A ADDITIONAL DISCUSSION OF SUBSPACE INFERENCE

A.1 LOSS OF MEASURE

When mapping into a lower-dimensional subspace of the true parameter space, we lose the ability to invert the transform and thus measure (i.e. volume of the distribution) is lost. Consider the following simple example in $\mathbb{R}^2$. First, form a spherical density, $p(x, y) = \mathcal{N}(0, I_2)$, and then fix x and y along a slice, such that $x - y = c$. The support of the resulting distribution has *no area*, since it represents a line with no width). For this reason, it is more correct to consider the subspace model (2) as a different model that shares many of the same functional properties as the fully parametrized model, rather than a re-parametrized version of the same model. Indeed, we cannot construct a Jacobian matrix to represent the density in the subspace.

A.2 POTENTIAL BENEFITS OF SUBSPACE INFERENCE

Quicker Exploration of the Posterior Reducing the dimensionality of parameter space enables significantly faster mixing of MCMC chains. For example, the expected number of likelihood evaluations needed for an accepted sample drawn using Metropolis-Hastings or Hamiltonian Monte Carlo (HMC) respectively grow as d^2 and $d^{5/4}$. If the dimensionality of the subspace grows as $K = \log d$, for example, then we would expect the runtime of Metropolis-Hastings to produce independent samples to grow at a modest $2 \log d$. We also note that the structure of the subspace may be much more amenable to exploration than the original posterior, requiring less time to effectively cover. For example, Maddox et al. (2019) show that the loss in the subspace constructed from the principal components of SGD iterates is approximately locally quadratic. On the other hand, if the subspace has a complicated structure, it can now be traversed using ambitious exploration methods which do not scale to higher dimensional spaces, such as parallel tempering (Geyer and Thompson, 1995).

Potential Lack of Degeneracies Given the restriction of DNNs into a subspace, we may expect that the subspace model concentrates to a single point in parameter space. This is in contrast to most DNNs, as due to the singularity of the Fisher information (e.g. Watanabe (2007)) and the fact that interconnected paths between global minima exist (Garipov et al., 2018), DNNs seem to concentrate in parameter space to connected point masses and gorges of global minima, where all solutions are exactly the same in function space. We expect then that the subspace model may be easier to theoretically analyze and could be more interpretable as a result.

B APPROXIMATE INFERENCE METHODS

We can use MCMC methods to approximately sample from $p(z|\mathcal{D})$, or we can perform a deterministic approximation $q(z|\mathcal{D}) \approx p(z|\mathcal{D})$, for example using Laplace or a variational approach, and then sample from q. We particularly consider the following methods, although there are many other possibilities. The inference procedure is an experimental design choice.

Slice Sampling As the dimensionality of the subspace K is low, gradient-free methods such as slice sampling (Neal et al., 2003) and elliptical slice sampling (ESS) (Murray et al., 2010) can be used to sample from the projected posterior distribution. Elliptical slice sampling is designed to have no tuning parameters, and only requires a Gaussian prior in the subspace. [7] For networks that cannot evaluate all of the training data in memory at a single time, it is easily possible to sum the loss over mini-batches computing a full log probability, without storing gradients.

NUTS The No-U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014) is an HMC method (Neal et al., 2011) that dynamically tunes the hyper-parameters (step-size and leapfrog steps) of HMC. [8] NUTS has the advantage of being nearly black-box: only a joint likelihood and its gradients need to be defined. However, full gradient calls are required, which can be difficult to cache and a constant factor slower than a full likelihood calculation.

Simple Variational Inference One can perform variational inference in the subspace using the fully-factorized Gaussian posterior approximation family for $p(z|\mathcal{D})$, from which we can sample to form a Bayesian model average. Fully-factorized Gaussians are among the simplest and the most common variational families. Unlike ESS or NUTS, VI can be trained with mini-batches (Hoffman et al., 2013), but is often practically constrained in the distributions it can represent.

RealNVP Normalizing flows, such as RealNVP (Dinh et al., 2017), parametrize the variational distribution family with invertible neural networks, for flexible non-Gaussian posterior approximations.

[7] We use the Python implementation at `https://github.com/jobovy/bovy_mcmc/blob/master/bovy_mcmc/elliptical_slice.py`.

[8] Implemented in Pyro (Bingham et al., 2018).

