Frequentist Uncertainty Quantification in Semi-Structured Neural Networks

Emilio Dorigatti
LMU Munich
HMGU Munich
MCML

Benjamin Schubert
HMGU Munich
TU Munich

Bernd Bischl
LMU Munich
MCML

David Rügamer
LMU Munich
MCML
TU Dortmund

Abstract

Semi-structured regression (SSR) models jointly learn the effect of structured (tabular) and unstructured (non-tabular) data through additive predictors and deep neural networks (DNNs), respectively. Inference in SSR models aims at deriving confidence intervals for the structured predictor, although current approaches ignore the variance of the DNN estimation of the unstructured effects. This results in an underestimation of the variance of the structured coefficients and, thus, an increase of Type-I error rates. To address this shortcoming, we present here a theoretical framework for structured inference in SSR models that incorporates the variance of the DNN estimate into confidence intervals for the structured predictor. By treating this estimate as a random offset with known variance, our formulation is agnostic to the specific deep uncertainty quantification method employed. Through numerical experiments and a practical application on a medical dataset, we show that our approach results in increased coverage of the true structured coefficients and thus a reduction in Type-I error rate compared to ignoring the variance of the neural network, naive ensembling of SSR models, and a variational inference baseline.

1 INTRODUCTION

Following the increase in data availability, there is a growing need for methods in the field of deep learning to meaningfully combine non-tabular data (e.g., image and/or text) with tabular data. Particularly in medicine, most recent studies include both medical images or genome sequence data and tabular patient information such as age, sex, body-mass index, or medication (see, e.g., Huang et al., 2022; Isobe et al., 2022; Zheng et al., 2020). Whereas the non-tabular data is modeled using a variety of deep neural network (DNN) architectures such as graph neural networks (see, e.g., Fritz et al., 2022) or (pre-trained) convolutional neural networks (see, e.g., Lassau et al., 2021), the tabular data is often concatenated in the last layer to learn a linear effect for these features. An alternative method is to extract the DNN's latent features learned in the penultimate layer and feed these into a regression model (see, e.g., Wolf et al., 2022, for a comparison of approaches). The simple concatenation of tabular features into the DNN (and, hence, linear effect assumption of these variables) is potentially not complex enough to represent real-world relationships. For example, the effect of medication does not always increase linearly with increased dosage.

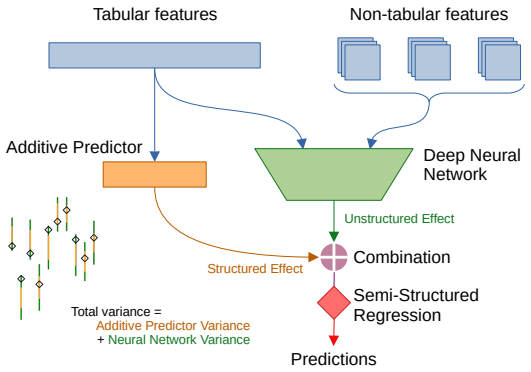

Figure 1: The semi-structured regression (SSR) framework makes it possible to learn from tabular and non-tabular features (e.g. image data or tensors) simultaneously, while providing principled inference for the additive predictor.

An approach to overcome these limitations is to combine the structured predictors of a flexible statistical regression model with deep (or unstructured) neural networks in a one-step semi-structured regression (SSR) network and jointly learn the effect of the tabular and non-tabular data (see, e.g., Kopper et al., 2021; Baumann et al., 2021; Kook et al., 2022).

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

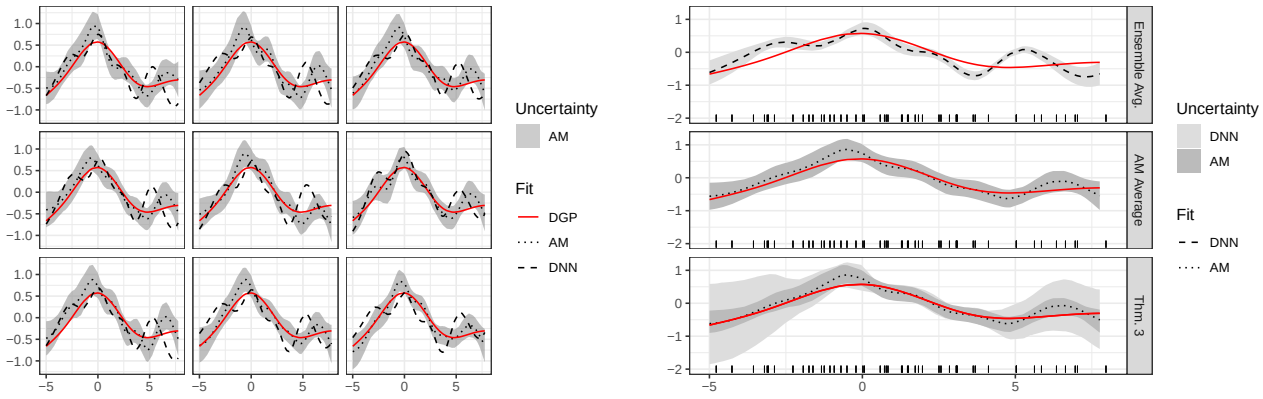

Figure 2: The structured additive predictor of nine SSR models is used to fit a thin plate regression spline on a simulated dataset (red line), and exhibits moderate overfitting and excessive wiggliness after the initial training with stochastic gradient descent (left, dashed lines). The predictions of the DNNs could be used to fit additive models (AM) *post hoc*, however, the resulting fits (left, dotted lines, and dark gray areas) are still too wiggly even after their smoothness penalty is tuned. Ensembling the initial SSR models does not result in either better point predictions or reliable confidence intervals (right top), while ensembling the AMs provides a better average fit with sometimes overly narrow confidence intervals (right middle). The best fit is instead achieved with our proposed method in Theorem 3: In dense regions of the data distribution, the DNN uncertainty is very low and does not affect confidence intervals, while at the extremes of the data distribution the confidence intervals are larger reflecting the increased DNN uncertainty (right bottom).

This approach allows one to incorporate non-linear effects of tabular data – such as splines – into the network while extending the class of generalized additive models (GAMs; Hastie and Tibshirani, 1990) by more complex neural network predictions or non-tabular data (Rügamer et al., 2023). The SSR framework can also be seen as a GAM where the DNN provides a sample-specific offset that captures the effect of the non-tabular features and all interactions not already codified in the design matrix of the additive predictor. This point of view is particularly interesting for medical applications where statistical models such as GAMs are well-accepted and used for decision-making in critical situations (see, e.g., Desquilbet and Mariotti, 2010).

Besides prediction on unseen data, an important use of predictive modeling is interpretation and understanding what features do and do not affect the response. While providing a framework with principled inference and interpretable results using the standard theoretical development of (generalized) linear models (Casella and Berger, 2021), the use of SSR models in research domains dominated by p-values and classical statistical models requires a well-defined and theoretical-founded quantification of uncertainty, i.e., valid statistical inference. DNN estimates are intrinsically uncertain for a multitude of reasons, including limited amount of training data, stochastic optimization procedures, multimodality of the likelihood landscape, etc. However, naive approaches such as last layer inference (see, e.g., Daxberger et al., 2021) or using the extracted features from a DNN in a regression model are known to ignore the variance in the DNN estimation of the unstructured effects, resulting in an estimator of the structured coefficients with artificially low variance and consequently increased Type-I (false positive) error rates. In reality, the variance in the estimation of the unstructured effects should be propagated to the structured coefficients in order to obtain confidence intervals of nominal coverage.

Our contributions: In this work, we quantify how the variance of the DNN estimation affects the distribution of the estimator of the structured effects of an SSR model, thus reducing Type-I error rates stemming from ignoring the DNN uncertainty. By abstracting the DNN as a random, normally-distributed offset, our framework can make use of any deep uncertainty quantification (DUQ) method that adheres to our distributional assumptions. Given such estimation, we derive exact expressions for the variance of the estimators of the structured coefficients for linear and additive semi-structured models and provide asymptotic results for the structured coefficients relating to generalized linear models (GLMs; Nelder and Wedderburn, 1972) and generalized additive semi-structured models. By analyzing the resulting analytical form of the variance in additive SSR models, we can answer two critical questions for practitioners – namely, what happens when DUQ is not performed, and how inference in the SSR model is affected by DNN pretraining. In both cases, we show that traditional inference methods based exclusively on residuals are almost surely correct in the large-sample limit.

We then conduct simulation studies to show the correctness of our framework, first by sampling from our assumed data-generating process, then by involving an actual DNN

with realistic, imprecise UQ and non-normally distributed predictions. In both cases, we show that the confidence intervals derived with our formulation result in increased coverage compared to the naive, uncertainty-unaware procedure and a naive ensembling of SSR models, eventually reaching nominal coverage in all situations when the DNN uncertainty is exactly quantified. We finally apply our framework to a real-world dataset of skin lesion images and show that including the DNN uncertainty greatly improves the predictive performance and avoids over-confident inference.

Our framework is based on frequentist theory and thus allows practitioners to provide inference statements – such as p-values and confidence intervals – in a well-established format without requiring priors to be specified. This allows researchers from other domains to use SSR models interchangeably with (generalized) linear models or GAMs, which fosters better comparisons and easier adaptation. To the best of our knowledge, correct inference for structured effects in semi-structured models has not been studied yet.

2 BACKGROUND

Many sources of uncertainty affect the quality of predictive models, ranging from data issues such as noise and non-representativeness to inference issues such as training and hyperparameter tuning, among others. Nonetheless, it is necessary to accurately quantify the uncertainty of predictive models in order to ensure that their predictions are fair, can be trusted, and are safe to use – especially in domains such as medicine and autonomous driving, where real harm could be done if real-world decisions are based on wrong, uncertain predictions (see, e.g., Begoli et al., 2019; Michelmore et al., 2020; Verma and Rubin, 2018).

Recent advances in deep learning made it possible to obtain accurate predictions for various unstructured data sources including images, text, and audio. However, DUQ is particularly difficult, as the theoretical foundations of the field are still being developed. Additionally, the considerable computational resources needed for exact UQ further complicate this approach. Due to these difficulties, a wide range of methods for DUQ have been proposed recently. For more information, we refer to Abdar et al. (2021); Gawlikowski et al. (2021) for an in-depth overview and Appendix A for further considerations on the topic. The lack of (theoretically) grounded DUQ methods and the need of practitioners in certain fields to provide reliable statistical significance statements motivates the necessity of SSR models and UQ thereof.

3 INFERENCE IN SSR MODELS

A semi-structured regression (SSR) model (Figure 1) is used to jointly learn from tabular features through an additive predictor and non-tabular data through a DNN. We distinguish between tabular and non-tabular data mainly by the modeling approach employed, where tabular data comes in the form of vectors and is amenable to structured (generalized) linear or additive models, while non-tabular data is generally represented as tensors and requires unstructured, non-linear modeling as done by (deep) neural networks. We assume a two-step fitting procedure for the SSR model where the DNN is first trained jointly with the additive predictor on a training dataset using a second, independent, held-out dataset for early stopping, then inference for the structured coefficients is performed on the validation set based on the DNN predictions as well as their associated (epistemic) uncertainty.[1]

3.1 Notation and Problem Setup

In this work, we focus on the second step outlined above, where inference on the structured coefficients is performed. We denote with $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, $\boldsymbol{X} = (\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_n^\top)^\top$ the design matrix of the tabular features $\boldsymbol{x}_i \in \mathbb{R}^d$ for the n validation samples, including (possibly) dummy encoding for factor variables and (possibly) basis expansion for splines, for a total of d features. We further denote with $\boldsymbol{\beta} \in \mathbb{R}^d$ the true parameters of the additive predictor and with $\boldsymbol{f} \in \mathbb{R}^n$ the true additive effect of the non-tabular features, so that the response to be predicted $\boldsymbol{y} = (y_1, \dots, y_n)^\top \in \mathbb{R}^n$ is generated from the linear predictor $\boldsymbol{\eta} := \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f}$. The true and unknown unstructured effect $\boldsymbol{f}$ is estimated by a DNN with suitable architecture and appropriate training procedure, whose predictions are denoted as $\boldsymbol{z} \in \mathbb{R}^n$ with covariance[2] $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$ derived by an unspecified DUQ method.

