Generalizing Orthogonalization for Models with Non-Linearities

David Rügamer [1 2] **Chris Kolb** [1 2] **Tobias Weber** [1 2] **Lucas Kook** [3] **Thomas Nagler** [1 2]

Abstract

The complexity of black-box algorithms can lead to various challenges, including the introduction of biases. These biases present immediate risks in the algorithms' application. It was, for instance, shown that neural networks can deduce racial information solely from a patient's X-ray scan, a task beyond the capability of medical experts. If this fact is not known to the medical expert, automatic decision-making based on this algorithm could lead to prescribing a treatment (purely) based on racial information. While current methodologies allow for the "orthogonalization" or "normalization" of neural networks with respect to such information, existing approaches are grounded in linear models. Our paper advances the discourse by introducing corrections for non-linearities such as ReLU activations. Our approach also encompasses scalar and tensor-valued predictions, facilitating its integration into neural network architectures. Through extensive experiments, we validate our method's effectiveness in safeguarding sensitive data in generalized linear models, normalizing convolutional neural networks for metadata, and rectifying pre-existing embeddings for undesired attributes.

1. Introduction

In the burgeoning landscape of artificial intelligence and deep learning, black-box algorithms have become a centerpiece for driving advances in many fields of application. These powerful and often inscrutable models offer impressive predictive capabilities, but their complexity also gives rise to challenges that cannot be overlooked. One of the most urgent contemporary challenges is the correction of unwanted behaviors in these algorithms. In particular, the presence of biases in model predictions and learned representations can lead to unintended consequences that transcend the technological sphere, impacting societal norms and ethical considerations. For example, Glocker et al. (2023b) and Weber et al. (2023) realized that predictions of convolutional networks trained for chest X-ray pathology classification are heavily affected by implicitly encoded racial information, leading to potentially inaccurate or unfair medical assessments.

In this paper, we stress the necessity of addressing these challenges and propose several solutions, focusing on what has been termed "orthogonalization" or "normalization" of neural networks in the literature (see, e.g., Lu et al., 2021; Rügamer, 2023). These approaches can be used to adjust predictions, prevent unequal treatment of population subgroups, protect from unintentionally revealing sensitive information, or ensure the interpretability of black-box models.

1.1. Related Literature

Classical Orthogonalization In contrast to research investigating network weight dependencies (see, e.g., Huang et al., 2020), the orthogonalization discussed in this work is usually motivated as a normalization operation or correction method. Lu et al. (2021), e.g., proposed the so-called metadata normalization network, which applies a layer-wise orthogonalization to remove metadata information from layers. Similarly, He et al. (2019) study an orthogonalization procedure to encourage fair learning by removing protected information from feature representations. To circumvent training instabilities caused by dependence on the batch size, Vento et al. (2022) propose to cast the metadata normalization as a bi-level problem optimized using a penalty approach. Kaiser et al. (2022) use the same idea to incorporate fairness in automated decision-making systems for labor market data. A related but different problem is addressed by Rügamer et al. (2023), who propose an orthogonalization cell to preserve effect identifiability and thereby interpretability. This idea was then also adapted to other model classes (Baumann et al., 2021; Kopper et al., 2021). Further, orthogonalization can also be used as a debiasing technique (e.g., in graph neural networks, Palowitch & Perozzi, 2020). Similar to our work, orthogonalization can be

[1]Department of Statistics, LMU Munich, Munich, Germany [2]Munich Center for Machine Learning (MCML), Munich, Germany [3]Institute for Statistics and Mathematics, Vienna University of Economics and Business, Vienna, Austria. Correspondence to: David Rügamer <david@stat.uni-muenchen.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Table 1. Estimated influence (coefficients) of sex, age, and race on the model's predictions when using self-supervised embeddings of the MIMIC-CXR dataset (Sellergren et al., 2022) on the label *Pleural Effusion*. The first row shows uncorrected values, the second row the results after classical orthogonalization, and the third row after applying our generalized orthogonalization. p-values in brackets indicate whether these coefficients represent a significant influence (p-values smaller than 0.05 show a ✗-sign, otherwise a ✓-sign).

	Sex (male)	Age	Race (Black)	Race (White)
Without correction	0.064 (9.3e-9) ✗	2.237 (< 2e-16) ✗	-0.483 (< 2e-16) ✗	-0.050 (0.0743) ✓
Classical orthogonalization $\mathcal{C}^l$	0.014 (0.192) ✓	0.453 (< 2e-16) ✗	-0.117 (< 2e-4) ✗	-0.015 (0.603) ✓
Generalized orthogonalization $\mathcal{C}^h$	0.001 (0.994) ✓	-0.011 (0.689) ✓	-0.001 (0.753) ✓	-0.000 (0.904) ✓

applied as part of a neural network (e.g., Lu et al., 2021) or post-model fitting (Rügamer, 2023). The latter approach also has a link to double and debiased machine learning in econometrics (see, e.g., Chernozhukov et al., 2018), where a related correction is performed to allow for inference statements about a treatment variable after removing (potentially non-linear) nuisance effects.

Fair Machine Learning Another related strand of literature is fair machine learning (Chen et al., 2022). As we discuss later in our work, methods to achieve fairness are closely linked to orthogonalization and vice versa. A wide range of notions and approaches exists to operationalize the concept of fairness (see, e.g., Mehrabi et al., 2021; Pessach & Shmueli, 2022, for literature reviews). Proposed methods are applied at different steps, e.g., in data preprocessing (Calmon et al., 2017) or post-processing (Hardt et al., 2016; Sattigeri et al., 2022; Xu et al., 2022), and have been derived for different model classes, such as generalized linear models (GLMs; Do et al., 2022) or kernel learning (Pérez-Suay et al., 2017). These approaches are usually implemented as constrained (see, e.g., Komiyama et al., 2018; Zafar et al., 2019) or regularized optimization problems (e.g., Scutari et al., 2022; Do et al., 2022) and require making a trade-off between model performance and the amount of achieved fairness. In contrast, our approach forces the model predictions to be uncorrelated with the protected information, thereby ensuring that they can not be inferred to any degree from the predictions using a linear model. While this potentially comes at the cost of a loss in model performance, a conservative correction routine that errs on the side of caution is required in many risk-averse domains such as medical applications with sensitive patient information (Glocker et al., 2023a), or when the goal is to achieve model identifiability (Rügamer, 2023). A limitation of presented fairness solutions is that the sensitive information is made explicit in the model under the assumption of a linear effect on the outcome (see, e.g., Scutari et al., 2022).

Despite the versatility of orthogonalization in practice, most existing methods share the same working principle which assumes linearity in both the model that needs to be corrected and the correction function. In many real-world situations, however, models include non-linearities that render the classical orthogonalization unable to fully correct the model (cf. Table 1 with details in Section 4.2.1).

1.2. Our Contribution

In this paper, we extend the orthogonalization operation to non-linearity in both the model that needs to be corrected and the correction function. We derive a correction routine that works for a variety of non-linear models, including GLMs and neural networks. Moreover, we show how to extend our approach to arbitrary tensor-valued predictions. This allows our method to be flexibly inserted anywhere in common neural network architectures and thereby further generalizes the idea of normalization. Our experimental results demonstrate the efficacy of our approach, showing how 1) sensitive information can be successfully protected in GLMs, 2) neural networks with non-linear activation functions can be normalized for metadata during training, and 3) orthogonalization effectively corrects pre-trained embeddings for unwanted attributes.

2. Orthogonalization

Given an input $\boldsymbol{X} \in \mathbb{R}^{n \times p}, p \leq n$ with full column rank, the linear predictor in a parametric learning model (or the pre-activation of a fully-connected neural network layer) is given by $\boldsymbol{X}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^p$ denotes the model's or layer's weights. In a linear model, we can write the least squares predictions $\hat{\boldsymbol{y}}$ as a function of the inputs $\boldsymbol{X}$ and targets $\boldsymbol{y}$:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \mathcal{P}_{\boldsymbol{X}}\boldsymbol{y},$$

where $\mathcal{P}_{\boldsymbol{X}} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$ is the projection matrix projecting $\boldsymbol{y}$ onto span$(\boldsymbol{X})$, the space spanned by the columns of $\boldsymbol{X}$. Instead of projecting onto span$(\boldsymbol{X})$, we can also project onto the orthogonal complement span$(\boldsymbol{X})^\perp$ using $\mathcal{P}_{\boldsymbol{X}}^\perp\boldsymbol{y} = (\boldsymbol{I}_n - \mathcal{P}_{\boldsymbol{X}})\boldsymbol{y}$, where $\boldsymbol{I}_n$ is the identity matrix. Hence, when left-multiplying a term by $\mathcal{P}_{\boldsymbol{X}}^\perp$, we *orthogonalize* this term (w.r.t. features $\boldsymbol{X}$). Ultimately — whether the motivation is debiasing, protecting information, or identifiability — the universal recipe for most approaches discussed in Section 1.1 is to use the described orthogonalization operation. Given protected features $\boldsymbol{X}$, which are not supposed to influence the modeling process, and a model that uses a different set of features $\boldsymbol{Z} \in \mathbb{R}^{n \times q}, q \geq p$ for prediction, we

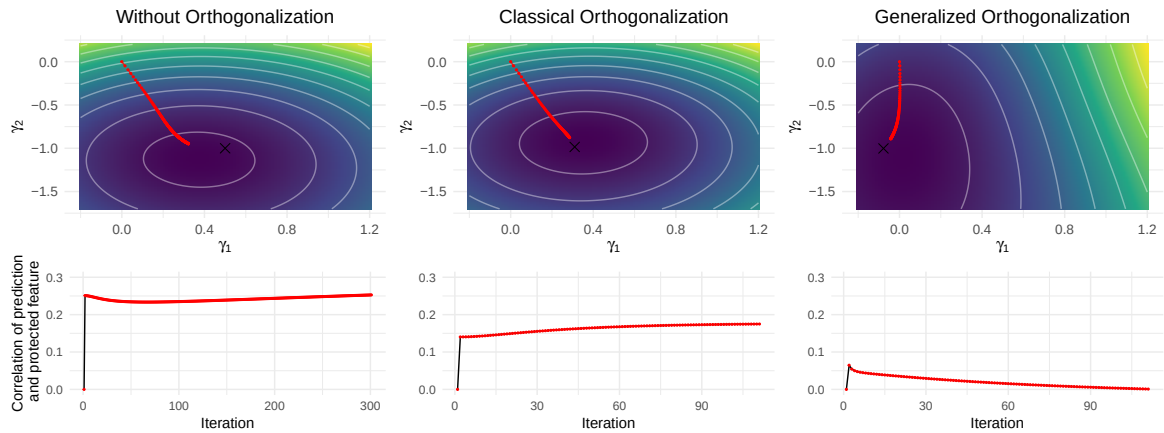

Figure 1. Exemplary optimization process for a fixed number of iterations (converging towards the black crosses) for logistic regression with features z_1, z_2, weights γ_1, γ_2, and one protected feature x_1 correlated with z_1. The upper row shows the loss surface and optimization path for the three different methods (columns) a small step size. The bottom row shows the correlation between the model's prediction and the protected feature along the optimization path, where only the generalized orthogonalization yields predictions uncorrelated with the sensitive information.

can "correct" the model (or, in this case equivalently, the features $\boldsymbol{Z}$) by fitting the model with $\boldsymbol{Z}^c := \mathcal{P}_{\boldsymbol{X}}^{\perp}\boldsymbol{Z}$ instead of $\boldsymbol{Z}$. This is equivalent to removing all linear $\boldsymbol{X}$-information in $\boldsymbol{Z}$. Note that $\boldsymbol{Z}$ might also contain features from $\boldsymbol{X}$, but the correction is supposed to work for unobserved relationships between $\boldsymbol{Z}$ and $\boldsymbol{X}$. We call this correction function $\mathcal{C}^l$ in the following.