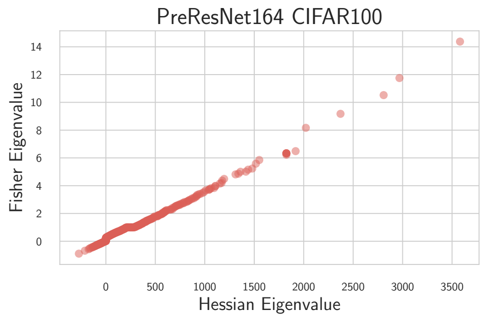


Figure 6: Plot of 300 eigenvalues of the Fisher and Hessian matrices for a PreResNet164 on CIFAR100. A clear separation exists between the top 20 or so eigenvalues and the rest, which are crowded together.

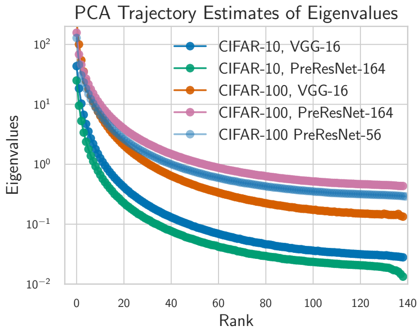


Figure 7: Eigenvalues of trajectory covariance (explained variance proportion) estimated from randomized SVD across three architectures on CIFAR-10 and CIFAR-100 plotted on a log-scale. The trajectory decays extremely quickly, decaying towards 0 around 10-20 setps.

C EIGEN-GAPS OF THE FISHER AND HESSIAN MATRICES

We can see similar behavior within the eigenvalues of both the Hessian and the empirical Fisher information matrix, at the end of training in Figure 6. To compute these eigenvalues, we used a GPU-enabled Lanczos method in GPyTorch (Gardner et al., 2018) on a pre-trained PreResNet164. We ran Lanczos for 100 steps, estimating 100 eigenvalues, before shifting by the maximum eigenvalue, and running for 200 steps, estimating 200 eigenvalues. Lanczos tends to converge from the "outside in" so to speak, see Chaper 7 of Demmel (1997) for theoretical guarantees, so that it ought to be possible to pick up eigengaps. As such, we would expect the training dynamics of SGD to primarily use these much larger eigenvalues, a finding shown empirically by Li et al. (2018b); Gur-Ari et al. (2019).

D ADDITIONAL REGRESSION UNCERTAINTY VISUALIZATIONS

In Figure 8 we present the predictive disribution plots for all the inference methods and subspaces. We additionally visualize the samples over poterior density surfaces for each of the methods in Figure 9.

E UCI REGRESSION EXPERIMENTAL DETAILS

E.1 SETUP

In all experiments, we replicated over 20 trials reserving 90% of the data for training and the other 10% for testing, following the set-up of Bui et al. (2016) and Wilson et al. (2016).

E.1.1 Gaussian test likelihood

In Bayesian model averaging, we compute a Gaussian estimator $\mathcal{N}(y|\hat{\mu}, \hat{\sigma}^2)$ based on sample statistics[9], where $\hat{\mu}(x) = \frac{1}{J}\sum_{i=1}^{J} \mu(x; w_i)$, $\hat{\sigma}^2(x) = \frac{1}{J}\sum_{i=1}^{J} \left(\sigma^2(x; w_i) + \mu(x; w_i)^2\right) - \hat{\mu}(x)^2$, and w_i are samples from the approximate posterior (see Section 3.2).

E.1.2 Small Regression

For the small UCI regression datasets, we use the architecture from Wu et al. (2019) with one hidden layer with 50 units. We manually tune learning rate and weight decay, and use batch size of $N/10$ where N is the dataset size. All models predict heteroscedastic uncertainty (i.e. output a variance). In Table 2, we compare subspace inference methods to deterministic VI (DVI, Wu et al. (2019)) and deep Gaussian processes with expectation propagation (DGP1-50 Bui et al. (2016)). ESS and VI in the PCA subspace outperform DVI on two out of five datasets.

