\dagger U_X^\top U_X D_X^{\dagger\top} V_X^\top) \tag{23}$$

$$= \sigma^2 tr(V_Z D_Z^2 V_Z^\top V_X D_X^{\dagger 2} V_X^\top) = \sigma^2 tr(D_X^{\dagger 2} V_X^\top V_Z D_Z^2 V_Z^\top V_X) \tag{24}$$

where $D_Z^2, D_X^{\dagger 2}$ are $D \times D$ diagonal matrices with diagonal values equal to the diagonal values of D_Z and $D_X^\dagger$ squared, respectively. The i^{th} diagonal entry of the matrix $V_X^\top V_Z D_Z^2 V_Z^\top V_X$ is:

$$\left[diag(V_X^\top V_Z D_Z^2 V_Z^\top V_X)\right]_i = \sum_{j=1}^{D} \lambda_{z,j}^2 \langle e_{x,i}, e_{z,j} \rangle^2 \tag{25}$$

Meaning that the entire expression will be equal to the value described:

$$\text{Risk}_{\text{OOD}}(\hat{w}) = \sigma^2 \sum_{i=1}^{D} \sum_{j=1}^{D} \frac{\lambda_{z,j}^2}{\lambda_{x,i}^2} \langle e_{x,i}, e_{z,j} \rangle^2 \mathbb{1}[\lambda_{x,i} > 0]. \tag{26}$$

C DERIVATION OF BIAS-VARIANCE DECOMPOSITION

SpAR produces a regressor of the following form:

$$w_{\text{proj}} = \hat{w} - \sum_{e \in S} \langle \hat{w}, e \rangle e \tag{27}$$

Where we are projecting out a set of eignevectors S from the pseudoinverse solution $\hat{w}$. We can substitute this into our expression for the OOD risk of a regressor to arrive at a bias-variance decomposition.

$$\text{Risk}_{\text{OOD}}(w_{\text{proj}}) = \mathbb{E}[\| Z w^* - Z(\hat{w} - \sum_{e_{z,j} \in S} \langle \hat{w}, e_{z,j} \rangle e_{z,j}) \|_2^2] \tag{28}$$

$$= \mathbb{E}[\| - Z V_X D_X^\dagger U_X^\top \epsilon + Z \sum_{e_{z,j} \in S} (\epsilon^\top X^{\dagger\top} e_{z,j} + w^{*\top} e_{z,j}) e_{z,j} \|_2^2] \tag{29}$$

$$= \mathbb{E}[\| - Z V_X D_X^\dagger U_X^\top \epsilon + Z \sum_{e_{z,j} \in S} \langle w^*, e_{z,j} \rangle e_{z,j} + Z \sum_{e_{z,j} \in S} \langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j} \rangle e_{z,j} \|_2^2] \tag{30}$$

We can further simplify this expression by using the fact that the eigenvectors in $Rows(V_Z^\top)$ form an orthonormal basis, and so the sum of their outer products forms an identity matrix. Formally, $\sum_{j=1}^{D} e_{z,j} e_{z,j}^\top = I$. Using this on the leftmost term in the sum, we have:

$$= \mathbb{E}[\| - Z \sum_{j=1}^{D} e_{z,j} e_{z,j}^\top V_X D_X^\dagger U_X^\top \epsilon + Z \sum_{e_{z,j} \in S} \langle w^*, e_{z,j} \rangle e_{z,j} + Z \sum_{e_{z,j} \in S} \langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j} \rangle e_{z,j} \|_2^2] \tag{31}$$

$$= \mathbb{E}[\| -Z\sum_{j=1}^{D}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j} + Z\sum_{e_{z,j}\in S}\langle w^*, e_{z,j}\rangle e_{z,j} + Z\sum_{e_{z,j}\in S}\langle V_X D_X^\dagger U_X^\top\epsilon, e_{z,j}\rangle e_{z,j}\|_2^2] \tag{32}$$

We can use the fact that $S \cup S^c$ form an orthogonal basis, where S^c is the complement set of eigenvectors. We are also assuming that we are only projecting out vectors from the Z right singular vector basis. This gives us:

$$\mathbb{E}[\| -Z\sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j} + Z\sum_{e_{z,j}\in S}\langle w^*, e_{z,j}\rangle e_{z,j}\|_2^2] = \mathbb{E}[\|V - B\|_2^2] \tag{33}$$

The euclidean norm $\|x\|_2 = \sqrt{x^\top x}$, and so we can consider the sum of products $V^\top V - 2V^\top B + B^\top B$. If we take the expectation over the error term ϵ, which has mean 0, we are left with only $V^\top V + B^\top B$.

