Bayesian Semi-structured Subspace Inference

Daniel Dold
HTWG Konstanz

David Rügamer
LMU Munich and MCML

Beate Sick
UZH and ZHAW Zurich

Oliver Dürr
HTWG Konstanz and TIDIT

Abstract

Semi-structured regression models enable the joint modeling of interpretable structured and complex unstructured feature effects. The structured model part is inspired by statistical models and can be used to infer the input-output relationship for features of particular importance. The complex unstructured part defines an arbitrary deep neural network and thereby provides enough flexibility to achieve competitive prediction performance. While these models can also account for aleatoric uncertainty, there is still a lack of work on accounting for epistemic uncertainty. In this paper, we address this problem by presenting a Bayesian approximation for semi-structured regression models using subspace inference. To this end, we extend subspace inference for joint posterior sampling from a full parameter space for structured effects and a subspace for unstructured effects. Apart from this hybrid sampling scheme, our method allows for tunable complexity of the subspace and can capture multiple minima in the loss landscape. Numerical experiments validate our approach's efficacy in recovering structured effect parameter posteriors in semi-structured models and approaching the full-space posterior distribution of MCMC for increasing subspace dimension. Further, our approach exhibits competitive predictive performance across simulated and real-world datasets.

1 INTRODUCTION

A linear model is inherently transparent and interpretable due to its model structure and underlying assumptions. When given features denoted as $\mathbf{x}$, the expected outcome $\mathbb{E}(y|\mathbf{x})$ for a variable of interest $y \in \mathbb{R}$ is estimated as a linear combination $\mathbf{x}^\top\boldsymbol{\theta}$ of features $\mathbf{x} \in \mathbb{R}^p$ and corresponding parameters $\boldsymbol{\theta} \in \mathbb{R}^p$. This interpretability is also preserved in the various extensions such as generalized linear models (GLMs; Nelder and Wedderburn, 1972) for non-Gaussian conditional outcome distributions or generalized additive models (GAMs; Wahba, 1990; Wood, 2017) for the inclusion of non-linearity via splines. While these extensions allow for a flexible definition of univariate or moderate-dimensional multivariate feature effects, they lack flexibility for complex higher-order interactions and are restricted to tabular features. A deep neural network (DNN), on the other hand, learns complex feature effects in a data-driven fashion and can work for different input modalities (e.g., image data). The fusion of a structured and interpretable statistical model with highly flexible DNNs thus has some attractive properties and has been investigated over the last 30 years (cf. Section 2.1).

While this combination is flexible and attractive from a modeling point of view, many properties of this so-called semi-structured regression (SSR) are yet still unexplored. One important aspect is their uncertainty quantification, particularly relevant in their application in the medical domain (Dorigatti et al., 2023). Although some of the more recent approaches account for aleatoric (Rügamer et al., 2023) or epistemic uncertainty in SSR models (Dürr et al., 2022; Dorigatti et al., 2023), all existing approaches do either not account for the epistemic uncertainty arising from the model's DNN part or assume this uncertainty to be given. Another significant but understudied challenge in traditional SSR models is the joint optimization of the two model components. On the one hand, DNNs can theoretically fit the training data perfectly, potentially leaving little to explain for the structured part(Zhang et al., 2017, 2021). Optimization of structured models such as GLMs, on the other hand, is typically done using more advanced second-order methods. This optimization asymmetry in SSR complicates the process of joint optimization.

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

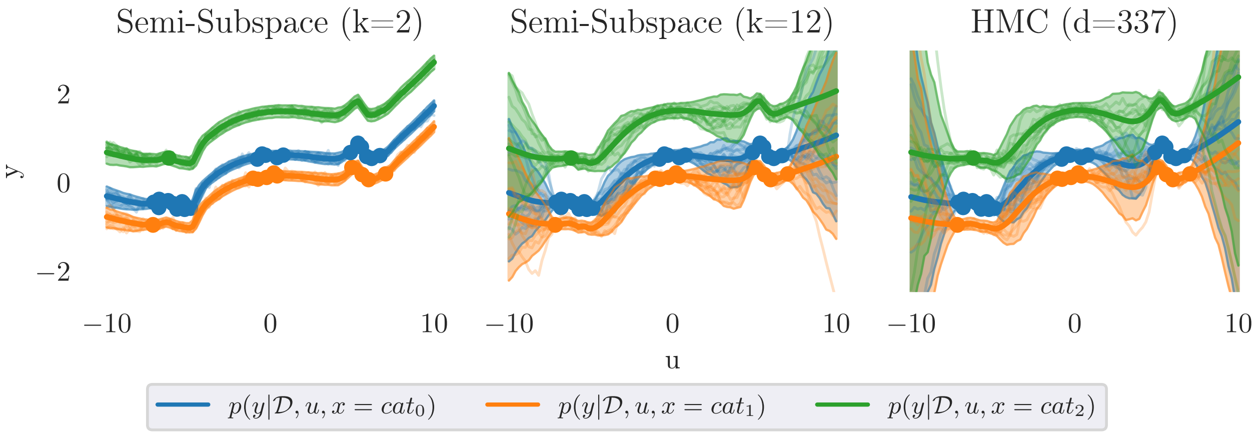

Figure 1: Comparison of semi-structured subspace inference and Hamilton Monte Carlo (HMC) for an SSR model. The SSR is defined as a combination of a linear shift induced by the categorical feature x (color code) and a non-linear trend in u (x-axis) modeled by a deep neural network (cf. Equation 1). Left/center: posterior predictive for dataset $\mathcal{D}$ and outcome y with a 2-dim. and 12-dim. subspace; right: posterior predictive of HMC without any approximation. Points represent the data, colored by their category of x, the solid line is the mean, and shading depicts the 95% Highest Density Interval.

The Bayesian paradigm offers a rigorous framework for quantifying uncertainty and Markov Chain Monte Carlo (MCMC) methods, relying on sampling rather than optimization, are often considered as the gold standard for inference in Bayesian neural networks (Wiese et al., 2023). Hence, a Bayesian variant of SSR models could provide inference statements and circumvent the aforementioned issues with the joint optimization of structured and DNN model parts. These sampling-based approaches, however, are computationally intensive and struggle with high-dimensional parameter spaces typical for DNNs.

Our Contribution In this work, we present *semi-structured subspace inference*, a sampling-based method that not only captures aleatoric and epistemic uncertainty in SSR models but also addresses the optimization asymmetry often observed in such models. Our method allows obtaining the posterior for every structured model parameter while accounting for the DNN's uncertainty. By using an adjustable subspace approximation of the DNN part, it is compatible with common MCMC methods. We show that semi-structured subspace inference 1) yields nearly the same posterior distribution as full-space MCMC methods for the structured model component, and 2) provides posterior predictive distributions of the quality of full-space inference even when using a highly-compressed subspace (see Figure 1). We further provide numerical evidence confirming the efficacy of our approach and superiority when compared to other Bayesian approximation methods.

2 RELATED WORK

Before introducing our method in Section 3, we briefly introduce SSR and Subspace inference in the following.

2.1 Semi-Structured Regression

The fusion of structured models from statistics and (deep) neural networks started with Ciampi and Lechevallier (1995, 1997), followed by extensions to model generalized additive neural networks (Potts, 1999; de Waal and du Toit, 2007; De Waal and Du Toit, 2011). In recent years, this combination has returned to the limelight under the name of *wide and deep learning* (Cheng et al., 2016) or semi-structured regression (SSR; Rügamer, 2023). Due to its flexibility, SSR has been adapted for various scenarios such as Deep GLMs (Tran et al., 2020), Deep Bayesian regression (Hubin et al., 2018), survival analysis (Pölsterl et al., 2020; Kopper et al., 2022), state space models (Amoura et al., 2011), transformation models (Baumann et al., 2021; Sick et al., 2021), ordinal (Kook et al., 2022b) or distributional regression (Rügamer et al., 2023). The question of how uncertainty can be quantified in a combination of an (unstructured) DNN and a structured regression model has however received not much attention. Only recently, Dorigatti et al. (2023) showed that for given DNN uncertainty, it is possible to derive the uncertainty for the structured model parameters in SSR models in a frequentistic manner. While their derived confidence intervals achieve nominal coverage when the deep uncertainty quantification method works well, they also point out the failure of the method if the uncertainty of the DNN is not well quantified. Their approach further leaves various points unanswered as it only focuses on the structured parameter uncertainty and cannot be embedded in a Bayesian setting despite many of the DNN uncertainty quantification methods being motivated in a Bayesian context (e.g., Daxberger et al., 2021; Izmailov et al., 2020). Another option to account for uncertainties and improve model performance are deep ensembles (Lakshminarayanan et al., 2017). While deep ensembling was adapted for semi-structured models (Kook et al., 2022a), it only accounts for algorithmic uncertainty in the DNN model part and cannot be considered fully Bayesian.

2.2 Bayesian Approximations and Subspace Inference

In complex Bayesian models that have many parameters, and where MCMC is not computationally feasible, Laplace approximation (Daxberger et al., 2021) provides a tractable alternative. This method approximates the posterior distribution with a Gaussian distribution centered at a single mode. However, this simplification neglects the potentially multimodal nature of the posterior in complex models. In contrast, subspace inference (Izmailov et al., 2020) provides an approach capable of capturing multiple modes in the posterior. This is achieved by defining a lower-dimensional subspace within the parameter space that can accommodate multiple modes. This subspace facilitates efficient posterior sampling using MCMC methods. Two methods were proposed to construct this subspace within the high-dimensional weight space of a neural network: the first employs principal component analysis on weights collected during the last training epochs, which typically corresponds to a single minimum in the loss space and hence captures a single mode of the posterior; the second method from Garipov et al. (2018) connects two local minima in the loss landscape using a quadratic Bézier curve, enriching the model's uncertainty by potentially capturing multiple posterior modes. Control points of the Bézier curve are determined by weights resulting from two independent training runs, with a third optimization refining the last control point to ensure all weights along the curve yield well-performing models. Izmailov et al. (2020) empirically showed that using a quadratic Bézier with two connected modes outperformed the principal component analysis approach. Thus our work exclusively adopts the Bézier approach for subspace inference. While effective, this method restricts the subspace dimension to two dimensions and necessitates multiple training runs. Wortsman et al. (2021) improved upon this by developing an algorithm that works within a single training run, but also uses the quadratic Bézier curves. To the best of our knowledge, there has been no further development building on Wortsman et al. (2021) or extending the Bézier curve approach originally introduced by Garipov et al. (2018). Recent work by Jantre et al. (2023) incorporates output information but remains confined to exploring a single mode of the posterior.

3 SEMI-STRUCTURED SUBSPACE INFERENCE

A semi-structured regression model is defined as an additive combination of a structured model part, capturing the interpretable effect of tabular input features $\mathbf{x} \in \mathbb{R}^p$, and an unstructured model part processing complex effects of a potentially complex input $\mathbf{u} \in \mathcal{U}$ through a DNN. While the class of models is not restricted to certain regression models and the structured model part can be flexibly defined, we here focus on mean regression approaches, where the mean μ of some distribution is modeled as a semi-structured predictor of the form

$$\mu = \mathbf{x}^\top \boldsymbol{\theta} + \mathrm{DNN}(\mathbf{u}). \qquad (1)$$

In (1), $\boldsymbol{\theta} \in \mathbb{R}^p$ are the interpretable parameters of the structured model part and DNN : $\mathcal{U} \to \mathbb{R}$ is parametrized with weights $\mathbf{w} \in \mathbb{R}^d$, where d is usually large. We use this simple definition for better readability but note that extending our approach to models with more complex structured effects such as splines or distribution regression approaches as discussed in Section 2.1 is straightforward. As the parameters $\boldsymbol{\theta}$ of the structured part play a special role, quantifying their uncertainty is one of our primary goals. One option to quantify the uncertainty is to employ a Bayesian approach.