By completely separating the estimation of $\boldsymbol{z}$ and $\boldsymbol{\Gamma}$ from the estimation of $\boldsymbol{\beta}$, we are able to propose a generic plug-and-play, future-proof framework that can use any DUQ method with no modifications, provided that it results in reliable UQ estimates. Finally, we note that providing a reliable estimation of $\boldsymbol{\Gamma}$ is still an open research problem (see Section 2) that we do not intend to tackle here.

DNN Abstraction and Assumptions: For tractability reasons, in our theoretical derivations, we assume that $\boldsymbol{z}$ is distributed as $\mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma})$ with full-rank covariance. We further assume $\boldsymbol{\Gamma}$ to represent the true uncertainty of the network, as in the case of unknown over-/under-estimation, it is not possible to derive theoretical guarantees on the nominal coverage of $\boldsymbol{\beta}$ without further assumptions. Despite our treatment of $\boldsymbol{\Gamma}$ as a generic matrix, we later argue that inference on held-out datasets (as described above) naturally results in diagonal covariance, which is considerably easier to obtain and deal with when working with large models or

[1] The aleatoric DNN uncertainty cannot be disentangled from the aleatoric uncertainty of the SSR model as a whole.

[2] Throughout the text, we refer to $\boldsymbol{\Gamma}$ alternatively as covariance of $\boldsymbol{z}$ and uncertainty of the DNN.

datasets, thus motivating our two-step approach. Moreover, we observe in practice that even when $\boldsymbol{z}$ and/or $\boldsymbol{\Gamma}$ are not perfectly estimated and the distributional assumptions are violated, our approach provides more accurate estimation of the structured coefficients over other baselines (see our experimental results in Section 4).

To develop our theory, we start by considering the simplest case where $\boldsymbol{y} = \boldsymbol{\eta}$, which allows us to derive exact expressions for the variance of $\hat{\boldsymbol{\beta}}$ (Section 3.2). We then proceed to the most general setting of $\mathbb{E}[\boldsymbol{y}] = g^{-1}(\boldsymbol{\eta})$ where only asymptotic expressions can be derived (Section 3.3).

3.2 Exact Inference in Additive SSR Models

Consider the following data-generating process:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon} \\ \boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma}) \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}) \end{aligned} \tag{1}$$

where $\boldsymbol{z}$ are the DNN predictions and $\boldsymbol{\beta}$ is estimated without further regularization. Under such a model, the additional variance of $\boldsymbol{z}$ affects inference for $\boldsymbol{\beta}$, as stated in Theorem 1.

Theorem 1 (Inference in linear models) *Given the data-generating process in Equation (1), an unbiased estimator for $\boldsymbol{\beta}$ is*

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{z})$$

whose variance is

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\sigma^2 \boldsymbol{I} + \boldsymbol{\Gamma}) \boldsymbol{X} (\boldsymbol{X}^\top \boldsymbol{X})^{-1}.$$

Given the residuals $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{z}$, an unbiased estimator of σ^2 is

$$\hat{\sigma}^2 = (\boldsymbol{r}^\top \boldsymbol{r} - tr(\boldsymbol{\Gamma}))/(n - d)$$

All proofs are found in Appendix B. The variance of $\hat{\boldsymbol{\beta}}$ can be recognized as the familiar ordinary least squares estimator with heteroscedastic errors (White, 1980), where, in this case, the sample-specific error stems from the DNN uncertainty. The additional variance in the estimator of $\hat{\boldsymbol{\beta}}$ is also propagated to the predictions, as stated in Theorem 2.

Theorem 2 (Prediction in linear models) *Given the data-generating process of Equation (1) and the estimator $\hat{\boldsymbol{\beta}}$ of Theorem 1, the variance of the predicted mean is*

$$\mathbb{V}[\boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{z}] = \sigma^2 \boldsymbol{H} + (\boldsymbol{I} - \boldsymbol{H}) \boldsymbol{\Gamma} (\boldsymbol{I} - \boldsymbol{H})^\top$$

where $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top$.

Looking at Theorem 1, we first notice that when $\sigma^2 \gg \text{tr}(\boldsymbol{\Gamma})/n$, the uncertainty from the DNN has very little effect, as most of the variance stems from the noise $\boldsymbol{\epsilon}$. Second, in the limit $\boldsymbol{\Gamma} \to \text{diag}(\gamma^2/n \cdot \boldsymbol{1})$, i.e., with constant uncertainty γ^2 from the DNN, the variance of the estimated parameters becomes $\mathbb{V}[\hat{\boldsymbol{\beta}}] = (\sigma^2 + \gamma^2)(\boldsymbol{X}^\top \boldsymbol{X})^{-1}$, whose scalar parameter can be estimated from the residual vector $\boldsymbol{r}$ using $\mathbb{E}[\boldsymbol{r}^\top \boldsymbol{r}]/(n - d) = \sigma^2 + \gamma^2$. In other words, with constant DNN uncertainty, inference based on ordinary linear model theory results in confidence intervals of nominal coverage. Therefore, our results are especially relevant when the uncertainty of the DNN 1) is of similar magnitude to or greater than the noise variance, 2) is highly variable among data points, and 3) has intricate covariance structure. These results can be specialized for the common case of a held-out validation dataset, shedding further light on the properties of SSR uncertainty.

3.2.1 Inference on Held-out Datasets

As introduced above, we assume a two-step fitting procedure where $\boldsymbol{z}$, $\boldsymbol{\Gamma}$ and $\boldsymbol{\beta}$ are estimated on a dataset different from that used to fit the SSR model. We observed empirically that in some cases and especially with small training sets, a naive estimation of $\boldsymbol{\beta}$ on the same training set used to train the SSR model can result in an artificial shrinkage, i.e., a bias towards zero, as noted by Rügamer et al. (2023) and illustrated in Appendix C. This bias can be avoided in two ways: (1) by warm-starting $\hat{\boldsymbol{\beta}}$ before training the SSR model as $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ and ignoring the non-tabular features, as suggested by Rügamer et al. (2023), or (2) by re-estimating $\hat{\boldsymbol{\beta}}$ after training the SSR model on a held-out dataset, such as the validation set used for early stopping. We prefer the latter approach, as performance on a sufficiently large and i.i.d. validation set is a reliable indicator of out-of-sample generalization and guarantees a diagonal covariance for $\boldsymbol{z}$:

Proposition 1 *When performing inference on a held-out dataset that is i.i.d. from the dataset used to fit the SSR model, the DNN predictions $\boldsymbol{z}$ are conditionally independent of each other given the training data. Furthermore, their uncertainty $\boldsymbol{\Gamma}$ is a diagonal matrix.*

The reason for this is that the DNN weights are kept fixed when predicting the unstructured effect of the held-out dataset – meaning that changes in one data point do not affect the predictions of other samples in this dataset. This proposition is extremely important for practical applications of our method, as it is considerably easier and computationally much cheaper to estimate a diagonal $\boldsymbol{\Gamma}$ instead of its full-rank counterpart. A diagonal covariance for $\boldsymbol{z}$ allows us to simplify the covariance expression of Theorem 1 and obtain two interesting asymptotic results.

Corollary 1 (Large-sample convergence) *In the large-sample limit, on a held-out dataset $\boldsymbol{X}$ i.i.d. from the dataset used to train the SSR model, the variance of $\hat{\boldsymbol{\beta}}$ can be estimated with*

$$\mathbb{V}(\hat{\boldsymbol{\beta}}) = \boldsymbol{r}^\top \boldsymbol{r}/(n - d) \cdot (\boldsymbol{X}^\top \boldsymbol{X})^{-1},$$

where $\boldsymbol{r}$ are the model's residuals on the held-out dataset.

In this situation, the sample-specific DNN uncertainty is "diluted" by a large number of observations and allows one to perform asymptotically correct inference directly from the residuals of the held-out dataset, since $\boldsymbol{r}^\top \boldsymbol{r}/(n-d)$ is the usual estimator of $\hat{\sigma}^2$ in traditional structured linear models. Similar reasoning allows us to derive another interesting corollary connecting uncertainty with DNN pre-training:

Corollary 2 (Small-sample pre-training) *Assume that the DNN was pre-trained on an external, independent dataset of m samples before fitting the SSR model. Assuming that $\gamma_i = \mathcal{O}(1/m)$ for all $i \in 1, \dots, n$, when estimating $\hat{\boldsymbol{\beta}}$ on the validation set, we have*

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = \mathcal{O}(\sigma^2 + tr(\boldsymbol{\Gamma})/\sqrt{m}) \cdot (\boldsymbol{X}^\top \boldsymbol{X})^{-1}.$$

In particular, if $\sqrt{m} = \mathcal{O}(n)$, then we have

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = \mathcal{O}(\boldsymbol{r}^\top \boldsymbol{r}/(n-d)) \cdot (\boldsymbol{X}^\top \boldsymbol{X})^{-1},$$

where the $\mathcal{O}(\cdot)$ notation in case of matrices refers to each element of the matrix.

In other words, relying exclusively on the residuals of the linear fit with small n and not modeling the DNN uncertainty results (in the worst case) in an underestimation of the variance of $\hat{\boldsymbol{\beta}}$. This variance can be reduced by pre-training the DNN on an external dataset, and when this dataset has size $\mathcal{O}(n^2)$, the resulting reduction in variance compensates for the underestimation arising from not quantifying the DNN uncertainty. However, the pre-training dataset must be relevant enough to the task at hand that the bound $\gamma_i = \mathcal{O}(1/m)$ still holds. While it is hard to establish formal results in this regard, the fact that pre-training is an effective and widely used technique suggests that this scaling bound can be reasonably achieved in practice. Although seemingly similar, note that Corollary 1 provides a convergence statement without DNN pre-training, while Corollary 2 provides an upper bound when pre-training was performed.

Being able to replace exact UQ with pre-training or completely ignoring DUQ with large datasets is incredibly useful for practitioners because proper DUQ is currently a difficult and open problem (Abdar et al., 2021), while data for pre-training is relatively easy to find aplenty in many domains. Even more conveniently, the required scaling of $m = \mathcal{O}(n^2)$ can be relaxed with large enough m and n, since at some point the irreducible aleatoric uncertainty σ^2 starts to dominate the residuals. This would result in confidence intervals that are slightly wider and more conservative but simultaneously narrow enough to allow reliable and practically useful inference, since their width asymptotically approaches zero.

These results may provoke a question regarding the value added by DUQ. First, note that Corollary 1 holds asymptotically, and Corollary 2 is a possibly very loose upper bound. Moreover, sound prediction intervals for $\boldsymbol{y}$ that consider the DNN uncertainty (Theorem 2) *cannot* be obtained from residuals alone, even asymptotically. Second, quantifying $\boldsymbol{\Gamma}$ allows decomposing the residual variance into (1) variance due to measurement noise and (2) variance from the DNN. This latter variance can be used to detect abnormal, out-of-distribution samples when applying the model in practice – for example, in biomedical applications where safety is critical and models must earn their trust from practitioners who are often not trained in statistical inference (Gaube et al., 2021).

