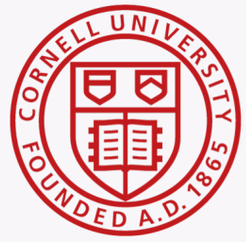
TWO APPROXIMATE SAMPLING METHODS FOR BAYESIAN DEEP LEARNING

WESLEY MADDOX

JOINT WORK WITH TIMUR GARIPOV, PAVEL IZMAILOV, POLINA KIRICHENKO, DMITRY VETROV, ANDREW GORDON WILSON



TWO APPROXIMATE SAMPLING METHODS FOR BAYESIAN DEEP LEARNING

WESLEY MADDOX

JOINT WORK WITH TIMUR GARIPOV, PAVEL IZMAILOV, POLINA KIRICHENKO, DMITRY VETROV, ANDREW GORDON WILSON

STRUCTURE

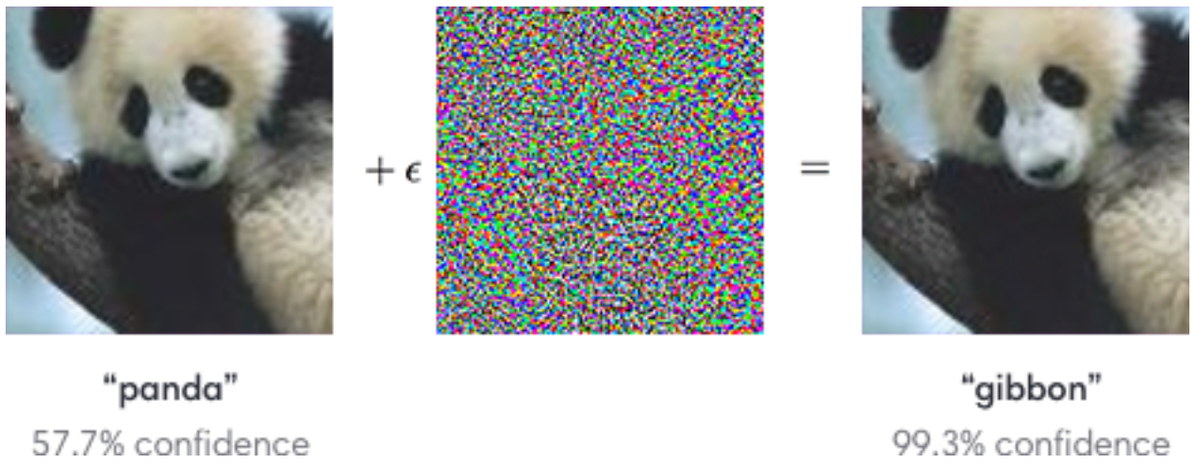
- Based off of:
  - *"A Simple Baseline for Bayesian Uncertainty in Deep Learning,"* **Maddox**, Garipov, Izmailov, Vetrov, Wilson. https://arxiv.org/abs/1902.02476. 2019
    - Code: https://github.com/wjmaddox/swa_gaussian
  - *"Subspace Inference for Bayesian Deep Learning,"* Izmailov, **Maddox**, Kirichenko, Garipov, Vetrov, Wilson. https://arxiv.org/abs/1907.07504. UAI, 2019
    - Code: https://github.com/wjmaddox/drbayes

STRUCTURE

- Introduction
- Stochastic Weight Averaging, Gaussian
- Subspace Inference

MOTIVATION

- Machine learning (ML) models are used for decision making, but accuracy isn't enough.
- Also should incorporate **uncertainty** quantification and **calibration.**



Is this a panda or a gibbon? From https://blog.openai.com/adversarial-example-research

WHY BAYESIAN METHODS?

- Allows **combining models** for **uncertainty quantification.**
  - Should give both **better predictions** and allow incorporation of prior knowledge.
- But....
  - Very high dimensional problems (complicated models).
  - Very large datasets.

BAYESIAN MODELING

- Likelihood: $p(\mathrm{Data}|\theta) = p(y|f(x;\theta))$ DNN, parameters θ
- Prior: $p(\theta)$
  - Possibly implicit to the training method
- Posterior: $p(\theta|\mathrm{Data}) \propto p(\mathrm{Data}|\theta)p(\theta)$ Form approximate posterior $\approx q(\theta)$
- Inference (Bayesian model averaging)

$$p(y^*|\mathrm{Data}) = \mathbb{E}_{p(\theta|\mathrm{Data})}p(y^*|\theta)$$
$$\approx \frac{1}{K}\sum_{k=1} p(y^*|\theta_k), \theta_k \sim q(\theta)$$

APPROXIMATE BAYESIAN INFERENCE

- How?