E.1.3 Large-Scale Regression

For the large-scale UCI regression tasks, we manually tuned hyper-parameters (batch size, learning rate, and epochs) to match the SGD DNN results in Table 1 of Wilson et al. (2016). Here, there is one significant difference which is that our networks use heteroscedastic uncertainty, while those networks use homoscedastic uncertainty (a fixed variance). However, we found that the results were similar in terms of RMSE, but fitting networks with heteroscedastic uncertainty allows for a principled comparison of test log-likelihood and calibration.

[9] This is the same estimator used in Wu et al. (2019) and Lakshminarayanan et al. (2017).

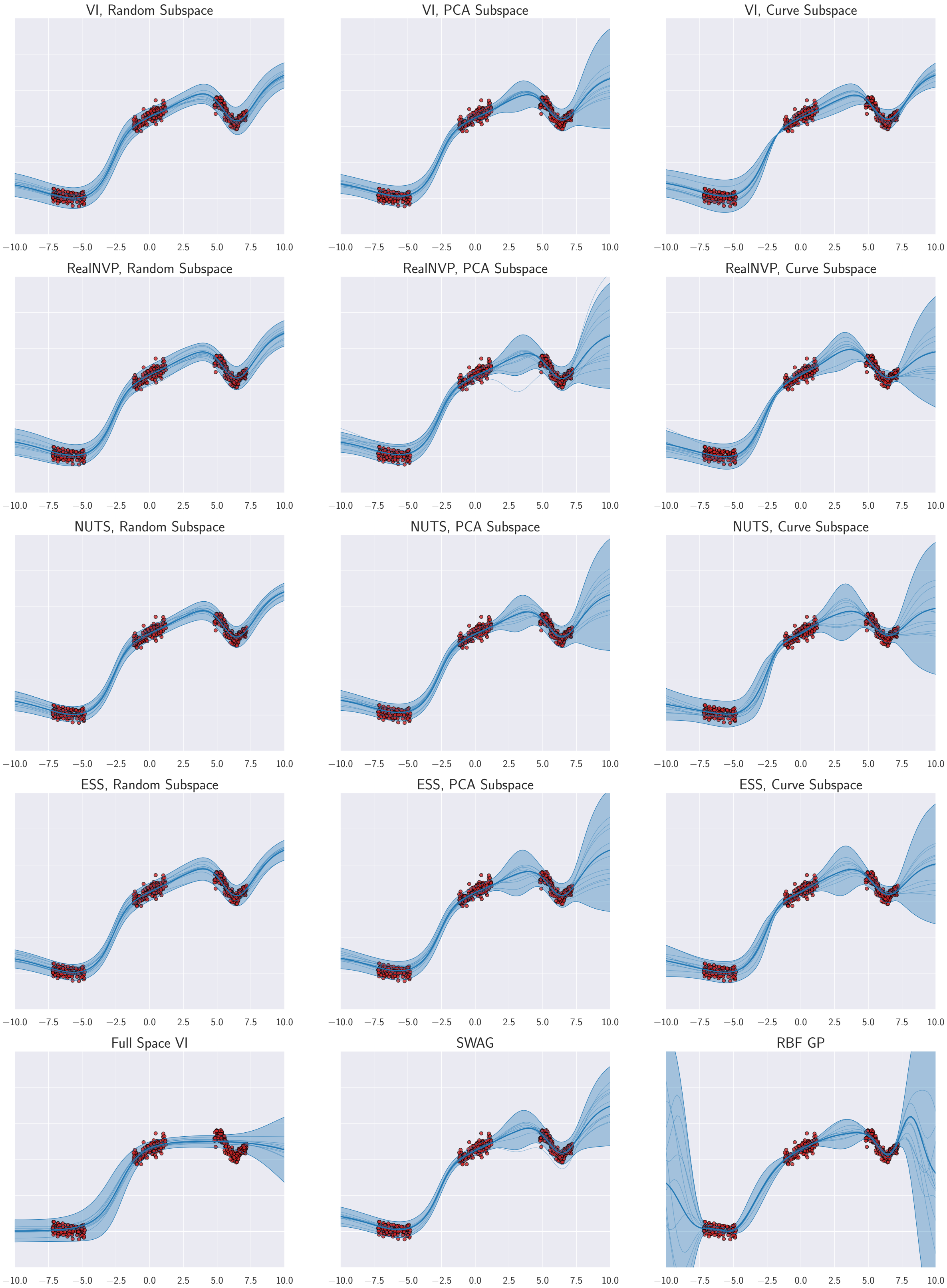


Figure 8: Regression predictive distributions across inference methods and subspaces. Data is shown with red circles, dark blue line shows predictive mean, lighter blue lines show sample predictive functions, and the shaded region represents ±3 standard deviation of predictive distribution at each point.