$V^\top V$ is the error term we are already familiar with (Theorem 1), restricted to the eigenvectors that weren't projected out:

$$V^\top V = (Z\sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j})^\top Z\sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j} \tag{34}$$

$$= (\sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j})^\top Z^\top Z\sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j} \tag{35}$$

We note that each vector $e_{z,j} \in S^c$ is an eigenvector of $Z^\top Z$ with eigenvalue $\lambda_{z,j}^2$.

$$= \sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle e_{z,j}^\top \sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle \lambda_{z,j}^2 e_{z,j} \tag{36}$$

$$= \sum_{e'_{z,j}\in S^c}\sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e'_{z,j}\rangle\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle \lambda_{z,j}^2 e'^\top_{z,j} e_{z,j} \tag{37}$$

Since S^c is a subset of an orthonormal basis, we know that $e'^\top_{z,j} e_{z,j} = 1$ iff $e'_{z,j} = e_{z,j}$. Otherwise, $e'^\top_{z,j} e_{z,j} = 0$.

$$= \sum_{e_{z,j}\in S^c}\langle V_X D_X^\dagger U_X^\top \epsilon, e_{z,j}\rangle^2 \lambda_{z,j}^2 = \sum_{e_{z,j}\in S^c}\epsilon^\top X^{\dagger\top} e_{z,j} e_{z,j}^T X^\dagger \epsilon \lambda_{z,j}^2 \tag{38}$$

In the expected loss, the expectation operator is applied to this expression, giving:

$$\mathbb{E}[V^\top V] = \mathbb{E}[\sum_{e_{z,j}\in S^c}\epsilon^\top X^{\dagger\top} e_{z,j} e_{z,j}^T X^\dagger \epsilon \lambda_{z,j}^2] \tag{39}$$

We can use the properties of the trace to isolate the label noise, as in Appendix B:

$$= \sum_{e_{z,j}\in S^c}\sigma^2 tr(e_{z,j}^\top X^\dagger X^{\dagger\top} e_{z,j})\lambda_{z,j}^2 \tag{40}$$

We can analyze the inner product of the vector $X^{\dagger\top} e_{z,j} = U_X D_X^{\dagger\top} V_X^\top e_{z,j}$ with itself:

$$e_{z,j}^\top X^\dagger X^{\dagger\top} e_{z,j} = \sum_{i=1}^{d} \sum_{k=1}^{d} \frac{1}{\lambda_{x,i}} \langle e_{z,j}, e_{x,i} \rangle \frac{1}{\lambda_{x,k}} \langle e_{z,j}, e_{x,k} \rangle u_{x,i}^\top u_{x,k} \mathbb{1}[\lambda_{x,i} > 0] \mathbb{1}[\lambda_{x,k} > 0] \tag{41}$$

Where $u_{x,i}$ is the i^{th} column of U_X, i.e. the i^{th} left singular vector of X. These left singular vectors also create an orthonormal basis, and so $u_{x,i}^\top u_{x,k} = 1$ iff $u_{x,i} = u_{x,k}$. Otherwise, $u_{x,i}^\top u_{x,k} = 0$. This ultimately gives us:

$$\mathbb{E}[V^\top V] = \sigma^2 \sum_{i=1}^{D} \sum_{j, e_{z,j} \in S^c} \frac{\lambda_{z,j}}{\lambda_{x,i}} \langle e_{x,i}, e_{z,j} \rangle^2 \mathbb{1}[\lambda_{x,i} > 0] \tag{42}$$

We can use similar reasoning to show that bias term $B^\top B$ is a simple expression relying on the true weight vector:

$$\mathbb{E}[B^T B] = B^T B = \sum_{e_{z,j} \in S} \langle w^*, e_{z,j} \rangle e_{z,j}^\top Z^\top Z \sum_{e_{z,j} \in S} \langle w^*, e_{z,j} \rangle e_{z,j} \tag{43}$$

$$\tag{44}$$

$$= \sum_{j, e_{z,j} \in S} \langle w^*, e_{z,j} \rangle^2 \lambda_{z,i}^2 \tag{45}$$