To see that $\mathcal{C}^l$ in fact removes the effect of protected features $\boldsymbol{X}$ from $\boldsymbol{Z}$ in the prediction, we can check whether the corrected predictions $\hat{\boldsymbol{y}}^c = \boldsymbol{Z}^c\hat{\boldsymbol{\gamma}}^c$, based on the orthogonalized features $\boldsymbol{Z}^c = \mathcal{P}_{\boldsymbol{X}}^{\perp}\boldsymbol{Z}$ and a corrected weight vector $\hat{\boldsymbol{\gamma}}^c \in \mathbb{R}^q$, can be linearly explained by the features $\boldsymbol{X}$ to any degree. This can be done by regressing $\hat{\boldsymbol{y}}^c$ on $\boldsymbol{X}$. If the correction is successful, the debiased effect

$$\hat{\boldsymbol{\beta}}^c := \arg\min_{\boldsymbol{\beta}^c} \|\hat{\boldsymbol{y}}^c - \boldsymbol{X}\boldsymbol{\beta}^c\|_2^2$$

of $\boldsymbol{X}$ in the linear model with outcome $\hat{\boldsymbol{y}}^c$ and features $\boldsymbol{X}$ should be $\hat{\boldsymbol{\beta}}^c \equiv \boldsymbol{0}$. Plugging in the terms of the ordinary least squares solution for both $\hat{\boldsymbol{\beta}}^c$ and $\hat{\boldsymbol{\gamma}}$, we get

$$\begin{aligned}\hat{\boldsymbol{\beta}}^c &= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\hat{\boldsymbol{y}}^c = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Z}^c\hat{\boldsymbol{\gamma}} \\ &= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\mathcal{P}_{\boldsymbol{X}}^{\perp}\boldsymbol{Z}(\boldsymbol{Z}^\top\mathcal{P}_{\boldsymbol{X}}^{\perp}\boldsymbol{Z})^{-1}\boldsymbol{Z}^\top\mathcal{P}_{\boldsymbol{X}}^{\perp}\boldsymbol{y},\end{aligned}$$

which yields $\boldsymbol{0}$ as $\boldsymbol{X}^\top\mathcal{P}_{\boldsymbol{X}}^{\perp} = \boldsymbol{0}_{p\times n}$. This confirms that we have successfully removed any <u>linear</u> effect of $\boldsymbol{X}$ from the predictions $\hat{\boldsymbol{y}}^c$ and corrected features $\boldsymbol{Z}^c$. So when $\boldsymbol{Z}$ is, e.g., a pre-trained embedding, using $\boldsymbol{Z}^c$ instead of $\boldsymbol{Z}$ in a downstream task removes the partial influence of protected features $\boldsymbol{X}$ contained in $\boldsymbol{Z}$, as we first project the original embedding onto a space that is orthogonal to the space spanned by the protected features $\boldsymbol{X}$.

3. Orthogonalization for Models with Non-linearities

While the classical orthogonalization allows correcting for metadata of images (Lu et al., 2021) or identifying contributions in additive predictors (Rügamer, 2023), all previous methods fail to provide a valid correction in the case of a non-linear, element-wise transformation h applied to $\boldsymbol{Z}\boldsymbol{\gamma}$, i.e., $\hat{\boldsymbol{y}} = h(\boldsymbol{Z}\boldsymbol{\gamma})$, such as in GLMs or most neural network layers (see Figure 1 for an example). In the following, we present an extension to previously proposed orthogonalization approaches that also encompass models with non-linear transformations of the linear predictor, thereby accounting for various important use cases. For simplicity, we motivate our approach using a GLM, but also discuss its embedding in neural architectures in Sections 3.2–3.4. Before deriving the theoretical details, we provide an instructive example in the following.

Example 1. Following Weber et al. (2023), assume we are given an embedding $\boldsymbol{Z} \in \mathbb{R}^{n\times q}$ from a medical imaging task and want to ensure that by using this embedding, we do not share any protected patient information encoded in $\boldsymbol{X}$. Assume that the embedding contains information from $\boldsymbol{X}$ (which represents a very real problem; see Glocker et al., 2023b). When predicting the patients' disease status using the embedding $\boldsymbol{Z}$ in a GLM with $\hat{\boldsymbol{y}} = h(\boldsymbol{Z}\hat{\boldsymbol{\gamma}})$, where h is the sigmoid function, we risk making decisions implicitly based on protected features. Instead, we have to come up with a corrected GLM routine such that the corrected non-linear model predictions $\hat{\boldsymbol{y}}^c$ do not utilize or leak patient information $\boldsymbol{X}$.

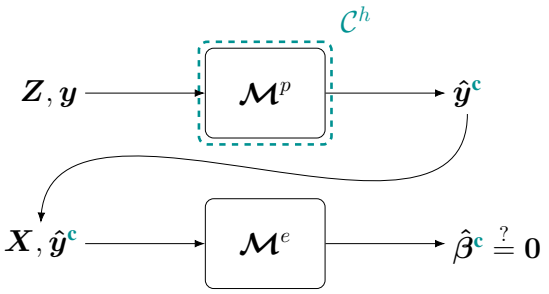

Figure 2. Workflow of generating predictions via model $\mathcal{M}^p$ and checking the influence of $\boldsymbol{X}$ on the resulting predictions $\hat{\boldsymbol{y}}$ using $\mathcal{M}^e$. Green parts indicate the orthogonalization applied to $\mathcal{M}^p$.

3.1. Types of Orthogonalizations

We first start by examining the properties of the original orthogonalization $\mathcal{C}^l$ when applied to models with non-linear activation functions. In particular, it is worth noting that using $\boldsymbol{Z}^c = \mathcal{P}_X^{\perp}\boldsymbol{Z}$ from Section 2 *can* be used in a GLM when the goal is to orthogonalize the model's additive predictor $\boldsymbol{\eta} = \boldsymbol{Z}\boldsymbol{\gamma}$. In this case, a corrected version $\boldsymbol{\eta}^c = \boldsymbol{Z}^c\boldsymbol{\gamma}$ will be orthogonal to the column space of $\boldsymbol{X}$ for any $\boldsymbol{\gamma}$ and hence, *before the non-linear transformation* h, "free" of any (linear) effect of protected features $\boldsymbol{X}$. In other words, regressing $\boldsymbol{\eta}^c$ on $\boldsymbol{X}$ will result in effects $\hat{\boldsymbol{\beta}}^c = \boldsymbol{0}$ as derived for the linear case in Section 2. In contrast, when transforming $\boldsymbol{\eta}^c$ with h, the model space becomes non-linear and regressing $\hat{\boldsymbol{y}} = h(\boldsymbol{\eta}^c)$ on $\boldsymbol{X}$ will generally not yield effects $\hat{\boldsymbol{\beta}}^c = \boldsymbol{0}$, i.e., the transformed predictor $h(\boldsymbol{\eta}^c)$ still potentially contains $\boldsymbol{X}$-information (cf. Appendix A for an illustrative explanation). As illustrated in Example 1, such non-linear transformations are ubiquitous in machine learning and employed in most classification tasks.

When using a non-linear activation function h in the prediction of $\boldsymbol{y}$ based on $\boldsymbol{Z}$, a natural choice is to also use the same non-linear transformation to check the model's prediction for any $\boldsymbol{X}$ influence. To better clarify the different types of models involved in an orthogonalization and to unify previous endeavors, we define the following terms for the workflow described in Figure 2:

Definition 1. Given features $\boldsymbol{Z} \in \mathbb{R}^{n\times q}$, a monotonic activation function h, loss function $\ell : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, and protected features stored in a full column rank matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, we define

- the **prediction model** $\mathcal{M}^p : \mathbb{R}^{n\times q} \to \mathbb{R}^n$, $\boldsymbol{Z} \mapsto \hat{\boldsymbol{y}} = h(\boldsymbol{Z}\hat{\boldsymbol{\gamma}})$, the model using $\boldsymbol{Z}$ to generate predictions $\hat{\boldsymbol{y}}$ for an outcome of interest $\boldsymbol{y}$, with parameters $\hat{\boldsymbol{\gamma}}$ learned by minimizing ℓ;
- the **correction routine** $\mathcal{C}^h$: A routine correcting $\mathcal{M}^p$ to produce corrected predictions $\hat{\boldsymbol{y}}^c$;
- the **evaluation model** $\mathcal{M}^e : \mathbb{R}^{n\times p} \times \mathbb{R}^n \to \mathbb{R}^p$, $(\boldsymbol{X}, \hat{\boldsymbol{y}}^c) \mapsto \hat{\boldsymbol{\beta}}^c$, the model minimizing $\ell(\hat{\boldsymbol{y}}^c, h(\boldsymbol{X}\hat{\boldsymbol{\beta}}^c))$ to check whether the influence of $\boldsymbol{X}$ on the corrected predictions $\hat{\boldsymbol{y}}^c$ from $\mathcal{M}^p$ has been successfully removed by $\mathcal{C}^h$.

The previous definition assumes that the activation function h is the same for all three components $\mathcal{M}^p$, $\mathcal{M}^e$, and $\mathcal{C}^h$.

Before introducing our new orthogonalization algorithm when the activation function h is used, we more formally define what we mean by "correcting for the influence of $\boldsymbol{X}$":

Definition 2. Given the setup and models in Definition 1, we say the model predictions $\hat{\boldsymbol{y}}$ of $\mathcal{M}^p$ are **corrected** (or **orthogonalized**) if $\mathcal{M}^e$ yields coefficient values $\hat{\boldsymbol{\beta}}^c = \boldsymbol{0}$. $\hat{\boldsymbol{y}}^c$ are then called **corrected predictions**.

While these definitions imply a similar goal as pursued in fair machine learning, our approach uses a slightly different measure of "unbiasedness" that coincides with the previously introduced normalization (Lu et al., 2021) as well as the orthogonalization for identifiability (Rügamer et al., 2023). In contrast, in fair machine learning, the level of unfairness is, e.g., defined as the proportion of the variance of $\hat{\boldsymbol{y}}$ that can be explained by $\boldsymbol{X}$ (see, e.g., Scutari et al., 2022, or further explanations in Appendix B). Using this notion, a model adhering to Definition 2 is perfectly fair, but the opposite is not always true. This is because fairness is achieved using a criterion different from the one in Definition 2. In addition, these approaches often do not fully enforce fairness to preserve other model properties, in most cases only use the L_2 loss for ℓ and assume a linear relationship between $\boldsymbol{X}$ and $\boldsymbol{y}$.

3.2. Orthogonalization for GLM-type Routines

We first extend the classical orthogonalization of Section 2 motivated by linear models to GLMs. This model class is described by a distributional assumption inducing a loss function ℓ defined by the corresponding negative log-likelihood and strictly monotone activation function h (equal to the so-called inverse link function in GLMs). The implied loss function comprises many common machine learning setups: binary classification using sigmoid activation, multi-class classification using a softmax-type activation, or, e.g., a count regression using the exp-activation function. When applying Definition 2 to these GLMs using their canonical link function, we say that the correction of a model $\mathcal{M}^p$ yielding $\hat{\boldsymbol{y}}^c$ is successful if the evaluation model $\mathcal{M}^e$ defined by the minimizer

$$\hat{\boldsymbol{\beta}}^c = \arg\min_{\boldsymbol{\beta}^c} \ell(\hat{\boldsymbol{y}}^c, h(\boldsymbol{X}\boldsymbol{\beta}^c)) \tag{1}$$

is a null model, i.e., $\hat{\boldsymbol{\beta}}^c \equiv \mathbf{0}$. Note that we here assume a model without intercept for better readability[1]. While the solution of (1) is known not to have an analytical solution in general (see, e.g., Nelder & Wedderburn, 1972), the GLM poses a convex optimization problem and can be solved efficiently with iteratively reweighted least squares or Newton-type procedures (Gill et al., 2019).

Importantly, the iterates of these Newton-type optimization procedures can be represented as $\hat{\boldsymbol{\beta}}^{[t]} = (\boldsymbol{X}^\top \boldsymbol{\Upsilon}^{[t]} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{\Upsilon}^{[t]} \boldsymbol{r}^{[t-1]}$, $t \in \mathbb{N}$, where the weights $\boldsymbol{\Upsilon}^{[t]} \in \mathbb{R}^{n \times n}$ are given by a diagonal matrix and $\boldsymbol{r}^{[t]} \in \mathbb{R}^n$ is a working response (see Appendix D for details). As both $\boldsymbol{\Upsilon}^{[t]}$ and $\boldsymbol{r}^{[t-1]}$ are fixed after fitting the evaluation model, we can make use of the approximate linearity of both $\mathcal{M}^p$ and $\mathcal{M}^e$ at convergence to derive our generalized orthogonalization. A helpful intermediate result is the following:

Lemma 3. *Given any GLM model $\mathcal{M}^p$ with predictions $\hat{\boldsymbol{y}}$, and GLM model $\mathcal{M}^e$ with features $\boldsymbol{X}$,*

$$\hat{\boldsymbol{y}}^c = \mathcal{P}_{\boldsymbol{X}}^{\perp} \hat{\boldsymbol{y}} + h(0) \mathbf{1}_n \tag{2}$$

defines corrected predictions yielding $\hat{\boldsymbol{\beta}}^c = \mathbf{0}$.

Lemma 3 is an interesting and important finding to construct a correction, stating that despite the non-linearity of $\mathcal{M}^e$, it is sufficient to (linearly) orthogonalize predictions of $\mathcal{M}^p$ using the orthogonal projection $\mathcal{P}_{\boldsymbol{X}}^{\perp}$. The derivation and proof can be found in Appendix E.1. However, as we discuss in the following paragraph, it is not straightforward to relate this finding to a corrected GLM model $\mathcal{M}^p$.