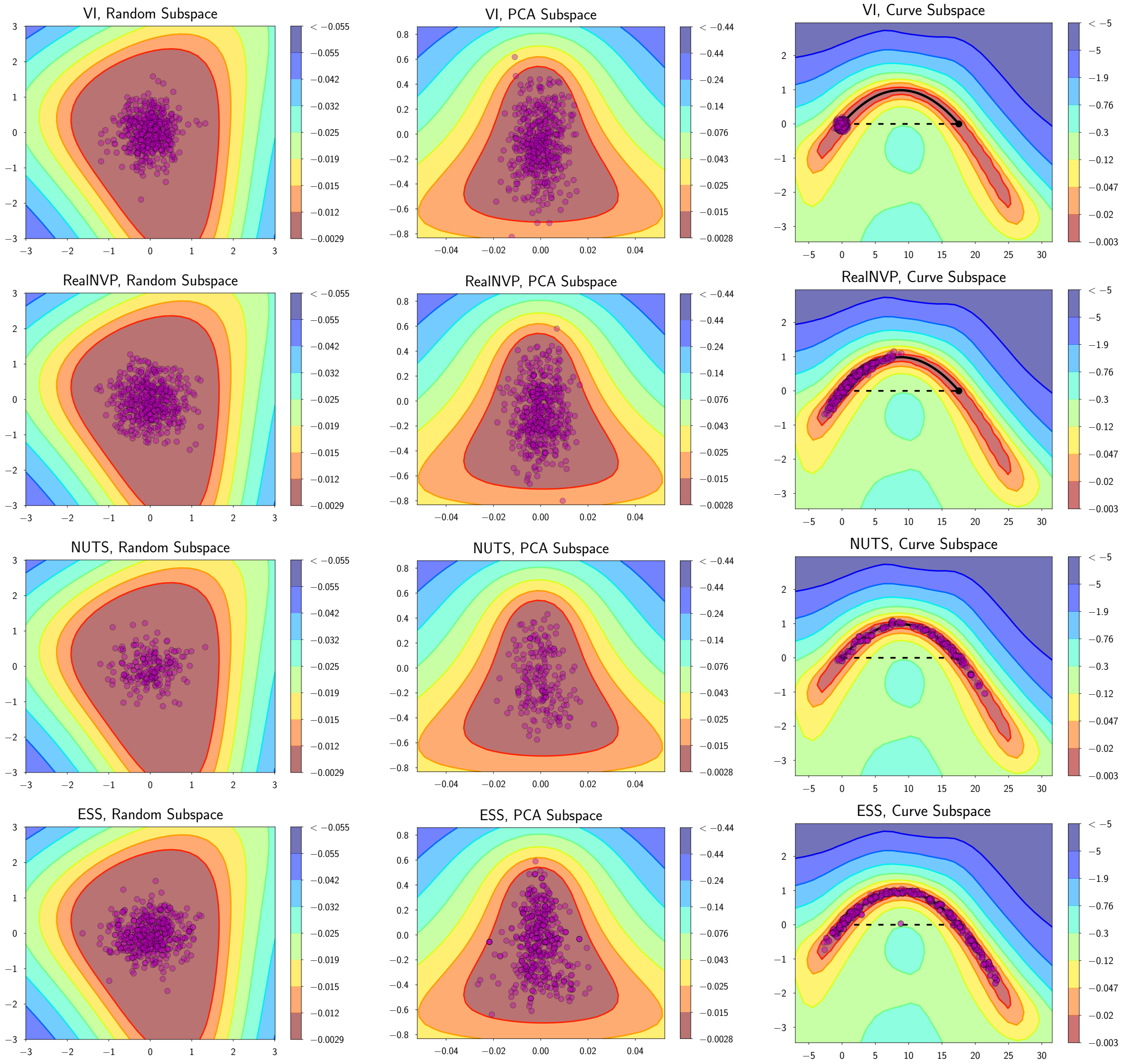


Figure 9: Posterior log-density surfaces and samples (magenta circles) for the synthetic regression problem across different subspaces and sampling methods.