Therefore, we have the following expression for the expected loss:

$$\mathbb{E}[\| Z w^* - Z(\hat{w} - \sum_{e_{z,j} \in S} \langle \hat{w}, e_{z,j} \rangle e_{z,j}) \|_2^2] = \mathbb{E}[V^\top V] + \mathbb{E}[B^\top B] \tag{46}$$

$$= \sigma^2 \sum_{i=1}^{D} \sum_{j, e_{z,j} \in S^c} \frac{\lambda_{z,j}}{\lambda_{x,i}} \langle e_{x,i}, e_{z,j} \rangle^2 \mathbb{1}[\lambda_{x,i} > 0] + \sum_{j, e_{z,j} \in S} \langle w^*, e_{z,j} \rangle^2 \lambda_{z,j}^2 \tag{47}$$

D PROOF OF THEOREM 3

In this section, we provide the proof of Theorem 3.

To start, we note that if $S_\phi = \{\}$, then the projected regressor created with this set is the pseudoinverse solution $\hat{w}$:

$$w_{projS_\phi} = \hat{w} - \sum_{e \in S_\phi} \langle \hat{w}, e \rangle e = \hat{w}. \tag{48}$$

Therefore, by Theorem 2, we know that the loss of this regressor will consist entirely of the variance terms associated with the eigenvectors. This is essentially a recovery of Theorem 1:

$$\mathbb{E}[\|Y_Z - Z\hat{w}\|_2^2] = \sum_{j=1}^{D} \sigma^2 \sum_{i=1}^{D} \frac{\lambda_{z,j}^2}{\lambda_{x,i}^2} \langle e_{x,i}, e_{z,j} \rangle^2 \mathbb{1}[\lambda_{x,i} > 0] = \sum_{j=1}^{D} \text{Var}_{z,j}. \tag{49}$$

We now compare this with the loss of the regressor created using the set S^*, which is:

$$w^*_{\text{proj}} = \hat{w} - \sum_{e \in S^*} \langle \hat{w}, e \rangle e \tag{50}$$

Again invoking Theorem 2, the expected loss of this estimator is:

$$\mathbb{E}[\|Y_Z - Zw^*_{\text{proj}}\|_2^2] = \sum_{e_{z,j} \in S^{*c}} \sigma^2 \sum_{i=1}^{D} \frac{\lambda_{z,j}^2}{\lambda_{x,i}^2} \langle e_{x,i}, e_{z,j} \rangle^2 \mathbb{1}[\lambda_{x,i} > 0] + \sum_{e_{z,j} \in S^*} \langle w^*, e_{z,j} \rangle^2 \lambda_{z,i}^2 \tag{51}$$

$$= \sum_{e_{z,j} \in S^{*c}} \text{Var}_{z,j} + \sum_{e_{z,j} \in S^*} \text{Bias}_{z,j} \tag{52}$$

We can now compare the two expected losses:

$$\mathbb{E}[\|Y_Z - Z\hat{w}\|_2^2] - \mathbb{E}[\|Y_Z - Zw^*_{\text{proj}}\|_2^2] = \sum_{e_{z,j} \in S^*} \text{Var}_{z,j} - \text{Bias}_{z,j} \tag{53}$$

From the definition of S^*, we know that for all $e_{z,j} \in S^*, \text{Var}_{z,j} \geq \text{Bias}_{z,j}$. Therefore, for all $e_{z,j} \in S^*, \text{Var}_{z,j} - \text{Bias}_{z,j} \geq 0$, and the sum of these terms will be greater than zero as well. This gives us:

$$\mathbb{E}[\|Y_Z - Z\hat{w}\|_2^2] - \mathbb{E}[\|Y_Z - Zw^*_{\text{proj}}\|_2^2] \geq 0 \implies \mathbb{E}[\|Y_Z - Z\hat{w}\|_2^2] \geq \mathbb{E}[\|Y_Z - Zw^*_{\text{proj}}\|_2^2] \tag{54}$$

E DISTRIBUTION OF $\widehat{\text{Bias}}$

In Section 3.4 we make statements about the distribution of $\widehat{\text{Bias}}$. In this section, we further explain our reasoning for these claims.