Deriving corrected coefficients In order to derive a corrected GLM routine for corrected coefficients $\hat{\gamma}^c$ that are consistent with $\hat{\boldsymbol{y}}^c$, a naïve approach is to simply invert their definition $\hat{\boldsymbol{y}}^c = h(\boldsymbol{Z}\hat{\gamma}^c)$, yielding

$$\hat{\gamma}^c = \boldsymbol{Z}^\dagger h^{-1}(\hat{\boldsymbol{y}}^c) = \boldsymbol{Z}^\dagger h^{-1}(\mathcal{P}_{\boldsymbol{X}}^{\perp} \hat{\boldsymbol{y}}),$$

where $\boldsymbol{Z}^\dagger = (\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top$ is the pseudo-inverse of $\boldsymbol{Z}$. However, as $\text{dom}(\mathcal{P}_{\boldsymbol{X}}^{\perp} \hat{\boldsymbol{y}})$ is not necessarily a subset of $\text{dom}(\hat{\boldsymbol{y}})$, h^{-1} might not be defined on this new domain. In classification tasks, for example, this could lead to predicted probabilities $\hat{\boldsymbol{y}}^c \notin (0, 1)$ for which the inverse sigmoid function is not defined. Since this problem will occur whenever non-linear functions h or h^{-1} are employed, we instead have to adapt the GLM optimization routine as described in the following.

Corollary 4. *Given activation function h, prediction model $\mathcal{M}^p$ and evaluation model $\mathcal{M}^e$ with features $\boldsymbol{X}$, the solution γ^c to the optimization problem*

$$\begin{aligned} &\arg\min_{\gamma} \ell(\boldsymbol{y}, h(\boldsymbol{Z}\gamma)) \\ &s.t. \ ||(\boldsymbol{X} - n^{-1} \mathbf{1}_n \mathbf{1}_n^\top \boldsymbol{X})^\top h(\boldsymbol{Z}\gamma)||_2^2 = 0 \end{aligned} \tag{3}$$

produces corrected coefficients satisfying $\hat{\boldsymbol{\beta}}^c = \mathbf{0}$.

The mathematical details and derivations are given in the Appendix E.2. While various routines exist to solve (3), the problem turns out to be hard in practice, in particular in applications that are concerned with fairness, such as the ones addressed in Section 4.1. We found the *modified differential multiplier method* (Platt & Barr, 1987) to be very effective and describe its details in Appendix E.2. Further, note that in the case of an identity activation function h, the derived optimization problem in Corollary 4 is also directly solved by using $\mathcal{C}^h = \mathcal{C}^l$.

3.3. Orthogonalization with Piece-wise Linear Activations

While the previous orthogonalization can be used in many different situations, intermediate network layers are often specified with piece-wise linear activation functions not comprised by the GLM framework, most notably the ReLU function. As for canonical activation functions in GLMs, the ReLU activation $\text{ReLU}(\boldsymbol{X}\boldsymbol{\beta}) = \boldsymbol{X}\boldsymbol{\beta} \circ \mathbb{1}(\boldsymbol{X}\boldsymbol{\beta} > 0)$ is applied element-wise, with $\circ$ denoting the Hadamard product and $\mathbb{1}(\cdot)$ an indicator function individually evaluated for all entries of the vector. While at first glance the definition of the ReLU activation might suggest that $\text{ReLU}(\boldsymbol{X}\boldsymbol{\beta}) \in \text{span}(\boldsymbol{X})$ and hence $\mathcal{C}^l$ from Section 3.1 is sufficient for successful orthogonalization, this is not the case as can be verified with a simple counterexample.

Example 2. Assume $\boldsymbol{X} = (1, -1)^\top \in \mathbb{R}^2$ and $\beta = 1$. Then $\text{ReLU}(\boldsymbol{X}\beta) = (1, 0) \notin \text{span}(\boldsymbol{X})$.

Although $\text{ReLU}(\boldsymbol{X}\boldsymbol{\beta})$ does not necessarily lie in $\text{span}(\boldsymbol{X})$, we can derive the following result

Theorem 5. *Given a prediction model $\mathcal{M}^p$ with ReLU activation, and an evaluation model $\mathcal{M}^e$ with ReLU activation and L_2 loss, a $\mathcal{C}^h$-orthogonalization of the form $\boldsymbol{Z}^c = \mathcal{P}_{\boldsymbol{X}}^{\perp} \boldsymbol{Z}$, will yield $\hat{\boldsymbol{\beta}}^c = \mathbf{0}$.*

The proof of Theorem 5 is given in the Appendix.

3.4. Orthogonalization for Tensor Predictions

We now show that our concept of orthogonalization is not restricted to regression models, but can also be used to remove information from learned representations in layers of neural networks. Such a correction should be able to deal with prediction models (or layers) with tensor-shaped predictions (e.g., present in early layers of convolutional networks) and we show how to adapt previous results accordingly. We first define the linear evaluation model $\mathcal{M}^e$ with L_2-loss ℓ as the solution of the following tensor-on-vector regression:

$$\hat{\mathfrak{B}}^c = \arg\min_{\mathfrak{B}^c} \ell(\hat{\mathfrak{Y}}^c, \boldsymbol{X} \times_1 \mathfrak{B}^c), \tag{4}$$

[1]The intercept does not necessarily have to be equal to zero, as it does not use any information from the protected features

where the outcome tensor $\hat{\mathfrak{Y}}^c \in \mathbb{R}^{n \times d_1 \times \cdots \times d_R}$ is the corrected version of an (intermediate) non-linear prediction $\hat{\mathfrak{Y}}$, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is a feature matrix as before, $\mathfrak{B}^c \in \mathbb{R}^{p \times d_1 \times \cdots \times d_R}$ a tensor-valued regression coefficient, and $\times_1$ denotes the 1-mode product. For a successful correction $\mathcal{C}^h$ from which we obtain $\hat{\mathfrak{Y}}^c$, it must hold $\hat{\mathfrak{B}}^c \equiv \mathbf{0}$ in (4), which can be achieved by projecting $\text{vec}(\hat{\mathfrak{Y}})$ onto $\text{span}(\boldsymbol{X})^\perp$ as before. Indeed, using a vectorized reformulation, we can prove that this yields corrected predictions (see Appendix E.4 for details):

Corollary 6. *Given the set-up around the evaluation model defined by* (4)*, the corrected predictions satisfying* $\hat{\mathfrak{B}}^c \equiv \mathbf{0}$ *are given by*

$$\hat{\mathfrak{Y}}^c = \mathcal{P}_{\boldsymbol{X}}^{\perp} \times_1 \hat{\mathfrak{Y}}. \tag{5}$$

While the result in (5) can be extended to GLM-type non-linearities, and hence an orthogonalization as the one in Section 3.2 can be derived, we here focus on the ReLU activation function, arguably the pervasive type of non-linearity in intermediate DNN layers. Hence, we consider an evaluation model $\mathcal{M}^e$ with ReLU activation defined by

$$\hat{\mathfrak{B}}^c = \arg\min_{\mathfrak{B}^c} \ell(\hat{\mathfrak{Y}}^c, \text{ReLU}(\boldsymbol{X} \times_1 \mathfrak{B}^c)), \tag{6}$$

and a prediction model $\mathcal{M}^p$ yielding predictions $\hat{\mathfrak{Y}} = \text{ReLU}(\bar{\mathfrak{Y}})$ with pre-activation $\bar{\mathfrak{Y}}$. In contrast to previous sections, we here focus on correcting the predictions and not the effects of $\mathcal{M}^p$, as predictions of an intermediate layer in a large neural network are not necessarily formed by a linear operation. We obtain the following result:

Corollary 7. *Assuming the evaluation model* (6)*, the corrected pre-activations* $\bar{\mathfrak{Y}}^c$ *yielding corrected predictions* $\hat{\mathfrak{Y}}^c$ *following ReLU activation are given by* $\bar{\mathfrak{Y}}^c = \mathcal{P}_{\boldsymbol{X}}^{\perp} \times_1 \bar{\mathfrak{Y}}$.

See Appendix E.5 for a proof. In other words, by multiplying the first dimension of a non-activated output of some layer with $\mathcal{P}_{\boldsymbol{X}}^{\perp}$ from the left before applying the ReLU activation, we obtain corrected predictions in the next layer.

4. Numerical Experiments

In Section 4.1, we first check whether sensitive information can be successfully protected in GLMs with our approach, using both synthetic and real-world datasets. We then show in various settings how our method corrects learned representations for potential biases (Section 4.2), and finally investigate its effectiveness as an online version applied during training (Section 4.3). The details of all experiments are provided in Appendix F. In all experiments, categorical information in the protected features is included in $\boldsymbol{Z}$ as one-hot encoded features with the first column removed (as we don't want to orthogonalize or penalize the model for a constant intercept term).

4.1. Generalized Linear Model

In the following, we focus on real-world applications but provide synthetic data examples in Appendix G.1, confirming that $\mathcal{C}^h$ works as intended for GLMs in a variety of settings.

4.1.1. Adult Income Data

Using the adult income data also investigated in Xu et al. (2022) to analyze algorithm fairness, we check the efficacy of our approach for protected features sex and race. We therefore first fit our prediction model, a GLM with features age, work class, education, marital status, relationship, and working hours per week, to predict the person's income binarized into $\leq$ 50k and $>$ 50k (as provided by the original dataset) and then correct for the protected features.

Results We find (Table 2, first row) that the income predictions are significantly influenced if the individual is white and/or male with a multiplicative increase in the odds-ratio of approximately exp(0.522) = 1.68 and exp(1.066) = 2.90 for earning more than 50k for white individuals and males, respectively. Similar results are inferred from Xu et al. (2022)'s approach. When applying our framework and adjusting for sex and race, we can see that all effects of these protected features are estimated to be close to zero and are non-significant.

Table 2. Estimated coefficients from $\mathcal{M}^e$ with corresponding p-values in brackets when not correcting the model (first column), when using Xu et al. (2022) (second column), and when correcting with $\mathcal{C}^h$ (last column). P-values smaller than 0.05 show a ✗-sign, otherwise a ✓-sign.

	w/o correction	Xu et al. (2022)	using $\mathcal{C}^h$
Sex (male)	1.066 (< 2e-16) ✗	1.255 (< 2e-16) ✗	-0.000 (0.998) ✓
Race (Asian)	0.263 (0.227) ✓	0.859 (2e-4) ✗	0.011 (0.954) ✓
Race (Black)	-0.094 (0.577) ✓	0.191 (0.319) ✓	0.008 (0.957) ✓
Race (Other)	-0.165 (0.594) ✓	-0.131 (0.706) ✓	-0.131 (0.609) ✓
Race (White)	0.522 (0.001) ✗	0.928 (4e-7) ✗	0.005 (0.969) ✓

4.1.2. Fairness Benchmark

While fairness methods target a slightly different goal than orthogonalization, it might still be interesting to see whether our criterion for orthogonalization (cf. Definition 2) can also be met by approaches suggested in the fairness literature. We therefore run a comprehensive analysis on fairness datasets with non-Gaussian outcomes and compare the results of various fairness methods with our orthogonalization. In the case of GLMs, closely related methods are fair (logistic / Poisson) regression with different optimization strategies. Komiyama et al. (2018) bounds the proportion of explained variance by protected features (we refer to this as *Komiyama1*). Another variant by the same authors first regresses the fitted values against the protected features (re-

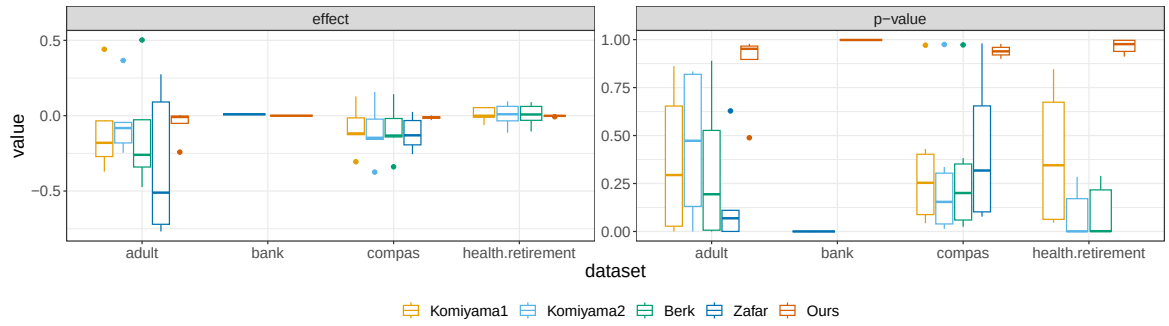

Figure 3. Comparison of orthogonalization properties of fairness methods in comparison with our approach. Ideally, methods should yield effects of zero (in the left plot) and large p-values (in the right plot). Missing boxes indicate that the method did not converge.

ferred to as *Komiyama2*). An alternative is the approach by Berk et al. (2017) enforcing individual fairness by penalizing pairs of observations that have different outcomes for the same protected features (referred to as *Berk*). A fourth approach we compare against is the one of Zafar et al. (2019) (referred to as *Zafar*) enforcing fairness by bounding the covariance of protected features and other predictors. We run these and our method on the benchmark datasets `adult`, `bank`, `compas`, and `health.retirement` provided in Scutari (2023) containing various protected features such as race, sex, or marriage.