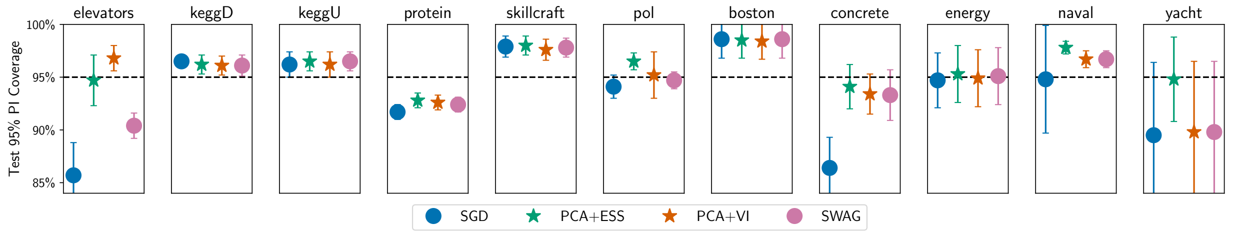


Figure 10: Coverage of 95% prediction interval for models trained on UCI datasets. In most cases, subspace inference produces closer to 95% coverage than models trained using SGD or SWAG.

Table 2: Unnormalized test log-likelihoods on small UCI datasets for Subspace Inference (SI), as well as direct comparisons to the numbers reported in deterministic variational inference (DVI, Wu et al. (2019)) and Deep Gaussian Processes with expectation propagation (DGP1-50, Bui et al. (2016)), and variational inference (VI) with the re-parameterization trick (Kingma et al., 2015).

dataset	N	D	SGD	PCA+ESS (SI)	PCA+VI (SI)	SWAG	DVI	DGP1-50	VI
boston	506	13	-2.752 ± 0.132	-2.719 ± 0.132	-2.716 ± 0.133	-2.761 ± 0.132	-2.41 ± 0.02	**-2.33** ± 0.06	-2.43 ±0.03
concrete	1030	8	-3.178 ± 0.198	-3.007 ± 0.086	**-2.994** ± 0.095	-3.013 ± 0.086	-3.06 ± 0.01	-3.13 ± 0.03	**-3.04** ±0.02
energy	768	8	-1.736 ± 1.613	-1.563 ± 1.243	-1.715 ± 1.588	-1.679 ± 1.488	**-1.01** ± 0.06	-1.32 ± 0.03	-2.38 ±0.02
naval	11934	16	6.567 ± 0.185	6.541 ± 0.095	**6.708** ± 0.105	6.708 ± 0.105	6.29 ± 0.04	3.60 ± 0.33	5.87 ±0.29
yacht	308	6	-0.418 ± 0.426	**-0.225** ± 0.400	-0.396 ± 0.419	-0.404 ± 0.418	-0.47 ± 0.03	-1.39 ± 0.14	-1.68 ±0.04

Table 3: RMSE on small UCI datasets. Subspace Inference (SI) typically performs comparably to SGD and SWAG.

	SGD	PCA+ESS (SI)	PCA+VI (SI)	SWAG
boston	3.504 ± 0.975	**3.453** ± 0.953	3.457 ± 0.951	3.517 ± 0.981
concrete	5.194 ± 0.446	5.194 ± 0.448	**5.142** ± 0.418	5.233 ± 0.417
energy	1.602 ± 0.275	1.598 ± 0.274	**1.587** ± 0.272	1.594 ± 0.273
naval	**0.001** ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000	**0.001** ± 0.000
yacht	0.973 ± 0.374	**0.972** ± 0.375	0.973 ± 0.375	0.973 ± 0.375

We additionally tried fitting models without a global variance parameter, but found that they were typically more over-confident than models with a global variance parameter.

Following Wilson et al. (2016), for the UCI regression tasks with more than 6,000 data points, we used networks with the following structure: [1000, 1000, 500, 50, 2], while for skillcraft, we used a network with: [1000, 500, 50, 2]. We used a learning rate of $1e-3$, doubling the learning rate of bias parameters, a batch size of 400, momentum of 0.9, and weight decay of $4e-3$, training for 200 epochs. For skillcraft and pol, we only trained for 100 epochs, while for skillcraft we used a learning rate of $5e-4$ and for keggD, we used a learning rate of $1e-4$. We additionally used a subspace prior of 1.0.