$$\widehat{\text{Bias}}_{z,j} = \langle \hat{w}, e_{z,j} \rangle^2 \lambda_{z,j}^2 = (w^{*T} e_{z,j} + \epsilon^\top X^{\dagger\top} e_{z,j})^2 \lambda_{z,j}^2. \tag{55}$$

We know that ϵ is a Gaussian vector with zero mean and spherical covariance. Therefore, $\epsilon^\top X^{\dagger\top} e_{z,j} \lambda_{z,j}$ would also have zero mean. For its covariance, we need only to multiply this expression by itself to recognize the expression from previous derivations:

$$\mathbb{E}[e_{z,j}^\top X^\dagger \epsilon \epsilon^\top X^{\dagger\top} e_{z,j} \lambda_{z,j}^2] \tag{56}$$

This expression is seen in the derivation of Theorem 2, where we show it is equal to $\text{Var}_{z,j}$. Therefore, the variance of $\epsilon^\top X^{\dagger\top} e_{z,j} \lambda_{z,j}$ is $\text{Var}_{z,j}$. With this in mind, we can rewrite this expression as a scaling of a standard normal random variable:

$$\epsilon^\top X^{\dagger\top} e_{z,j}\lambda_{z,j} = \sqrt{\mathrm{Var}_{z,j}}\beta, \quad \beta \sim \mathcal{N}(0,1) \tag{57}$$

With this in mind, we can also easily describe the distribution of $\langle \hat{w}, e_{z,j}\rangle\lambda_{z,j}$:

$$\langle \hat{w}, e_{z,j}\rangle\lambda_{z,j} = w^{*T} e_{z,j}\lambda_{z,j} + \epsilon^\top X^{\dagger\top} e_{z,j}\lambda_{z,j} \tag{58}$$

Which is a Gaussian random variable plus a constant, which shifts the mean of the Gaussian. This gives us the two distributions we list in Section 3.4:

$$\epsilon^\top X^{\dagger\top} e_{z,j}\lambda_{z,j} \sim \mathcal{N}(0, \mathrm{Var}_{z,j}), \quad \langle \hat{w}, e_{z,j}\rangle\lambda_{z,j} \sim \mathcal{N}(\sqrt{\mathrm{Bias}_{z,j}}, \mathrm{Var}_{z,j}). \tag{59}$$

We would next like to explain the claims made in Case 2 of Section 3.4. Specifically, we make claims about the distribution of $\widehat{\mathrm{Bias}}_{z,j}$ when $\widehat{\mathrm{Bias}}_{z,j} \approx (\epsilon^\top X^{\dagger\top} e_{z,j})^2\lambda_{z,j}^2$:

$$\widehat{\mathrm{Bias}}_{z,j} \approx (\epsilon^\top X^{\dagger\top} e_{z,j})^2\lambda_{z,j}^2 = (\sqrt{\mathrm{Var}_{z,j}}\beta)^2 = \mathrm{Var}_{z,j}\beta^2 \tag{60}$$

$$\beta \sim \mathcal{N}(0,1), \quad \beta^2 \sim \chi^2(df = 1) \tag{61}$$

We therefore know in this case that $\widehat{\mathrm{Bias}}_{z,j}$ is the scaling of a chi-squared random variable. By properties of CDFs, we know that $\Pr(\mathrm{Var}_{z,j}\beta^2 \le \alpha) = \Pr(\beta^2 \le \frac{\alpha}{\mathrm{Var}_{z,j}})$, and therefore we know that the inverse CDF of $\mathrm{Var}_{z,j}\beta^2$ will be $\mathrm{CDF}^{-1}_{\chi^2_{df=1}}(\alpha) \times \mathrm{Var}_{z,j}$.