Results The results are depicted in Figure 3 showing that all fairness approaches show significant feature effects with effect strengths often different from 0. In contrast, our approach adheres to the orthogonality requirement from Definition 2, yielding effects close to 0 with no significant impact. As stated earlier, this does not imply unfairness of the other approaches, but demonstrates that fairness approaches cannot be simply used as a substitute to achieve orthogonality. As orthogonality might potentially decrease model performance, Table 3 additionally compares the model performances with and without the correction. Since most of the datasets contain the protected features also in $\boldsymbol{Z}$, i.e., as features for the prediction model, the prediction performance with orthogonalization will naturally decrease (if not, then the protected features would likely not bias the model and raise fairness issues in the first place). However, as can be seen in Table 3, the decrease in performance is often negligible with a maximum of 2% decrease in accuracy for compas.

Table 3. Test set prediction performances with/out correction.

Dataset (Metric)	w/o Corr.	w/ Corr.
adult (Accuracy)	0.854	0.833
bank (Accuracy)	0.899	0.887
compas (Accuracy)	0.739	0.724
health (Root mean squared error)	0.750	0.789

4.2. Orthogonalization of Learned Representations

4.2.1. Comparing $\mathcal{C}^l$ and $\mathcal{C}^h$ on Image Embeddings

Going back to our instructive Example 1, we use self-supervised embeddings for the MIMIC Chest X-Ray dataset (Johnson et al., 2019; Sellergren et al., 2022) known to have encoded race, sex and age (Glocker et al., 2023b), and try to remove this implicit information using orthogonalization. Following Glocker et al. (2023b), we first validate that meta-information such as race, in fact, is influencing predictions in a downstream task. As $\mathcal{M}^p$ we use a GLM to predict whether the patient suffers from pleural effusion using only the embedding as features. We then check whether a $\mathcal{C}^l$-orthogonalization is sufficient to remove this information, but in contrast to the Adult data application, check on the level of probabilities where $\mathcal{C}^l$ should fail. Finally, we apply $\mathcal{C}^h$-orthogonalization to obtain the corrected predictions.

Results Table 1 (page 2) summarizes the results, confirming our hypotheses and demonstrating that our approach effectively removes the meta-information from embeddings. In this case, removing the features even results in a slight improvement in prediction performance with a 0.02 increase in AUC.

4.2.2. Post-hoc Orthogonalization for Neural Representations

As another demonstration, we analyze the race and age bias in a learned neural representation of faces from the UTKFace dataset (Zhang et al., 2017), where previous literature suggests, among other biases, a misalignment for black women (see, e.g., Krishnan et al., 2020). We first train a ResNet50 (He et al., 2016) on the provided images to classify the sex of the depicted person. We then check the influence of race and age on the predictions using a logistic regression. In order to correct for biases, we extract the weights in the ResNet's penultimate layer and apply the $\mathcal{C}^h$-orthogonalization by fitting a corrected GLM (following Corollary 4) with learned representations as features and sex

as the binary outcome. Finally, we again check the influence of race and age on the model's predictions.

Results Results are summarized in Table 4, again depicting coefficients and p-values, and confirm the effectiveness of our approach in blinding the network for certain features (here age and race). The orthogonalization in this case comes at the cost of an AUC reduction from 0.831 (w/o correction) to 0.826 (with $\mathcal{C}^h$), which seems to be a reasonable trade-off to correct for age and race effects.

Table 4. Estimated coefficients from $\mathcal{M}^e$ with corresponding p-values in brackets when not correcting the model (first column) and when correcting the model for age and race (second column). Significant influences are highlighted in bold for an α-level of 0.05.

	w/o correction	$\mathcal{C}^h$ orthogonalization
Age	-0.017 (< 2e-16) ✗	-0.000 (0.976) ✓
Race (Black)	-0.352 (8.68e-10) ✗	0.009 (0.872) ✓
Race (Indian)	-0.620 (< 2e-16) ✗	0.008 (0.894) ✓
Race (Other)	-0.315 (4.05e-05) ✗	-0.002 (0.979) ✓
Race (White)	-0.377 (5.65e-13) ✗	-0.008 (0.872) ✓

4.2.3. Word Embeddings

As a last example, we demonstrate that our approach is not limited to tabular and image data. As, e.g., discussed in Bolukbasi et al. (2016), text embeddings, as well, can have biases encoded, and addressing this property seems of great relevance given the rise of GPT(-like) models which have also been shown to not always be neutral in their generation (see, e.g., Ray, 2023). To show that the proposed method also works for the correction of text embeddings in a non-linear model, we use the movies review dataset (Maas et al., 2011), which we enhance with the gender information of the protagonist and the movie director. Using an LSTM model and a learned text embedding $\boldsymbol{Z}$ (see Appendix F for details), we try to predict the number of reviews for each movie (i.e., a Poisson-distributed outcome). We do this once without correction and once by correcting the embeddings $\boldsymbol{Z}$ of size 100×100 of every movie description following the results from Corollary 7 using the available gender information as $\boldsymbol{X}$.

Results When not correcting the model's prediction, we obtain p-values <2e-16 for both gender effects, suggesting a significant influence of gender on the predictions $\hat{\boldsymbol{y}}$. After correction, p-values are 0.993 and 1.0, implying a successful embedding correction with only a small increase in root mean squared error of 1% compared to the model without correction.

4.3. Online Orthogonalization

Finally, we test how our approach performs when applied in an online fashion by iteratively optimizing a small CNN and orthogonalizing the network with respect to some fixed features $\boldsymbol{X}$ as proposed in the Metadata Normalization approach (Lu et al., 2021). In other words, we perform the proposed orthogonalization during training. Inspired by Clever Hans predictors (Lapuschkin et al., 2019) and orthogonalization in distribution shift situations (Chen et al., 2023), we colorize the MNIST data by reshaping the grayscale image to an RGB image where one of the three channels is filled with the original grayscale pixel values and the remaining two channels filled with zeros, i.e., set to black. The binary prediction task is to classify a subset of the MNIST dataset reduced to digits 0 and 9. For the train set, we intentionally create only red images of 0s by assigning the pixel values to the first channel. For the images of 9s we randomly assign either the second (green) or third (blue) channel. In the counterfactual test set, images of both classes are colored randomly in green or blue, but not red. As a result, the color red is predictive for the train but not the test set. We then use the color information "red", denoted as the binary feature $\boldsymbol{X}$, in the correction model to normalize the CNN's predictions. In other words, we train a CNN on RGB images but blind the network w.r.t. the color red (or in fact any) color information by orthogonalizing the first convolutional layer's latent features w.r.t. $\boldsymbol{X}$.

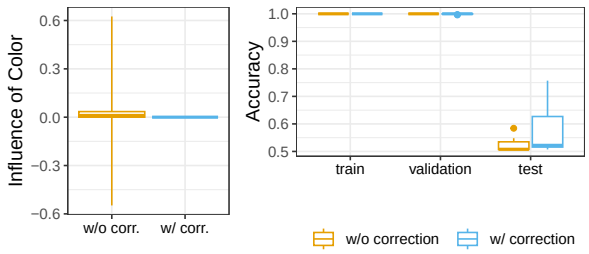

Figure 4. Estimated influence of the color red on the convolutional layer's predictions (left plot) and train-, validation- and test-accuracy of the two models (with/out correction).

Results Figure 4 summarizes the results of the experiment showing that our method correcting for the color red in fact yields no influence of the metadata information on the hidden layer's weights (left plot) and further illustrates how the model without correction is fooled by the training/validation data by focusing on the information of the color channels instead of the pixel values. This leads to a poor test performance of not much more than 50% accuracy. The corrected model shows a better test performance as it learned to focus on the image content and not the channel information.

5. Discussion

In this work, we introduced a novel correction routine that extends the concept of orthogonalization to non-linear and

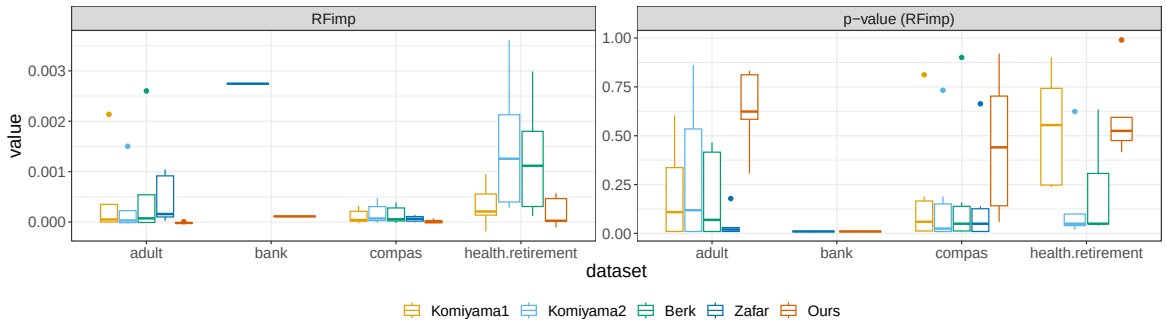

Figure 5. Comparison of orthogonalization properties of fairness methods in comparison with our approach when extending the notion of orthogonality beyond linear feature effects. Ideally, methods should yield random forest feature importance (RFimp) of zero (in the left plot) and large p-values (in the right plot). Missing boxes indicate that the method did not converge.

tensor-valued models. Our experimental results confirm that our approach is effective in correcting sensitive data in linear models and neural networks, as well as rectifying pre-existing embeddings for undesired attributes.

Caveat We want to emphasize that the given examples and results in this paper are limited by the representativeness of the datasets used and should not be the sole basis for decision-making in applications involving medical imaging or text applications. Furthermore, as "protection of information" can be defined in various ways, our presented methods do not offer guarantees of absolute protection for sensitive information in every situation. In particular, note that orthogonalization does not imply stochastic independence of the involved random variables, which means that some form of residual dependence could persist in theory.

Limitations and future research The proposed algorithms require the number of observations n to be larger than the number of features p (or intermediate layer predictions in a network), and that there are more features q in the prediction model than in the evaluation model (p). While this is usually the case, extensions to applications for $n < p$ or $q < p$ are a challenging yet interesting future research direction.

The proposed orthogonalization also requires knowledge of features $\boldsymbol{Z}$ related to protected information. Performing such an operation to remove implicit biases is another interesting approach brought up by an anonymous reviewer but would require simultaneously learning the protected feature and being able to evaluate whether these present potential biases.

6. Outlook

A possible extension suggested by one of the anonymous reviewers is to make the class of evaluation models more flexible, allowing it to go beyond linear effects of protected features on the additive predictor scale, i.e., fitting an evaluation model $h(g(X))$ where g is, e.g., a non-linear function. This would yield a more general notion of orthogonalization than presented in this paper but is also more challenging to analyze theoretically. In order to check how robust existing approaches and ours are w.r.t. non-linear evaluation models, we run the fairness benchmark again and use a random forest as an evaluation model to explain the prediction model's output using protected information. We then extract the importance of each feature together with a p-value for the importance values using a permutation test. Results are depicted in Figure 5, showing a very similar pattern to Figure 3. Interestingly, our method seems to work well in terms of this new orthogonalization notion, only yielding significant feature importances for the bank dataset. Results for all other methods suggest that protected features play a significant role in the random forest's explanation of the prediction model's output.

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Baumann, P. F., Hothorn, T., and Rügamer, D. Deep conditional transformation models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 3–18. Springer, 2021.

Berk, R., Heidari, H., Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman

is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K. R. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

Chen, A. S., Lee, Y., Setlur, A., Levine, S., and Finn, C. Project and probe: Sample-efficient domain adaptation by interpolating orthogonal features. *arXiv preprint arXiv:2302.05441*, 2023.

Chen, Y., Li, Y., Weller, A., et al. Scalable infomin learning. *Advances in Neural Information Processing Systems*, 35: 2226–2239, 2022.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters, 2018.

Do, H., Putzel, P., Martin, A. S., Smyth, P., and Zhong, J. Fair generalized linear models with a convex penalty. In *International Conference on Machine Learning*, pp. 5286–5308. PMLR, 2022.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Gill, J., Gill, J. M., Torres, M., and Pacheco, S. M. T. *Generalized linear models: a unified approach*, volume 134. Sage Publications, 2019.

Glocker, B., Jones, C., Bernhardt, M., and Winzeck, S. Algorithmic encoding of protected characteristics in chest x-ray disease detection models. *Ebiomedicine*, 89, 2023a.

Glocker, B., Jones, C., Roschewitz, M., and Winzeck, S. Risk of bias in chest radiography deep learning foundation models. *Radiology: Artificial Intelligence*, 5(6), 2023b.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Y., Burghardt, K., and Lerman, K. Learning fair and interpretable representations via linear orthogonalization. *arXiv preprint arXiv:1910.12854*, 2019.