3.2.2 Penalized SSR Models

A very similar analysis as shown above carries over to penalized least squares – including in particular Ridge regression and additive models – by explicitly handling the penalty term. We consider a setup similar to that described in the previous section: for additive models, the design matrix $\boldsymbol{X}$ includes suitable basis expansions for all smooth terms of interest. We denote the penalty matrix as $\boldsymbol{S}_{\boldsymbol{\lambda}}$, with $\boldsymbol{\lambda}$ controlling the amount of regularization. In case of Ridge regression, we set $\boldsymbol{\lambda} = \lambda^2 \mathbf{1}$ and thus $\boldsymbol{S}_{\boldsymbol{\lambda}} = \lambda^2 \boldsymbol{I}$, while for additive models, we use $\boldsymbol{S}_{\boldsymbol{\lambda}} = \sum_i \lambda_i \boldsymbol{S}_i$ with λ_i and $\boldsymbol{S}_i$ controlling the smoothness of the i-th smooth, whose coefficients are gathered by $\boldsymbol{S}_i$ to form the penalty. With this setup, inference in additive SSR models is performed as shown in Theorem 3.

Theorem 3 (Additive inference) *Given the data-generating process in Equation (1), the penalized estimator for $\boldsymbol{\beta}$ with penalty $\boldsymbol{S}_{\boldsymbol{\lambda}}$ is*

$$\hat{\boldsymbol{\beta}} = \boldsymbol{P}(\boldsymbol{y} - \boldsymbol{z}),$$

where $\boldsymbol{P} := (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1} \boldsymbol{X}^\top$. The estimator variance is

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = \boldsymbol{P}(\sigma^2 \boldsymbol{I} + \boldsymbol{\Gamma})\boldsymbol{P}^\top,$$

and an estimator of σ^2 is

$$\hat{\sigma}^2 = (\boldsymbol{r}^\top \boldsymbol{r} - tr(\boldsymbol{\Gamma}))/(n-d),$$

where $\boldsymbol{r}$ is the vector of residuals $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{z}$.

The optimal penalties $\boldsymbol{\lambda}$ can now be chosen based on (generalized) cross-validation by deriving a Bayesian posterior for $\boldsymbol{\beta}$ and tuning $\boldsymbol{\lambda}$ via marginal or restricted likelihood maximization, or via the Demmler-Reinsch orthogonalization (Ruppert et al., 2003).

In this case, prediction intervals are not straightforward to obtain in a frequentist setting due to the bias in $\hat{\boldsymbol{\beta}}$ introduced by $\boldsymbol{S}_{\boldsymbol{\lambda}}$. An empirical Bayesian approach treats $\boldsymbol{S}_{\boldsymbol{\lambda}}$ as a prior for $\boldsymbol{\beta}$ and results in $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \mathbb{V}[\hat{\boldsymbol{\beta}}])$, allowing further inference and probabilistic predictions.

3.3 Asymptotic Inference in Generalized SSR Models

We now extend our theory to the more general case of SSR models with an exponential response distribution. In this

case, the conditional mean of the response is expressed as a function of the linear predictor:

$$\begin{aligned} \mathbb{E}[\boldsymbol{y}|\boldsymbol{X}] &= g^{-1}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f}), \\ \boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma}), \end{aligned} \tag{2}$$

where, again, we first show how to estimate $\boldsymbol{\beta}$ without additional (explicit) regularization. In contrast to our previous approach, we now fix the DNN's prediction $\hat{\boldsymbol{z}}$ of $\boldsymbol{z}$ and model the remaining variation in its prediction using a random effect $\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$. This can be seen as a reformulation of the linear regression case with $\boldsymbol{z} \overset{d}{=} \hat{\boldsymbol{z}} + \boldsymbol{b}$, used to derive Theorem 4 for inference in SSR models with exponential response (and which allows for straightforward computation, as elaborated afterward).

Theorem 4 (GLM Inference) *Inference in a semi-structured GLM can be performed by solving the equivalent generalized linear mixed model (GLMM):*

$$\begin{aligned} \mathbb{E}[\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{z}] &= g^{-1}(\boldsymbol{X}\boldsymbol{\beta} + \hat{\boldsymbol{z}} + \boldsymbol{b}) \\ \boldsymbol{b} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}) \end{aligned}$$

with response distribution $\mathcal{D}$, fixed offset $\hat{\boldsymbol{z}}$, and random effects $\boldsymbol{b}$. Let $\boldsymbol{W}$ denote the GLM weights and ϕ the scale parameter for $\mathcal{D}$. Then, $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{b}}$ can be found by penalized iterated least squares, producing the asymptotic relationship (Wood, 2006, Equation 3.21):

$$\begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{b}} \end{pmatrix} \overset{a}{\sim} \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{0} \end{pmatrix}, \phi \begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X} & \boldsymbol{X}^\top \boldsymbol{W} \\ \boldsymbol{W}\boldsymbol{X} & \boldsymbol{W} + \phi \boldsymbol{\Gamma}^{-1} \end{pmatrix}^{-1} \right).$$

As most existing routines for GLMM fitting do not readily work with an externally given variance estimate for the random offset, we present an adaption to the Fisher Scoring algorithm (Breslow and Clayton, 1993) in Appendix D that makes use of conventional GLMM software.

3.3.1 Generalized Additive SSR Models

The previous considerations can be extended to include non-linear structured effects into the additive predictor $\boldsymbol{\eta}$. The resulting model (a semi-structured GAM) allows the incorporation of more flexible yet interpretable univariate or low-dimensional multivariate smooth effects (while other, higher-order interactions can be represented with the DNN part of the SSR model). While GAMs for Gaussian response can be estimated as stated in Theorem 3, a more flexible approach is to treat the smooth terms as random effects (see, e.g., Ruppert et al., 2003). This allows GAMs and their (co-)variance terms in $\boldsymbol{\Gamma}$ to be estimated by a linear mixed model solver or, in the general case, using an iterative mixed model procedure (Breslow and Clayton, 1993). Having framed semi-structured GLMs as a mixed model (Theorem 4), the extension of our approach to semi-structured GAMs, resulting in a mixed GAM (GAMM), is therefore straightforward and covered by the same theoretical results.

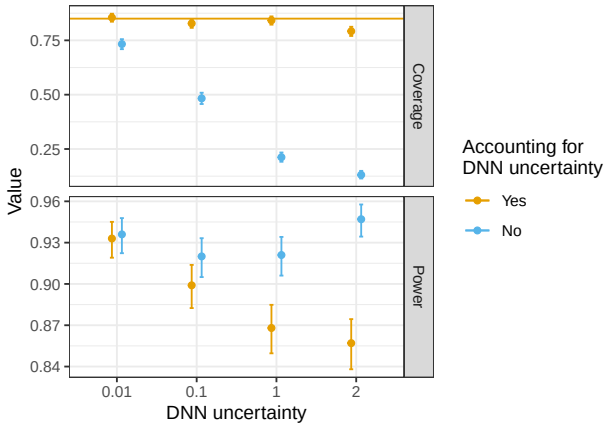

Figure 3: Coverage (top) and power (bottom) of confidence intervals for $\boldsymbol{\beta}$ with a simulated DNN giving correlated predictions with variance of γ^2 (x-axis) and when accounting (orange) and neglecting (blue) the DNN uncertainty in Theorem 4.

4 EXPERIMENTAL RESULTS

We introduce simulation studies on synthetic datasets to validate our theoretical development with simulated (Section 4.1) and real (Section 4.2) DNNs, then use a real-world data set to demonstrate our framework's practical applicability and advantages (Section 4.3). In both cases, we are mainly interested in the confidence intervals for $\boldsymbol{\beta}$, which should ideally reach the nominal coverage level α. Unless otherwise specified, we used $\alpha = 0.15$ and show 90% confidence intervals in the plots.

4.1 Data-Generating Process Investigation

We first simulate a dataset following the data-generating process of Equation (2) in order to test theoretical properties of our assumed model with known DNN predictions and uncertainty. For this, we use $n = 100$ observations, $d = 4$ structured coefficients, and a Poisson response distribution. We sample both $\boldsymbol{\beta}$ and $\boldsymbol{f}$ from standard Normal distributions. We use $\boldsymbol{\Gamma} = \gamma \boldsymbol{P}$ to sample $\boldsymbol{z}$, where $\gamma^2 \in \{0.01, 0.1, 1, 2\}$ and $\boldsymbol{P}$ is a full-rank matrix of random correlations.[3] We simulate 250 random datasets for each value of γ^2, and, for each dataset, we estimate $\boldsymbol{\beta}$ and compute the empirical power and coverage of the confidence intervals resulting from the asymptotic result of Theorem 4. We compare two scenarios: (1) when the true γ^2 is used, i.e., when correctly accounting for uncertainty in the DNN predictions, and (2) when the estimation is performed with $\gamma^2 = 0$, as is the case when no UQ is performed.

[3] Given a random $n \times n$ matrix $\boldsymbol{A}$ with $\boldsymbol{A}_{ij} \sim \mathcal{N}(3/4, 1)$, we compute $\boldsymbol{P}_{ij} = \boldsymbol{A}_i^\top \boldsymbol{A}_j (\boldsymbol{A}_{ii}^\top \boldsymbol{A}_{ii} \boldsymbol{A}_{jj}^\top \boldsymbol{A}_{jj})^{-1/2}$, resulting in correlations between 0.3 and 0.4.

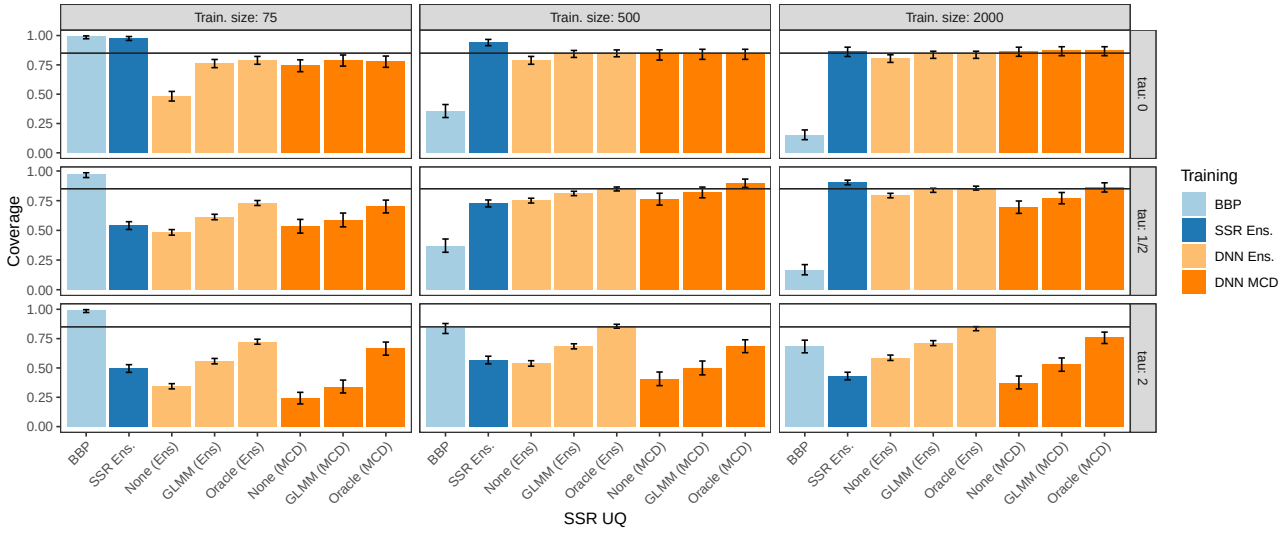

Figure 4: Coverage (y-axis) of confidence intervals for $\boldsymbol{\beta}$ for different ways of training the SSR model and the DNN, for different number of training samples (columns) and magnitude of the unstructured effect (rows). Bayes-by-backprop (BBP, light blue) and an ensemble of SSR models (SSR Ens., dark blue) directly provide confidence intervals and do not require our framework. Ensembles of DNNs (DNN Ens., light orange) and Monte Carlo Dropout (DNN MCD, dark orange) both provide an estimation of $\boldsymbol{z}$ and $\boldsymbol{\Gamma}$ that is used by our GLMM. Ignoring these $\boldsymbol{\Gamma}$ results in two uncertainty-unaware baselines using the predictions of a single network or the average prediction of an ensemble. Finally, combining $\boldsymbol{z}$ and $\boldsymbol{f}$ allows us to compute the exact uncertainty associated with $\boldsymbol{z}$ and thus derive an oracle baseline using our GLMM method with this uncertainty.