Naïve Approximation Methods Translating SSR models into a Bayesian framework requires some form of approximation as classical MCMC is infeasible for the model's unstructured DNN part. A naïve approach to implement approximation techniques such as Laplace approximation or subspace inference for SSR models would be to treat the parameters from the structured model part $\boldsymbol{\theta}$ without special attention. This would mean simply extending the d-dimensional DNN weight space with the p-dimensional space of $\boldsymbol{\theta}$ and applying the original approximation to the combined $(d + p)$-dimensional space without further modification. However, by not differentiating between the small parameter set $\boldsymbol{\theta}$ and the overparameterized weights of a DNN, these naive approaches constrain the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ in its shape (e.g., a unimodal Gaussian distribution for Laplace approximation) or flexibility (cf. Figure 3 for subspace inference).

Semi-Structured Subspace Inference Our method is an extension of subspace inference, originally introduced by Izmailov et al. (2020), tailored specifically for SSR models. The core premise of our approach is a dimensionality reduction applied to the weight vector $\mathbf{w}$ of the neural network to a k-dimensional subspace, allowing sampling in $\mathbb{R}^k \times \mathbb{R}^p$, with $k \ll d$, instead of $\mathbb{R}^d \times \mathbb{R}^p$. In the following, we elaborate on how our method incorporates elements from Izmailov et al. (2020) while introducing our own adaptations and enhancements to better suit SSR models. The design of the sampling space is guided by the following criteria. First, the subspace must be sample-efficient, concentrating on regions of the loss landscape with low loss values. Second, it must encompass a diverse set of weight configurations that correspond to small loss values. Third, the subspace should facilitate smooth and rapid traversal between distinct low-loss regions, thereby enabling the exploration of a diverse set of solutions. Finally, the subspace construction must be aware of the structured model part $\mathbf{x}^\top\boldsymbol{\theta}$ as $\mathbf{x}^\top\boldsymbol{\theta}$ can have a significant impact on the loss landscape of the DNN component.

Algorithm 1 Subspace construction

```
Initialize weights $\mathbf{p}_0, \ldots, \mathbf{p}_k$ and $\boldsymbol{\theta}$ randomly
while validation loss still reducible do
    for each minibatch $\mathcal{B}$ of training data $\mathcal{D}$ do
        Sample $t \sim U(0,1)$
        Compute $\mathcal{L}(\mathbf{b}_{\boldsymbol{\Lambda}}(t), \boldsymbol{\theta})$ and gradients $\nabla\mathcal{L}$
        Update $\mathbf{p}_0, \ldots, \mathbf{p}_k$ and $\boldsymbol{\theta}$ using any
            SGD variant (e.g., Adam)
    end for
end while
return optimized $\mathbf{p}_0^*, \ldots, \mathbf{p}_k^*$ and $\boldsymbol{\theta}^*$
```

3.1 Construction of the Approximate Sampling Space

In line with these guiding principles, we introduce a parametric path $\mathbf{b}_{\boldsymbol{\Lambda}} : [0, 1] \to \mathbb{R}^d$ that interconnects weight vectors $\mathbf{p}_l$, $l = 0 \ldots k$, of $k + 1$ neural network parametrizations, within the d-dimensional DNN weight space (see Figure 2). This path is formulated using a Bézier curve with the weight vectors $\mathbf{p}_l$ serving as control points:

$$\mathbf{b}_{\boldsymbol{\Lambda}}(t) = \sum_{l=0}^{k} \binom{k}{l} (1-t)^{k-l} t^l \mathbf{p}_l. \tag{2}$$

The parameterization of the curve is given by $\boldsymbol{\Lambda} := (\mathbf{p}_0, \ldots, \mathbf{p}_k)$. Each point $\mathbf{b}_{\boldsymbol{\Lambda}}(t)$ together with the parameters $\boldsymbol{\theta}$ comprises a parameter set of an SSR model with loss $\mathcal{L}(\mathbf{b}_{\boldsymbol{\Lambda}}(t), \boldsymbol{\theta})$. Empirical evidence, as outlined in Garipov et al. (2018), indicates the presence of a low-loss valley between distinct SGD solutions. Following their approach, we minimize the functional $L(\boldsymbol{\theta}, \boldsymbol{\Lambda})$ defined as

$$L(\boldsymbol{\theta}, \boldsymbol{\Lambda}) = \int_0^1 \mathcal{L}(\mathbf{b}_{\boldsymbol{\Lambda}}(t), \boldsymbol{\theta})\, dt. \tag{3}$$

We compute an unbiased estimate of the objective (3) by sampling $t \sim U(0, 1)$ and update $(\boldsymbol{\theta}, \boldsymbol{\Lambda})$ via minibatch gradient descent (Algorithm 1). The optimal parameters $\boldsymbol{\Lambda}^*$ define a k-dimensional (affine) subspace in $\mathbb{R}^d$

$$\text{AffSpan}(\boldsymbol{\Lambda}^*) = \left\{ \mathbf{p}_0^* + \sum_{i=1}^{k} \varphi_i' \Delta\mathbf{p}_i^* \;\middle|\; \varphi_i' \in \mathbb{R} \right\}, \tag{4}$$

where $\Delta\mathbf{p}_i^* = (\mathbf{p}_i^* - \mathbf{p}_0^*)$, and $\boldsymbol{\Lambda}^* = (\mathbf{p}_0^*, \ldots, \mathbf{p}_k^*)$, which includes the estimated low-loss valley given by the Bézier curve (see Supplementary Material B.1 for a proof). Figure 2 illustrates this concept for $k = 2$ and $d = 3$. In contrast to Izmailov et al. (2020), our approach streamlines subspace construction by training the Bézier curve model in a single stage, eliminating the need for sequential training of $\mathbf{p}_0^*$, $\mathbf{p}_2^*$, and $\mathbf{p}_1^*$ with fixed $\mathbf{p}_0^*$ and $\mathbf{p}_2^*$.

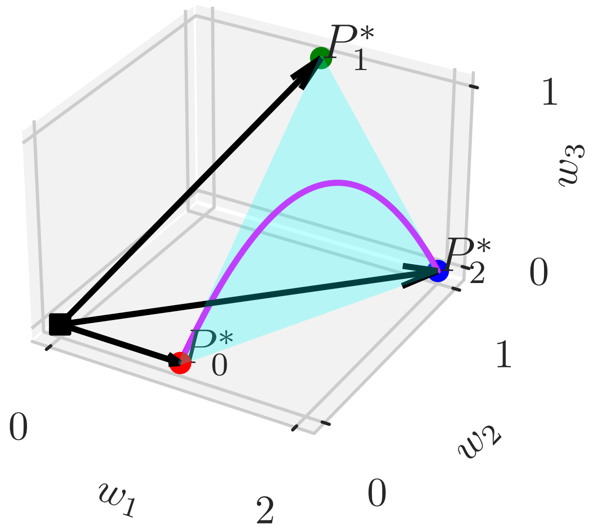

Figure 2: Bézier curve (magenta) in three-dimensional weight space, controlled by optimized points $\mathbf{p}_0^*, \mathbf{p}_1^*, \mathbf{p}_2^*$, which form a two-dimensional subspace AffSpan$(\mathbf{p}_0^*, \mathbf{p}_1^*, \mathbf{p}_2^*)$ indicated by the cyan triangle that includes the Bézier curve. The difference vectors $\mathbf{p}_1^* - \mathbf{p}_0^*$ and $\mathbf{p}_2^* - \mathbf{p}_0^*$ spanning the affine subspace are shown in green.

3.2 Subspace Sampling

In order to approximate the DNN part's posterior, we sample weights in the k-dimensional affine subspace defined in (4). To ensure compatibility with the methodology presented in Izmailov et al. (2020) for the case $k = 2$, we adopt an orthogonal coordinate system for sampling, as opposed to directly sampling φ' in (4). Specifically, we perform a translation to center the controlling points $\mathbf{p}_i^*$ around their mean vector, given by $\bar{\mathbf{p}} = \frac{1}{k+1}\sum_{i=0}^{k}\mathbf{p}_i^*$. Subsequently, we perform a principal component analysis on the centered points to construct an orthogonal projection matrix, denoted as $\mathbf{\Pi} : \mathbb{R}^k \to \mathbb{R}^d$. This matrix $\mathbf{\Pi} \in \mathbb{R}^{d\times k}$ encapsulates the first k principal components of the centered dataset. Given a sample vector $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_k) \in \mathbb{R}^k$, we can then transform $\boldsymbol{\varphi}$ into a weight vector $\mathbf{w}$ in the d-dimensional weight space of the neural network via:

$$\mathbf{w} = \bar{\mathbf{p}} + \mathbf{\Pi}\boldsymbol{\varphi}. \tag{5}$$

Together with the structured parameters $\boldsymbol{\theta}$, the sampling procedure hence involves generating samples from a tuple $(\boldsymbol{\varphi}, \boldsymbol{\theta})$. This allows us to compute the likelihood contribution of the DNN part and the structured part as:

$$p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\varphi}) = p(\mathcal{D}|\boldsymbol{\theta}, \mathbf{w} = \bar{\mathbf{p}} + \mathbf{\Pi}\boldsymbol{\varphi}). \tag{6}$$

Since both p and k are reasonably small, our method highly supports efficient sampling with MCMC algorithms including sophisticated algorithms such as HMC. The sampling space design inherently captures multiple low-loss regions, fostering effective sampling and diverse solution exploration. Within this space, through the Bézier curve, a low-loss pathway is embedded to ensure smooth transitions during sampling. Importantly, this design integrates the interpretable parameter $\boldsymbol{\theta}$ into the sampling space without being directly affected by the approximation.

Priors As in Izmailov et al. (2020), we model the vectors $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ using independent multivariate normal distributions $\boldsymbol{\varphi} \sim \mathcal{N}(0, \boldsymbol{I}_k\sigma_\varphi)$ and $\boldsymbol{\theta} \sim \mathcal{N}(0, \boldsymbol{I}_p\sigma_\theta)$. Although a Gaussian prior was found to be adequate for $\boldsymbol{\theta}$ in our studies, our framework accommodates the use of more complex priors. This flexibility is particularly beneficial for interpreting the structured model component parameterized by $\boldsymbol{\theta}$. To bring the lower-dimensional subspace priors in line with conventional Bayesian DNN priors, Izmailov et al. (2020) discussed the use of temperature scaling. A detailed discussion for SSR models is provided in the Supplementary Material A.

4 NUMERICAL EXPERIMENTS

We now illustrate the advantages of our framework on four experiments where we compare our approach with 1) a naïve Bayesian SSR approximation on a simple regression toy experiment; 2) ground truth results derived from HMC on simulated data; 3) MCMC and approximation methods on benchmark datasets, and 4) various SSR approaches on a complex medical dataset. For the latter, we employ the Elliptic Slice Sampler (Murray et al., 2010). For all other cases, full batch processing is possible and we hence choose HMC to sample in the subspace as it typically results in a larger effective sample size. Further details and experimental results can be found in the Supplementary Material Section C and D. Due to the large parameter space and the inherent symmetry in neural network weights, it becomes challenging to compare the posterior of different Bayesian approximations Wiese et al. (2023). Instead, these approximations are typically evaluated based on their posterior predictive performance. We do so in Sections 4.3 and 4.4. In addition, the architecture of our SSR model enables a direct comparison of posteriors of the structured parameters, which is done in Sections 4.1 and 4.2.

4.1 Comparison with Naïve Subspace Inference

In this first experiment, we aim to compare the posterior distributions derived from the structured model component of our approach against those from a naïve subspace approximation (described in Paragraph 3). To this end, we adapt the synthetic dataset $\{f(u_i), u_i\}_{i=1}^{n_{\text{train}}}$ from Izmailov et al. (2020) with noisy nonlinear function f. For each data point i, we then incorporate a structured effect by randomly choosing a category vector $\mathbf{x}_i \in \{(0,0)^\top, (1,0)^\top, (0,1)^\top\}$ to shift $f(u_i)$ by an offset $\mathbf{x}_i^\top\boldsymbol{\theta}^*$ with $\boldsymbol{\theta}^* = (-0.5, 1)^\top$, resulting in the final training dataset $\{y_i = f(u_i) + \mathbf{x}_i^\top\boldsymbol{\theta}^*, u_i, \mathbf{x}_i\}_{i=1}^{n_{\text{train}=35}}$.