Huang, L., Liu, L., Zhu, F., Wan, D., Yuan, Z., Li, B., and Shao, L. Controllable orthogonalization in training dnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6429–6438, 2020.

Johnson, A. E., Pollard, T. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Peng, Y., Lu, Z., Mark, R. G., Berkowitz, S. J., and Horng, S. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Kaiser, P., Kern, C., and Rügamer, D. Uncertainty-aware predictive modeling for fair data-driven decisions. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022.

Komiyama, J., Takeda, A., Honda, J., and Shimao, H. Nonconvex optimization for regression with fairness constraints. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2737–2746. PMLR, 10–15 Jul 2018.

Kopper, P., Pölsterl, S., Wachinger, C., Bischl, B., Bender, A., and Rügamer, D. Semi-structured deep piecewise exponential models. In *Survival Prediction-Algorithms, Challenges and Applications*, pp. 40–53. PMLR, 2021.

Krishnan, A., Almadan, A., and Rattani, A. Understanding fairness of gender classification algorithms across gender-race groups. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1028–1035. IEEE, 2020.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

Lu, M., Zhao, Q., Zhang, J., Pohl, K. M., Fei-Fei, L., Niebles, J. C., and Adeli, E. Metadata normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10917–10927, 2021.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

Nelder, J. A. and Wedderburn, R. W. Generalized linear models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 135(3):370–384, 1972.

Palowitch, J. and Perozzi, B. Debiasing graph representations via metadata-orthogonal training. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 435–442, 2020. doi: 10.1109/ASONAM49781.2020.9381348.

Pérez-Suay, A., Laparra, V., Mateo-García, G., Muñoz-Marí, J., Gómez-Chova, L., and Camps-Valls, G. Fair kernel learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 339–355. Springer, 2017.

Pessach, D. and Shmueli, E. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

Platt, J. and Barr, A. Constrained differential optimization. In *Neural Information Processing Systems*, 1987.

Ray, P. P. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 2023.

Rügamer, D. A new PHO-rmula for improved performance of semi-structured networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29291–29305. PMLR, 23–29 Jul 2023.

Rügamer, D., Kolb, C., and Klein, N. Semi-structured distributional regression. *The American Statistician*, pp. 1–12, 2023.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Sattigeri, P., Ghosh, S., Padhi, I., Dognin, P., and Varshney, K. R. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. *Advances in Neural Information Processing Systems*, 35: 35894–35906, 2022.

Scutari, M. fairml: A statistician's take on fair machine learning modelling. *arXiv preprint arXiv:2305.02009*, 2023.

Scutari, M., Panero, F., and Proissl, M. Achieving fairness with a simple ridge penalty. *Statistics and Computing*, 32 (5):77, 2022.

Sellergren, A. B., Chen, C., Nabulsi, Z., Li, Y., Maschinot, A., Sarna, A., Huang, J., Lau, C., Kalidindi, S. R., Etemadi, M., et al. Simplified transfer learning for chest radiography models using less data. *Radiology*, 305(2): 454–465, 2022.

Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., and Sarawagi, S. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

Vento, A., Zhao, Q., Paul, R., Pohl, K. M., and Adeli, E. A penalty approach for normalizing feature distributions to build confounder-free models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 387–397. Springer, 2022.

Weber, T., Ingrisch, M., Bischl, B., and Rügamer, D. Unreading race: Purging protected features from chest x-ray embeddings. *arXiv preprint arXiv:2311.01349*, 2023.

Wood, S. N. *Generalized additive models: an introduction with R*. CRC press, 2017.

Xu, Y., He, H., Shen, T., and Jaakkola, T. S. Controlling directions orthogonal to a classifier. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=DIjCrlsu6Z.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

A. Simplified Problem Statement

To illustrate our problem, consider two vectors $x = (0, 1)$ and $y = (1, 0)$, which are orthogonal as $x^\top y = 0$. However, after transformation $h(x)$ and $h(y)$ are not orthogonal anymore for non-linear transformations h as depicted in Figure 6.

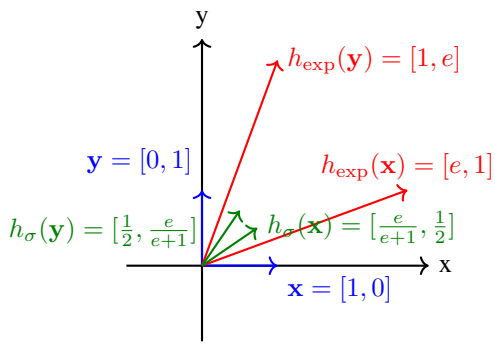

Figure 6. Graphical depiction of orthogonality without transformation (blue vectors) and after transformation with $h_{\exp}$ (red) or the sigmoid function h_σ (green).

B. Connection to Fair Machine Learning

When comparing the original orthogonalization approach in Lu et al. (2021); Rügamer et al. (2023) and Komiyama et al. (2018), both approaches use corrected features

$$\boldsymbol{Z}^c = \mathcal{P}_{\boldsymbol{X}}^{\perp} \boldsymbol{Z} \tag{7}$$

in their model equation. The main difference is that fair machine learning approaches incorporate the protected features into the analysis as well, i.e., estimate the model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}^c\boldsymbol{\gamma}^c + \boldsymbol{\varepsilon} \tag{8}$$

with residual term $\boldsymbol{\varepsilon}$, whereas the prediction model $\mathcal{M}^p$ in the orthogonalization approach does not necessarily include the features $\boldsymbol{X}$ in the model. If $\boldsymbol{X}$ is incorporated and without further constraints on the model, the approach of Komiyama et al. (2018) and the orthogonalization would lead to the same effect estimates (e.g., when using orthogonalization for identifiability as in Rügamer, 2023). Fair machine learning, however, offers a trade-off between model performance and (un-)fairness by regularizing the effects $\boldsymbol{\beta}$ of $\boldsymbol{X}$ in (8) using an explained variance formulation:

$$\min_{\boldsymbol{\beta}, \boldsymbol{\gamma}^c} \mathbb{E}[(\boldsymbol{y} - \hat{\boldsymbol{y}})^2] \quad s.t. \ \frac{\text{Var}(\boldsymbol{X}\boldsymbol{\beta})}{\text{Var}(\hat{\boldsymbol{y}})} \leq r \tag{9}$$

with predictions $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{Z}^c\hat{\boldsymbol{\gamma}}^c$ and some constraint $r \in [0, 1]$. This approach and reformulations, such as the one of Scutari et al. (2022) as a ridge-penalized optimization problem, hence do not provide the full unpenalized effects $\boldsymbol{\beta}$ as obtained by the orthogonalization.

Apart from the difference between the two approaches induced by the regularization, the generalization of fair machine learning approaches to models with activation functions differs further. In particular, as, e.g, suggested by Scutari et al. (2022) the orthogonal predictors $\boldsymbol{Z}^c$ are only used on the pre-activation scale, i.e., the fair model uses $\boldsymbol{Z}^c$ in the GLM by incorporating it only once in the additive predictor: $\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{Z}) = h(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}^c\boldsymbol{\gamma}^c)$. As discussed in the main part of our paper and in the previous section, this does not imply orthogonality of $h(\boldsymbol{X}\boldsymbol{\beta})$ and $h(\boldsymbol{Z}\boldsymbol{\gamma})$ as required in our work.

A slightly different approach is used by Xu et al. (2022) who correct a classifier trained with all variables by another classifier trained only on the protected variables. Their approach, however, does not debias representations but only finds directions for predictions that are orthogonal.

C. Other Notions of Orthogonality

In the following, we briefly describe connections to other notions of orthogonality that are used in different fields of machine learning research.

C.1. Domain Generalization

The objective of the domain generalization (DG) problem is to train models to be robust against (or orthogonal to) domain (or covariate) shifts (Ganin et al., 2016; Shankar et al., 2018). While our orthogonalization approach and DG have different goals, viewing orthogonalization through the lens of DG is an interesting idea brought up by an anonymous reviewer. In particular, distributionally robust optimization (DRO; Sagawa et al., 2019) aims to ensure that predictive accuracy is robust w.r.t. changes in the distribution of the sensitive features. Our orthogonalization approach, in contrast, aims to ensure that the sensitive features do not affect the predictions (in the sense of a zero coefficient in the evaluation model). Note, however, that DRO does not provide any guarantees regarding our goal.

C.2. Orthogonality of Predictions and Embeddings

Another interesting question brought up by an anonymous reviewer is how our notion of orthogonality relates to the following two concepts:

- **Concept 1**: requiring that the predictions do not change when the sensitive attributes are changed;
- **Concept 2**: requiring that the embeddings are free from the influence of sensitive information.

Concept 2 is a special case of our proposed notion if the model is linear. Concept 1 can also coincide with our definition if sensitive attributes are part of the prediction model. In our setting, however, this is not necessarily the case. Other (fairness) approaches aligning with concept 1 assume knowledge of the causal relationship between sensitive and non-sensitive attributes, which we also do not require.

D. Background on GLMs

We briefly introduce several quantities used in the estimation of GLMs, but refer to Wood (2017) for more detailed explanations and derivations. Let $g = h^{-1}$ be the link function. The weights $\boldsymbol{\Upsilon}$ are defined as

$$\boldsymbol{\Upsilon} = \text{diag}(\alpha(\mu_i)/\{g'(\mu_i)^2 V(\mu_i)\}) \tag{10}$$

where $\mu_i = h(\boldsymbol{x}_i^\top \boldsymbol{\beta})$ and

$$\alpha(\mu_i) = 1 + (y_i - \mu_i)\{V'(\mu_i)/V(\mu_i) + g''(\mu_i)/g'(\mu_i)\} \tag{11}$$

and

$$V(\mu_i) = \text{Var}(y)/\phi \tag{12}$$

with ϕ denoting the distribution's scale parameter. The working response $\boldsymbol{r}$ is defined as

$$\boldsymbol{r} = \boldsymbol{G}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{13}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$ and

$$\boldsymbol{G} = \text{diag}(g'(\mu_i)/\alpha(\mu_i)). \tag{14}$$

In particular, note that in the case of using *Fisher weights*, i.e., the expected Hessian matrix, $\alpha(\mu_i) = 1$ and we have the following simplification:

$$\boldsymbol{G}^{-1}\boldsymbol{\Upsilon}^{-1/2} = \text{diag}(V(\mu_i)^{1/2}), \tag{15}$$

i.e., a function of μ_i only. We can use this result in later derivations.

D.1. GLM in Detail

The above derivation for GLMs can be made more explicit for our purposes. As noted before, using the Fisher weights implies $\alpha(\mu_i) = 1$. As $\mu_i = h(x_i^\top \beta) \equiv h(0)$ in case of a successful correction, this implies

$$\boldsymbol{G} = \text{diag}(g'(\mu_i)) = (h^{-1})'(0) \cdot \boldsymbol{I}. \tag{16}$$

In other words, multiplying with $\boldsymbol{G}$ only scales all observations by a constant. Similarly, (15) reduces to a constant multiplicative factor of $V(h(0)^{1/2})$.

E. Proofs and Derivations

E.1. Proof of Lemma 3

To prove Lemma 3, we use the final iterate representation of weights $\hat{\boldsymbol{\beta}}^c$ in a GLM and note that the evaluation model (first without intercept) must yield

$$\hat{\boldsymbol{\beta}}^c = (\boldsymbol{X}^\top \boldsymbol{\Upsilon} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{\Upsilon} \boldsymbol{r} = (\boldsymbol{X}^\top \boldsymbol{\Upsilon} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{\Upsilon} \boldsymbol{G}(\hat{\boldsymbol{y}}^c - \boldsymbol{\mu}) \overset{!}{=} \mathbf{0}. \tag{17}$$

Consequently, it must also hold $\boldsymbol{\mu} = h(\boldsymbol{X}\hat{\boldsymbol{\beta}}^c) = h(\mathbf{0})$. The equality in (17) yields two possible solutions: either $\hat{\boldsymbol{y}}^c = \boldsymbol{\mu}$, which is undesirable as it implies that we cannot predict anything after removing the influence of $\boldsymbol{X}$, or we can construct $\hat{\boldsymbol{y}}^c$ so that when first subtracting $\boldsymbol{\mu}$ and then projecting it onto the space spanned by the columns of $\widetilde{\boldsymbol{X}} = \boldsymbol{\Upsilon}^{1/2}\boldsymbol{X}$, we obtain the zero vector (see details below). Solving for $\hat{\boldsymbol{y}}^c$, we get the following result.