Results: Without considering uncertainty from the DNN, coverage becomes progressively worse as the simulated DNN predictions become more uncertain,[4] while nominal coverage can be achieved in all cases when accounting for DNN uncertainty using our GLMM approach (Figure 3). However, accounting for the uncertainty decreases the power of the test with increased DNN influence as structured effects naturally become more uncertain. We can thus conclude that our approach yields valid inference with nominal coverage and is well aware of the DNN prediction uncertainty. Not accounting for the additional uncertainty gives overconfident results that never yield nominal coverage while also not being aware of the increased DNN uncertainty. In this case, we also observe no change in power for varying DNN prediction influence.

4.2 Power and Coverage for Semi-Structured Uncertainty Quantification

We now turn to the application of our framework under controlled settings to investigate its power and coverage properties in practice. We generate $\boldsymbol{f}$ in Equation (2) from a randomly initialized DNN comprising of one hidden layer with eight neurons and $\tanh$ activation, and normalize both $\boldsymbol{X}\boldsymbol{\beta}$ and $\boldsymbol{f}$ to unit standard deviation before generating the observations from a Poisson distribution:

$$y_i \sim Poi(\exp(\boldsymbol{x}_i^\top \boldsymbol{\beta} + \tau f_i)), \quad i = 1, \dots, n, \tag{3}$$

where we used $\tau \in \{0, \frac{1}{2}, 2\}$ to modulate the influence of the unstructured part on the response, including a scenario where no real unstructured effect is present ($\tau = 0$). For each value of τ, we generate 150 training datasets with 75, 500, and 2000 samples and fit an ensemble of $K = 25$ SSR models respectively for each dataset. The ensembles are formed by DNNs with two hidden layers of 12 neurons each and ReLU activation, and an external validation set is used both for early stopping and to estimate $\hat{\boldsymbol{\beta}}$, $\boldsymbol{z}$, and $\boldsymbol{\Gamma}$ (see Appendix E for further details).

Uncertainty Quantification: We evaluate in total eight different approaches for the UQ of structured effects $\hat{\boldsymbol{\beta}}$. As alternative approaches to UQ for SSR models, we consider ensembles (Lakshminarayanan et al., 2017), which are frequently found to be among the best-performing methods in the field (Wilson and Izmailov, 2020) despite their simplicity, thus representing a likely choice of current SSR users seeking to perform UQ. We also consider Bayes-by-backprop (BBP; Blundell et al., 2015), a variational inference approach using a mean field approximation for all parameters of the SSR model. To quantify uncertainty with our proposed method, we consider two different ways of obtaining the uncertainty from the DNN: ensembles and Monte Carlo Dropout (MCD; Gal and Ghahramani, 2015).

[4] Note that Corollary 1 refers to OLS models and thus does not apply here.

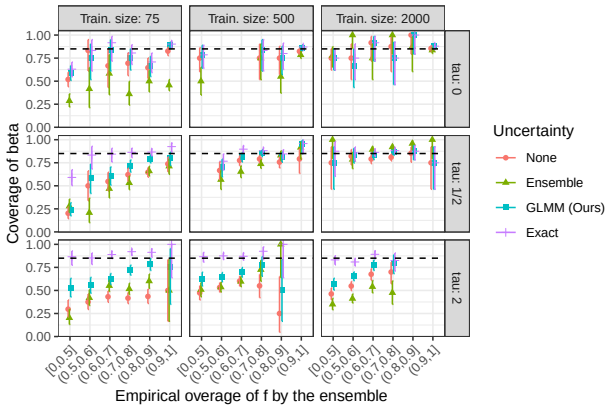

Figure 5: Comparison of coverage of $\boldsymbol{\beta}$ (y-axis) for different methods (colors) along different coverage values of $\boldsymbol{f}$ (x-axis) on datasets with different sizes (columns) and or varying DNN influence (rows).

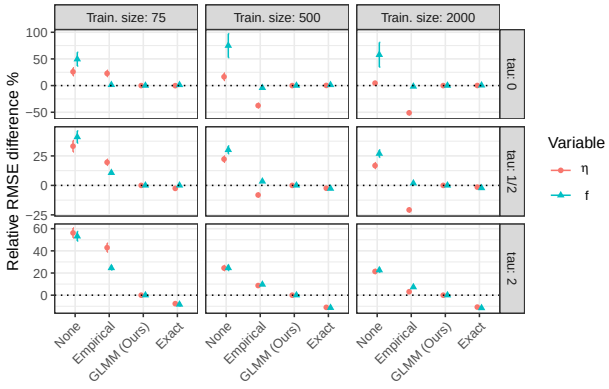

Figure 6: Relative difference in RMSE (y-axis) of the different UQ methods (x-axis) when predicting $\boldsymbol{\eta}$ (red circles) and $\boldsymbol{f}$ (blue triangles) compared to the GLMM uncertainty.

Both methods can be seen as producing samples from the posterior weight distribution of the DNN, from which we can derive both $\boldsymbol{z}$ and $\boldsymbol{\Gamma}$ (diagonal, as per Proposition 1) as their mean and variance, respectively. Moreover, having access to the true unstructured effect $\boldsymbol{f}$ (which is unavailable in real-world applications) allows us to simulate perfect DUQ by an oracle using the exact uncertainty $\boldsymbol{\Gamma}^{(e)}$ associated with $\boldsymbol{z}$:

$$\boldsymbol{\Gamma}_{ii}^{(e)} = \frac{1}{K-1}\sum_{k=1}^{K}(z_{ik} - f_i)^2, \tag{4}$$

where z_{ik} is the prediction for the i-th example by the k-th ensemble network / Monte Carlo sample, and $\boldsymbol{\Gamma}_{ij} = 0$ for $i \neq j$. Finally, we consider two uncertainty-unaware baselines obtained by discarding $\boldsymbol{\Gamma}$ from the Monte Carlo dropout approximation and aforementioned DNN ensembles. The former corresponds to fitting an SSR model with a single DNN and is understood to be the standard SSR training procedure with no additional UQ considerations, and the latter sheds light on how improved estimation of $\boldsymbol{z}$ can on its own improve inference of $\hat{\boldsymbol{\beta}}$.

Improved Inference of $\boldsymbol{\beta}$: The most immediately visible trend is an improvement in coverage as the size of the training set and the magnitude of the unstructured effect τ decrease. In all cases, the oracle GLMM provided higher coverage than the GLMM with estimated uncertainty, which in turn was better than the models that ignored DNN uncertainty (Figure 4). Confirming previous observations in the literature (Wilson and Izmailov, 2020), ensembles produce more reliable uncertainty estimates, which translates to better coverage from our GLMM models compared to MCD uncertainty. However, except in difficult scenarios with a small training set ($n = 75$), the oracle GLMMs reached nominal coverage (with ensemble uncertainty) or close to it (with MCD uncertainty with $\tau = 2$). Moreover, reaching nominal coverage with the oracle uncertainty but imperfect point predictions suggests that biased estimation of $\boldsymbol{f}$ by $\boldsymbol{z}$ does not negatively affect inference on $\boldsymbol{\beta}$, given that such a bias is reflected correctly in $\boldsymbol{\Gamma}$. This is an important finding, as in principle, it does not preclude the use of *post-hoc* DUQ methods – meaning that practitioners do not have to alter the training procedure of their DNN. These observations confirm that our approach is theoretically correct given perfect UQ, that a diagonal approximation to the uncertainty $\boldsymbol{\Gamma}$ is sufficient, and that violations of the normality assumption on the DNN predictions do not prevent our method from notably increasing coverage.

As for the baselines, the behavior of BBP is strongly dependent on both n and τ, where an increase in n decreases coverage and an increase in τ increases coverage. The ensemble of SSR models also presents highly variable behavior – considerably under-covering with $\tau = 2$ but generally reaching closer to the nominal coverage level as n increases. After controlling for τ and n, BBP did not significantly increase the coverage odds (CO) compared to the uncertainty-unaware baseline (CO=1.02, $p = 0.73$), and the ensemble of SSR models performed similarly as the GLMM with MCD uncertainty (CO=0.97, $p = 0.62$). Compared to the uncertainty-unaware baseline, our GLMM improved the CO by 35% using MCD uncertainty and 76% using the ensemble uncertainty, and our GLMM with ensemble uncertainty increased the CO by 36% compared to the SSR ensemble and by 40% compared to MCD uncertainty (all $p < 10^{-16}$).

Impact of DUQ on the Coverage of $\boldsymbol{\beta}$: We next investigate the connection between the quality of the estimated DNN uncertainty by the ensemble and the quality of $\hat{\boldsymbol{\beta}}$. We use $\boldsymbol{z}$ and $\text{diag}(\boldsymbol{\Gamma})$ to derive 85% confidence intervals for $\boldsymbol{f}$, and we compute the quality of the DNN uncertainty for a dataset as the average coverage of $\boldsymbol{f}$ across all data points.

	Pen.	Age (M)	Age (F)	aR^2	DE
No img.	.0002	.0028	.0019	.13	.30
No unc.	.0002	.032	.017	.11	.32
Ours	.1000	.044	.046	**.31**	**.39**

Table 1: Comparison of three models trained on the melanoma dataset, one using only structured features ("no img."), an SSR model without uncertainty ("no unc.") and one using our GLMM. The columns show the penalty for the patient random effect, the p-values of smooth age terms for male (M) and female (F) patients, the adjusted R^2 of the model, and the deviance explained (DE).

We then bin the coverage of $\boldsymbol{f}$, and for each bin, we compute the coverage of $\boldsymbol{\beta}$ achieved by the four UQ methods. For all methods except for the oracle UQ, coverage of $\boldsymbol{\beta}$ tends to increase as the DNN uncertainty is quantified more precisely, and again our GLMM provides confidence intervals with coverage closest to the nominal level (Figure 5) – significantly higher than the ensemble and GLM intervals.

Improved Inference of $\boldsymbol{f}$ and $\boldsymbol{y}$: We now analyze the effect of (D)UQ on prediction accuracy, again using the uncertainty derived from the ensemble. Correcting $\boldsymbol{z}$ with the random effects estimated by the GLMM provides new estimators $\hat{\boldsymbol{f}} = \hat{\boldsymbol{z}} + \hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{\eta}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{f}}$ that are more accurate than just using $\boldsymbol{z}$, reducing the root mean squared error (RMSE) of $\boldsymbol{\eta}$ and $\boldsymbol{f}$ by 20% or more compared to not performing DUQ and by about 10% compared to when the empirical method is used (Figure 6). The reduction in RMSE is particularly evident on smaller training datasets with a large unstructured effect, i.e., when DNN uncertainty plays a larger role. However, with large training sets and a weak unstructured effect, the empirical method achieves better predictions of $\boldsymbol{\eta}$.