  Deep Learning Version

  - Laplace: $p(\theta|y) \approx N(\theta_{MAP}, (H(\theta_{MAP}) + \lambda I)^{-1})$

    KFAC Laplace, Ritter et al, 2018

  - Variational Bayes:

    $$p(\theta|y) \approx N(\mu, S)$$

    Bayes by Backprop, Blundell et al, 2015

    MC Dropout, Gal & Ghahramani, 2016

- Markov Chain Monte Carlo

  Stochastic Gradient HMC, Chen et al, 2014

A SIMPLE BASELINE FOR BAYESIAN UNCERTAINTY IN DEEP LEARNING

SUMMARY

- Stochastic Weight Averaging **(Izmailov et al, UAI, 2018)** computes first moment of weights given from SGD iterates with a modified learning rate schedule.
- We propose to keep the variance as well to form a **Gaussian approximation in weight space.**
- Sample from Gaussian to compute **Bayesian model averages and estimate uncertainty.**
- **Theoretically motivated** from results on SGD & relation of iterates to Gaussian distribution (Ruppert, 1992 and Mandt et al, 2017).

THEORETICAL MOTIVATION

THEORETICAL MOTIVATION: SGD AS APPROXIMATE BAYESIAN INFERENCE (MANDT ET AL. 2017)

Log Joint Distribution $$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N} -\log p(x_n|\theta) - \frac{1}{N}p(\theta)$$

Minibatch Loss $$\hat{\mathcal{L}}(\theta) = \frac{1}{S}\sum_{n=1}^{S} -\log p(x_n|\theta) - \frac{1}{SN}p(\theta) \quad S << N$$

Gradient Estimates $$\hat{g}(\theta) = \nabla_\theta \hat{L}_S(\theta)$$

Parameter Updates (SGD) $$\theta_{t+1} = \theta_t - \epsilon \hat{g}_S(\theta)$$

SGD AS APPROXIMATE BAYESIAN INFERENCE (MANDT ET AL. 2017)

- **Assumptions**
  - 1) Gradient noise is Gaussian

$$\widehat{g}_S(\theta) \approx g(\theta) + \frac{1}{\sqrt{S}}\Delta g(\theta), \quad \Delta g(\theta) \sim \mathcal{N}(0, C(\theta))$$

  - 2) Gradient covariance is approximately constant & full rank

$$C(\theta) \approx C = BB^\top$$

SGD AS APPROXIMATE BAYESIAN INFERENCE (MANDT ET AL. 2017)

- **Assumptions**
  - 3) We can approximate the finite difference equation with a stochastic differential equation.

$$\Delta\theta(t) = \theta(t+1) - \theta(t) = -\epsilon g(\theta(t)) + \frac{\epsilon}{\sqrt{S}} B \Delta W, \quad \Delta W \sim \mathcal{N}(0, I)$$

  - 4) SGD iterates are in a region well approximated by a quadratic.

$$\mathcal{L}(\theta) = \frac{1}{2}\theta^\top A \theta, \quad A > 0$$

SGD AS APPROXIMATE BAYESIAN INFERENCE (MANDT ET AL 2017)

- Assumptions 1 -> 4 yield a multivariate Ornstein-Uhlenbeck process

$$d\theta(t) = -\epsilon A\theta(t)dt + \frac{1}{\sqrt{S}}\epsilon B dW(t)$$

- With a Gaussian stationary distribution

$$q(\theta) \approx \exp\left\{-\frac{1}{2}\theta^\top \Sigma^{-1}\theta\right\}.$$

- Covariance satisfies $\Sigma A + A\Sigma = \frac{\epsilon}{S}BB^\top$

SGD AS APPROXIMATE BAYESIAN INFERENCE (MANDT ET AL 2017)

- **Theorem:** Under assumptions 1-4, the optimal constant learning rate that minimizes the KL divergence from the stationary distribution of SGD to the posterior is:

$$\epsilon^* = 2\frac{S}{N}\frac{D}{\mathrm{Tr}(BB^\top)},$$

  - D: dimensionality of parameters

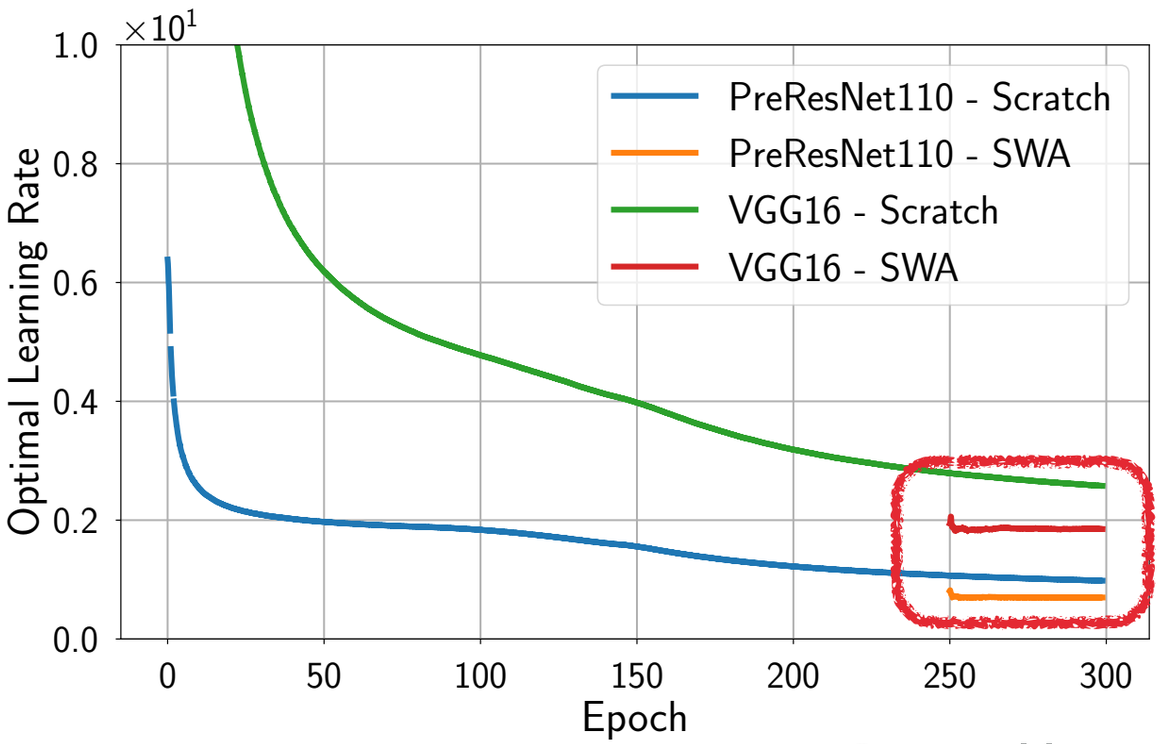
- **But…. Do Assumptions 1-4 actually hold for DNNs?**

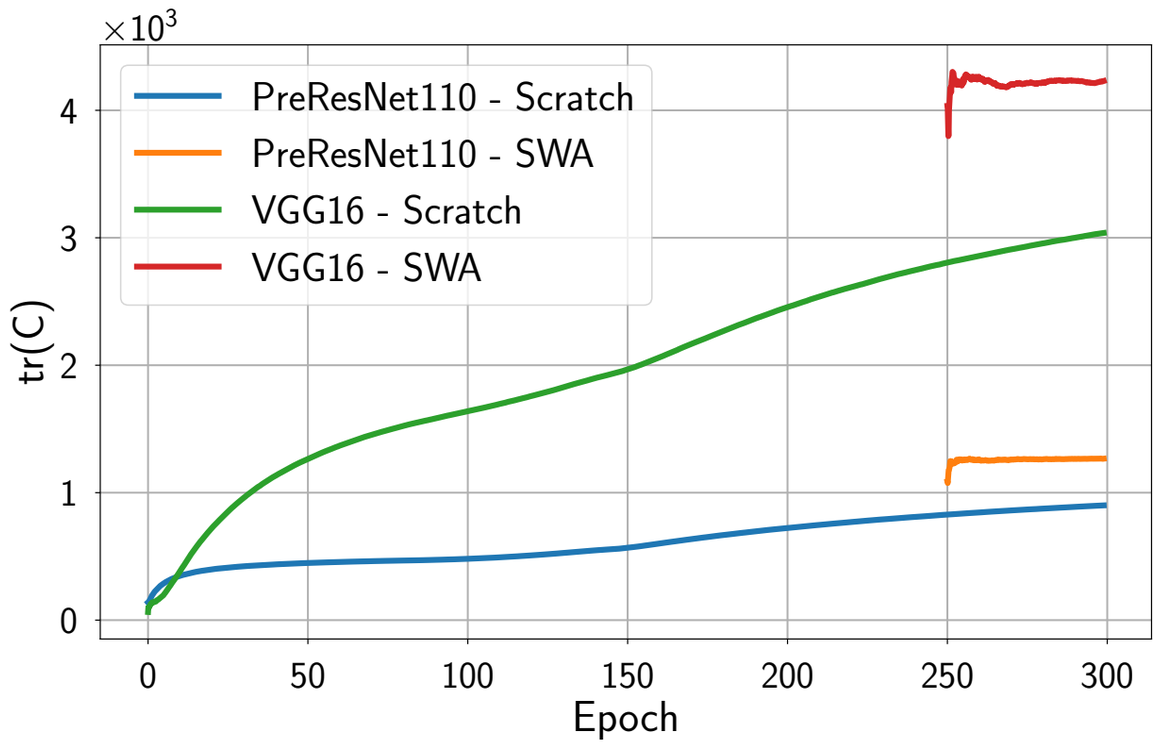
Note that KL(q|f) is minimized but isn't zero :

But iterate averaging samples from the true posterior

DO THE ASSUMPTIONS HOLD?