Knowing the structure of the data-generating process, we model the simulated data by a corresponding SSR as in Equation (1). For DNN(u) we choose a simple network with two fully-connected layers, each with 16 neurons and ReLU activation, and a linear output layer. The inference is done by sampling weights from a k-dimensional subspace for the unstructured model part while sampling from the full space of $\boldsymbol{\theta} = (\theta_0, \theta_1)^\top$ for the structured model part. The naïve approach does not differentiate between the two model parts and applies subspace inference on the combined (w, θ)-space.

Results As shown in Figure 3, the naïve subspace approach fails to represent the true posterior accurately, this is especially visible along the θ_1 direction. We further visualize the resulting posterior predictive in Figure 1. While the choice of k does not influence the predictive distribution's mean, we find that $k = 2$ produces a too narrow distribution compared to the gold standard obtained via full space HMC, and hence an overconfident uncertainty measure for both in- and out-of-distribution data. Notably, when increasing to $k = 12$, which is still much less than $d = 337$, we obtain almost the same level of uncertainty as HMC despite reducing the space by a factor of around 28. In contrast, the Laplace approximation underestimates the epistemic uncertainty (cf. Supplementary Material, Section D.1). This is also reflected in the log pointwise predictive density (LPPD)[1] evaluated on $n_{\text{test}} = 365$ test data points, where the Laplace approximation achieved an LPPD of 1.0 and our Semi-Subspace approach achieves an LPPD of 1.14, while Semi-Subspace($k = 12$) with an LPPD of 1.27 is even slightly better than HMC (LPPD 1.26) running on the full parameter space.

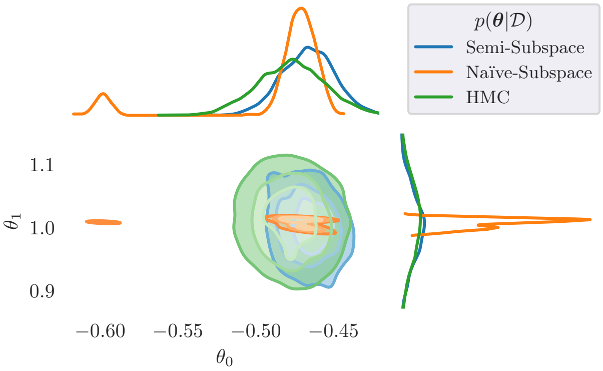

Figure 3: Posterior of the parameters in the structured model part using the naïve subspace approximation with $k = 4$ (Naïve-Subspace), our approach with $k = 2$ and $p = 2$ (Semi-Subspace), and HMC running in the full parameter space. The top and the right plot shows the marginal posterior distribution, whereas the center plot visualizes the bivariate distribution using a kernel density estimator based on 4000 samples from 10 HMC chains.

4.2 Simulation Study

While simple, the previously analyzed data-generating process allows us to systematically investigate the behavior of our subspace approach in comparison to the gold standard full space HMC method. As the previous results suggest, our subspace approach has the potential to generate a very similar posterior distribution of the structured model part $p(\boldsymbol{\theta}|\mathcal{D})$ as HMC would do for the entire parameter space. To test this, we extended the data generation of our previous study with two outcome distributions (Poisson and Normal), a larger input space $\mathbf{u} \in \mathbb{R}^4$ and $\mathbf{x} \in \mathbb{R}^3$, and different subspace dimensions $k = 2, 4, 8, 12, 16$. We conduct 50 simulation runs for every configuration, with different data sets generated in each run. Each data point $(y_i, \mathbf{x}_i, \mathbf{u}_i)$ was generated using the following data generating process: First, we sample $\mathbf{u}_i \sim N(0, \boldsymbol{I_4})$ and $\mathbf{x}_i \sim N(0, \boldsymbol{I_3})$. Next, we randomly initialize the SSR model with parameters $\boldsymbol{\theta}^* \in \mathbb{R}^3$ and $\mathbf{w}^* \in \mathbb{R}^{336}$, using a similar architecture as in our first experiment. Finally we choose $y_i \sim \mathcal{N}(f(\mathbf{u}_i) + \mathbf{x}_i^\top \boldsymbol{\theta}^*, 1)$ for the Normal outcome distribution case and $y_i \sim \text{Pois}(\rho(f(\mathbf{u}_i) + \mathbf{x}_i^\top \boldsymbol{\theta}^*))$ with ρ the exponential function for the Poisson case. We model the synthetic data using an SSR model with the same architecture as in the data generation process.

Results We examine the results focusing on the differences in the mean and standard deviation of the posterior between our method and HMC. Figure 4 shows the results for the Poisson distribution. We find that our subspace approach yields an unbiased posterior mean irrespective of k and that the difference to HMC reduces to zero for increasing subspace dimension (left plot). While biased in the distribution's variance, increasing the subspace alleviates this discrepancy and a larger subspace produces almost the same posterior variance as HMC. Small k, in contrast, leads to overly confident uncertainty quantification. We thus argue that the subspace dimension k can be chosen as large as computationally feasible, as it consistently improves the distribution quality.

In order to analyze previous results in light of parameter uncertainty quantification, we further check the calibration of the posterior. To this end, we compute the amount of coverage of the true parameter $\boldsymbol{\theta}^*$ from the data generation process by the derived α-credibility intervals for different nominal levels $\alpha \in (0, 1)$. This is visualized by plotting the theoretical coverage α against the empirical sample coverage using the obtained posterior (cf. Figure 5 for θ_1). Results clearly show that for increasing subspace dimensions, calibration improves, and already for $k = 12$ or $k = 16$, coverage is not notably different from the one obtained by using HMC with 372 dimensions.

4.3 UCI Benchmark

While the previous experiments assess our SSR subspace approximation by checking the coverage of structured model parameters, we now evaluate the general applicability of our subspace approach using

[1] To be comparable with Wiese et al. (2023), we divided each reported LPPD by the number of data points

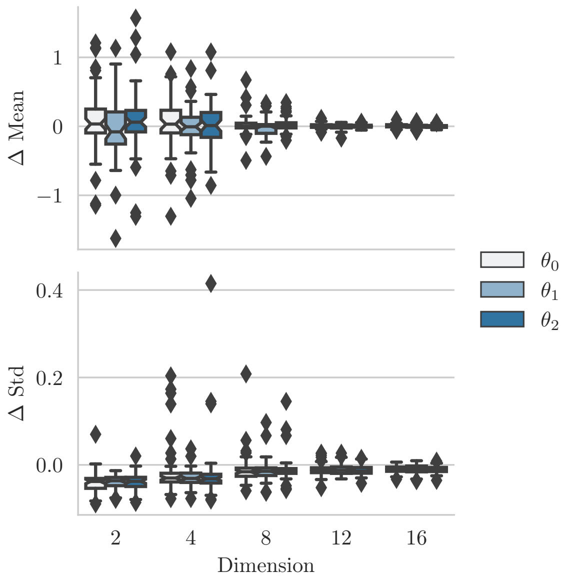

Figure 4: Posterior mean (top) and standard deviation (bottom) of our approximation method compared to the gold standard HMC. The boxplots show differences between the learned distribution's mean/standard deviation of our approach minus the respective statistic using HMC for the 50 simulation repetitions. The x-axis depicts the different subspace dimensions k used in our approach and each color represents one of the three parameters in θ.

the benchmark datasets and methods investigated in Wiese et al. (2023). The authors show how to efficiently use multiple chains to capture different posterior modes and thereby achieve superior performance by capturing most of the relevant parts of the posterior. The obtained results are hence (close to) an oracle performance and can be used to check how well our approximation is working. Their benchmark comprises three simulated datasets and six datasets from the UCI machine learning repository Dua et al. (2017). We use the same data splits, model architectures, and data pre-processing as in Wiese et al. (2023), allowing for a direct comparison with their MCMC approach as well as results provided for Laplace approximation (Daxberger et al., 2021) and Deep Ensembles (Lakshminarayanan et al., 2017). We run our method using $k = 2$ and $k = 5$.

Results Table 1 summarizes the results, showing that our method outperforms both the Laplace approximation and deep ensembles on all provided datasets while being often very close to the gold standard MCMC approach. We further see that an increase in subspace dimension can notably improve predictive performance (Diabetes, ForestF, Yacht).

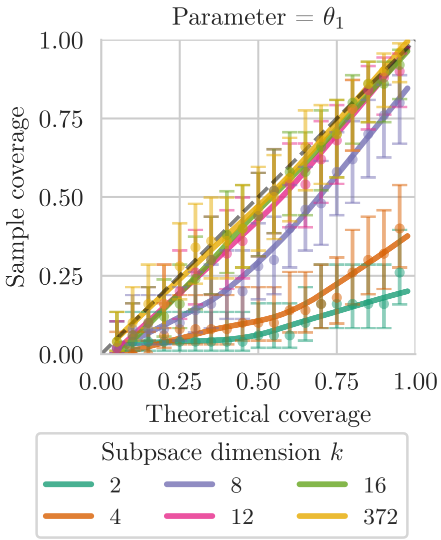

Figure 5: Coverage comparison of credibility intervals derived from the posterior $p(\theta_1|\mathcal{D})$ using different subspace dimensions k (colors). The theoretical coverage (x-axis) across different values in $(0, 1)$ is plotted against the sample coverage (y-axis), based on the empirical ratio of the credibility interval containing the true parameter. Whiskers represent the 95% Wilson confidence interval.

4.4 Application to Melanoma Data

Finally, we apply our method to a real-world melanoma dataset (International Skin Imaging Collaboration, 2020) containing 33,058 patient records of $3 \times 128 \times 128$ RGB color images of skin lesions as well as additional metadata such as the patient's age. The primary objective of this dataset is to predict the presence or absence of malignant skin lesions. We follow the approach by Dürr et al. (2022) and process the images using a basic convolutional neural network (see Supplementary Material C for details) while modeling the patient's age as a linear effect θ_{age}. Next to a comparison with MCMC using only the age information, we compare against the transformation model approach by Dürr et al. (2022), and the Laplace approximation (using the last layer approach). The data is split into six data folds as in previous works.

Results In Figure 6, we compare the posterior distribution $p(\theta_{\text{age}}|\mathcal{D})$ obtained by the different methods. Results suggest that the inclusion of image information decreases the effect of age when comparing the different methods to the results of MCMC which only uses age information. The results from Dürr et al. (2022) (CNN + BF-VI) are not in line with all other methods, yielding a differently shaped distribution. While our approach is similar to the Laplace approximation in

Table 1: Normalized expected test log pointwise predictive density (LPPD; larger is better) with the network architecture introduced by Wiese et al. (2023), comprising three hidden layers with 16 neurons. The values within parentheses represent the standard errors of the predictive density per data point. The best method, excluding MCMC (representing an approximate upper bound), and all methods within one standard error of the best method are highlighted in bold.

dataset	MCMC	Subspace (k=2)	Subspace (k=5)	Deep Ens.	Laplace Appr.
DI	0.91 (±0.09)	**0.82** (± 0.10)	**0.77** (± 0.09)	-2.02 (±0.02)	-1.81 (±0.01)
DR	0.95 (±0.08)	**0.82** (± 0.13)	**0.86** (± 0.12)	-2.20 (±0.02)	-2.33 (±0.00)
Airfoil	0.92 (±0.05)	**-0.28** (± 0.12)	**-0.19** (± 0.09)	-2.17 (±0.01)	-3.57 (±0.18)
Concrete	0.26 (±0.07)	**-0.53** (± 0.20)	**-0.55** (± 0.17)	-2.03 (±0.01)	-4.36 (±0.47)
Diabetes	-1.18 (±0.08)	-2.40 (± 0.28)	**-1.21** (± 0.08)	-2.09 (±0.04)	-2.61 (±0.00)
Energy	2.07 (±0.46)	**1.43** (± 0.14)	**1.57** (± 0.15)	-1.99 (±0.02)	-1.39 (±0.06)
ForestF	-1.43 (±0.45)	-1.90 (± 0.19)	**-1.38** (± 0.07)	-2.20 (±0.02)	-2.80 (±0.00)
Yacht	3.31 (±0.21)	-0.69 (± 1.90)	**1.49** (± 0.51)	-2.18 (±0.03)	-2.69 (±0.00)

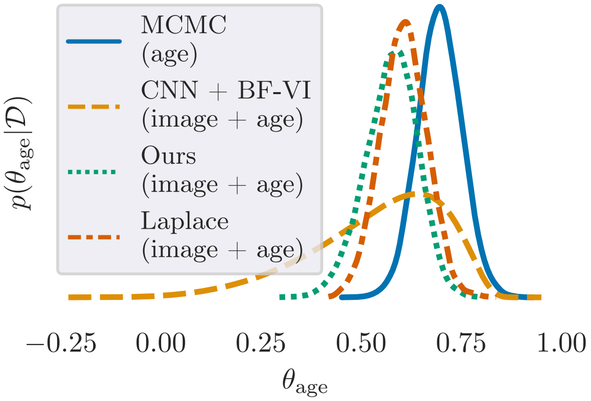

Figure 6: Posterior $p(\theta_{\text{age}}|\mathcal{D})$ obtained using different methods on the Melanoma dataset. Distributions are based on a KDE smoother combining all samples from all training folds.

terms of the posterior $p(\theta_{\text{age}}|\mathcal{D})$ it surpasses both the Laplace approximation and the transformation model approach in terms of negative log-likelihood (see Table 2).