4.3 Real-World Application

We finally apply our framework to the ISIC 2020 Challenge dataset (Rotemberg et al., 2021), which contains 32,531 dermoscopic images of skin lesions and information on the patient, with the goal of predicting whether the lesion is benign or malignant. As there could be several lesions associated with the same patient, we fit a binomial GAMM that includes a patient-specific random effect, a term for the site where the lesion was found, two smooth terms for the age of male and female patients, and the image of the lesion as non-tabular input modeled by a convolutional DNN. We omit all patients with less than four lesion images and use the remaining 1,409 patients (25,461 lesions, 389 malignant) to fit an ensemble of ten DNNs via stratified cross-validation by performing a grid search on the smoothing parameter for the patient random effect and selecting the best based on the model's deviance explained on a held-out dataset of 100 patients (3,217 lesions, 59 malignant). Finally, we use another held-out dataset of 100 patients (2,950 lesions, 54 malignant) for the final model fit and report results on this dataset (see Appendix F for more details).

The baseline that omits the lesion image yields an adjusted R^2 of 13% and explains 30% of the deviance, while the model that includes the average ensemble prediction *without* its uncertainty obtains an adjusted R^2 of 11% and 32% of deviance explained. Instead, our approach of incorporating the uncertainty of the ensemble obtains $R^2 = 31\%$ and explains 39% of the deviance (Table 1). The penalty associated with our model for the patient random effect is considerably larger compared to the uncertainty-unaware model, suggesting that the latter mistakenly attributes some of the DNN uncertainty to patient variability. As expected, our model presents larger p-values, reducing over-confidence by accounting for the DNN uncertainty (since the lesion site is never found to be significant by any model). Finally, using each network in the ensemble to build a separate SSR model results in adjusted R^2 values ranging from 8% to 11% and explains deviance values ranging from 24% to 31%.

5 DISCUSSION AND CONCLUSION

We showed how to perform valid statistical inference for the structured coefficients of a SSR model with theoretically-sound foundations, reducing the rate of Type-I errors and improving predictive performance. By analyzing the asymptotic variance-covariance of the structured coefficients we also showed that for additive SSR models explicit DNN uncertainty quantification is unnecessary (in the large-sample limit), and that (for small datasets) it can be replaced for pre-training on another larger (but relevant) dataset.

Experiments on simulated and real-world data confirmed the correctness of our approach as well as its superiority over alternative SSR UQ methods, including naive ensembling of SSR models and a variational approximation to the structured coefficients. However, it is important to note that the empirical performance of our method is inevitably tied to the performance of the specific method employed to quantify the DNN uncertainty, in the sense that bad estimation of the DNN output cannot be overcome by our method. Empirical benchmarking of several DUQ methods in conjunction with our framework thus constitutes an important future research direction. Further societal impacts of our work are discussed in Appendix G.

In conclusion, theoretically grounded and asymptotically correct statistical inference with our method reduces the rate of false discoveries and misleading interpretations of structured effects, thereby increasing the trustworthiness of SSR models and their applicability in safety-critical scenarios.

Acknowledgements

The authors thank the anonymous reviewers and Chris Kolb for their valuable suggestions. E. D. was supported by the Helmholtz Association under the joint research school "Munich School for Data Science - MUDS" (Award Number HIDSS-0006).

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P. W., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion*, 76:243–297.

Alaa, A. M. and van der Schaar, M. (2020). Discriminative jackknife: Quantifying uncertainty in deep learning via higher-order influence functions. *ArXiv*, abs/2007.13481.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2014). Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*.

Baumann, P. F. M., Hothorn, T., and Rügamer, D. (2021). Deep conditional transformation models. In *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 3–18, Cham. Springer International Publishing.

Begoli, E., Bhattacharya, T., and Kusnezov, D. (2019). The need for uncertainty quantification in machine-assisted medical decision making. *Nat Mach Intell*, 1(1):20–23.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *ICML*.

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25.

Casella, G. and Berger, R. L. (2021). *Statistical inference*. Cengage Learning.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20103.

Desquilbet, L. and Mariotti, F. (2010). Dose-response analyses using restricted cubic spline functions in public health research. *Statistics in medicine*, 29(9):1037–1057.

Fritz, C., Dorigatti, E., and Rügamer, D. (2022). Combining graph neural networks and spatio-temporal disease models to improve the prediction of weekly covid-19 cases in germany. *Scientific Reports*, 12(1):1–18.

Gal, Y. and Ghahramani, Z. (2015). Bayesian convolutional neural networks with bernoulli approximate variational inference. *ArXiv*, abs/1506.02158.

Gaube, S., Suresh, H., Raue, M., Merritt, A., Berkowitz, S. J., Lermer, E., Coughlin, J. F., Guttag, J. V., Colak, E., and Ghassemi, M. (2021). Do as ai say: susceptibility in deployment of clinical decision-aids. *NPJ digital medicine*, 4(1):1–8.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., Kruspe, A. M., Triebel, R., Jung, P., Roscher, R., Shahzad, M., Yang, W., Bamler, R., and Zhu, X. (2021). A survey of uncertainty in deep neural networks. *ArXiv*, abs/2107.03342.

Hastie, T. and Tibshirani, R. (1990). Generalized additive models. *Monographs on statistics and applied probability. Chapman & Hall*, 43:335.

Huang, S.-C., Chen, C.-C., Lan, J., Hsieh, T.-Y., Chuang, H.-C., Chien, M.-Y., Ou, T.-S., Chen, K.-H., Wu, R.-C., Liu, Y.-J., et al. (2022). Deep neural network trained on gigapixel images improves lymph node metastasis detection in clinical settings. *Nature Communications*, 13(1):1–14.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLOS Medicine*, 2(8).

Isobe, T., Takagi, M., Sato-Otsubo, A., Nishimura, A., Nagae, G., Yamagishi, C., Tamura, M., Tanaka, Y., Asada, S., Takeda, R., et al. (2022). Multi-omics analysis defines highly refractory ras burdened immature subgroup of infant acute lymphoblastic leukemia. *Nature communications*, 13(1):1–16.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. (2021). What are bayesian neural network posteriors really like? In *International Conference on Machine Learning*.

Kook, L., Herzog, L., Hothorn, T., Dürr, O., and Sick, B. (2022). Deep and interpretable regression models for ordinal outcomes. *Pattern Recognition*, 122:108263.

Kopper, P., Pölsterl, S., Wachinger, C., Bischl, B., Bender, A., and Rügamer, D. (2021). Semi-structured deep piecewise exponential models. In *Proceedings of AAAI Spring Symposium on Survival Prediction – Algorithms, Challenges, and Applications, PMLR*, pages 40–53.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*.

Lassau, N., Ammari, S., Chouzenoux, E., Gortais, H., Herent, P., Devilder, M., Soliman, S., Meyrignac, O., Talabard, M.-P., Lamarque, J.-P., et al. (2021). Integrating deep learning ct-scan model, biological and clinical variables to predict severity of covid-19 patients. *Nature communications*, 12(1):1–11.

Maddox, W. J., Garipov, T., Izmailov, P., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. In *Neural Information Processing Systems*.

Michelmore, R., Wicker, M., Laurenti, L., Cardelli, L., Gal, Y., and Kwiatkowska, M. (2020). Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

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

Rotemberg, V. M., Kurtansky, N. R., Betz-Stablein, B., Caffery, L. J., Chousakos, E., Codella, N. C. F., Combalia, M., Dusza, S. W., Guitera, P., Gutman, D., Halpern, A. C., Kittler, H., Köse, K., Langer, S. G., Liopryis, K., Malvehy, J., Musthaq, S., Nanda, J., Reiter, O., Shih, G., Stratigos, A. J., Tschandl, P., Weber, J., and Soyer, H. P. (2021). A patient-centric dataset of images and metadata for identifying melanomas using clinical context. *Scientific Data*, 8.

Rügamer, D., Kolb, C., Fritz, C., Pfisterer, F., Kopper, P., Bischl, B., Shen, R., Bukas, C., Barros de Andrade e Sousa, L., Thalmeier, D., Baumann, P. F. M., Kook, L., Klein, N., and Müller, C. L. (2023). deepregression: a flexible neural network framework for semi-structured deep distributional regression. *Journal of Statistical Software*, 105(1):1–31.

Rügamer, D., Kolb, C., and Klein, N. (2023). Semi-structured distributional regression – extending structured additive models by arbitrary deep neural networks and data modalities. *The American Statistician*, 0(ja):1–25.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press.

Shrout, P. E. and Rodgers, J. L. (2018). Psychology, science, and knowledge construction: Broadening perspectives from the replication crisis. *Annual Review of Psychology*, 69(1):487–510.

Taylor, J. and Tibshirani, R. J. (2015). Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634.

Verma, S. and Rubin, J. (2018). Fairness definitions explained. In *Proceedings of the International Workshop on Software Fairness*. ACM.

Wang, H. and Yeung, D.-Y. (2016). A survey on bayesian deep learning. *ACM Computing Surveys (CSUR)*, 53:1 – 37.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning*.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):817.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2015). Deep kernel learning. *ArXiv*, abs/1511.02222.

Wilson, A. G. and Izmailov, P. (2020). Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708.

Wolf, T. N., Pölsterl, S., and Wachinger, C. (2022). Daft: A universal module to interweave tabular data and 3d images in cnns. *NeuroImage*, 260:119505.

Wood, S. N. (2006). *Generalized additive models: an introduction with R*. chapman and hall/CRC.

Zheng, X., Yao, Z., Huang, Y., Yu, Y., Wang, Y., Liu, Y., Mao, R., Li, F., Xiao, Y., Wang, Y., et al. (2020). Deep learning radiomics can predict axillary lymph node status in early-stage breast cancer. *Nature communications*, 11(1):1–9.

A DEEP UNCERTAINTY QUANTIFICATION

Given the difficulty of exact DUQ, practitioners must often resort to cheaper approximations, the most common of which are Bayes-by-backprop (BBP, Blundell et al., 2015), Monte Carlo Dropout (MCD, Gal and Ghahramani, 2015), Deep Ensembles (Lakshminarayanan et al., 2017), the Laplace approximation (Daxberger et al., 2021), Stochastic Gradient Langevin Dynamics (SGLD, Welling and Teh, 2011), Stochastic Weight Averaging (SWAG, Maddox et al., 2019), Deep Kernel Learning (DKL, Wilson et al., 2015), the Jackknife (Alaa and van der Schaar, 2020), and many others. Given all these competing approaches, it is not trivial to understand when they work well and when they do not, with each method having their own strength and weaknesses and considerable effort spent on benchmarks and comparisons (Wilson and Izmailov, 2020; Izmailov et al., 2021; Abdar et al., 2021; Wang and Yeung, 2016; Gawlikowski et al., 2021). The necessity of benchmarking our method along with all these DUQ approaches is therefore unavoidable.

From a high level perspective, DUQ methods can be partitioned into those focusing on *weight-space* uncertainty and those focusing on *functional* uncertainty: while methods in the former class (including, e.g., BBP, SGLD and SWAG) try to derive a distribution for the model's parameters, methods in the latter class (e.g., MCD, DKL, and the Jackknife) only focus on deriving a distribution for the model's predictions. However, functional uncertainty methods cannot be applied as-is to our goal of deriving an estimator for the structured coefficients of a SSR model, thus excluding a significant portion of the literature on DUQ. Since our method only makes use of the DNN predictions and their distribution, it opens the door to using functional DUQ methods in SSR inference. Weight-space methods to quantify the uncertainty of the DNN predictions usually require repeatedly sampling from the posterior distribution of the weights and running a separate forward pass for each sample, aggregating the model's predictions into their mean and variance. As argued in Proposition 1, it is not necessary to quantify the entire variance-covariance matrix on held-out datasets, and as shown in Section 4.2, our method is robust towards non-normality of the predictions.

Another distinction of DUQ methods is between *ad hoc* and *post hoc* methods, where the former type of method estimates uncertainty during DNN training with specific procedures and the latter derives the uncertainty after the DNN is fully trained. BBP and DKL are examples of *ad hoc* methods, while Laplace and SWAG are examples of *post hoc* approaches. The method we proposed is also part of the *post hoc* category, as we require a trained DNN to derive an estimator for $\boldsymbol{\beta}$, although the DNN uncertainty can be derived using both method types during the first stage of training the SSR model.