Lemma 8. *Consider a prediction model $\mathcal{M}^p$ with activation function h using data $\boldsymbol{Z}$ to produce predictions $\hat{\boldsymbol{y}}$. Given weights $\boldsymbol{\Upsilon}$ and protected features $\boldsymbol{X}$, corrected predictions satisfying $\hat{\boldsymbol{\beta}}^c = \mathbf{0}$ are given by*

$$\hat{\boldsymbol{y}}^c = \boldsymbol{G}^{-1}\boldsymbol{\Upsilon}^{-1/2}\mathcal{P}^\perp_{\widetilde{\boldsymbol{X}}}\hat{\boldsymbol{y}} + h(\mathbf{0}). \tag{18}$$

We can easily confirm that this yields a desired solution by plugging in (18) into (17). Further note that $\boldsymbol{\mu} = h(0) \cdot \mathbf{1}_n$, i.e., shifts by a constant $c_2 := h(0)$ and the multiplication from the left with $\boldsymbol{G}^{-1}\boldsymbol{\Upsilon}^{-1/2}$ does only scale by a constant factor $c_1 := V(h(0))^{1/2}$ (see Section D.1) but not rotate $\mathcal{P}^\perp_{\widetilde{\boldsymbol{X}}}\hat{\boldsymbol{y}}$, i.e.,

$$\hat{\boldsymbol{y}}^c = \boldsymbol{G}^{-1}\boldsymbol{\Upsilon}^{-1/2}\mathcal{P}^\perp_{\widetilde{\boldsymbol{X}}}\hat{\boldsymbol{y}} + \boldsymbol{\mu} \overset{(15)}{=} V(h(0))^{1/2}\mathcal{P}^\perp_{\widetilde{\boldsymbol{X}}}\hat{\boldsymbol{y}} + h(\mathbf{0}) = c_1\mathcal{P}^\perp_{\widetilde{\boldsymbol{X}}}\hat{\boldsymbol{y}} + c_2\mathbf{1}_n \tag{19}$$

with constants c_1, c_2. Any multiplicative constant does not change the result. We can therefore further simplify the previous expression from (18) to

$$\hat{\boldsymbol{y}}^c = \mathcal{P}^\perp_{\boldsymbol{X}}\hat{\boldsymbol{y}} + c_2\mathbf{1}_n. \tag{20}$$

This is due to the fact that $\mathcal{P}^\perp_{\widetilde{\boldsymbol{X}}}$ is the projection matrix of $\widetilde{\boldsymbol{X}} = \boldsymbol{\Upsilon}^{1/2}\boldsymbol{X}$ with $\boldsymbol{\Upsilon}^{1/2}$ being $\boldsymbol{\Upsilon}^{1/2} = \text{diag}(1/(g'(\mu_i)\sqrt{V(\mu_i)})) \equiv c_3 \cdot \boldsymbol{I}$ for some constant c_3 as $\boldsymbol{\mu} = h(\mathbf{0}) = const. \cdot \mathbf{1}_n$, and hence both $g'(\mu_i)$ and $V(\mu_i)$ are constant. As a consequence, we have $\mathcal{P}_{\widetilde{\boldsymbol{X}}} = c_3\boldsymbol{X}(c_3^2\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}c_3 = \mathcal{P}_{\boldsymbol{X}}$. For a similar reason, we can ignore c_1 as $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top c_1\mathcal{P}^\perp_{\boldsymbol{X}}\hat{\boldsymbol{y}} = c_1(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\mathcal{P}^\perp_{\boldsymbol{X}}\hat{\boldsymbol{y}} = \mathbf{0}$.

Model with Intercept Similar results can be derived for a model with intercept. First, note that $\boldsymbol{\mu} = h(\beta_0\mathbf{1}_n)$. Hence, c_1 and c_2 are fixed constants $c_2 = h(\beta_0)$ and $c_1 = V(h(\beta_0))^{1/2}$. Hence, this yields again (20), now just with different constants c_1 and c_2. In an intercept-only GLM with canonical link function, the intercept can be computed using only the response values, allowing to derive the closed-form solution $\hat{\beta}_0 = h^{-1}(\bar{\hat{y}})$ with $\bar{\hat{y}} = n^{-1}\sum_i \hat{y}_i$. As the intercept in the evaluation model is of no particular interest, one can alternatively center the predictions $\hat{\boldsymbol{y}}$ and subsequently drop the constant c_2. We will assume centered predictions in the following and use the correction $\hat{\boldsymbol{y}}^c = \mathcal{P}^\perp_{\boldsymbol{X}}\hat{\boldsymbol{y}}$.

E.2. Details and Derivation of Corollary 4

In the following, we describe how to derive the constrained optimization problem and subsequently our preferred way of solving it.

Deriving the constrained optimization problem To see that we can write problem

$$\hat{\boldsymbol{\gamma}}^c = \arg\min_{\boldsymbol{\gamma}^c} \ell(\boldsymbol{y}, h(\boldsymbol{Z}\boldsymbol{\gamma}^c)) \; s.t. \; \hat{\boldsymbol{\beta}}^c = \mathbf{0} \tag{21}$$

with

$$\hat{\boldsymbol{\beta}}^c = \arg\min_{\boldsymbol{\beta}^c} \ell(\hat{\boldsymbol{y}}^c = h(\boldsymbol{Z}\hat{\boldsymbol{\gamma}}^c), h(\boldsymbol{X}\boldsymbol{\beta}^c)) \tag{22}$$

as

$$\begin{aligned} \hat{\boldsymbol{\gamma}}^c &= \arg\min_{\boldsymbol{\gamma}^c} \ell(\boldsymbol{y}, h(\boldsymbol{Z}\boldsymbol{\gamma}^c)) \\ s.t. \; &||(\boldsymbol{X} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{X})^\top h(\boldsymbol{Z}\boldsymbol{\gamma}^c)||_2^2 = 0, \end{aligned} \tag{23}$$

note that the statement in Lemma 3 can be simplified by recognizing that (17) will also hold true if $\boldsymbol{X}^\top(\hat{\boldsymbol{y}}^c - \boldsymbol{\mu}) = \boldsymbol{X}^\top(\hat{\boldsymbol{y}}^c - c_2\mathbf{1}_n) = \mathbf{0}$. The second term $\boldsymbol{X}^\top c_2\mathbf{1}_n$ is always equal to zero if we center $\boldsymbol{X}$, i.e., use $(\boldsymbol{X} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{X})$. Hence, if $(\boldsymbol{X} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{X})^\top\hat{\boldsymbol{y}}^c = \mathbf{0}$ is fulfilled, (17) will also hold true. Now, instead of using $(\boldsymbol{X} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{X})^\top h(\boldsymbol{Z}\boldsymbol{\gamma}^c) = \mathbf{0}$ as a constraint, introducing p auxiliary variables when using the Lagrangian multiplier method (see details below), we can instead also use the constraint $||(\boldsymbol{X} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{X})^\top h(\boldsymbol{Z}\boldsymbol{\gamma}^c)||_2^2 = 0$, which implies the former, but only requires one Lagrangian multiplier variable.

Modified differential multiplier method (MDMM) Following Platt & Barr (1987), we first reformulate the constrained optimization problem (23) using Lagrangian multipliers, yielding

$$\arg\min_{\boldsymbol{\gamma},\lambda} \ell(\boldsymbol{y}, h(\boldsymbol{Z}\boldsymbol{\gamma})) + \lambda \underbrace{||(\boldsymbol{X} - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{X})^\top h(\boldsymbol{Z}\boldsymbol{\gamma})||_2^2}_{=:\mathcal{A}(\boldsymbol{\gamma})} =: \arg\min_{\boldsymbol{\gamma},\lambda} \mathcal{E}(\boldsymbol{\gamma}, \lambda). \tag{24}$$

In order to optimize (24), we can use the basic differential multiplier method (BDMM) by performing gradient descent to optimize $\mathcal{E}$ w.r.t. $\boldsymbol{\gamma}$ but gradient ascent for the auxiliary term $\lambda\mathcal{A}$ w.r.t. λ, i.e., updating the parameters in each iteration $t \in \mathbb{N}$ as

$$\begin{aligned} \boldsymbol{\gamma}^{[t]} &= \boldsymbol{\gamma}^{[t-1]} - \nu^{[t]}\nabla_{\boldsymbol{\gamma}}\mathcal{E}(\boldsymbol{\gamma}^{[t-1]}; \lambda^{[t-1]}), \\ \lambda^{[t]} &= \lambda^{[t-1]} + \nu^{[t]}\mathcal{A}(\lambda^{[t-1]}; \boldsymbol{\gamma}^{[t-1]}), \end{aligned} \tag{25}$$

with learning rate $\nu^{[t]}$. The *modified* version MDMM of the BDMM suggested by Platt & Barr (1987) combines the Lagrangian method with the penalty method augmenting the overall objective by a penalty term

$$\mathcal{E}_P(\boldsymbol{\gamma}, \lambda) := \mathcal{E}(\boldsymbol{\gamma}, \lambda) + \frac{\zeta}{2}\mathcal{A}(\boldsymbol{\gamma})^2 \tag{26}$$

with damping factor ζ (that defaults to 1), and thus adds another term $\zeta\mathcal{A}(\boldsymbol{\gamma})$ to the gradient descent direction in (25).

E.3. Proof of Theorem 5

Consider the ReLU activation function defined as $h(x) = x \cdot \mathbb{1}(x \geq 0)$. Using the fact that the MSE-optimal $\hat{\boldsymbol{\beta}}^c$ is found by

$$\hat{\boldsymbol{\beta}}^c = \arg\min_{\boldsymbol{\beta}^c} \frac{1}{n}||h(\boldsymbol{Z}\boldsymbol{\gamma}) - h(\boldsymbol{X}\boldsymbol{\beta}^c)||_2^2, \tag{27}$$

we can first expand the objective which is minimized in (27) to

$$\frac{1}{n}\left(h(\boldsymbol{Z}\boldsymbol{\gamma})^\top h(\boldsymbol{Z}\boldsymbol{\gamma}) - 2h(\boldsymbol{Z}\boldsymbol{\gamma})^\top h(\boldsymbol{X}\boldsymbol{\beta}^c) + h(\boldsymbol{X}\boldsymbol{\beta}^c)^\top h(\boldsymbol{X}\boldsymbol{\beta}^c)\right). \tag{28}$$

The first dot product in (28) is constant w.r.t. $\boldsymbol{\beta}^c$ and the last dot product only determines the scaling of $\hat{\boldsymbol{\beta}}^c$, hence we focus on the mixed term. If, after feature correction $\boldsymbol{Z}^c = \mathcal{P}_{\boldsymbol{X}}^\perp\boldsymbol{Z}$, the mixed term

$$h(\boldsymbol{Z}^c\boldsymbol{\gamma})^\top h(\boldsymbol{X}\boldsymbol{\beta}^c) = 0, \tag{29}$$

the optimal $\hat{\boldsymbol{\beta}}^c$ in (27) will also be zero.

We can exploit properties of the ReLU function to show that this is indeed the case, using the fact that for any $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ it holds

$$\begin{aligned} \boldsymbol{a}^\top\boldsymbol{b} = \sum_i a_i b_i &= \sum_i \left[h(a_i)h(b_i) + h(-a_i)h(b_i) + h(a_i)h(-b_i) + h(-a_i)h(-b_i)\right] \\ &= \sum_i h(a_i)h(b_i) + \sum_i h(-a_i)h(b_i) + \sum_i h(a_i)h(-b_i) + \sum_i h(-a_i)h(-b_i) \\ &= h(\boldsymbol{a})^\top h(\boldsymbol{b}) + h(-\boldsymbol{a})^\top h(\boldsymbol{b}) + h(\boldsymbol{a})^\top h(-\boldsymbol{b}) + h(-\boldsymbol{a})^\top h(-\boldsymbol{b}), \end{aligned} \tag{30}$$

which allows us to rewrite the left-hand side of (29) as

$$\begin{aligned}
& h(\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) = (\boldsymbol{Z^c\gamma})^\top(\boldsymbol{X\beta}^c) - h(-\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) - h(\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) - h(-\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) \\
\iff & h(\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) = \underbrace{\boldsymbol{\gamma}^\top \boldsymbol{Z}^\top \mathcal{P}_{\boldsymbol{X}}^\perp \boldsymbol{X\beta}^c}_{=0} - h(-\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) - h(\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) - h(-\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) \\
\iff & h(\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) = -h(-\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) - h(\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) - h(-\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) \\
\iff & h(\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) + h(-\boldsymbol{Z^c\gamma})^\top h(\boldsymbol{X\beta}^c) + h(\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) + h(-\boldsymbol{Z^c\gamma})^\top h(-\boldsymbol{X\beta}^c) = 0.
\end{aligned} \tag{31}$$

Since $h(\cdot) \geq 0$ for all inputs, the left side of the equation consists of a sum of products of non-negative terms, implying that all product terms must be equal to zero.