Table 2: Mean area under the ROC Curve (AUC) and LPPD values (standard errors in brackets; not available for CNN + BF-VI) across the six data folds for the different methods (rows)

	AUC	LPPD
CNN + BF-VI	0.82 (±0.03)	-0.076 (NA)
Laplace	0.795 (±0.004)	-0.076 (±0.001)
Semi-Subspace(k=2)	**0.841** (±0.003)	**-0.072** (±0.001)

Optimization Aspects It's worth noting that during our empirical analysis, we encountered difficulties fitting the Laplace SSR model, requiring careful tuning of the learning rate, see Supplementary Material D.4 for a detailed discussion. We attribute this observation to the optimization asymmetry when training SSR models, where the optimization of structured model parameters is not treated differently from the one of neural network parameters. We note that compared to the Laplace approximation, our method is less affected by the optimization asymmetry in SSR, as we do not directly rely on a specific learning rate or optimizer (except for the construction of the sampling space). We find this to be a major advantage of our method compared to Laplace approximation or other optimization-based SSR methods.

5 CONCLUSION

We have presented a method to address two critical challenges inherent in SSR models, uncertainty quantification and optimization. Our approach notably improves over naïve approximation methods in terms of posterior distribution quality while outperforming other approximation methods in predictive posterior performance. We also find that with a sufficiently large subspace dimension k, our method comes remarkably close to replicating the posterior distribution and posterior predictive distribution achieved by gold standard MCMC techniques. Additionally, our method enables a deeper analysis of parameter uncertainty within the structured model component, accounting for the uncertainty propagated through the DNN part. This nuanced understanding of uncertainty provides valuable insights, particularly in domains like medical diagnostics, where model interpretability is crucial. Furthermore, our work sheds light on the optimization asymmetry in SSR and makes inference more robust as it mitigates the challenges arising from this asymmetry. One limitation of our current method is the notable memory usage due to the storage of $(k+1) \times d$ parameters in $\mathbf{\Pi}$ and $\bar{\mathbf{p}}$ during the sampling phase, or in $\mathbf{\Lambda}$ during the training phase. A possible alleviation of this issue could be achieved by sequentially computing the necessary values in Equations (2) and (5), trading performance with memory demand. Another limitation is the high computational demand, inherent in sampling-based approaches, requiring one forward pass per posterior sample. This could be addressed by treating only parts of the network as Bayesian.

In summary, our method not only boosts the predictive performance of SSR models but also serves as a comprehensive framework for understanding and quantifying uncertainty. It achieves results that are comparable to those of HMC, while also addressing fitting challenges commonly encountered in other SSR methods that rely solely on optimization. This makes our approach a valuable asset across a diverse array of applications.

Acknowledgements

We thank the reviewers for their valuable feedback. This work was supported by the Federal Ministry of Education and Research of Germany (BMBF) in the project “DeepDoubt” (grant no. 01IS19083A) and by the Carl-Zeiss-Stiftung in the project ”DeepCarbPlanner” (grant no. P2021-08-007).

References

Amoura, K., Wira, P., and Djennoune, S. (2011). A state-space neural network for modeling dynamical nonlinear systems. In *IJCCI (NCTA)*, pages 369–376.

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

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10. ACM.

Ciampi, A. and Lechevallier, Y. (1995). Designing neural networks from statistical models: A new approach to data exploration. In *KDD*, pages 45–50.

Ciampi, A. and Lechevallier, Y. (1997). Statistical models as building blocks of neural networks. *Communications in statistics-theory and methods*, 26(4):991–1009.

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.

de Waal, D. A. and du Toit, J. V. (2007). Generalized additive models from a neural network perspective. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 265–270. IEEE.

De Waal, D. A. and Du Toit, J. V. (2011). Automation of generalized additive neural networks for predictive data mining. *Applied Artificial Intelligence*, 25(5):380–425.

Dorigatti, E., Schubert, B., Bischl, B., and Rügamer, D. (2023). Frequentist Uncertainty Quantification in Semi-Structured Neural Networks. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 1924–1941. PMLR.

Dua, D., Graff, C., et al. (2017). Uci machine learning repository. *URL http://archive. ics. uci. edu/ml.*

Dürr, O., Hörling, S., Dold, D., Kovylov, I., and Sick, B. (2022). Bernstein Flows for Flexible Posteriors in Variational Bayes. *arXiv preprint arXiv:2202.05650.*

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018). Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Hubin, A., Storvik, G., and Frommlet, F. (2018). Deep Bayesian regression models. *arXiv preprint arXiv:1806.02160.*

International Skin Imaging Collaboration (2020). Siim-isic 2020 challenge dataset. Accessed on September 28, 2023.

Izmailov, P., Maddox, W. J., Kirichenko, P., Garipov, T., Vetrov, D., and Wilson, A. G. (2020). Subspace Inference for Bayesian Deep Learning. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 1169–1179. PMLR.

Jantre, S., Urban, N. M., Qian, X., and Yoon, B.-J. (2023). Learning active subspaces for effective and scalable uncertainty quantification in deep neural networks. *arXiv preprint arXiv:2309.03061.*

Kook, L., Götschi, A., Baumann, P. F., Hothorn, T., and Sick, B. (2022a). Deep interpretable ensembles. *arXiv preprint arXiv:2205.12729.*

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

Kopper, P., Wiegrebe, S., Bischl, B., Bender, A., and Rügamer, D. (2022). DeepPAMM: Deep Piecewise Exponential Additive Mixed Models for Complex Hazard Structures in Survival Analysis. In *Advances in Knowledge Discovery and Data Mining*

(PAKDD), pages 249–261. Springer International Publishing.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 541–548. JMLR Workshop and Conference Proceedings.

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

Potts, W. J. (1999). Generalized additive neural networks. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 194–200.

Pölsterl, S., Sarasua, I., Gutiérrez-Becker, B., and Wachinger, C. (2020). A wide and deep neural network for survival analysis from anatomical shape and tabular clinical data. *Communications in Computer and Information Science*, page 453–464.

Rügamer, D. (2023). 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*, pages 29291–29305. PMLR.

Rügamer, D., Kolb, C., and Klein, N. (2023). Semi-structured distributional regression. *The American Statistician*, 0(0):1–12.

Sick, B., Hothorn, T., and Dürr, O. (2021). Deep transformation models: Tackling complex regression problems with neural network based transformation models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2476–2481. IEEE.

Tran, M.-N., Nguyen, N., Nott, D., and Kohn, R. (2020). Bayesian deep net GLM and GLMM. *Journal of Computational and Graphical Statistics*, 29(1):97–113.

Wahba, G. (1990). *Spline models for observational data*. SIAM.

Wiese, J. G., Wimmer, L., Papamarkou, T., Bischl, B., Günnemann, S., and Rügamer, D. (2023). Towards efficient posterior sampling in deep neural networks via symmetry removal. In *Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*. Springer International Publishing.

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

Wortsman, M., Horton, M. C., Guestrin, C., Farhadi, A., and Rastegari, M. (2021). Learning neural network subspaces. In *International Conference on Machine Learning*, pages 11217–11227. PMLR.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Bayesian Semi-structured Subspace Inference: Supplementary Materials

A TEMPERED POSTERIOR FOR SSR MODELS

In the following section, we will discuss how we can apply a temperature parameter to improve predictive performance and what the unique challenges are in the context of SSR models.

As highlighted by Izmailov et al. (2020), subspace inference can yield overly confident uncertainty estimates. This overconfidence may stem from the fact that the prior is defined within the subspace of dimension k rather than the larger dimension d. Consequently, reducing the parameter space through subspace construction has a noticeable impact on the posterior distribution. To mitigate this effect, Izmailov et al. (2020) proposed the application of a temperature parameter $T > 0$. This parameter scales the likelihood according to the following Equation:

$$p_T(\boldsymbol{\theta}, \boldsymbol{\varphi}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\varphi})^{1/T} p(\boldsymbol{\theta}, \boldsymbol{\varphi}) \tag{7}$$

Here, a temperature smaller than one shifts the posterior towards the maximum likelihood estimate, while a temperature larger than one moves the posterior closer to the prior distribution. Their findings suggest that using a temperature parameter can improve predictive performance and, potentially, the quality of uncertainty estimates.

However, with this proposed method the marginal posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ will be influenced by the temperature parameter. This is problematic for SSR models as $\boldsymbol{\theta}$ is not affected by the subspace approximation, hence there is no reason to modify its marginal posterior with a temperature parameter. To tackle this challenge, we devised a novel approach. First, we split the joint posterior distribution into two parts, which can be expressed as follows:

$$\begin{aligned} p_T(\boldsymbol{\theta}, \boldsymbol{\varphi}|\mathcal{D}) &= p(\boldsymbol{\theta}|\boldsymbol{\varphi}, \mathcal{D})\, p_T(\boldsymbol{\varphi}|\mathcal{D}) \\ &= \frac{p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\varphi})\, p(\boldsymbol{\theta}|\boldsymbol{\varphi})}{p(\mathcal{D}|\boldsymbol{\varphi})} \frac{p(\mathcal{D}|\boldsymbol{\varphi})^{\frac{1}{T}}\, p(\boldsymbol{\varphi})}{p_T(\mathcal{D})} \end{aligned} \tag{8}$$

The first part of this equation represents the conditioned posterior for our interpretable parameters, denoted as $\boldsymbol{\theta}$, while the second part reflects the posterior of the neural network, $p(\boldsymbol{\varphi}|\mathcal{D})$, where we apply the temperature parameter. If $p(\boldsymbol{\theta}|\mathcal{D})$ and $p(\boldsymbol{\varphi}|\mathcal{D})$ are independent, the temperature parameter won't influence the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ in this approach. This can be shown by simply rewriting Equation (9)

$$\begin{aligned} p_T(\boldsymbol{\theta}|\mathcal{D}) &= \int p_T(\boldsymbol{\theta}, \boldsymbol{\varphi}|\mathcal{D}) d\boldsymbol{\varphi} \\ &= \int p(\boldsymbol{\theta}|\boldsymbol{\varphi}, \mathcal{D})\, p_T(\boldsymbol{\varphi}|\mathcal{D}) d\boldsymbol{\varphi} \\ &= p(\boldsymbol{\theta}|\mathcal{D}) \int p_T(\boldsymbol{\varphi}|\mathcal{D}) d\boldsymbol{\varphi} && \text{, if}\, p(\boldsymbol{\theta}, \boldsymbol{\varphi}|\mathcal{D}) = p(\boldsymbol{\theta}|\mathcal{D}) p(\boldsymbol{\varphi}|\mathcal{D}) \\ &= p(\boldsymbol{\theta}|\mathcal{D}) \quad \blacksquare \end{aligned} \tag{9}$$

However, the proposed approach necessitates the computation of the marginal likelihood, represented as $p(\mathcal{D}|\boldsymbol{\varphi})$. We chose to numerically integrate the likelihood function according to the following equation:

$$p(\mathcal{D}|\boldsymbol{\varphi}) = \int p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\varphi})\, p(\boldsymbol{\theta}|\boldsymbol{\varphi})\, d\boldsymbol{\theta} \tag{10}$$

It's important to note that this numerical integration depends on the choice of technical parameters like integration step size and range, making misconfigurations of these parameters a potential source of inaccurate results. Additionally, this approach is only feasible when the dimension of the structural parameters is relatively small, which allows for manageable numerical integration. Given these constraints, applying a temperature parameter to SSR models proves to be a challenging endeavor.