B PROOFS

B.1 Proof of Theorem 1

The data generating process is $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon}$ with $\boldsymbol{f}$ being the true, unknown unstructured effect and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$. The unstructured effect $\boldsymbol{f}$ is estimated by an uncertainty-aware DNN as $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{f}, \boldsymbol{\Gamma})$ with known $\boldsymbol{\Gamma}$.

The estimator for $\boldsymbol{\beta}$ is obtained from $\boldsymbol{y} - \boldsymbol{z} = \boldsymbol{X}\boldsymbol{\beta}$ by multiplying both sides by $(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$ to the left. This estimator is unbiased:

$$\begin{aligned}
\mathbb{E}[\hat{\boldsymbol{\beta}}] &= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\mathbb{E}[\boldsymbol{y} - \boldsymbol{z}] && (5)\\
&= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} - \boldsymbol{f}) && (6)\\
&= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\beta}. && (7)
\end{aligned}$$

B.1.1 Confidence Intervals

To compute the variance of $\hat{\boldsymbol{\beta}}$, we leverage the identity $\mathbb{V}(\hat{\boldsymbol{\beta}}) = \mathbb{E}[\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top] - \mathbb{E}[\hat{\boldsymbol{\beta}}]\mathbb{E}[\hat{\boldsymbol{\beta}}]^\top$. From Equation (5) it follows $\mathbb{E}[\hat{\boldsymbol{\beta}}]\mathbb{E}[\hat{\boldsymbol{\beta}}]^\top = \boldsymbol{\beta}\boldsymbol{\beta}^\top$, while the first term is:

$$\begin{aligned}
\mathbb{E}[\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top] &= \mathbb{E}[(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{z})(\boldsymbol{y} - \boldsymbol{z})^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}] && (8)\\
&= (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top \cdot \mathbb{E}[(\boldsymbol{y} - \boldsymbol{z})(\boldsymbol{y} - \boldsymbol{z})^\top] \cdot \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1} && (9)
\end{aligned}$$

Its middle term expands to

$$\begin{aligned}
\mathbb{E}[(\boldsymbol{y} - \boldsymbol{z})(\boldsymbol{y} - \boldsymbol{z})^\top] &= \mathbb{E}[(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon} - \boldsymbol{z})(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon} - \boldsymbol{z})^\top] && (10)\\
&= \mathbb{E}[\boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\beta}^\top\boldsymbol{X} + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top + (\boldsymbol{f} - \boldsymbol{z})(\boldsymbol{f} - \boldsymbol{z})^\top + 2\boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\epsilon}^\top + 2\boldsymbol{X}\boldsymbol{\beta}(\boldsymbol{f} - \boldsymbol{z})^\top + 2(\boldsymbol{f} - \boldsymbol{z})\boldsymbol{\epsilon}^\top] && (11)\\
&= \boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\beta}^\top\boldsymbol{X} + \sigma^2\boldsymbol{I} + \boldsymbol{\Gamma}, && (12)
\end{aligned}$$

where we used the fact that $\boldsymbol{f} - \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma})$ and is independent of $\boldsymbol{\epsilon}$. Then,

$$\mathbb{E}[\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top] = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top \cdot (\boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\beta}^\top\boldsymbol{X}^\top + \sigma^2\boldsymbol{I} + \boldsymbol{\Gamma}) \cdot \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1} \tag{13}$$

$$= \boldsymbol{\beta}\boldsymbol{\beta}^\top + (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top(\sigma^2\boldsymbol{I} + \boldsymbol{\Gamma})\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}, \tag{14}$$

which implies that

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top(\sigma^2\boldsymbol{I} + \boldsymbol{\Gamma})\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1} \tag{15}$$

$$= \sigma^2(\boldsymbol{X}^\top\boldsymbol{X})^{-1} + (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{\Gamma}\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}, \tag{16}$$

where the second expression makes it clear how much variance is added by additionally estimating $\boldsymbol{f}$ through $\boldsymbol{z}$.

B.1.2 Residuals

The residuals are $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{z} = \boldsymbol{\epsilon} + (\boldsymbol{f} - \boldsymbol{z})$, and since $\boldsymbol{\epsilon} \perp\!\!\!\perp (\boldsymbol{f} - \boldsymbol{z})$, they are distributed as $\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I} + \boldsymbol{\Gamma})$. Thus,

$$\begin{aligned}\mathbb{E}[\boldsymbol{r}^\top\boldsymbol{r}] &\overset{(a)}{=} \operatorname{tr}\left\{(\sigma^2\boldsymbol{I} + \boldsymbol{\Gamma})\right\} \\ &\overset{(b)}{=} n\sigma^2 + \operatorname{tr}(\boldsymbol{\Gamma}),\end{aligned} \tag{17}$$

where $\operatorname{tr}(\cdot)$ is the trace operator, (a) follows due the quadratic form of the inner product of residuals with zero mean and covariance as in Equation (5), and (b) follows as $\operatorname{tr}(\cdot)$ is a linear mapping.

B.2 Proof of Theorem 2

The prediction intervals are constructed based on $\mathbb{V}[\boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{z}]$, computed using the variance decomposition. First, note that $\mathbb{E}[\boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{z}] = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f}$, thus,

$$\mathbb{V}[\boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{z}] = \mathbb{E}[(\boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{z})(\boldsymbol{X}\hat{\boldsymbol{\beta}} + \boldsymbol{z})^\top] - (\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f})(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f})^\top \tag{18}$$

$$= \sigma^2\boldsymbol{H} + (\boldsymbol{H} - \boldsymbol{I})\boldsymbol{\Gamma}(\boldsymbol{H} - \boldsymbol{I})^\top. \tag{19}$$

B.3 Proof of Proposition 1

Let $\mathcal{D} = \mathcal{P}_\mathcal{X} \times \mathcal{P}_\mathcal{U} \times \mathcal{P}_\mathcal{Y}$ be the data distribution with $\mathcal{P}_\mathcal{X}$, $\mathcal{P}_\mathcal{U}$ and $\mathcal{P}_\mathcal{Y}$ the distributions of, respectively, the tabular features, non-tabular features and response, and let $(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{U}}, \tilde{\boldsymbol{Y}}) \sim \mathcal{D}^n$ be the dataset used to fit the SSR model. Further, let $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ be the DNN predictions for two test data points $\boldsymbol{u}_i, \boldsymbol{u}_j \sim \mathcal{P}_\mathcal{U}$ independent of each other and of $\tilde{\boldsymbol{U}}$. Note that these predictions are obtained after the SSR model is fit on $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{U}}$ to predict $\tilde{\boldsymbol{Y}}$ and its parameters are fixed.

Then, Proposition 1 states that $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are conditionally independent of each other, given $(\boldsymbol{u}_i, \boldsymbol{u}_j, \tilde{\boldsymbol{X}}, \tilde{\boldsymbol{U}}, \tilde{\boldsymbol{y}})$. This can be proved by noting that changing $\boldsymbol{u}_i$ (respectively, $\boldsymbol{u}_j$) will not affect $\boldsymbol{z}_j$ (respectively, $\boldsymbol{z}_i$), as the entire SSR model (including the DNN) is fixed. Therefore, given a held-out dataset $(\boldsymbol{X}, \boldsymbol{U}, \boldsymbol{Y}) \sim \mathcal{D}^m$, the covariance $\boldsymbol{\Gamma}$ of $\boldsymbol{z}$ is a diagonal matrix. Moreover, the entries on the diagonal are i.i.d. because $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ are i.i.d. themselves.

B.4 Proof of Corollary 1

We assume that Corollary 1 is applied to a held-out dataset with tabular features $\boldsymbol{X}$ that was not used for training the DNN so that Proposition 1 applies, i.e., the predictions of any two samples are conditionally independent, given the training set and their features. Therefore:

$$\mathbb{E}[(\boldsymbol{X}^\top\boldsymbol{\Gamma}\boldsymbol{X})_{ij}] = \sum_{k=1}^{n}\sum_{\ell=1}^{n} x_{ki}x_{\ell j}\mathbb{E}[\Gamma_{k\ell}] \tag{20}$$

$$\overset{(a)}{=} \sum_{k=1}^{n} x_{ki}x_{kj}\mathbb{E}[\Gamma_{kk}] \tag{21}$$

$$:= \gamma^2\boldsymbol{X}^\top\boldsymbol{X} \tag{22}$$

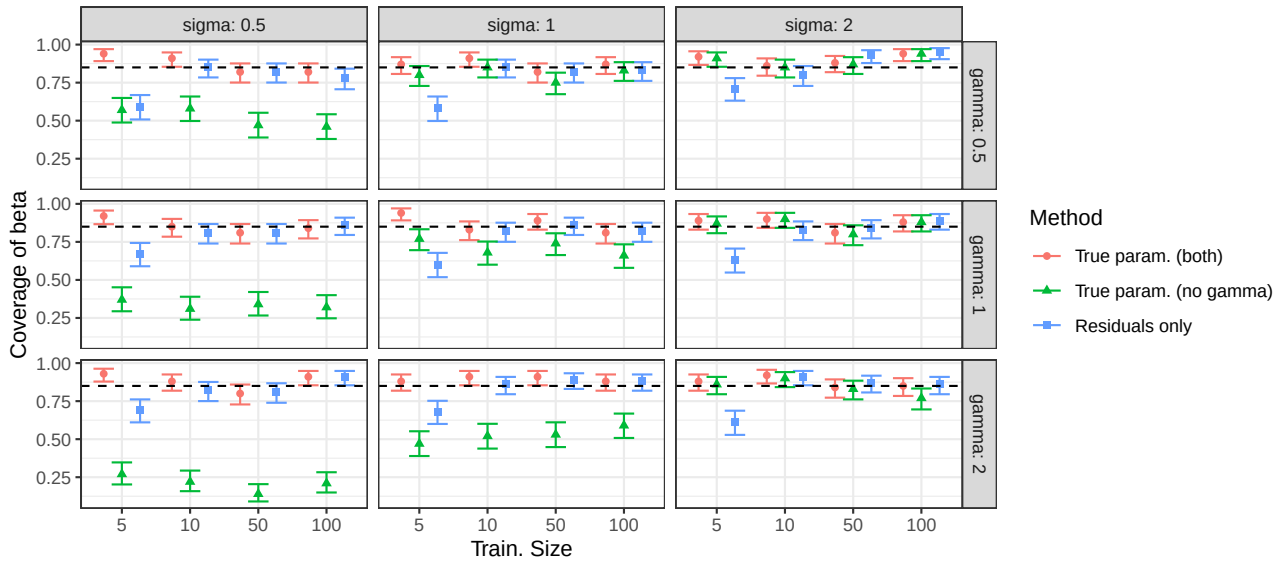

Figure 7: Demonstration of Corollary 1 on simulated data generated following the same protocol of Section 4.1 (using correlated $\boldsymbol{z}$) but with Gaussian response and identity link. The coverage of $\boldsymbol{\beta}$ is computed when using the true σ^2 and $\boldsymbol{\Gamma}$ to estimate $\mathbb{V}[\hat{\boldsymbol{\beta}}]$ (red circles), when only $\sigma^2\boldsymbol{I}$ is used (green triangles), and when $\boldsymbol{r}^\top\boldsymbol{r}/(n-d)$ is used (blue squares).

where step (a) follows because $\boldsymbol{\Gamma}$ is diagonal due to Proposition 1 and we use $\gamma^2 := \mathbb{E}[\Gamma_{kk}]$. Since the elements of $\boldsymbol{\Gamma}$ are i.i.d. (Proposition 1), we can use the strong law of large numbers to estimate γ^2 with $\text{tr}(\boldsymbol{\Gamma})/n$ with almost sure convergence guarantees. Using this estimator with Equation (22) and substituting into Equation (16) completes the proof in combination with Equation (17).