- Assumption 2:



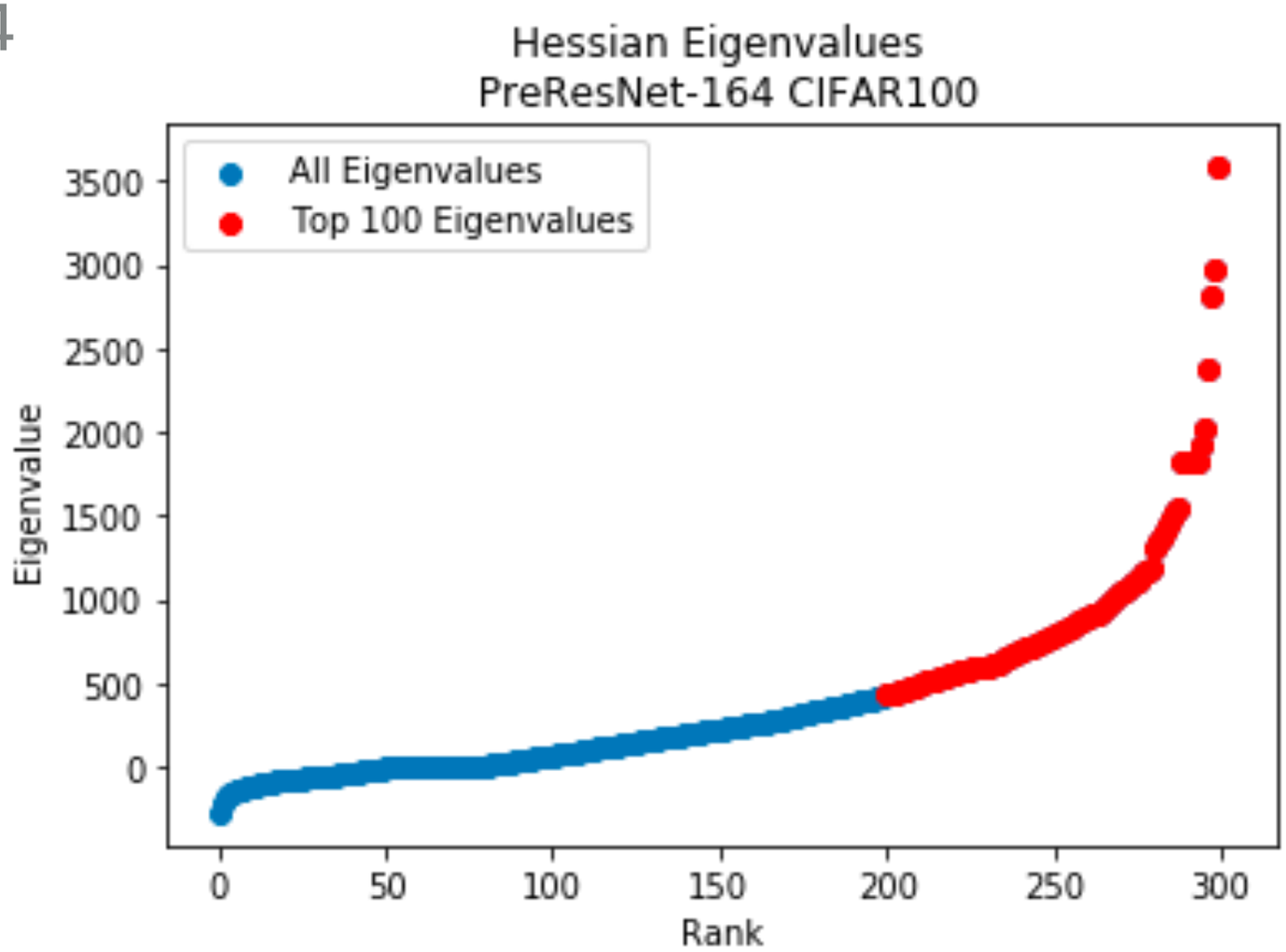


Optimal lr = 1-3

- Constant near end of training but 10-20x too large.

DO THE ASSUMPTIONS HOLD?