In Table 5, we report RMSE results compared to two types of approximate Gaussian processes (Salimbeni et al., 2018; Yang et al., 2015); note that the results for OrthVGP are reproduced from Appendix Table F of Salimbeni et al. (2018) but scaled by the standard deviation of the respective dataset. For the comparisons using Bayesian final layers (Riquelme et al., 2018), we trained SGD nets with the same architecture and used the second-to-last layer (ignoring the final 2 hidden unit layer as it performed considerably worse) for the Bayesian approach and then followed the same hyper-parameter setup as in the authors' codebase [10] with $a = b = 6$ and $\lambda = 0.25$.

We repeated each model over 10 random train/test splits; each test set consisted of 10% of the full dataset. All data was pre-processed to have mean zero and variance one.

F IMAGE CLASSIFICATION RESULTS

For the experiments on CIFAR datasets we are following the framework of (Maddox et al., 2019). We report the negative log-likelihood and accuracy for our method and baselines in Tables 8 and 9.

[10] `https://github.com/tensorflow/models/tree/master/research/deep_contextual_bandits`

Table 4: Calibration on small-scale UCI datasets for Subspace Inference (SI). Bolded numbers are those closest to 95% of the predicted coverage.

	N	D	SGD	PCA+ESS (SI)	PCA+VI (SI)	SWAG
boston	506	13	0.986 ± 0.018	0.985 ± 0.017	**0.984** ± 0.017	0.986 ± 0.018
concrete	1030	8	0.864 ± 0.029	**0.941** ± 0.021	0.934 ± 0.019	0.933 ± 0.024
energy	768	8	0.947 ± 0.026	0.953 ± 0.027	**0.949** ± 0.027	**0.951** ± 0.027
naval	11934	16	**0.948** ± 0.051	0.978 ± 0.006	0.967 ± 0.008	0.967 ± 0.008
yacht	308	6	0.895 ± 0.069	**0.948** ± 0.040	0.898 ± 0.067	0.898 ± 0.067

Table 5: RMSE comparison amongst methods on larger UCI regression tasks, as well as direct comparisons to the numbers reported in deep kernel learning with a spectral mixture kernel (DKL, (Wilson et al., 2016)), orthogonally decoupled variational GPs (OrthVGP, Salimbeni et al. (2018)), FastFood kernel GPs (FF, Yang et al. (2015) from Wilson et al. (2016)), and Bayesian final layers (NL, Riquelme et al. (2018)). Subspace based inference typically outperforms SGD and approximate GPs and is competitive with DKL.

dataset	N	D	SGD	PCA+ESS (SI)	PCA+VI (SI)	SWAG	DKL	OrthVGP	FF	NL
elevators	16599	18	0.092 ± 0.003	0.093 ± 0.004	0.090 ± 0.002	0.090 ± 0.002	**0.084** ± 0.02	0.0952	0.089 ± 0.002	0.097 ± 0.003
keggD	48827	20	0.121 ± 0.003	0.137 ± 0.031	0.137 ± 0.032	0.138 ± 0.032	**0.10** ± 0.01	0.1198	0.12 ± 0.00	0.121 ± 0.005
keggU	63608	27	0.125 ± 0.024	0.125 ± 0.023	0.125 ± 0.023	0.125 ± 0.023	**0.11** ± 0.00	0.1172	0.12 ± 0.00	0.122 ± 0.008
protein	45730	9	0.443 ± 0.009	**0.440** ± 0.007	0.444 ± 0.009	0.447 ± 0.011	0.46 ± 0.01	0.46071	0.47 ± 0.01	0.445 ± 0.008
skillcraft	3338	19	0.284 ± 0.015	0.286 ± 0.016	0.276 ± 0.015	0.298 ± 0.015	**0.25** ± 0.00		**0.25** ± 0.02	0.253 ± 0.05
pol	15000	26	3.018 ± 0.310	2.446 ± 0.151	**2.427** ± 0.161	2.452 ± 0.156	3.11 ± 0.07	6.61749	4.30 ± 0.2	4.09 ± 1.25