In our initial analysis, we applied our proposed temperature adjustment to the likelihood function of an SSR model according to Equation 8. We conducted this analysis on the melanoma dataset and on the adapted synthetic dataset from Izmailov et al. (2020) as described in Section 4.1, where in both settings the parameter θ is of low dimension allowing for numerical integration. In both experiments, we observed an undesired influence of the temperature parameter manifesting in a broadening on the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ with increasing temperature (cf. Figure 7). We attribute this influence to the dependence between $p(\boldsymbol{\theta}|\mathcal{D})$ and $p(\boldsymbol{\varphi}|\mathcal{D})$. Additionally, we observed only marginal performance improvements in our initial experiments when adjusting the temperature parameter. However, these slight gains are outweighed by the significant computational difficulties and the impact of the temperature on the posterior $p(\boldsymbol{\theta}|\mathcal{D})$. These difficulties led us to decide against its adaptation.

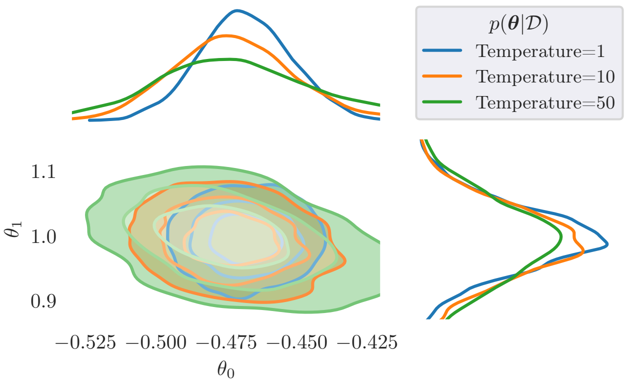

Figure 7: Samples from the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ for different temperature T parameters on the adapted synthetic dataset described in section 4.1. The contour lines represent the 0.25, 0.5, and 0.75 high-density interval (HDI).

B ADDITIONAL PROOFS

B.1 The Bézier Curve Resides in the Affine Subspace

We aim to demonstrate that every point on the Bézier curve $B_\lambda(t)$, which serves as an approximation of the low-energy valley, can be obtained through our chosen sampling procedure. Specifically, our sampling approach ensures that each sample $\boldsymbol{\varphi}$ belongs to the affine subspace spanned by the control points $\mathbf{p}_0^*, \mathbf{p}_1^*, \ldots, \mathbf{p}_k^*$.

$$\text{AffSpan}(\boldsymbol{\Lambda}^*) = \left\{ \mathbf{p}_0^* + \sum_{i=1}^{k} \varphi_i' \Delta\mathbf{p}_i^* \,\middle|\, \varphi_i' \in \mathbb{R} \right\}. \tag{11}$$

To demonstrate that the Bézier curve $B_\lambda(t)$ resides within this affine subspace, we perform the following algebraic manipulations:

$$\begin{aligned}
B_\lambda(t) &= \sum_{l=0}^{k} \binom{k}{l} (1-t)^{k-l} t^l \mathbf{p}_l^* \\
&= (1-t)^k \mathbf{p}_0^* + \sum_{l=1}^{k} \binom{k}{l} (1-t)^{k-l} t^l \mathbf{p}_l^* \\
&= (1-t)^k \mathbf{p}_0^* + \sum_{l=1}^{k} \binom{k}{l} (1-t)^{k-l} t^l (\mathbf{p}_l^* - \mathbf{p}_0^* + \mathbf{p}_0^*) \\
&= \mathbf{p}_0^* \cdot \underbrace{\left(\sum_{l=0}^{k} \binom{k}{l} (1-t)^{k-l} t^l \right)}_{=1} + \sum_{l=1}^{k} \underbrace{\binom{k}{l} (1-t)^{k-l} t^l}_{=\alpha_l} (\mathbf{p}_l^* - \mathbf{p}_0^*) \\
&= \mathbf{p}_0^* + \sum_{l=1}^{k} \alpha_l (\mathbf{p}_l^* - \mathbf{p}_0^*),
\end{aligned}$$

where $\alpha_l = \binom{k}{l}(1-t)^{k-l}t^l$ for l in $1, \ldots, k$. The last equation clearly indicates that $B_\lambda(t)$ is contained within the affine subspace $\mathrm{AffSpan}(\mathbf{\Lambda}^*)$.

C EXPERIMENTAL SETUP

The code and notebooks with further experimental settings for this project are available under https://github.com/doldd/Bayesian_Semi_Sub.

C.1 General Experimental Setup

Below, we detail how we train the model to construct the subspace. To train the model, we optimized the entire SSR model, where only the weights of the DNN part are controlled via the Bézier curve. In all experiments, we used the Adam optimizer with a learning rate of 1e-4 and weight decay of 1e-4 for the medical dataset, a learning rate of 5e-3 with zero weight decay for the UCI benchmark, and a learning rate of 0.0025 and weight decay of 1e-3 for the Simulation and toy data. After a fixed number of epochs, we selected the model with the lowest validation loss and used this model to construct the subspace. We also verified that the training was long enough for the model to enter the overfitting region and adapted the number of epochs accordingly.

C.2 Sampling Setup

To generate the posterior samples, we used the NUTS implementation from Pyro in cases where we could evaluate all data in one step, experiments described in Section 4.1, 4.2, and 4.3. For the melanoma dataset (see Section 4.4), an elliptic slice sampler (ESS) was used. In the simulation study and melanoma experiment, we ran 10 chains, discarded 400 warmup samples, and collected 800 samples per chain. In the UCI experiment, we also draw 10 chains with 200 warmup samples and 600 samples per chain. In the NUTS environment, the step size was optimized during the warmup phase to achieve a target acceptance probability of 0.8. The first proposal was drawn out of the prior distribution.

To conduct inference on the full parameter space, we employed the NUTS implementation from NumPyro. The number of chains, warmup, and collected samples remained the same as in the subspace setting. Due to poorer sampling performance compared to our subspace approximation, as indicated by *r_hat*, we reduced the target acceptance probability to 0.6 to enhance explorativeness.

C.3 CNN Model Architecture

The following listing defines the PyTorch model which we used to process the medical dataset in Experiment 4.4. The following SSR model architecture consists of a CNN (DNN listing) and a linear structured model part (structured_model listing).

Listing 1: DNN definition used for the melanoma data in Section 4.4

```
(DNN): Sequential(
    (0): Sequential(
        (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1))
        (1): Tanh()
        (2): MaxPool2d(kernel_size=2, stride=2, padding=0,
                       dilation=1, ceil_mode=False)
        (3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1))
        (4): Tanh()
        (5): MaxPool2d(kernel_size=2, stride=2, padding=0,
                       dilation=1, ceil_mode=False)
        (6): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
        (7): Tanh()
        (8): MaxPool2d(kernel_size=2, stride=2, padding=0,
                       dilation=1, ceil_mode=False)
        (9): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1))
        (10): Tanh()
        (11): MaxPool2d(kernel_size=2, stride=2, padding=0,
                       dilation=1, ceil_mode=False)
        (12): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
        (13): Tanh()
        (14): MaxPool2d(kernel_size=2, stride=2, padding=0,
                       dilation=1, ceil_mode=False)
    )
    (1): Sequential(
        (0): Flatten(start_dim=1, end_dim=-1)
        (1): Linear(in_features=512, out_features=128, bias=True)
        (2): Tanh()
        (3): Linear(in_features=128, out_features=128, bias=True)
        (4): Tanh()
        (5): Linear(in_features=128, out_features=1, bias=True)
    )
)
(structured_model): Linear(in_features=1, out_features=1, bias=False)
```

D ADDITIONAL RESULTS

In this section, we provide further results from our experiments.

D.1 Additional Results from the Toy Data

Here, we present additional results for the naïve Laplace approximation on the toy dataset. Figure 8 shows the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ and Figure 9 the corresponding posterior predictive distribution obtained by applying the naïve Laplace approximation to the last layer and the structured model component. In the posterior distribution, we see some deviations between naïve Laplace approximation and HMC. However, the posterior predictive distribution is comparable to our approach with a subspace dimension of two. If we compare to HMC or our method with a subspace dimension of 12 it is still too narrow in terms of epistemic uncertainty (See Figure 1).

To complement the experiment on the toy dataset, we present additional results in the posterior distribution of the parameters from the structured model component in Figure 10. The results for HMC and Semi-Subspace ($k = 2$) are identical to those shown in Figure 3, and we extended the results with our Semi-Subspace model, utilizing a subspace dimension of 12. The corresponding posterior predictive was shown in Figure 1. This comparison aligns with the findings from our simulation study, indicating that increasing the subspace dimension leads to posterior distributions that closely resemble those obtained with HMC.

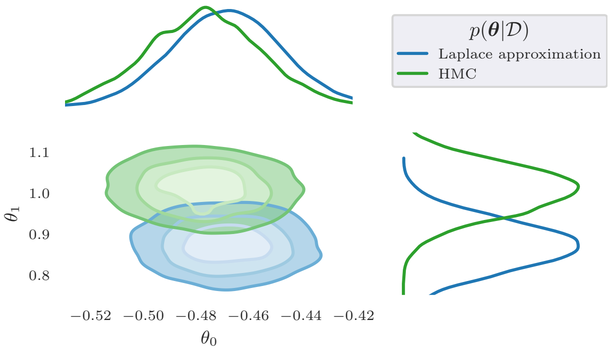

Figure 8: Comparison of the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ between ground truth HMC and Laplace Approximation on the regression dataset adapted from Izmailov et al. (2020). This posterior distribution corresponds to the posterior predictive shown in Figure 9

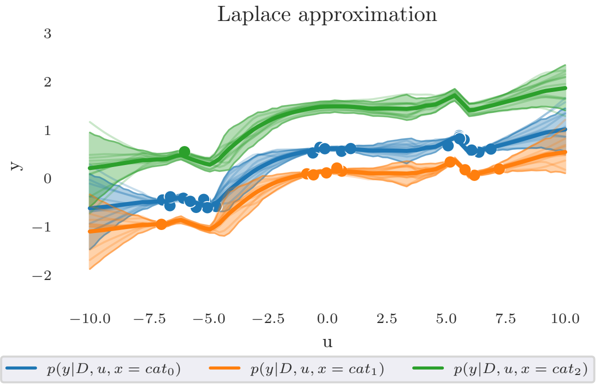

Figure 9: Posterior predictive using the Laplace approximation on the toy dataset. We applied the Laplace approximation on the last layer of the DNN part and on the two-dimensional $\boldsymbol{\theta}$ parameters of the structured model part. Data and model architecture are the same as described in Section 4.1

.