F PROOF OF PROPOSITION 1

First, we will restructure $\widehat{\mathrm{Bias}}_{z,j}$ as the scaling of a non-central chi-squared random variable. From Equation 59, we know the distribution of $\sqrt{\widehat{\mathrm{Bias}}_{z,j}}$, which we can write in terms of a Gaussian random variable with non-zero mean:

$$\sqrt{\widehat{\mathrm{Bias}}_{z,j}} = \langle \hat{w}, e_{z,j}\rangle\lambda_{z,j} \sim \mathcal{N}(\sqrt{\mathrm{Bias}_{z,j}}, \mathrm{Var}_{z,j}) \tag{62}$$

$$\implies \sqrt{\widehat{\mathrm{Bias}}_{z,j}} = \sqrt{\mathrm{Var}_{z,j}}\delta, \quad \delta \sim \mathcal{N}(\frac{\sqrt{\mathrm{Bias}}}{\sqrt{\mathrm{Var}_{z,j}}}, 1) \tag{63}$$

We therefore know that δ^2 is distributed according to a non-central chi-squared distribution:

$$\widehat{\mathrm{Bias}}_{z,j} = (\sqrt{\mathrm{Var}_{z,j}}\delta)^2 = \mathrm{Var}_{z,j}\delta^2, \quad \delta^2 \sim \chi^2_\lambda(df = 1, \lambda = \frac{\mathrm{Bias}_{z,j}}{\mathrm{Var}_{z,j}}) \tag{64}$$

Furthermore, we know the CDF of this variable as $\Pr(\mathrm{Var}_{z,j}\delta^2 \le \alpha) = \Pr(\delta^2 \le \frac{\alpha}{\mathrm{Var}_{z,j}})$.

We include an eigenvector $e_{z,j}$ in our set S if $\widehat{\mathrm{Bias}}_{z,j} \le \mathrm{CDF}^{-1}_{\chi^2_{df=1}}(\alpha) \times \mathrm{Var}_{z,j}$. The probability of this event occurring is given by the CDF of $\widehat{\mathrm{Bias}}_{z,j}$, which is the following:

$$\Pr(\widehat{\text{Bias}}_{z,j} \leq \text{CDF}^{-1}_{\chi^2_{df=1}}(\alpha) \times \text{Var}_{z,j}) = 1 - Q_{\frac{1}{2}}(\sqrt{\frac{\text{Bias}_{z,j}}{\text{Var}_{z,j}}}, \frac{\sqrt{\text{Var}_{z,j}}\sqrt{\text{CDF}^{-1}_{\chi^2_{df=1}}(\alpha)}}{\sqrt{\text{Var}_{z,j}}}) \tag{65}$$

$$= 1 - Q_{\frac{1}{2}}(\sqrt{\frac{\text{Bias}_{z,j}}{\text{Var}_{z,j}}}, \sqrt{\text{CDF}^{-1}_{\chi^2_{df=1}}(\alpha)}) \tag{66}$$

G PROOF OF LEMMA 1

Proposition 1 gives us an expression for the probability that a given eigenvector is included in the set S. Lemma 1 will use this proposition to demonstrate the tail behaviour of this expression. We will first note that since the expression in Proposition 1 is a CDF, it is continuous. Therefore, in order to find its limits at 0 and ∞, we need only be able to evaluate the expression at these values.

We will first show that:

$$\Pr(e_{z,j} \in S) \xrightarrow{\frac{\text{Bias}_{z,j}}{\text{Var}_{z,j}} \to \infty} 0 \tag{67}$$

This is a special value of the Marcum Q function (Sun & Baricz, 2008). Specifically, $Q_{\frac{1}{2}}(\infty, b) = 1$ for any b. Therefore:

$$\Pr(e_{z,j} \in S) = 1 - Q_{\frac{1}{2}}(\infty, \sqrt{\text{CDF}^{-1}_{\chi^2_{df=1}}(\alpha)}) = 1 - 1 = 0 \tag{68}$$

We will next show that:

$$\Pr(e_{z,j} \in S) \xrightarrow{\frac{\text{Bias}_{z,j}}{\text{Var}_{z,j}} \to 0} \alpha \tag{69}$$

This is another special value of the Marcum Q function (Sun & Baricz, 2008). Specifically:

$$Q_{\frac{1}{2}}(0, b) = \frac{\Gamma(\frac{1}{2}, \frac{b^2}{2})}{\Gamma(\frac{1}{2})} \tag{70}$$