Table 6: Normalized test log-likelihoods on larger UCI datasets. Subspace methods outperform an approximate GP approach (OrthVGP), SGD, and Bayesian final layers (NL), typically often out-performing SWAG.

dataset	N	D	SGD	PCA+ESS (SI)	PCA+VI (SI)	SWAG	OrthVGP	NL
elevators	16599	18	−0.538 ± 0.108	−0.397 ± 0.05	**-0.380** ± 0.041	−0.395 ± 0.030	-0.4479	0.803 ± 0.04
keggD	48827	20	0.985 ± 0.022	0.995 ± 0.104	0.988 ± 0.106	0.984 ± 0.114	**1.0224**	0.675 ± 0.05
keggU	63608	27	0.700 ± 0.046	**0.707** ± 0.032	0.702 ± 0.043	**0.707** ± 0.038	0.7007	0.664 ± 0.05
protein	45730	9	−0.861 ± 0.027	**-0.834** ± 0.021	−0.849 ± 0.025	−0.861 ± 0.031	-0.9138	−0.619 ± 0.01
skillcraft	3338	19	−1.147 ± 0.035	−1.159 ± 0.034	**-1.109** ± 0.036	−1.181 ± 0.032	−0.05 ± 0.05	
pol	15000	26	1.290 ± 0.1834	**1.737** ± 0.043	1.728 ± 0.076	1.680 ± 0.075	0.1586	−2.84 ± 0.226

Table 7: Calibration on large-scale UCI datasets. Bolded numbers are those closest to 95 % of the predicted coverage).

dataset	N	D	SGD	PCA+ESS (SI)	PCA+VI (SI)	SWAG
elevators	16599	18	0.857 ± 0.031	$\mathbf{0.947} \pm 0.024$	0.968 ± 0.012	0.904 ± 0.012
keggD	48827	20	0.965 ± 0.002	0.962 ± 0.009	$\mathbf{0.961} \pm 0.009$	0.961 ± 0.01
keggU	63608	27	$\mathbf{0.962} \pm 0.012$	0.965 ± 0.009	$\mathbf{0.962} \pm 0.012$	0.965 ± 0.009
protein	45730	9	0.917 ± 0.007	$\mathbf{0.928} \pm 0.007$	0.926 ± 0.007	0.924 ± 0.007
skillcraft	3338	19	0.979 ± 0.010	0.980 ± 0.009	$\mathbf{0.976} \pm 0.010$	0.978 ± 0.009
pol	15000	26	0.941 ± 0.011	0.965 ± 0.008	$\mathbf{0.952} \pm 0.022$	0.947 ± 0.008

Table 8: NLL for various versions of subspace inference, SWAG, temperature scaling, and dropout.

Dataset	Model	PCA + VI (SI)	PCA + ESS (SI)	SWA	SWAG	KFAC-Laplace	SWA-Dropout	SWA-Temp
CIFAR-10	VGG-16	0.2052 ± 0.0029	0.2068 ± 0.0029	0.2621 ± 0.0104	$\mathbf{0.2016} \pm 0.0031$	0.2252 ± 0.0032	0.2328 ± 0.0049	0.2481 ± 0.0245
CIFAR-10	PreResNet-164	0.1247 ± 0.0025	0.1252 ± 0.0018	0.1450 ± 0.0042	$\mathbf{0.1232} \pm 0.0022$	0.1471 ± 0.0012	0.1270 ± 0.0000	0.1347 ± 0.0038
CIFAR-10	WideResNet28x10	0.1081 ± 0.0003	0.1090 ± 0.0038	0.1075 ± 0.0004	0.1122 ± 0.0009	0.1210 ± 0.0020	0.1094 ± 0.0021	$\mathbf{0.1064} \pm 0.0004$
CIFAR-100	VGG-16	0.9904 ± 0.0218	1.015 ± 0.0259	1.2780 ± 0.0051	$\mathbf{0.9480} \pm 0.0038$	1.1915 ± 0.0199	1.1872 ± 0.0524	1.0386 ± 0.0126
CIFAR-100	PreResNet-164	$\mathbf{0.6640} \pm 0.0025$	0.6858 ± 0.0052	0.7370 ± 0.0265	0.7081 ± 0.0162	0.7881 ± 0.0025		0.6770 ± 0.0191
CIFAR-100	WideResNet28x10	$\mathbf{0.6052} \pm 0.0090$	0.6096 ± 0.0072	0.6684 ± 0.0034	0.6078 ± 0.0006	0.7692 ± 0.0092	0.6500 ± 0.0049	0.6134 ± 0.0023