E.4. Proof of Corollary 6

Using the $\text{vec}(\cdot)$ operation, we can first reformulate the evaluation model with uncorrected predictions $\text{vec}(\hat{\mathfrak{Y}}) \in \mathbb{R}^{n \cdot d}$, $d = \prod_{r=1}^{R} d_r$, as

$$\text{vec}(\hat{\mathfrak{Y}}) = \text{vec}(\boldsymbol{X} \times_1 \mathfrak{B}^c) = \text{vec}(\boldsymbol{X} \underbrace{\text{mat}(\mathfrak{B}^c)}_{p \times d}) = (\boldsymbol{I}_d \otimes \boldsymbol{X})\text{vec}(\mathfrak{B}^c). \tag{32}$$

The orthogonal projection matrix of $(\boldsymbol{I}_d \otimes \boldsymbol{X})$ is given by $(\boldsymbol{I}_d \otimes \mathcal{P}_{\boldsymbol{X}}^\perp)$. Premultiplying $\text{vec}(\hat{\mathfrak{Y}})$ by this term yields

$$(\boldsymbol{I}_d \otimes \mathcal{P}_{\boldsymbol{X}}^\perp)\text{vec}(\hat{\mathfrak{Y}}) = \text{vec}(\mathcal{P}_{\boldsymbol{X}}^\perp \underbrace{\text{mat}(\hat{\mathfrak{Y}})}_{n \times d}) = \text{vec}(\underbrace{\mathcal{P}_{\boldsymbol{X}}^\perp \times_1 \hat{\mathfrak{Y}}}_{:=\hat{\mathfrak{Y}}^c}). \tag{33}$$

The solution $\hat{\mathfrak{B}}^c$ to the evaluation model using corrected predictions $\hat{\mathfrak{Y}}^c$ can then be found by minimizing

$$||\text{vec}(\hat{\mathfrak{Y}}^c) - \text{vec}(\boldsymbol{X} \times_1 \mathfrak{B}^c)||_2^2, \tag{34}$$

over $\mathfrak{B}^c$. The closed-form solution is given by

$$\text{vec}(\hat{\mathfrak{B}}^c) = ((\boldsymbol{I}_d \otimes \boldsymbol{X})^\top(\boldsymbol{I}_d \otimes \boldsymbol{X}))^{-1}(\boldsymbol{I}_d \otimes \boldsymbol{X})^\top \text{vec}(\hat{\mathfrak{Y}}^c) = ((\boldsymbol{I}_d \otimes \boldsymbol{X})^\top(\boldsymbol{I}_d \otimes \boldsymbol{X}))^{-1} \underbrace{(\boldsymbol{I}_d \otimes \boldsymbol{X})^\top(\boldsymbol{I}_d \otimes \mathcal{P}_{\boldsymbol{X}}^\perp)}_{=\boldsymbol{0}} \text{vec}(\hat{\mathfrak{Y}}), \tag{35}$$

proving that $\hat{\mathfrak{B}}^c \equiv \boldsymbol{0}$.

E.5. Proof of Corollary 7

Similar to the proof of Theorem 5, we start with the solution of the evaluation model given by

$$\hat{\mathfrak{B}}^c = \arg\min_{\mathfrak{B}^c} \frac{1}{n}||h(\text{vec}(\bar{\mathfrak{Y}}^c)) - h(\text{vec}(\boldsymbol{X} \times_1 \mathfrak{B}^c))||_2^2, \tag{36}$$

where $h(x) = x \cdot \mathbb{1}(x \geq 0)$ denotes the ReLU function, and focus on the mixed term $h(\text{vec}(\bar{\mathfrak{Y}}^c))^\top h(\text{vec}(\boldsymbol{X} \times_1 \mathfrak{B}^c))$ when expanding the argument. Noting that this is again a dot product of vectors, we can use the same argument as before in (30) for the proof of Theorem 5. In particular, the non-activated mixed term including the correction evaluates to

$$\text{vec}(\mathcal{P}_{\boldsymbol{X}}^\perp \times_1 \bar{\mathfrak{Y}})^\top \text{vec}(\boldsymbol{X} \times_1 \mathfrak{B}^c) = (\text{vec}\bar{\mathfrak{Y}})^\top \underbrace{(\boldsymbol{I}_d \otimes \mathcal{P}_{\boldsymbol{X}}^\perp)(\boldsymbol{I}_d \otimes \boldsymbol{X})}_{\boldsymbol{0}} \text{vec}(\mathfrak{B}^c) = 0, \tag{37}$$

and all other terms are again activated using the ReLU activation (and hence must be non-negative, but since their sum must be zero, all terms must be zero).

F. Numerical Experiment Details

The code for reproducing results can be found on the first author's Github repository.

Table 5. Overview of all datasets used. The reference category refers to the category used as a reference in the $\mathcal{M}^e$ when encoding categorical variables.

	n_{train}	n_{test}	p	q	reference category
Adult	30162	-	5	45	female, White
Bank	40195	-	1	49	-
Compas	5855	-	6	13	female, African-American
Health Retirement	38653	-	4	25	female, married, Black
MNISTred	11872	1989	1	28×28	red
MIMIC	181342	3041	4	111	female, Asian
UTKfaces	16514	4129	5	256	Asian
Movie Reviews	34167	8538	2	100×100	female, female (main actor, movie director)

F.1. Adult Income Prediction

For $\mathcal{M}^p$ we use the features age, employment status, education, marital status, relationship, and hours per week. Protected features are given in Table 5. All models are GLMs with canonical link function and binomial distribution, i.e., ℓ is the binary cross-entropy loss and $h()$ the sigmoid function.

F.2. Fairness Benchmark

For all methods, we set the unfairness amount to 0.05 (as 0 would result in an infinite penalty).

F.3. Chest X-ray Embeddings

We utilize the publicly available embeddings[2] of the CXR foundation model proposed by (Sellergren et al., 2022). The train and test partition is based on the recommended splits of the original MIMIC-CXR database.

Before using the embedding in a classifier, we apply singular value decomposition (SVD) to reduce the risk of overfitting in the 1376-dimensional embedding. We find that $q = 111$ dimensions are the best trade-off between explained variance (98%) and sparsity. The prediction model uses only these q features while the correction function and evaluation model are defined by age, sex, and race.

F.4. Face Recognition

The embedding of the UTKFace dataset consists of the activations in the penultimate layer in a ResNet-50. Similar to the Chest X-ray application, we first reduce the 2048-dimensional embedding to $q = 32$ features using an SVD. These q features explain 95% of the embedding's variance. The train and test split is based on a random 80/20 partitioning. We use these embedding features in the prediction model while using age and race for the correction function and evaluation model.

F.5. Movie Reviews

We first extract the first actor's sex and the director's sex from the provided metadata. We then tokenize the movie descriptions using 1000 words and an embedding size of 100 with a maximum length of 100 words for each description. Padding is done at the end of sentences with less than 100 words. We further remove stop words and punctuations from sentences and transform all texts to lowercase. We use the vote count as a Poisson-distributed outcome and partition the data into 80/20% for training and testing.

The LSTM model is defined by an embedding layer with embedding size 100, an LSTM layer with 50 units and ReLU activation, a dropout layer with 0.1 dropout rate, a dense layer with 25 units and ReLU activation, a dropout layer with 0.2 dropout rate, a dense layer with 5 units and ReLU activation, a dropout layer with 0.3 dropout rate, and a final dense layer with 1 unit and exponential activation. The network is trained for a maximum of 1000 epochs with early stopping using Adam with a learning rate of 1e-6, a batch size of 128, and Poisson loss.

For the orthogonalization, we use the same architecture but insert an orthogonalization operation between the embedding layer and the LSTM layer. As features $\boldsymbol{Z}$ we use only the movie descriptions, whereas the protected features $\boldsymbol{X}$ consist of the gender of the protagonist and director.

[2]https://doi.org/10.13026/pxc2-vx69

F.6. Online Orthogonalization

We subsample the MNIST dataset to observations with labels 0 or 9. Based on the coloring described in the main part of the paper, we train a CNN defined by a Conv2D layer with 8 filters, kernel size of 3×3, and ReLU activation, followed by the orthogonalization operation for the network that uses a correction. The output (with or without correction) is then fed into another Conv2D layer with 16 filters, kernel size of 3×3, and ReLU activation, followed by Max pooling of size 2×2 and a flattening operation. Finally, the classification head comprises a dense layer with 64 units and ReLU activation followed by the output layer with one unit and sigmoid activation.

The prediction model only uses the images and, if corrected, also the color information to perform the orthogonal projection for training. The networks are trained for 100 epochs with early stopping and a batch size of 128. Early stopping is based on a 20% validation split and a patience of 25. The evaluation model is optimized using quadratic programming and only uses the color red information.

F.7. Runtimes

Table 6 compares the runtimes of different methods on the fairness benchmark dataset.

Table 6. Runtimes on the fairness benchmark dataset (in seconds).

Method	adult	bank	compas	health.retirement
Komiyama1	3.62	2.37	0.65	1.00
Komiyama2	5.66	2.68	1.19	14.17
Berk	1.13	2.59	1.18	3.52
Zafar	15.02	0.56	12.02	-
Ours	1.11	13.20	0.48	3.84

For the colorized MNIST dataset, the average fitting time until early stopping (with standard deviation in brackets) of both networks was 1.08 (0.528) minutes with orthogonalization and 0.40 (0.006) minutes without orthogonalization. We can see that the orthogonalization takes on average longer to "converge". This can mainly be attributed to the fact that it runs for more iterations and not the additional operations required to perform the orthogonalization.

G. Additional Results

G.1. Synthetic Data

First, we simulate data from a Bernoulli and Poisson distribution to check whether our approach provides the proposed protection for features $\boldsymbol{X}$ using a canonical activation function h and either $\mathcal{C}^l$ or $\mathcal{C}^h$. We generate features $\boldsymbol{X}$ from a standard normal distribution and generate features $\boldsymbol{Z}$ correlated to $\boldsymbol{X}$ using the transformation $\boldsymbol{X} = \rho \boldsymbol{Z}_{[:,1:q]} + \boldsymbol{E}$, where $\rho \in \{0, 1, 2\}$ and $\boldsymbol{E}$ are realizations of a standard normal distribution with $\dim(\boldsymbol{E}) = \dim(\boldsymbol{Z})$. We examine different settings for $p \in \{5, 10\}$, $q \in \{10, 100\}$, and $n \in \{200, 1000, 5000, 10000\}$, and repeat every setting 10 times with different random seed.

Results Figure 7 shows the results of the simulation for the classification task and $\rho = 2$ (the count response task in the Appendix is qualitatively the same) with and without correction for the model's pre-activation (corrected with $\mathcal{C}^l$) and post-activation values (corrected with $\mathcal{C}^h$). Results confirm that not only are coefficients estimated to be close to zero after correction, but their p-value is also close to 1 in most cases, suggesting no significant influence of protected features in the predicted value $\hat{\boldsymbol{y}}^c$. In contrast, without correction, coefficients tend to be non-zero and "highly significant" (with a very small p-value).

Similar results are obtained when using $\rho = 0$ or $\rho = 1$ for a logistic regression model (cf. Figure 8 and 9, respectively) and also for $\rho \in \{0, 1, 2\}$ for a count regression (Figures 10-12).

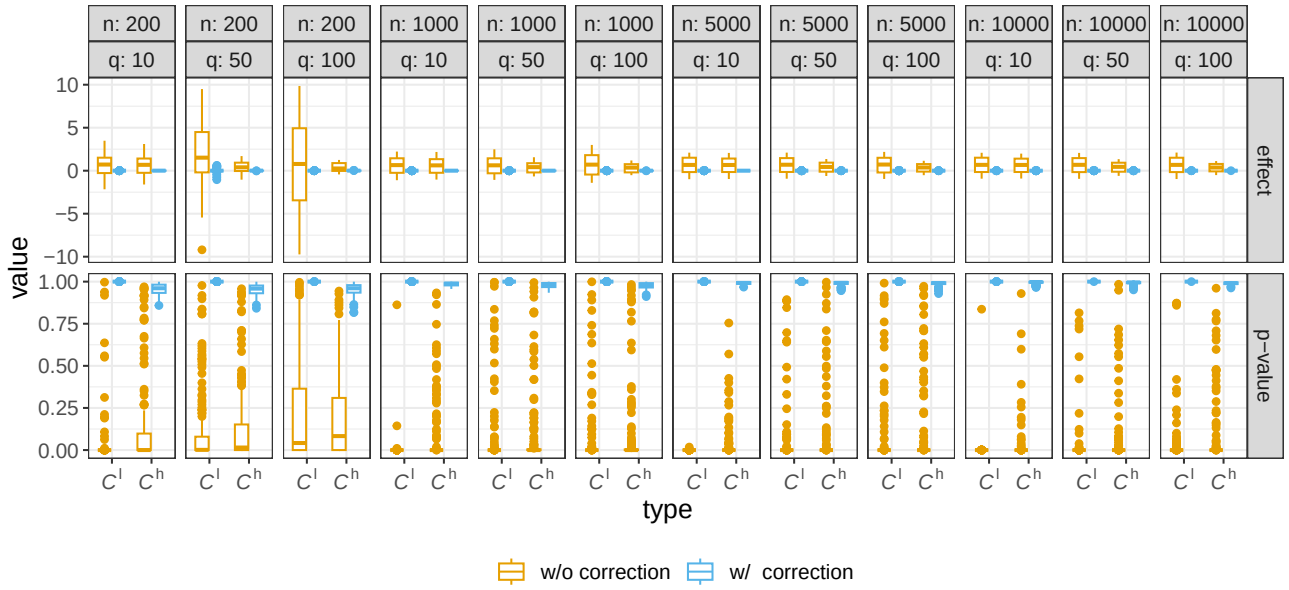

Figure 7. Distribution of estimated coefficients (top row) and corresponding p-values (bottom row) for different data sizes n and features q (columns) for the binary classification task (Bernoulli distribution). Boxplots summarize all different values of p and the 10 simulation repetitions.