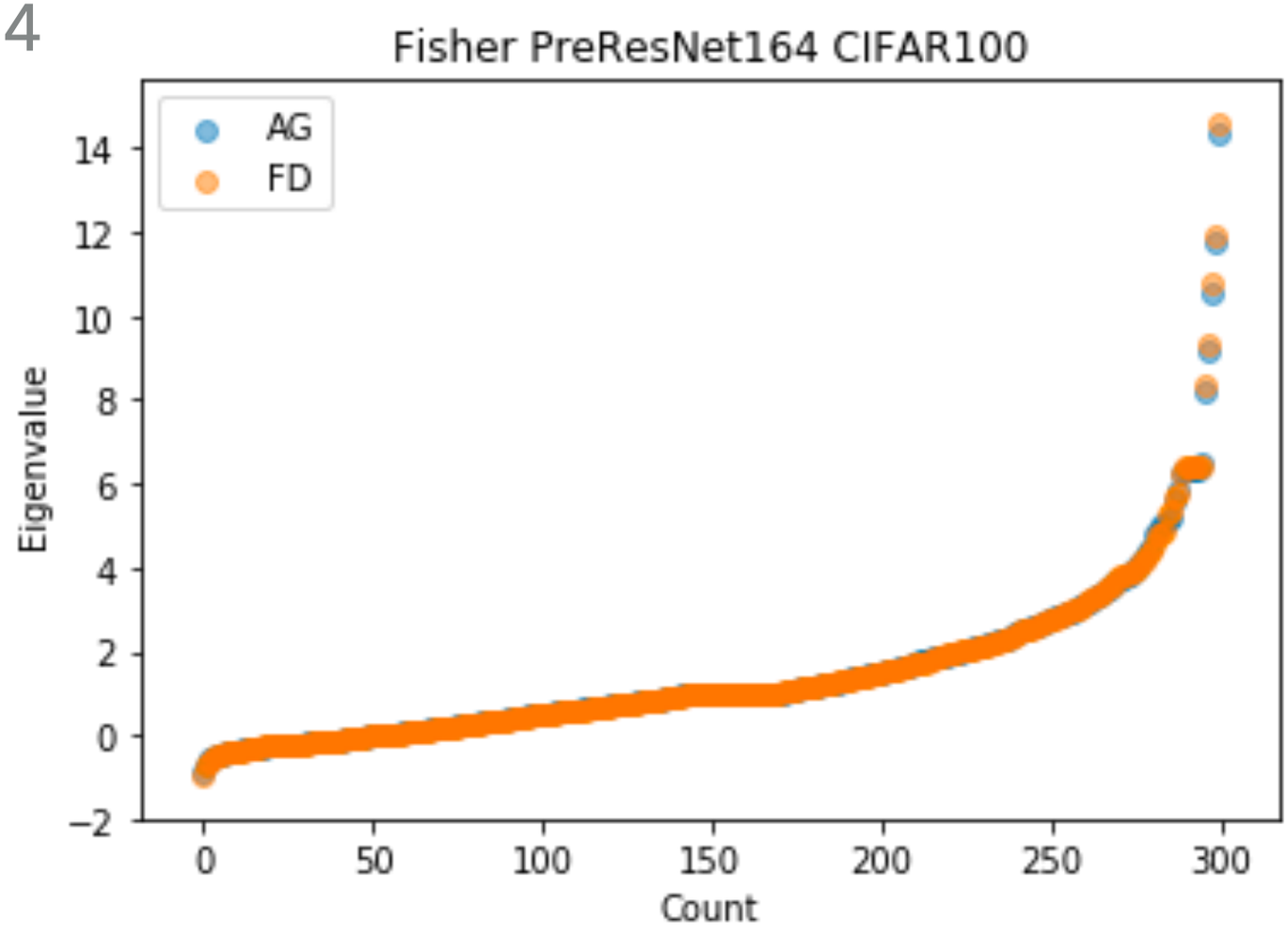
- Assumption 4



- Max: 3580, Min: -**272.** Using Lanczos (GPyTorch)

DO THE ASSUMPTIONS HOLD?