For any b. Here, Γ with one argument is the gamma function and Γ with two arguments is the upper incomplete gamma function. By properties of gamma functions, we know that if γ is the lower incomplete gamma function, then $\Gamma(\frac{1}{2}, \frac{b^2}{2}) + \gamma(\frac{1}{2}, \frac{b^2}{2}) = \Gamma(\frac{1}{2})$. Using this property, and by letting $b = \sqrt{\text{CDF}^{-1}_{\chi^2_{df=1}}(\alpha)}$, we have the following:

$$\Pr(e_{z,j} \in S) = 1 - Q_{\frac{1}{2}}(0, b) = \frac{\Gamma(\frac{1}{2}, \frac{b^2}{2}) + \gamma(\frac{1}{2}, \frac{b^2}{2})}{\Gamma(\frac{1}{2})} - \frac{\Gamma(\frac{1}{2}, \frac{b^2}{2})}{\Gamma(\frac{1}{2})} \tag{71}$$

$$= \frac{\gamma(\frac{1}{2}, \frac{b^2}{2})}{\Gamma(\frac{1}{2})} = \text{CDF}_{\chi^2_{df=1}}(\text{CDF}^{-1}_{\chi^2_{df=1}}(\alpha)) = \alpha \tag{72}$$

Where we have used the observation that the leftmost expression in Equation 72 is the CDF for a chi-squared distribution with one degree of freedom.

H ADDITIONAL TRAINING DETAILS

For our experiments in Section 4, we adapt the code provided by Yao et al. (2022) in this Github repo: `https://github.com/huaxiuyao/C-Mixup`. While training, we perform early stopping on a validation set evaluation metric. For PovertyMap, this procedure is seen in the original work of Koh et al. (2021). We also use the hyperparameters provided in the appendix of Yao et al. (2022)'s work, including the following learning rates and bandwidth parameters for C-Mixup:

Table 4: Hyperparameters used for training models responsible for the results in Section 4.

Hyperparameter	CommunitiesAndCrime	SkillCraft	PovertyMap
Learning Rate	1e-3	1e-2	1e-3
Bandwidth	1.0	5e-4	0.5

We additionally make the modification to train models without a bias term in the final linear layer. This is due to the fact that SpAR assumes a regressor that does not use a bias.

Models are trained using Tesla T4 GPUs from NVIDIA. Tabular and synthetic experiments take less than 10 minutes to run for a single seed and hyperparameter setting. PovertyMap experiments take roughly 3 hours to run when training ERM and roughly 15 hours to run when training C-Mixup.

I TABULAR DATA RESULTS WITH BASE HYPERPARAMETERS

In this section, we provide the table of results that Figure 3 is based upon. This is the performance of the models using the hyperparameters described in Table 4. The results are included in Table 5.

J HYPERPARAMETER SEARCH

For hyperparameter tuning, we perform random search over the learning rate and the bandwidth used in C-Mixup. Specifically, we search over learning rates using the following formula for the learning rate lr and bandwidth bw:

$$lr = base_{lr} * 10^{u}, u \sim Unif(-1, 1) \tag{73}$$

$$bw = base_{bw} * 10^{u}, u \sim Unif(-1, 1) \tag{74}$$

where $base_{lr}$ and $base_{bw}$ are the values described in Table 4 for each dataset. We test out 10 randomly selected hyperparameter settings for both ERM and C-Mixup, and select the settings that yield the best validation performance. Those hyperparameter settings selected for C-Mixup are presented in Table 6 and hyperparameter settings selected for ERM are presented in Table 7.

K TUNED BASELINES

Using the hyperparameters presented in Tables 6 and 7 which were selected hyperparameter search process described in Section J, we benchmark the performance of ERM and C-Mixup models across 10 seeds for the tabular datasets and the 5 data folds for PovertyMap. We report results for PovertyMap and the tabular datasets in Tables 9 and 8, respectively.

We find that SpAR can achieve superior worst group performance than any other method presented in either Tables 9 or 8, or in Section 4. For C-Mixup on CommunitiesAndCrime, we see that tuning hyperparameters on the validation set yields poorer performance (Table 8) than using the hyperparameters presented in Yao et al. (2022)'s work (Table 5). However, we can see that a SpAR model is able to achieve the best worst-group RMSE of any model on this dataset, 0.161.

Table 5: **Tabular data.** OOD RMSE averaged across 10 seeds for models using the hyperparameters described in Table 4.