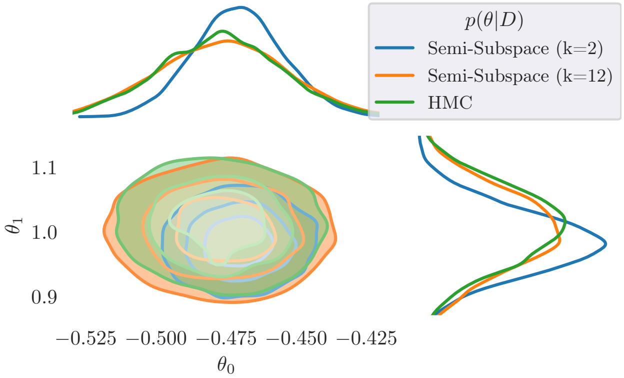

Figure 10: Posterior of the parameters in the structured model part using our approach with $k = 2$ and $k = 12$ (Semi-Subspace) compared with HMC running in the full parameter space. The top and right plots show the marginal posterior distribution, whereas the center plot visualizes the bivariate distribution using a kernel density plot based on 4000 samples from 10 HMC chains.

D.2 Additional Results from the Simulation Study

We continue with further results of our simulation study. Figure 11 and Figure 13 show the results by using a normal outcome distribution, where the parameters μ is modeled, instead of the Poisson distribution. These Figures also validate our thesis, that increasing the subspace dimension reduces the error in the first two moments of the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ and improves its uncertainty quality.

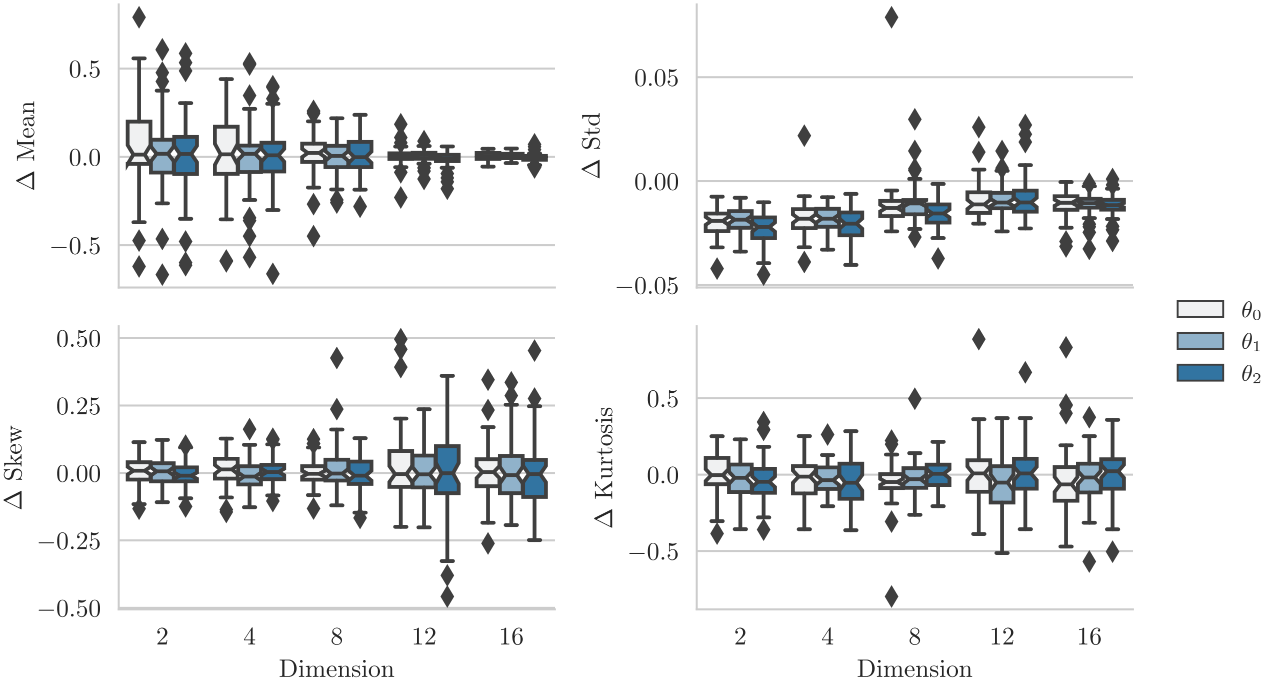

Figure 11: A detailed comparison of the differences in the first four moments between our approximation method and the gold standard HMC when utilizing a normal outcome distribution in our simulation study. The boxplots show differences between the learned distribution's moment of our approach minus the respective moment using HMC for the 50 simulation repetitions. The x-axis depicts the different subspace dimensions k used in our approach and each color represents one of the three parameters in $\boldsymbol{\theta}$.

In Figure 5 we visualized the influence of the subspace dimensions on the posterior calibration. This was shown by depicting one parameter θ_1 out of the three-dimensional parameter space. For the sake of completeness, we provide in the following Figure 12 the calibration comparison for the entire parameter space $\boldsymbol{\theta}$ from the structured model. In a parallel analysis to the previous one, Figure 13 showcases the calibration comparison using the Normal outcome distribution. The results demonstrate well-calibrated distributions with increasing subspace dimension k. Overall, this observation holds regardless of the outcome distribution or data.

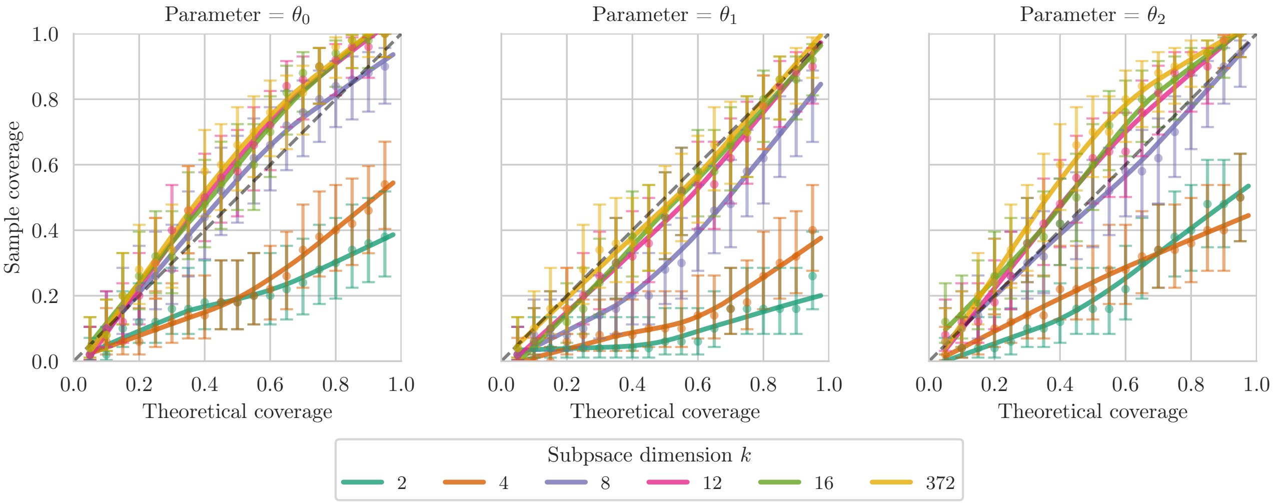

Figure 12: Coverage comparison of credibility intervals derived from the posterior $p(\theta_1|\mathcal{D})$ using different subspace dimensions k (colors) of all parameters instead of only picking θ_1 as shown in Figure 5. The theoretical coverage (x-axis) across different values in $(0, 1)$ is plotted against the sample coverage (y-axis), based on the empirical ratio of the credibility interval containing the true parameter. Whiskers represent the 95% Wilson confidence interval.

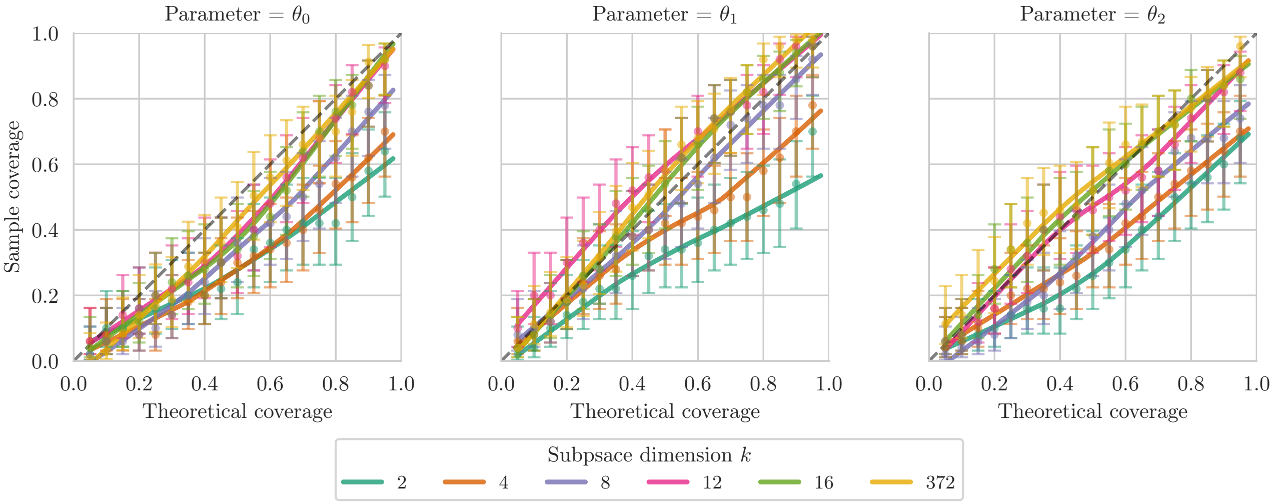

Figure 13: Identical analyses as in Figure 12 but with Normal outcome distribution instead of Poisson distribution.

The following figures (Figures 16, 17, 14, and 15) depict common sampling evaluation metrics, namely, effective sample size (ESS_{bulk}) and r_hat, from our simulation study. In most cases, r_hat for all parameters $\boldsymbol{\theta}$ was smaller than 1.1, indicating an accurate sampling process. Instances where r_hat exceeded 1.1 often correlated with a slightly bimodal subspace, where a single chain has difficulty exploring both modes in a sufficient amount of time. In practical applications, addressing this challenge might involve further training during the subspace construction. Overall, we observe that sampling becomes more challenging in higher dimensions (smaller ESS_{bulk} and larger r_hat cf. Figure 16, 17, 14, and 15), but it was still feasible to produce valid samples.

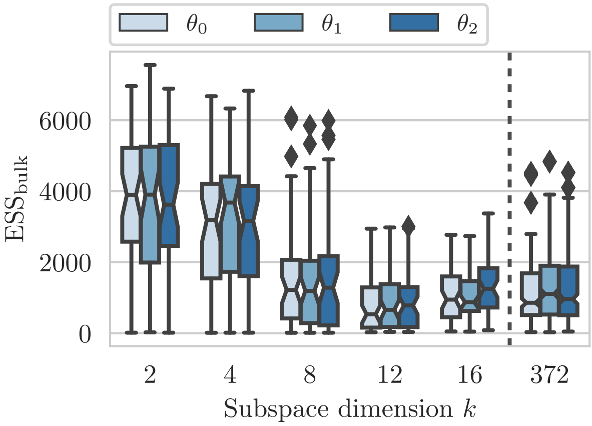

Figure 14: ESS of all parameters $\boldsymbol{\theta}$ of the structured model part. The three parameters are depicted in different colors. Subspace with dimension 372 refers to running HMC on the entire parameter space. Each Boxplot contains the ESS of the 50 different runs from the simulation study using the Poisson outcome distribution.

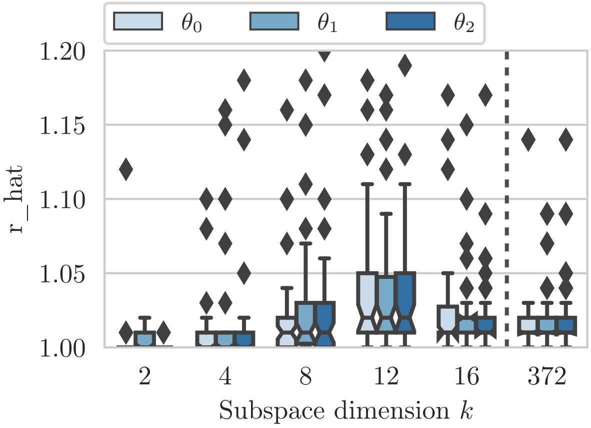

Figure 15: Same simulation study as shown in Figure 14 but with the r_hat metric.