Plots in Figure 7 using simulated data confirm that the corollary holds even for relatively modest sample sizes. When γ^2 is comparable or larger compared to σ^2, the DNN uncertainty dominates and ignoring it completely results in under-coverage (green triangles), while using the residuals $\boldsymbol{r}$ provides nominal coverage in all cases where $n \geq 10$. However, with real DNNs, the number of required samples may be considerably larger.

B.5 Proof of Corollary 2

As in the proof of Corollary 1, we assume that a held-out dataset is used, and thus, we treat $\boldsymbol{\Gamma}$ as diagonal with elements $\gamma_1^2, \ldots, \gamma_m^2$ grouped into the vector $\boldsymbol{\gamma}$. Consider the singular-value decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, with $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}$. Then, $\boldsymbol{X}^\top\boldsymbol{\Gamma}\boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{\Gamma}\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$. Let $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$ be the i-th and j-th columns of $\boldsymbol{U}$, then we have

$$(\boldsymbol{U}^\top\boldsymbol{\Gamma}\boldsymbol{U})_{ij} = (\boldsymbol{U}^\top\text{diag}(\boldsymbol{\gamma}^2)\boldsymbol{U})_{ij} = \sum_k u_{ki}u_{kj}\gamma_k^2 = \langle \boldsymbol{U}_i \odot \boldsymbol{U}_j, \boldsymbol{\gamma}^2 \rangle \leq \sqrt{\left(\sum_k u_{ki}^2 u_{kj}^2\right) \cdot \left(\sum_k \gamma_k^4\right)}, \tag{23}$$

where $\odot$ is the element-wise (Hadamard) product, $\langle \cdot, \cdot \rangle$ is the dot-product, and the last step is due to the Cauchy-Schwartz inequality. Note that in general, $\sum_k a_k^2 \leq (\sum_k a_k)^2$ when $a_k \geq 0$ for all k. Let m be the number of samples used to pre-train the DNN. Then using $a_k = \gamma_k^2$ and assuming that $\gamma_k^2 = \mathcal{O}(1/m)$ (a reasonable assumption, formally proven in many common cases in statistical inference), we have that $\mathcal{O}(\sum_k \gamma_k^4) = \mathcal{O}(n/m^2)$ and $\mathcal{O}(\sum_k \gamma_k^2) = \mathcal{O}(n/m)$, and thus

$$\mathcal{O}\left(\sum_k \gamma_k^4\right) = \mathcal{O}\left(\frac{1}{m}\left(\sum_k \gamma_k^2\right)^2\right). \tag{24}$$

This allows us to continue the derivation as

$$(\boldsymbol{U}^\top \text{diag}(\gamma^2)\boldsymbol{U})_{ij} = \mathcal{O}\left(\sqrt{\frac{1}{m}\left(\sum_k \gamma_k^2\right)^2 \cdot \left(\sum_k u_{ki}u_{kj}\right)^2}\right) \tag{25}$$

$$= \mathcal{O}\left(\frac{1}{\sqrt{m}}\sum_k \gamma_k^2 \cdot \sum_k u_{ki}u_{kj}\right) \tag{26}$$

$$= \mathcal{O}\left(\text{tr}(\boldsymbol{\Gamma})\sqrt{m} \cdot (\boldsymbol{U}^\top \boldsymbol{U})_{ij}\right) \tag{27}$$

$$= \mathcal{O}\left(\text{tr}(\boldsymbol{\Gamma})/\sqrt{m} \cdot \boldsymbol{I}_{ij}\right), \tag{28}$$

where we could elide the square root in Equation (25), as both terms are non-negative before squaring. This implies that

$$\boldsymbol{X}^\top \boldsymbol{\Gamma} \boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top \text{diag}(\gamma^2)\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \tag{29}$$

$$= \mathcal{O}\left(\text{tr}(\boldsymbol{\Gamma})/\sqrt{m}\right) \cdot \boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V} \tag{30}$$

$$= \mathcal{O}\left(\text{tr}(\boldsymbol{\Gamma})/\sqrt{m}\right) \cdot \boldsymbol{X}^\top \boldsymbol{X}. \tag{31}$$

Plugging this into the expression for $\mathbb{V}[\hat{\boldsymbol{\beta}}]$ gives the desired result. Further assuming that $\sqrt{m} = \mathcal{O}(n)$ gives $\mathcal{O}(\sigma^2 + \text{tr}(\boldsymbol{\Gamma})/\sqrt{m}) = \mathcal{O}(\sigma^2 + \text{tr}(\boldsymbol{\Gamma})/n) = \mathcal{O}(\boldsymbol{r}^\top \boldsymbol{r}/n)$, as argued in the proof of Theorem 1.

B.6 Proof of Theorem 3

The coefficients are estimated by solving the standard penalized least squares problem:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{z}|| + \boldsymbol{\beta}^\top \boldsymbol{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta} = (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{z}) \tag{32}$$

Due to the penalty, we now obtain a biased estimate of $\boldsymbol{\beta}$:

$$\mathbb{E}\left[\hat{\boldsymbol{\beta}}\right] = (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon} - (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}\boldsymbol{X}^\top \boldsymbol{f} \tag{33}$$

$$= (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}(\boldsymbol{X}^\top \boldsymbol{X})\boldsymbol{\beta} \tag{34}$$

To compute the variance of such estimate, we use the equality $\mathbb{V}[\hat{\boldsymbol{\beta}}] = \mathbb{E}\left[\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top\right] - \mathbb{E}\left[\hat{\boldsymbol{\beta}}\right]\mathbb{E}\left[\hat{\boldsymbol{\beta}}\right]^\top$. Thus:

$$\mathbb{E}\left[\hat{\boldsymbol{\beta}}\right]\mathbb{E}\left[\hat{\boldsymbol{\beta}}\right]^\top = (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}(\boldsymbol{X}^\top \boldsymbol{X})\boldsymbol{\beta}\boldsymbol{\beta}^\top(\boldsymbol{X}^\top \boldsymbol{X})(\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1} \tag{35}$$

and:

$$\mathbb{E}\left[\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top\right] = (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}\boldsymbol{X}^\top \mathbb{E}\left[(\boldsymbol{y} - \boldsymbol{z})(\boldsymbol{y} - \boldsymbol{z})^\top\right]\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1} \tag{36}$$

$$= (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\beta}\boldsymbol{\beta}^\top \boldsymbol{X}^\top + \sigma^2 \boldsymbol{I} + \boldsymbol{\Gamma})\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}, \tag{37}$$

which gives:

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = (\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}\boldsymbol{X}^\top(\sigma^2 \boldsymbol{I} + \boldsymbol{\Gamma})\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X} + \boldsymbol{S}_{\boldsymbol{\lambda}})^{-1}. \tag{38}$$

$$\tag{39}$$

C ESTIMATION OF STRUCTURED COEFFICIENTS

We observed empirically that a naive estimation of $\hat{\boldsymbol{\beta}}$ on the same training set used to train the SSR model can (in some cases and especially with small training sets) result in an artificial shrinkage, i.e., a bias towards zero, as already noted by Rügamer et al. (2023). This bias can be avoided in two ways: (1) by warm-starting $\hat{\boldsymbol{\beta}}$ before training the SSR model as $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$, as suggested by Rügamer et al. (2023), or (2) by re-estimating $\hat{\boldsymbol{\beta}}$ after training the SSR model on a

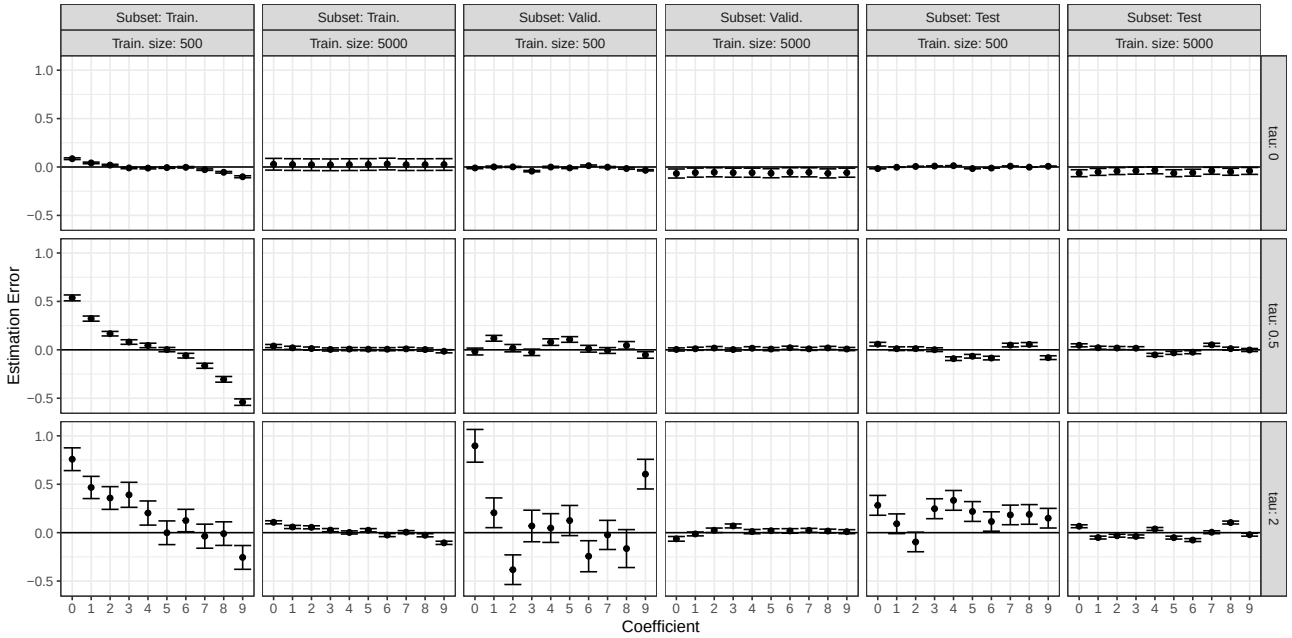

Figure 8: Estimation error (y-axis) for each coefficient (x-axis) when an OLS model is fit on tabular features and DNN predictions $\boldsymbol{z}$ without uncertainty on the same training set used to fit the SSR model (subset: train.), on the validation set used for early stopping (subset: valid.), and on an independent test set (subset: test), for training sets of size 500 and 5,000 samples.

held-out dataset. For simplicity, in all experiments we used the same validation set we used for early stopping, and while this may introduce some bias we have not observed any adverse effects.

We showcase this effect with a simulation study on a synthetic dataset constructed by drawing two random numbers a and b for each sample i, and generating the response as $y_i = (a - 5.5) + \tau(b - 5.5)$, where $\tau \in \mathbb{R}$. As tabular features, we encode a as a one-hot vector of dimension 10, so that the true coefficient β_j, $j \in \{1, \dots, 10\}$ has value $j - 5.5$. As non-tabular features, we encode b with a handwritten image of the digit n randomly sampled from the MNIST dataset. For each value of $\tau \in \{0, 1/2, 2\}$, we generate 25 datasets with 500 samples and 25 datasets with 5,000 samples, and fit an SSR model using a DNN with three convolutional layers with 32, 64 and 128 filters respectively of size 3×3 and identical padding, with leaky ReLU activation and interleaved with 2×2 max-pooling layers, after which we perform global average pooling, place a hidden fully-connected layer of 64 neurons and leaky ReLU activation, and the final output layer. We train the SSR model with a batch size of 128 and initial learning rate of 0.01, decimating it when no improvement in validation mean squared error (MSE) is observed for five epochs and early stopping of the training when the validation MSE does not improve for more than eight epochs. After the SSR model is trained, we fit an OLS model on training, validation, and test sets using the tabular features and the DNN predictions as fixed offset, and we compute the estimation error of $\hat{\boldsymbol{\beta}}$. Note that here we do not use any DUQ method, as we intend to demonstrate a general danger (unwanted shrinkage) when training SSR models.