- Assumption 4



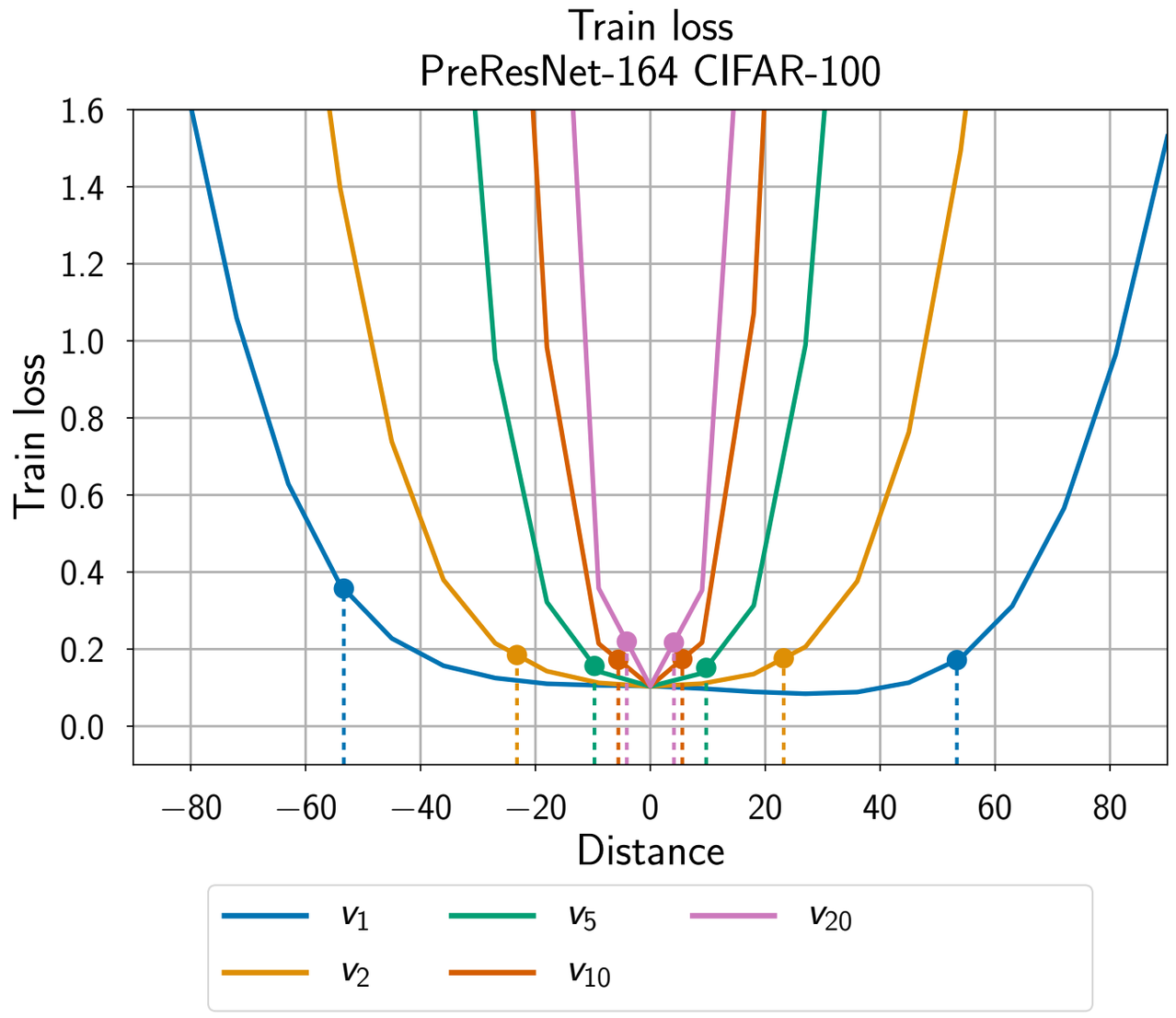
- Max:14, Min: -1 Using Lanczos (GPyTorch)