.

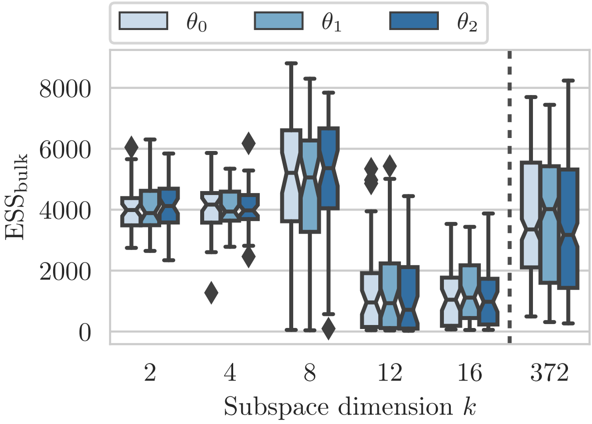

Figure 16: Same analysis as shown in Figure 14 but using the data from the simulation study with the normal outcome distribution.

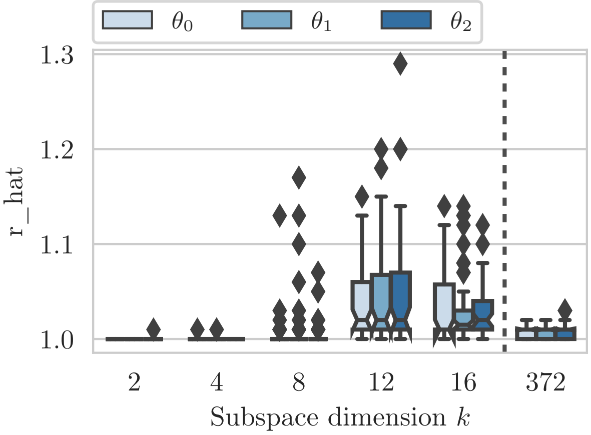

Figure 17: Same simulation study as shown in Figure 16 but with the r_hat metric.

.

D.3 Evaluating the Arc Length of the Bézier Curve in Subspace Construction

The optimization of the Bézier curve in subspace construction theoretically risks the curve collapsing into a single point or a tightly confined area of low loss. This would yield a minimal-length curve, leading to a degenerated subspace characterized by almost identical configurations. However, the inherent stochasticity in optimization and the vastness of the parameter space likely prevent such unfavorable outcomes.

To assess this, we measured the arc length of the Bézier curve after optimization finished in our simulation study (Figure 18). The curve lengths were consistently greater than zero and increased as we added more control points. Additionally, we tracked the curve length dynamics during optimization (Figure 19). These results showed no signs of the curve collapsing into a single local minimum, suggesting the subspace's robustness. Nonetheless, further investigation into the curve's length behavior would be valuable.

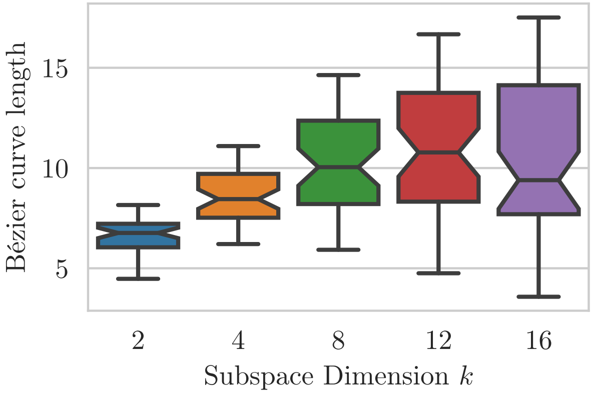

Figure 18: Boxplot of the Bézier curve arc length computed from the 50 different models used in the simulation study with the Poisson outcome distribution in Section 4.2. The x-axis represents the subspace dimension k which contains $k + 1$ control points of the Bézier curve.

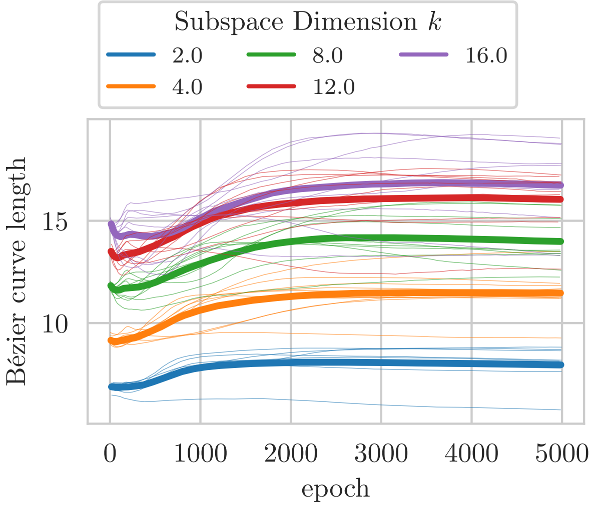

Figure 19: Dynamic of the Bézier curve length during ten different optimizations in the simulation study using the Poisson outcome distribution. Each thin line characterizes the dynamic of a single training and the bold line represents the mean in each subspace dimension k.

D.4 Additional Results from the Melanoma Dataset

In Figure 6, we present the overall posterior $p(\theta_{\text{age}}|\mathcal{D})$ by pooling all samples from the six folds. For a more detailed examination, we break down the analysis in Figure 20 to visualize the posterior distribution separately for each fold, rather than aggregating the samples. We observe that in some folds (2, 3, and 6), the posterior distribution of the Laplace approximation closely resembles our approach. However, folds one and four exhibit slight differences, but we did not observe a consistent trend. Notably, the expectation of the posterior remains relatively stable, with fold one being the exception. This suggests that pooling the folds, as shown in Figure 6, does not significantly alter the distribution's shape.

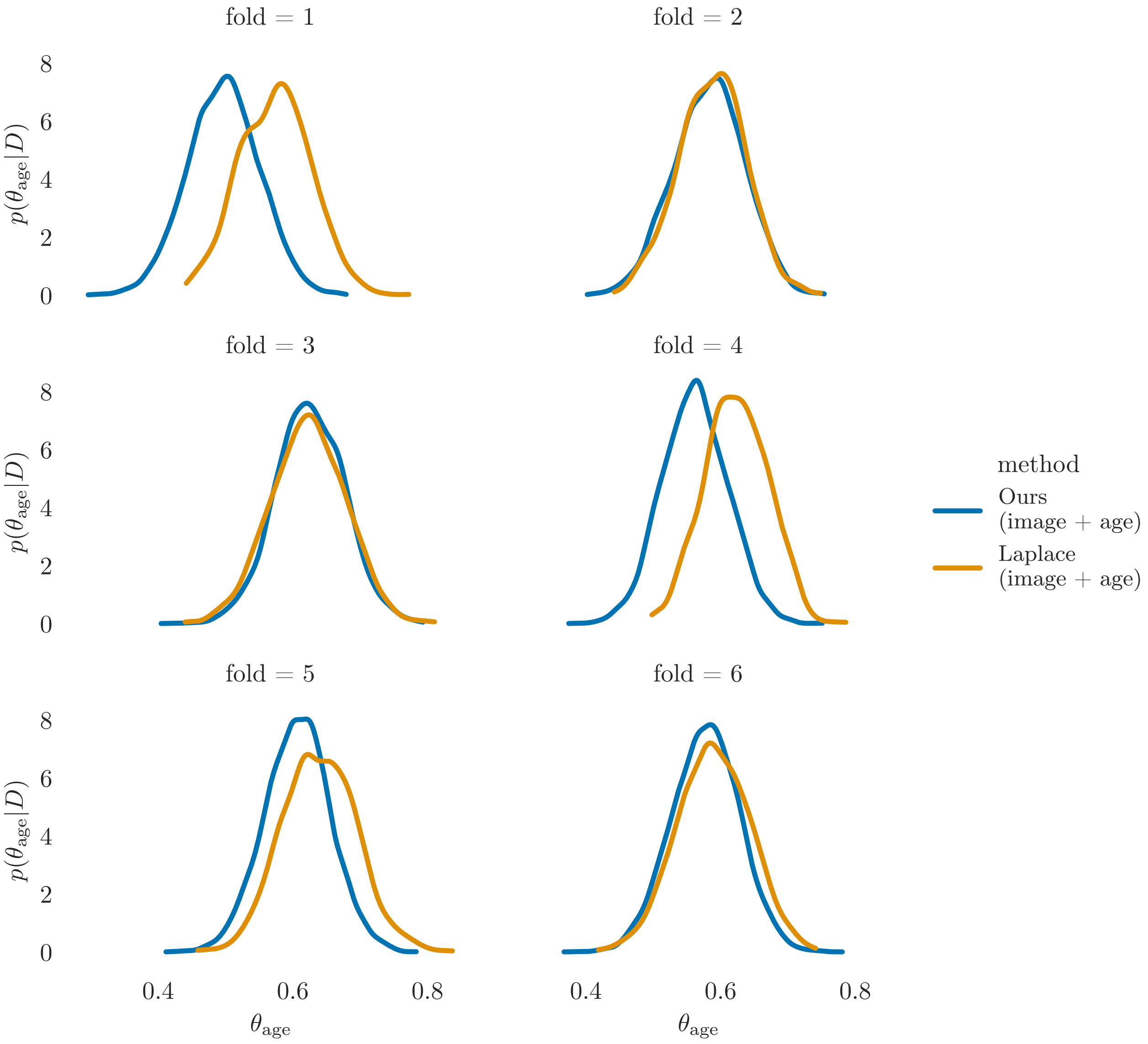

Figure 20: Posterior $p(\theta_{\text{age}}|\mathcal{D})$ for each of the six folds in the melanoma dataset. The color-coding distinguishes between our method and the Laplace approximation (using a learning rate of 5e-3). The corresponding loss curve is displayed in Figure 21. According to this loss curve, we select the model with the lowest validation loss and accomplished the Laplace approximation to generate the shown posterior.

D.5 Optimization Asymmetry

In the following discussion, we investigate the optimization asymmetry present in SSR models which are optimized using gradient-based methods. Optimization asymmetry, as we define it, suggests that the parameters $\boldsymbol{\theta}$ of the structured model part require different optimization strategies compared to the weights $\mathbf{w}$ of a DNN.

To investigate this issue we trained two SSR models, where each one was trained with a different learning rate on the melanoma dataset. In the first experiment, we trained a model with a learning rate of 5e-3, as utilized in our primary work. Analyzing the corresponding validation loss curve (refer to Figure 21), one could argue for a decrease in the learning rate due to a peak and fast optimization in the early optimization phases. Thus, we tested a lower learning rate of 1e-4. This leads to a smoother loss curve and is, therefore, more trustworthy at first sight (as shown in Figure 22). However, the lower learning rate shows an optimization issue for the parameter θ_{age} in the structured model part.

To highlight this issue, we take a look at the posterior $p(\theta_{\text{age}}|\mathcal{D})$ and compare the posterior approximation of our approach against the Laplace approximation obtained from the trained SSR model which could be affected by optimization issues. We repeated the training over the six folds from the melanoma dataset with different weight initialization. As in all experiments, we selected the model with the lowest validation loss to perform the Laplace approximation.

Figure 20 displays the results from our primary work, which employed the larger learning rate, while Figure 23 illustrates the posterior $p(\theta_{\text{age}}|\mathcal{D})$ using the lower learning rate. In Figure 23 we see the optimization issue because the Laplace approximation exhibits significantly higher variance in the expectation of the posterior across the six folds compared to our approach and the results shown in Figure 20.