A bias towards zero is clearly visible on the training set with 500 samples and $\tau = 2$, especially for the estimation of true coefficients with large absolute value, and decreases to almost zero as the influence of the non-tabular features vanishes as well as when the training set becomes larger (Figure 8). No systematic bias is observed on validation or test sets, although with small training sets, the estimation error is quite large.

D FITTING PROCEDURE FOR GLMMs AND GAMMs

Given that the conditional distribution of $\boldsymbol{y}$ given the features is in the exponential family, finding the Maximum Likelihood estimator for $\boldsymbol{\beta}$ for fixed $\hat{\boldsymbol{z}}$ and $\boldsymbol{b} = \hat{\boldsymbol{b}}$ in the model

$$\begin{aligned} \mathbb{E}[\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{z}] &= g^{-1}(\boldsymbol{X}\boldsymbol{\beta} + \hat{\boldsymbol{z}} + \boldsymbol{b}) \\ \boldsymbol{b} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Gamma}) \end{aligned} \tag{40}$$

is a convex problem and can be solved using the Fisher Scoring algorithm (see, e.g., Nelder and Wedderburn, 1972). Further, given fixed $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{z}}$, the optimization of (40) is also a convex problem in $\boldsymbol{b}$ and can also be solved using Fisher Scoring. When the only random effect in the GLMM is the DNN, the procedure proposed by Breslow and Clayton (1993) can be applied. This iterative procedure is initialized by taking $\boldsymbol{\mu}_0 = \boldsymbol{y} + \boldsymbol{\kappa}$ and $\boldsymbol{\eta}_0 = g(\boldsymbol{\mu}_0)$, with $\boldsymbol{\kappa}$ small constants to avoid numerical problems when applying g and g', and the procedure is repeated until $\hat{\boldsymbol{\beta}}_n$ and $\hat{\boldsymbol{b}}_n$ converge. The new estimates $\hat{\boldsymbol{\beta}}'$ and $\hat{\boldsymbol{b}}'$ of $\boldsymbol{\beta}$ and $\boldsymbol{b}$ are computed from the pseudo-responses $\boldsymbol{Y}_n$ as follows:

$$\hat{\boldsymbol{\eta}}_n = \boldsymbol{X}\hat{\boldsymbol{\beta}}_n + \hat{\boldsymbol{z}} + \hat{\boldsymbol{b}}_n \tag{41}$$

$$\hat{\boldsymbol{\mu}}_n = g^{-1}(\hat{\boldsymbol{\eta}}_n) \tag{42}$$

$$\boldsymbol{Y}_n = \hat{\boldsymbol{\eta}}_n - \hat{\boldsymbol{z}} + \text{diag}(\boldsymbol{y} - \hat{\boldsymbol{\mu}}_n)g'(\hat{\boldsymbol{\mu}}_n) \tag{43}$$

$$\boldsymbol{V}_n = \boldsymbol{W}_n^{-1}\phi + \boldsymbol{\Gamma} \tag{44}$$

$$\hat{\boldsymbol{\beta}}_{n+1} = (\boldsymbol{X}^\top \boldsymbol{V}^{-1} + n\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{V}_n^{-1}\boldsymbol{Y}_n \tag{45}$$

$$\hat{\boldsymbol{b}}_{n+1} = \boldsymbol{\Gamma}\boldsymbol{V}_n^{-1}(\boldsymbol{Y}_n - \boldsymbol{X}\hat{\boldsymbol{\beta}}_{n+1}), \tag{46}$$

where $\boldsymbol{W}_n$ are the GLM weights and ϕ is the scale parameter of $\boldsymbol{y}$.

For designs with more random effects or smooth terms, a more general approach is needed. As the optimization problem in $\boldsymbol{\beta}, \boldsymbol{b}$ is also convex, a cyclic coordinate-wise optimization routine (see, e.g., Boyd et al., 2004) alternating the two Fisher Scoring steps can be used to find the joint optimum. Similarly, a GAM formulation of our SSR model can be optimized using the same routine, as the defined penalties for $\boldsymbol{b}$ and smooth functions are separable. In particular, such an alternating procedure allows using readily available software for the optimization of $\boldsymbol{\beta}$ (by introducing $\hat{\boldsymbol{z}} + \hat{\boldsymbol{b}}$) as an offset, while the optimization of $\boldsymbol{b}$ given $\boldsymbol{\Gamma}$ and other effects is straightforward to implement.

This leads to Algorithm 1, which can be used to estimate SSR models based on individual optimization routines `opt1` and `opt2` (such as Fisher Scoring) for the optimization problem in $\boldsymbol{\beta}$ and $\boldsymbol{b}$, respectively (also returning the Hessian $\mathcal{H}$ of the GLM/GAM). If the model in (40) contains additional random effects based on structured features, a GLMM fitting routine such as in Bates et al. (2014) can be used in the optimization step `opt1`.

Algorithm 1 SSR Optimization for Inference

Input: $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \hat{\boldsymbol{z}}, \boldsymbol{\Gamma}$; small constant ξ;
Set $\hat{\boldsymbol{b}}^{(0)} = \boldsymbol{0}$; $i = 1$; $\delta = \xi + 1$;
while $\xi < \delta$ **do**
  $\hat{\boldsymbol{\beta}}^{(i)} = \texttt{opt1}(\boldsymbol{b}^{(i-1)}, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \hat{\boldsymbol{z}})$;
  $\hat{\boldsymbol{b}}^{(i)}, \mathcal{H} = \texttt{opt2}(\boldsymbol{\beta}^{(i)}, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\lambda}, \hat{\boldsymbol{z}}, \boldsymbol{\Gamma})$;
  $\delta = \max\left\{|\hat{\boldsymbol{\beta}}^{(i)} - \hat{\boldsymbol{\beta}}^{(i-1)}|, |\hat{\boldsymbol{b}}^{(i)} - \hat{\boldsymbol{b}}^{(i-1)}|\right\}$;
  $i = i + 1$;
end while
Output: $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^{(i-1)}, \hat{\boldsymbol{b}} = \hat{\boldsymbol{b}}^{(i-1)}, \mathcal{H}$

E POISSON SIMULATION DETAILS

We simulate the unstructured effect f_i of sample i by randomly sampling a vector $\boldsymbol{u}_i$ of $d_u = 8$ normally and independently distributed features, i.e., $u_{ij} \sim \mathcal{N}(0, 1)$. We then transform this vector with a randomly initialized DNN h with one hidden layer with eight neurons and $\tanh$ activation and linear output activation to obtain $f_i := h(\boldsymbol{u}_i)$, and we finally normalize $\boldsymbol{f}$ to unit standard deviation. We also sample tabular features x_{ij} and β_j from $\mathcal{N}(0, 1)$ and obtain the linear predictor $\eta_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \tau f_i$, removing all outliers $|\eta_i| \geq 3$ before obtaining the observations $y_i \sim Poi(\exp(\eta_i))$.

To fit the dataset, we use a DNN with three hidden layers of 12 units each and ReLU activation. We initialize the structured coefficients as $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$ and train the SSR model for 75 epochs with batch size of 128, weight decay of 10^{-5} and learning rate of 0.075, halving the learning rate if the validation loss did not improve in the last five epochs and early stopping if the validation loss on an external dataset of 100 samples did not improve in the last ten epochs.

F REAL-WORLD APPLICATION DETAILS

Dataset The original dataset contains 33,126 data points and 2,056 patients, from which we omit 595 data points with missing information as well as all patients with less than four lesions, resulting in a final data set with 32,236 data points and 1,759 patients.

Tabular features We model the tabular features using dummy encoding for the site where the lesion image was taken (head/neck: 1,751 images, upper extremity:4,835 images, lower extremity: 8,223 images, oral/genital: 118 images, palms/soles: 365 images, torso: 16,336 images), the sex of the patient (female: 15,340 images, male: 16,288 images), the age of the patient (mean: 48, median: 50, first quartile: 40, third quartile: 60, minimum: 10, maximum: 40), an interaction term between sex and age, and a patient-specific random effect.

Non-tabular features We resize all images to 128×128 and normalize the intensity of each channel separately between 0 and 1. The effect of the skin lesion image was modeled with a convolutional DNN that used two blocks of two 3×3 convolutional layers followed by 2×2 max pooling, the first block having 16 filters and the second block having 32 filters. After the second max pooling, the filters are flattened into a vector of size 26,912 units, followed by a fully connected layer with 256 units and finally the output layer. Each layer is followed by ReLU activation and dropout with $p = 0.3$ except for the output layer that had linear activation and no dropout.

Model training We use a random selection of 1,409 patients for training the SSR models, performing a grid search on the penalty of the patient random effect using the values $\{1 \times 10^{-1}, 2 \times 10^{-1}, 5 \times 10^{-1}, 1 \times 10^{-2}, 2 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$. For each value, we performed ten-fold stratified cross-validation, training the SSR model with a batch size of 1,024, learning rate of 0.001, weight decay of 0.0001, exponential learning rate decay with exponent 0.99, and clipping gradients to a maximum norm of 10. We monitored the area under the precision-recall curve (AUPRC) on the validation set and early-stopped training when the AUPRC did not improve for 20 epochs.

Penalty tuning and model testing For each fold, we use the model with the highest validation AUPRC to predict the unstructured effect of the lesion image of the remaining 340 patients, thus obtaining the predictions of ten networks for each random effect penalty and computing $\boldsymbol{z}$ as the average ensemble prediction and $\boldsymbol{\Gamma}$ as a matrix with the variance of the predictions as diagonal. We split the 340 patients into a tuning subset of 170 patients and a testing subset of 170 patients. For each penalty value, we use the tuning subset to fit a GLMM with $\boldsymbol{z}$ as a fixed offset and select the model with the largest explained deviance. We then use the predictions of this model on the testing subset to fit another GLMM, whose results are presented in the main text as the model without uncertainty. Similarly, we use both $\boldsymbol{z}$ and $\boldsymbol{\Gamma}$ and the fitting procedure of Appendix C to fit a model on the tuning subset for each penalty value, selecting the penalty which explains the largest deviance and using the corresponding predictions on the test set to refit and interpret the model.

G SOCIETAL IMPACT

With its rigorous theoretical foundations, our work contributes to less heuristic and more principled applications of deep learning, resulting in more reliable and trustworthy outcomes. As opposed to traditional, uncertainty-unaware SSR models that provide overconfident statements, the main outcome of our framework is wider confidence intervals for the structured coefficients, resulting in more conservative inference and thus reduced false discovery rate. In light of the reproducibility crisis plaguing certain research areas (Shrout and Rodgers, 2018), we thus hope that our work will reduce the effort wasted by the research community as a result of incorrect inferences of early works.

With our work, we also hope to facilitate the use of deep learning techniques in applications that still rely on classical statistical inference to control the risk of harmful decisions, without sacrificing UQ and interpretability of the inferred effects. We also hope that our method would benefit other additional use-cases of important practical concern, such as using the DNN uncertainty for anomaly detection and thus refraining from predicting anomalous inputs or appropriately warning end-users about the issue. However, as a methodological paper, many ethical issues related to our work are highly dependent on the specifics of the practical application that it is applied to, and especially in data-driven fields, practitioners should take extreme care in the way hypotheses are generated and tested to avoid misleading results (Ioannidis, 2005; Taylor and Tibshirani, 2015).