EMPIRICAL FINDINGS

1D surface along eigenvectors of approx. posterior

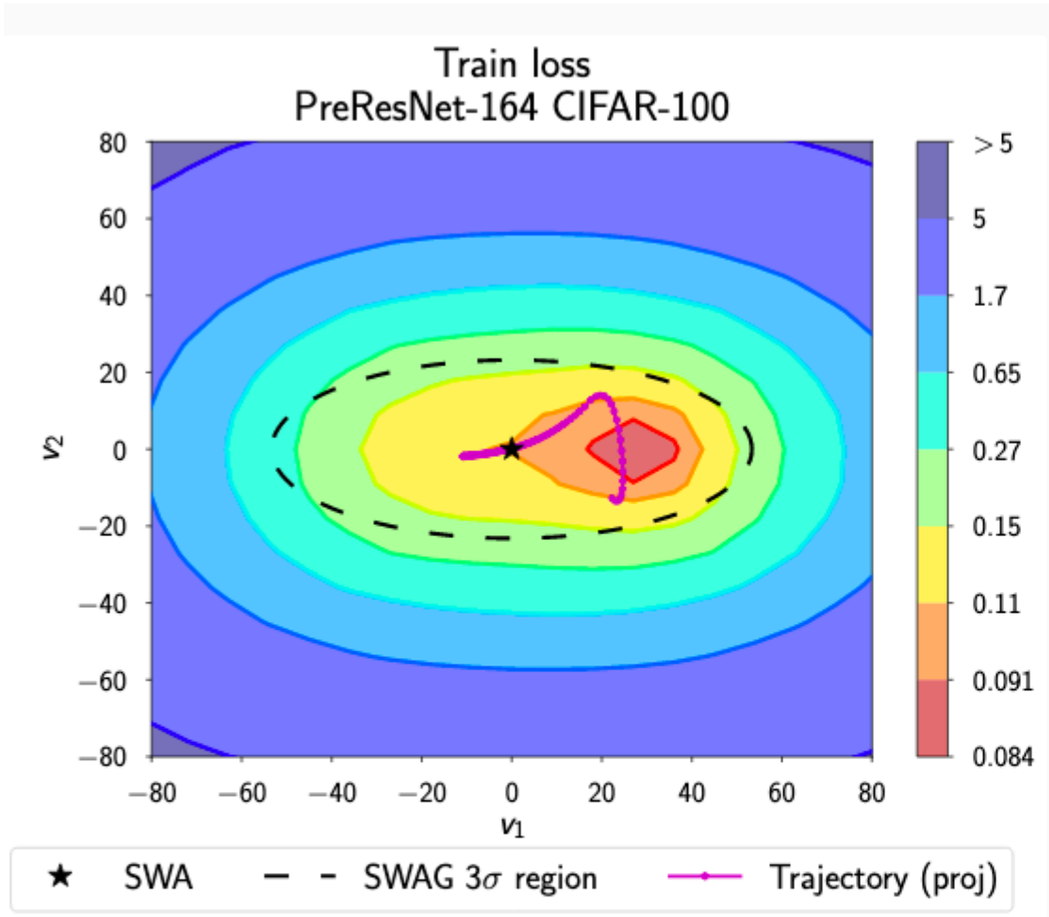
$$\phi(t) = \mathcal{L}(\theta_{\text{SWA}} + t \cdot \frac{v_i}{\|v_i\|})$$

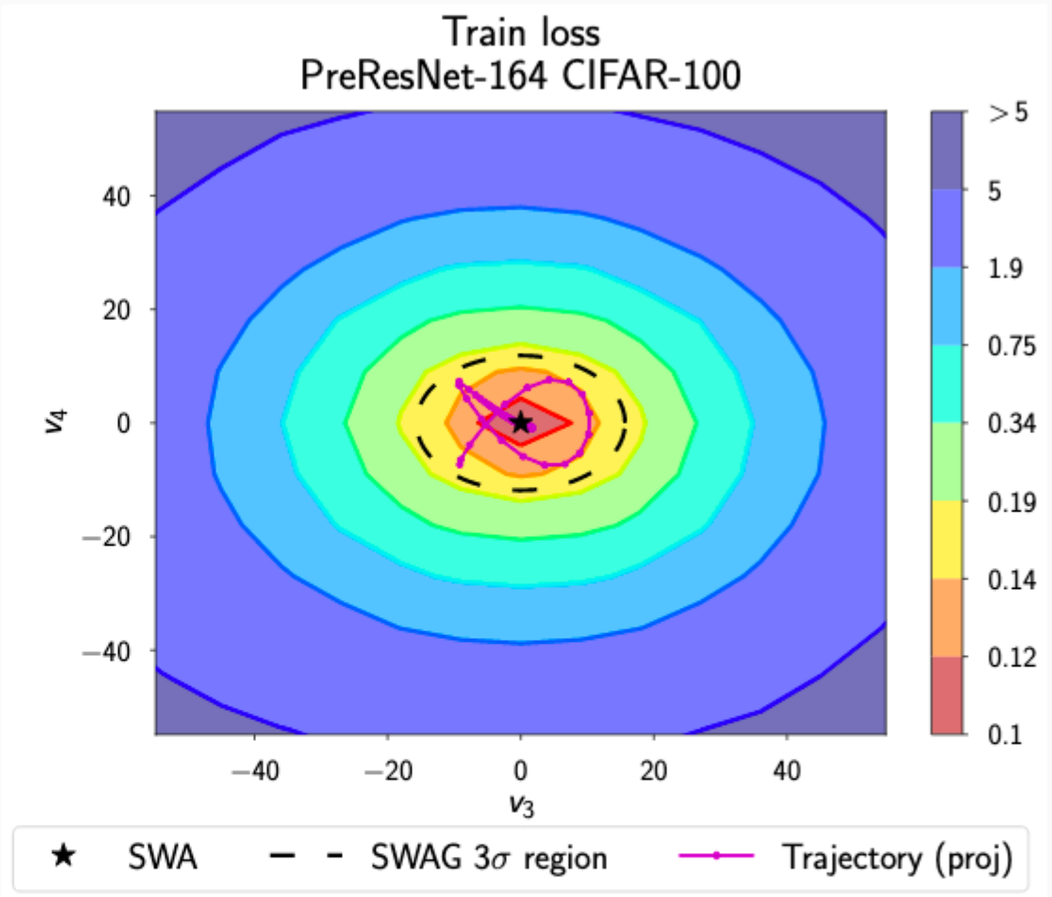
$$\Sigma_{\text{low-rank}} = \frac{1}{K-1}\widehat{D}\widehat{D}^\top = V\Lambda V^T$$