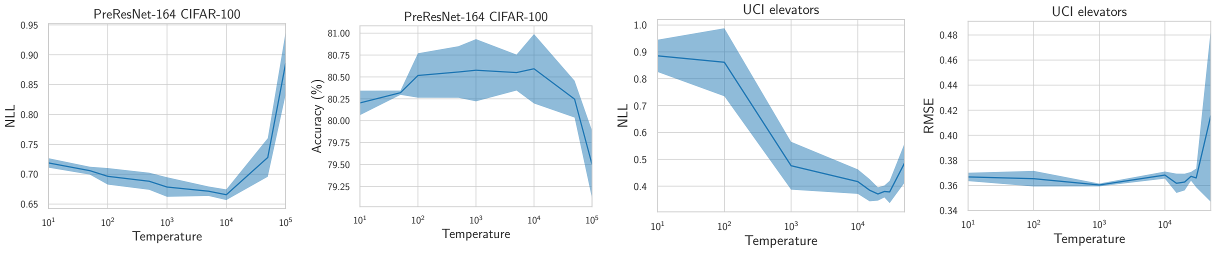
F.1 EFFECT OF TEMPERATURE

In this section we study the effect of temperature parameter T defined in (4) on the performance of subspace inference. We run elliptical slice sampling in a 5-dimensional PCA subspace for a PreResNet-164 on CIFAR-100. We show test performance as a function of temperature parameter in Figure 11 panels (a) and (b). Bayesian model averaging achieves strong results in the range $10^3 \leq T \leq 10^4$. We also observe that the value T has a larger effect on uncertainty estimates and consequently NLL than on predictive accuracy.

We then repeat the same experiment on UCI elevators using the setting described in Section 5.2.1. We show the results in Figure 11 panels (c), (d). Again, we observe that the performance is almost constant and close to optimal in a certain range of temperatures, and the effect of temperature on likelihood is larger compared to RMSE.

Table 9: Accuracy for various versions of subspace inference, SWAG, temperature scaling, and dropout.

Dataset	Model	PCA + VI (SI)	PCA + ESS (SI)	SWA	SWAG	KFAC-Laplace	SWA-Dropout	SWA-Temp
CIFAR-10	VGG-16	93.61 ± 0.02	**93.66** ± 0.08	93.61 ± 0.11	93.60 ± 0.10	92.65 ± 0.20	93.23 ± 0.36	93.61 ± 0.11
CIFAR-10	PreResNet-164	95.96 ± 0.13	95.98 ± 0.09	96.09 ± 0.08	96.03 ± 0.02	95.49 ± 0.06	**96.18** ± 0.00	96.09 ± 0.08
CIFAR-10	WideResNet28x10	96.32 ± 0.03	96.38 ± 0.05	**96.46** ± 0.04	96.32 ± 0.08	96.17 ± 0.00	96.39 ± 0.09	96.46 ± 0.04
CIFAR-100	VGG-16	**74.83** ± 0.08	74.62 ± 0.37	74.30 ± 0.22	74.77 ± 0.09	72.38 ± 0.23	72.50 ± 0.54	74.30 ± 0.22
CIFAR-100	PreResNet-164	80.52 ± 0.18	**80.54** ± 0.13	80.19 ± 0.52	79.90 ± 0.50	78.51 ± 0.05		80.19 ± 0.52
CIFAR-100	WideResNet28x10	**82.63** ± 0.26	82.49 ± 0.23	82.40 ± 0.16	82.23 ± 0.19	80.94 ± 0.41	82.30 ± 0.19	82.40 ± 0.16



(a) (b) (c) (d)

Figure 11: **(a)**: Test negative log-likelihood and **(b)**: accuracy as a function of temperature in (4) for PreResNet-164 on CIFAR-100. **(c)**: Test negative log-likelihood and **(d)**: RMSE as a function of temperature for our regression architecture (see Section 5.2) on UCI Elevators. We used ESS in a 5-dimensional PCA subspace to construct this plot. The dark blue line shows mean and shaded region shows standard deviation over 3 independent runs of the procedure.