We argue that this asymmetry arises because the change a parameter must undergo from initialization to maximum likelihood solution is typically much larger for $\boldsymbol{\theta}$ compared to $\mathbf{w}$. Our intuition behind this is, that it exists an arbitrary number of solutions for $\mathbf{w}$, including those closer to the initialization point. Therefore, optimizing $\boldsymbol{\theta}$ may necessitate a greater number of optimization steps or larger learning rates compared to $\mathbf{w}$. We observed this in our experiments as the expectation of $p(\theta_{\text{age}}|\mathcal{D})$ still show changes when optimization is finished according to the validation loss. Thus, in the epoch where we stop training due to the signs of overfitting, the parameter θ_{age} of the structured model part still shows a dependence on the initial value. So while a lower learning rate produces a smoother learning curve, the increased learning rate leads to more stable results for the structured part (cf. Figure 20 vs. 23).

This highlights the challenges of optimizing SSR models, which we attribute to what we refer to as *optimization asymmetry*.

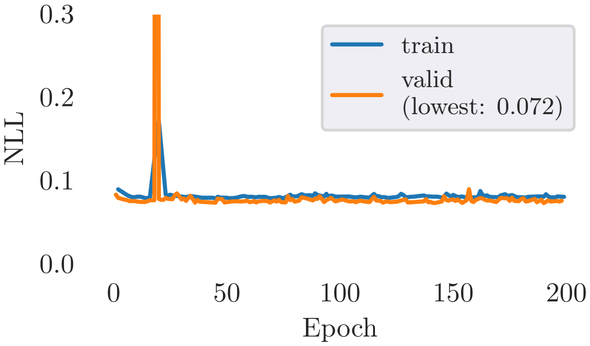

Figure 21: Validation and training negative log-likelihood (NLL) (Lower is better). This loss curve corresponds to the Laplace approximation training on fold 3 with the larger learning rate 5e-3

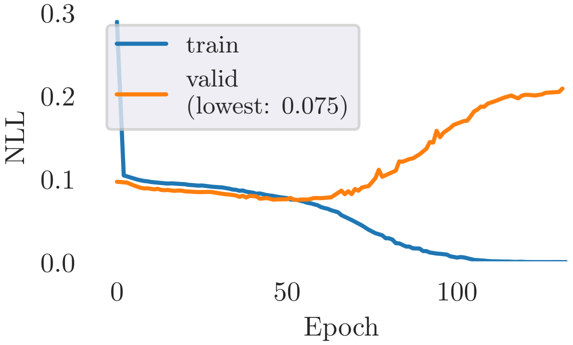

Figure 22: Same validation and training negative log-likelihood as shown in Figure 21 while learning rate for the SSR model was 1e-4 compared to 5e-3

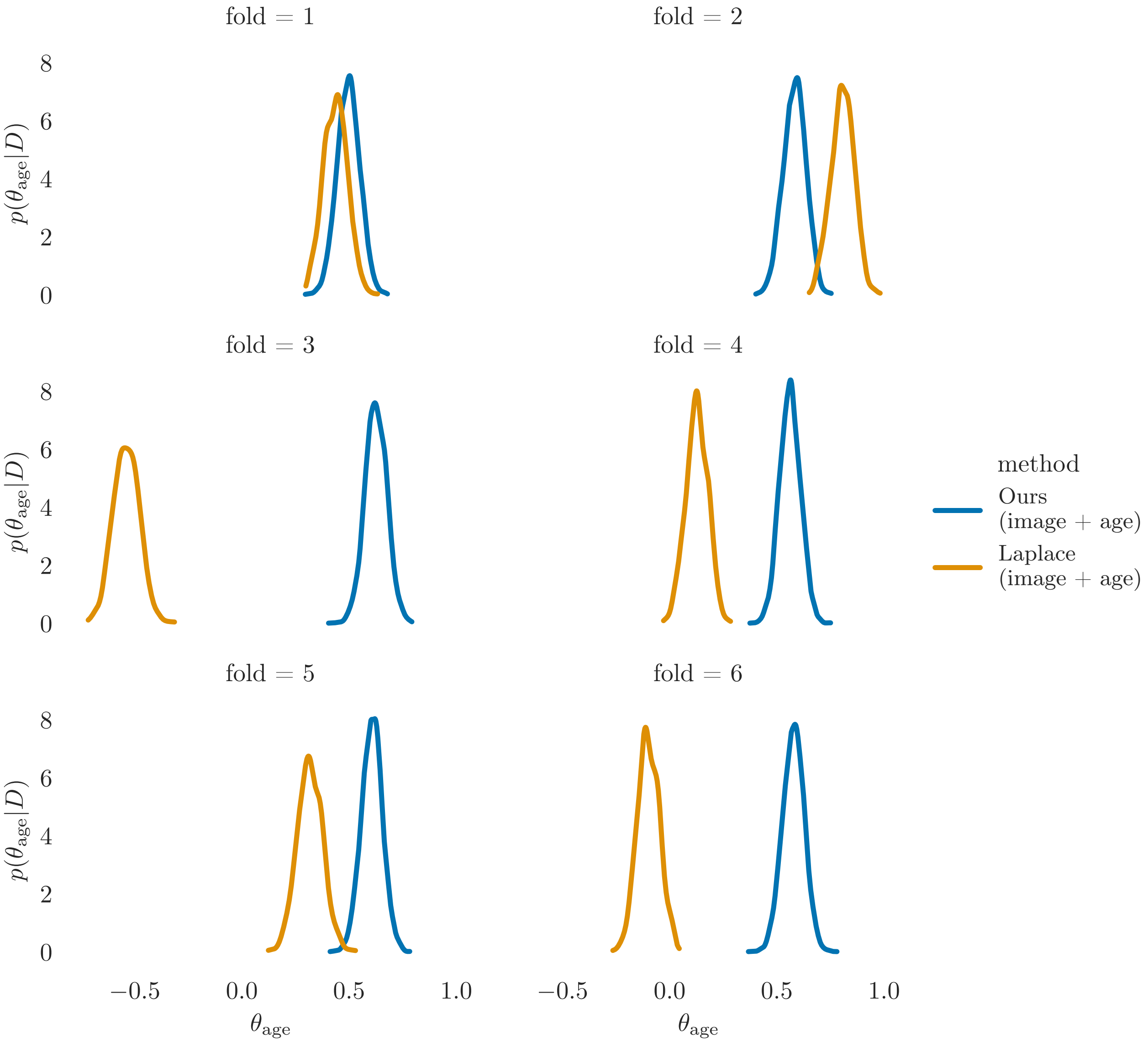

Figure 23: Same analysis as shown in Figure 20, but the Laplace approximation model is trained with a smaller learning rate (1e-4). However, according to the validation loss (cf. Figure 22), the Laplace approximation model was still able to achieve overfitting

D.6 Additional Results with the smaller Network Architecture from Wiese et al. (2023) on the UCI Benchmark

For the sake of completeness, we also conducted our benchmark study using the 'smaller network f1' proposed by Wiese et al. (2023) on the simulated and UCI datasets. Table 3 confirms our previous results, showing that increasing the subspace dimension improves predictive performance. Additionally, with this network architecture, our approach is capable of achieving performance close to HMC on almost every dataset, and it outperforms in most cases the other two approximation methods.

Table 3: Normalized expected test log pointwise predictive density (LPPD; larger is better) with the "smaller network f1" introduced by Wiese et al. (2023), comprising a single hidden layer with three neurons. The values within parentheses represent the standard errors of the predictive density per data point. The best method, excluding MCMC (representing an approximate upper bound), and all methods within one standard error of the best method are highlighted in bold.

dataset	MCMC	Subspace (k=2)	Subspace (k=5)	Deep Ens.	Laplace Appr.
DS	-0.53 (±0.09)	**-0.58 (± 0.11)**	**-0.60 (± 0.11)**	**-0.58 (±0.11)**	**-0.57 (±0.10)**
DI	0.79 (±0.06)	0.51 (± 0.06)	**0.60 (± 0.05)**	**0.56 (±0.06)**	0.53 (±0.07)
DR	0.64 (±0.10)	-0.39 (± 0.11)	**0.57 (± 0.12)**	-1.46 (±0.06)	-27.39 (±3.65)
Airfoil	-0.74 (±0.04)	**-0.88 (± 0.05)**	**-0.83 (± 0.06)**	-1.62 (±0.03)	-1.78 (±0.13)
Concrete	-0.41 (±0.05)	**-0.53 (± 0.06)**	**-0.50 (± 0.06)**	-1.59 (±0.03)	-14.49 (±1.02)
Diabetes	-1.20 (±0.07)	-1.24 (± 0.09)	**-1.17 (± 0.06)**	-1.47 (±0.07)	-1.46 (±0.09)
Energy	0.92 (±0.04)	-0.05 (± 0.07)	**0.62 (± 0.08)**	-1.76 (±0.02)	-31.74 (±1.88)
ForestF	-1.37 (±0.07)	-1.47 (± 0.08)	**-1.37 (± 0.07)**	-1.60 (±0.06)	-2.39 (±0.16)
Yacht	1.90 (±0.16)	**1.13 (± 0.47)**	**1.20 (± 0.44)**	-1.14 (±0.14)	-5.60 (±1.51)

D.7 Time Consumption

In the following analysis, we discuss the additional time consumption associated with using a larger subspace dimension. We focus solely on the time spent during the training of the subspace construction. To investigate this, we optimized a plain SSR model (k=0), which serves as our baseline, and trained our Semi-Subspace models for $k = 1, 3, 7, 15$ using the Algorithm 1 stopping after 50 epochs. Note that the weights of these four Semi-Subspace models are controlled using the Bézier curve, as outlined in Equation 2. We also excluded data preprocessing and model instantiation from the time computation. The following time computations were carried out on a *NVIDIA GeForce RTX 3080 Ti* GPU device. Additionally, we optimized the time consumption by fully utilizing the GPU capacity through pre-loading data onto its storage.

If we compare the time consumption of the plain SSR model (k=0) with our Semi-Subspace models we observe an initial offset of around $2.259s - 2.219s = 0.039s$. In addition, we see that our Semi-Subspace model scales linearly with k with a moderate slope.

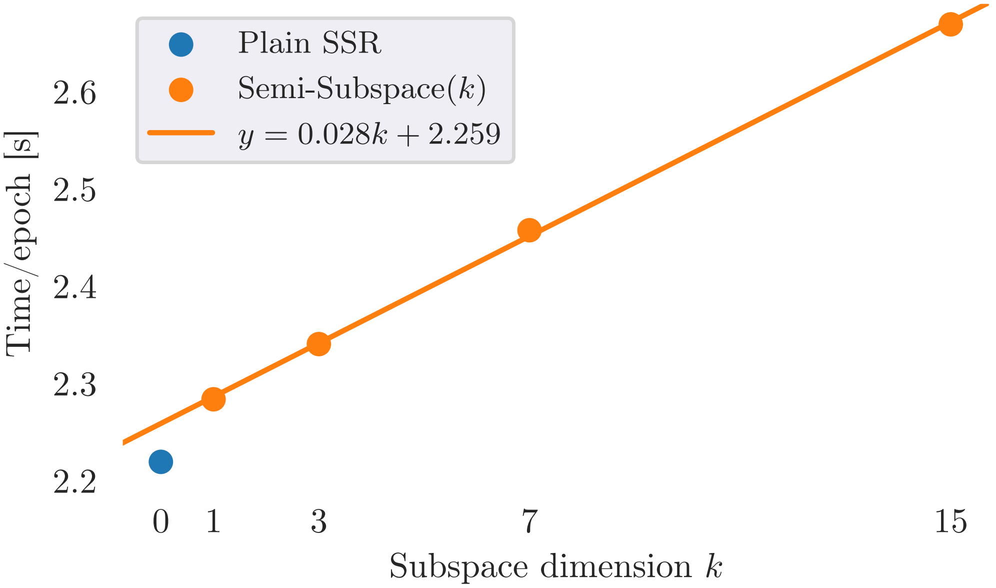

Figure 24: Time consumption required to optimize the SSR model per epoch depending on the subspace dimension k. $k = 0$ symbolizes the time consumption to train a plain SSR, whereas $k > 0$ depicts the training of the Semi-Subspace model with respective $k + 1$ control points.