SkillCraft		
Method	Average RMSE (↓)	Worst Group RMSE (↓)
ERM	6.273 ± 0.384	8.933 ± 1.338
ERM + OLS	6.884 ± 0.860	11.156 ± 3.892
ERM + SpAR (Ours)	**6.049** ± 0.379	**8.317** ± 1.327
C-Mixup	6.319 ± 0.450	8.713 ± 1.106
C-Mixup + OLS	7.070 ± 0.898	11.747 ± 3.450
C-Mixup + SpAR (Ours)	**6.038** ± 0.705	**8.343** ± 1.563

CommunitiesAndCrime		
Method	Average RMSE (↓)	Worst Group RMSE (↓)
ERM	0.134 ± 0.006	0.166 ± 0.014
ERM + OLS	0.142 ± 0.004	0.175 ± 0.012
ERM + SpAR (Ours)	**0.133** ± 0.002	**0.163** ± 0.009
C-Mixup	**0.131** ± 0.005	0.162 ± 0.016
C-Mixup + OLS	0.140 ± 0.003	0.175 ± 0.010
C-Mixup + SpAR (Ours)	0.133 ± 0.002	**0.161** ± 0.004

Table 6: Tuned hyperparameters used for training C-Mixup models. 4.

Hyperparameter	CommunitiesAndCrime	SkillCraft	PovertyMap
Learning Rate	0.003630376073213171	0.023276939100527687	0.003630376073213171
Bandwidth	0.35090148857968506	0.0013316008334250096	0.17545074428984253

Table 7: Tuned hyperparameters used for training ERM models.

Hyperparameter	CommunitiesAndCrime	SkillCraft	PovertyMap
Learning Rate	0.008246671732726021	0.023276939100527687	0.003630376073213171

Table 8: **Tabular data.** OOD RMSE averaged across 10 seeds for models using tuned hyperparameters.

SkillCraft		
Method	Average RMSE (↓)	Worst Group RMSE (↓)
ERM	**5.917** ± 0.620	8.308 ± 1.915
ERM + OLS	6.548 ± 0.915	10.219 ± 3.123
ERM + SpAR (Ours)	6.083 ± 0.681	**8.193** ± 1.212
C-Mixup	**5.816** ± 0.558	8.371 ± 1.611
C-Mixup + OLS	6.535 ± 0.822	10.297 ± 2.362
C-Mixup + SpAR (Ours)	5.833 ± 0.580	**7.922** ± 1.043

CommunitiesAndCrime		
Method	Average RMSE (↓)	Worst Group RMSE (↓)
ERM	**0.133** ± 0.004	**0.161** ± 0.010
ERM + OLS	0.149 ± 0.018	0.184 ± 0.032
ERM + SpAR (Ours)	0.134 ± 0.007	0.164 ± 0.013
C-Mixup	0.133 ± 0.003	0.171 ± 0.012
C-Mixup + OLS	0.144 ± 0.011	0.177 ± 0.019
C-Mixup + SpAR (Ours)	**0.132** ± 0.004	**0.164** ± 0.008

Table 9: **PovertyMap-WILDS.** Average OOD all-group and worst-group Spearman r across 5 splits for models using tuned hyperparameters.

Method	$r_{all}(\uparrow)$	$r_{wg}(\uparrow)$
ERM	0.798 ± 0.052	0.518 ± 0.076
ERM + SpAR (Ours)	**0.799** ± 0.045	**0.522** ± 0.080
C-Mixup	**0.806** ± 0.031	0.523 ± 0.083
C-Mixup + SpAR (Ours)	0.803 ± 0.038	**0.528** ± 0.087

L SENSITIVITY OF ALPHA HYPERPARAMETER

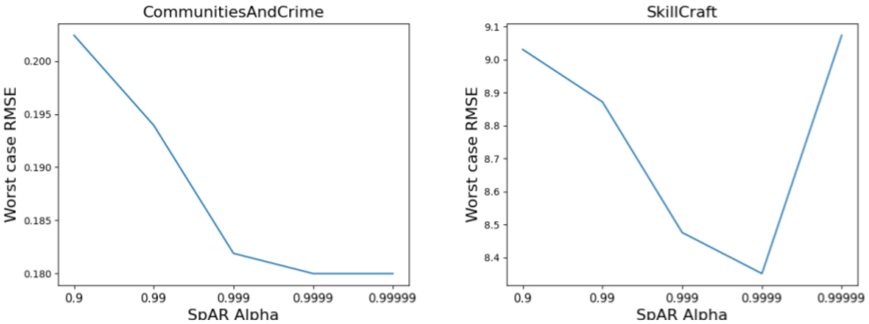

Figure 4: **Hyperparameter sensitivity** SpAR performance as a function of α on tabular datasets.