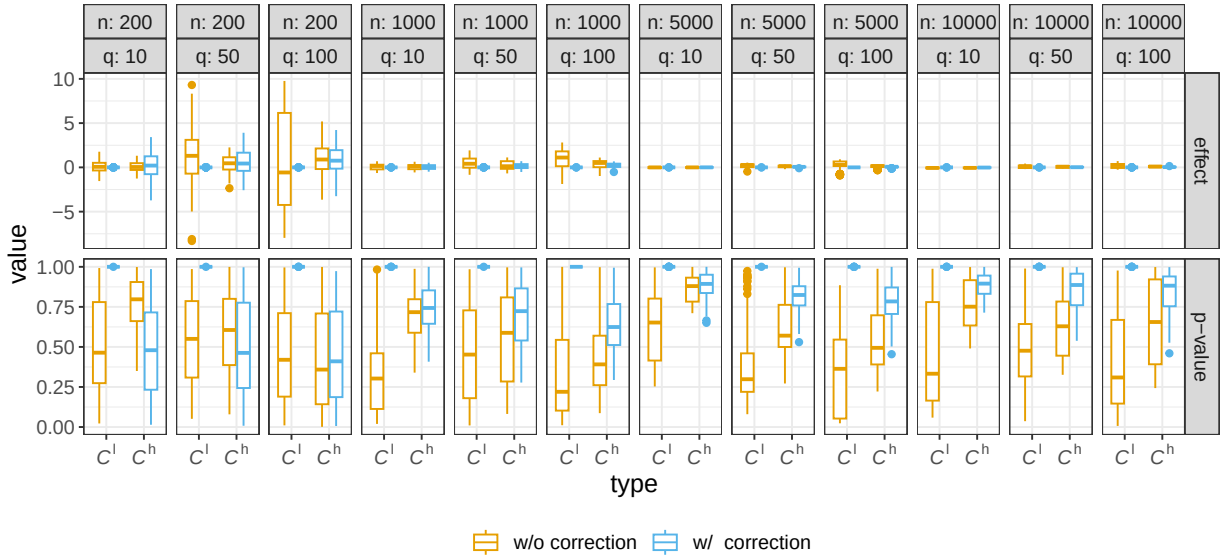

Figure 8. Distribution of estimated coefficients (top row) and corresponding p-values (bottom row) for different data sizes n (columns) for the classification task and $\rho = 0$. Boxplots summarize all different values of p, q, and the 10 simulation repetitions.

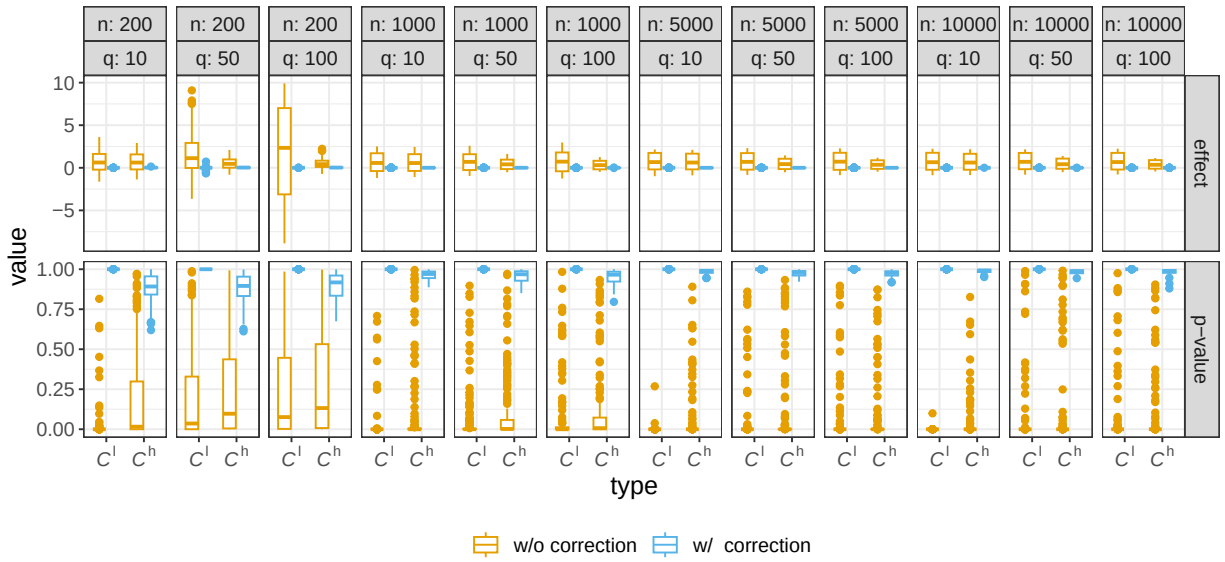

Figure 9. Distribution of estimated coefficients (top row) and corresponding p-values (bottom row) for different data sizes n (columns) for the classification task and $\rho = 0$. Boxplots summarize all different values of p and the 10 simulation repetitions.

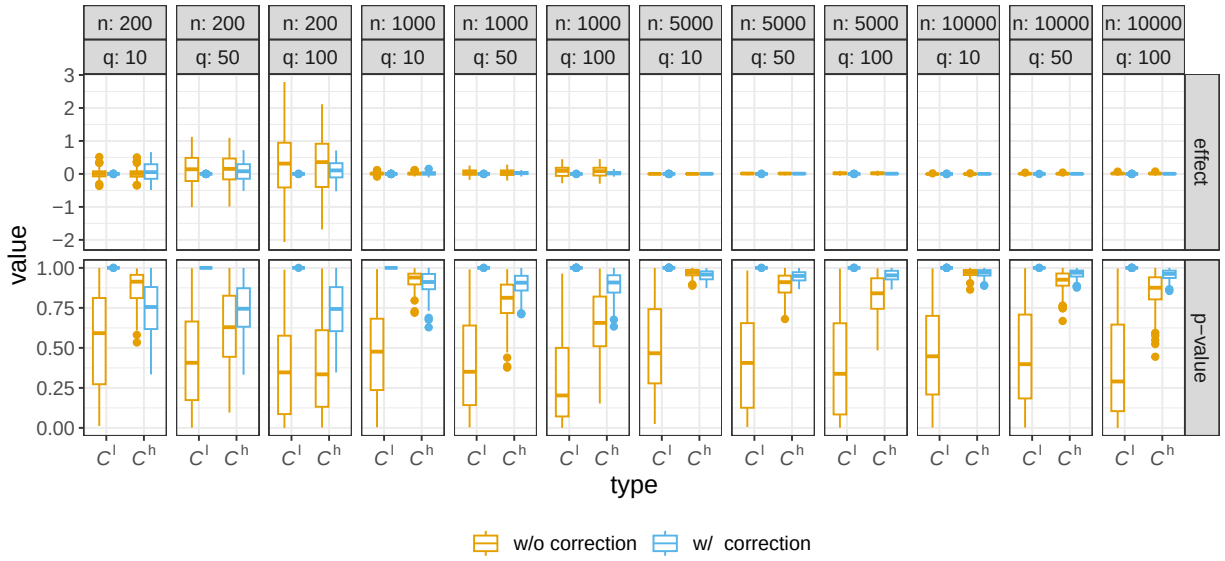

Figure 10. Distribution of estimated coefficients (top row) and corresponding p-values (bottom row) for different data sizes n (columns) for the count task (Poisson) and $\rho = 0$. Boxplots summarize all different values of p and the 10 simulation repetitions.

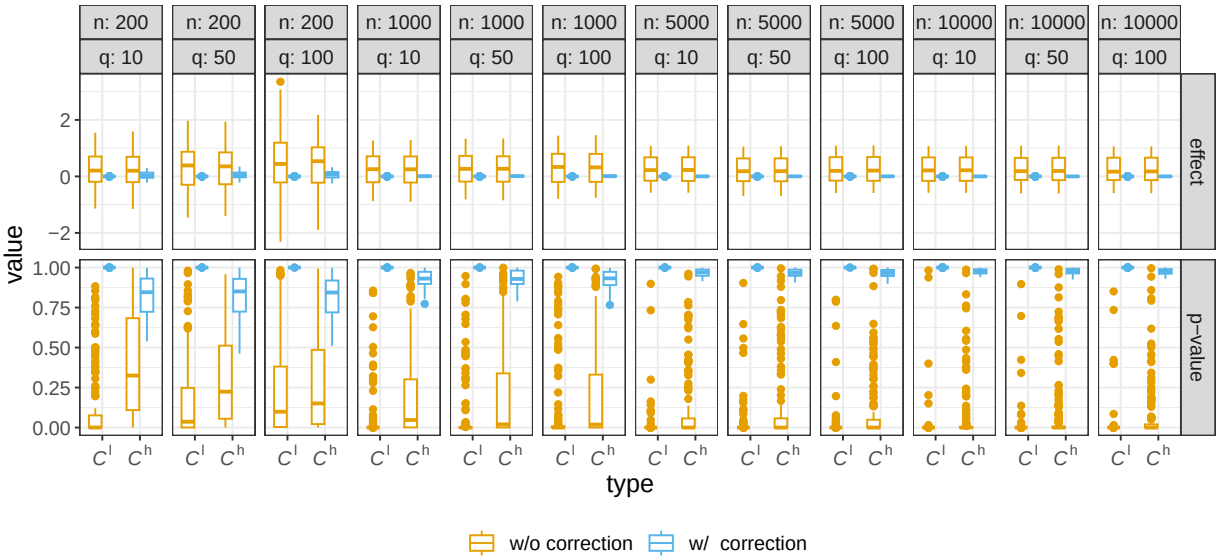

Figure 11. Distribution of estimated coefficients (top row) and corresponding p-values (bottom row) for different data sizes n (columns) for the count task (Poisson) and $\rho = 1$. Boxplots summarize all different values of p and the 10 simulation repetitions.

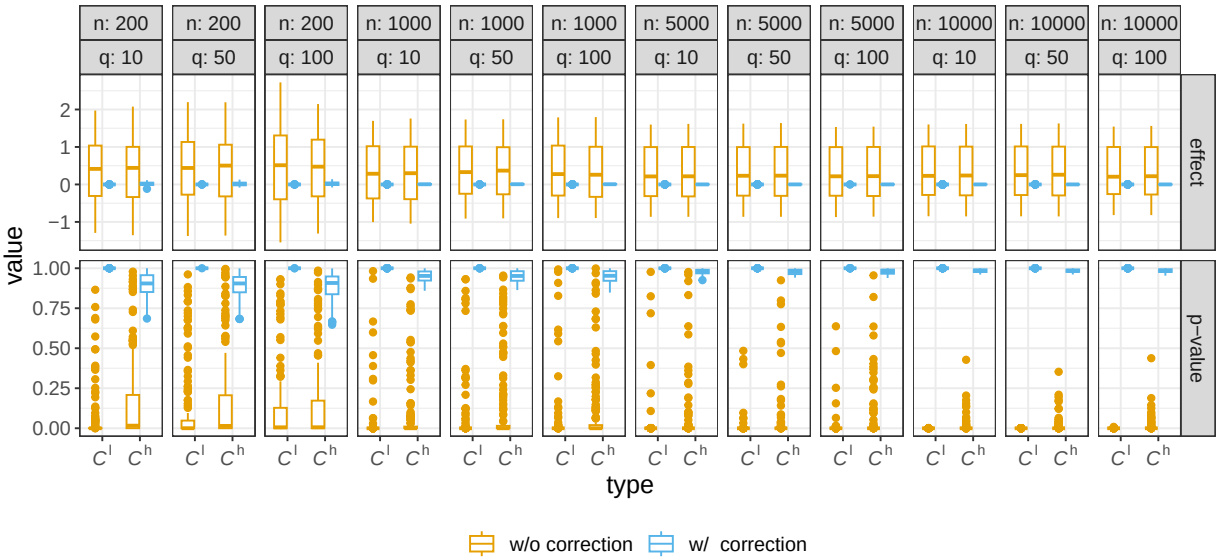

Figure 12. Distribution of estimated coefficients (top row) and corresponding p-values (bottom row) for different data sizes n (columns) for the count task (Poisson) and $\rho = 2$. Boxplots summarize all different values of p and the 10 simulation repetitions.