Throughout this work, we use a single setting of α for each of our experiments. Our specific setting of α=0.999 was selected using a minimal amount of tuning on a single seed of a single experiment. This value was then used on every seed of every dataset, regardless of potential improvements. To achieve a more complete understanding of SpAR's sensitivity to α, we conduct an experiment measuring OOD performance as a function of α when SpAR is applied to an ERM base model on the SkillCraft and CommunitiesAndCrime datasets. See Figure 4 for results. We see that on the CommunitiesAndCrime dataset, a higher α than 0.999 could have resulted in superior worst case performance. Meanwhile, on SkillCraft, we clearly see that setting α too close to 1 can result in very poor worst group performance. Expression 13 in the paper indicates that as α tends towards zero, the regressor produced by SpAR will more closely resemble the solution produced by OLS. Specifically, fewer eigenvectors will be projected out from the OLS solution. Conversely, as α tends towards one, the regressor produced by SpAR will tend towards the zero vector. This can be seen as a tradeoff between the cases where no Spectral Inflation is expected and where Spectral Inflation is expected to occur along every right singular vector.

In general, selecting α using validation set performance can have mixed results, as SpAR is intended to produce a regressor for a specific evaluation set (namely, the OOD test set, not the ID validation set). Future work could investigate the interesting question of how α could be selected based on the amount of spectral inflation presented in the train/evaluation data.

M COMPUTATIONAL COST

The computational cost of SpAR comes from collecting the representations (running forward passes for every train and test example) and performing SVD, with the former step dominating the cost. Notably, it is much less cumbersome than other adaptation techniques. Computing the SVD of the matrix can be done in polynomial time, and we find in practice that performing this one-time post-hoc adaptation is quite efficient relative to other methods that must compute a regularizer or augment data on each training iteration (see Table 10).

Table 10: **Measured train time** on PovertyMap. Each model is trained on a NVIDIA Tesla T4 GPU. In-processing methods and SpAR use a large pool of unlabeled data that are distinct from the test set, but come from the same distribution Sagawa et al. (2021).

Method	Average RMSE
ERM	3h22m ± 0h22m
C-Mixup	14h58 ± 1h01m
DARE-GRAM	5h58m ± 0h26m
ERM + SpAR (Ours)	4h11m ± 0h33m
SpAR only	0h40m ± 0h18m

N LIMITATIONS AND BROADER IMPACTS

SpAR is designed for covariate shift, and its ability to handle other types of distribution shift (such as concept shift) is not known analytically. To be more specific, we assume that the targets have a the same linear relationship (via the ground truth weight w^*) with inputs X and Z, and that X and Z are covariate-shifted. A subtle issue here is that when X and Z are internal representations of some neural net, we require that the difference P and Q is captured in terms of a covariate shift *in the representation space*, which may or may not correspond to a covariate shift in the original input space (which could be some high-dimensional vector, e.g. pixels).

Empirically, however, we successfully apply SpAR to several real-world datasets without assurance that they exhibit only covariate shift, and find promising results. The spectral inflation property that we observe in real data (Figure 2) may be relevant to other distribution shifts as well, although this remains to be seen in future studies. Identifying covariate shift within a datasets is an active area of work (Ginsberg et al., 2022) that complements our efforts in this paper.

Our research seeks to improve OOD generalization with the hopes of ensuring ML benefits are distributed more equitably across social strata. However, it is worthwhile to be self-reflexive about the methodology we use when working towards this goal. For example, for the purposes of comparing against existing methods from the literature, we use the Communities and Crime dataset, where average crime rates are predicted based on statistics of neighborhoods, which could include demographic information. This raises a potential fairness concern: even if we have an OOD-robust model, it may not be fair if it uses demographic information in its predictions. While this is not the focus of our paper, we note that the research community is in the process of reevaluating tabular datasets used for benchmarking (Ding et al., 2021; Bao et al., 2021).