EMPIRICAL FINDINGS

2D surface along eigenvectors of approx. posterior



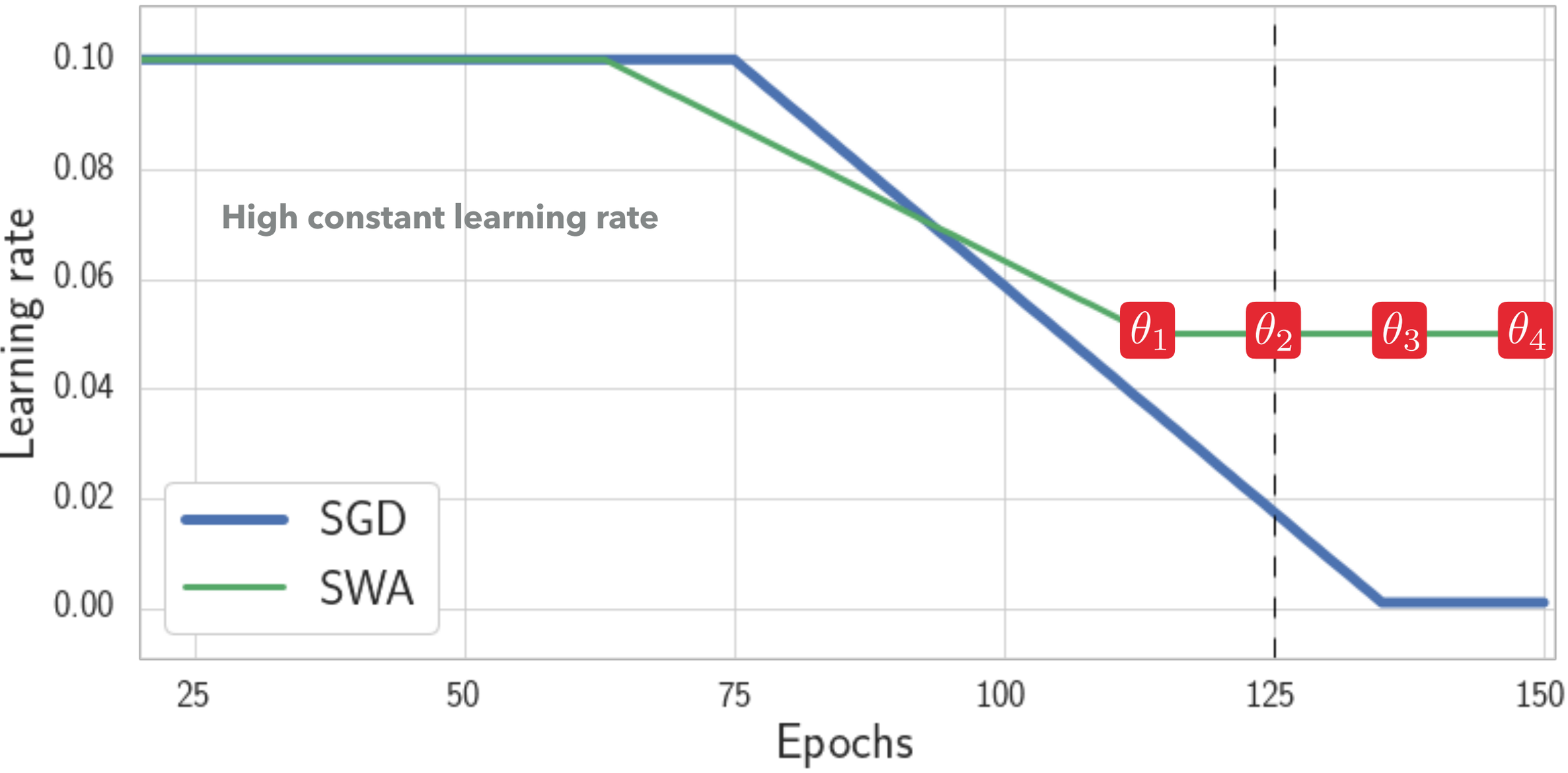


From fitting a Gaussian

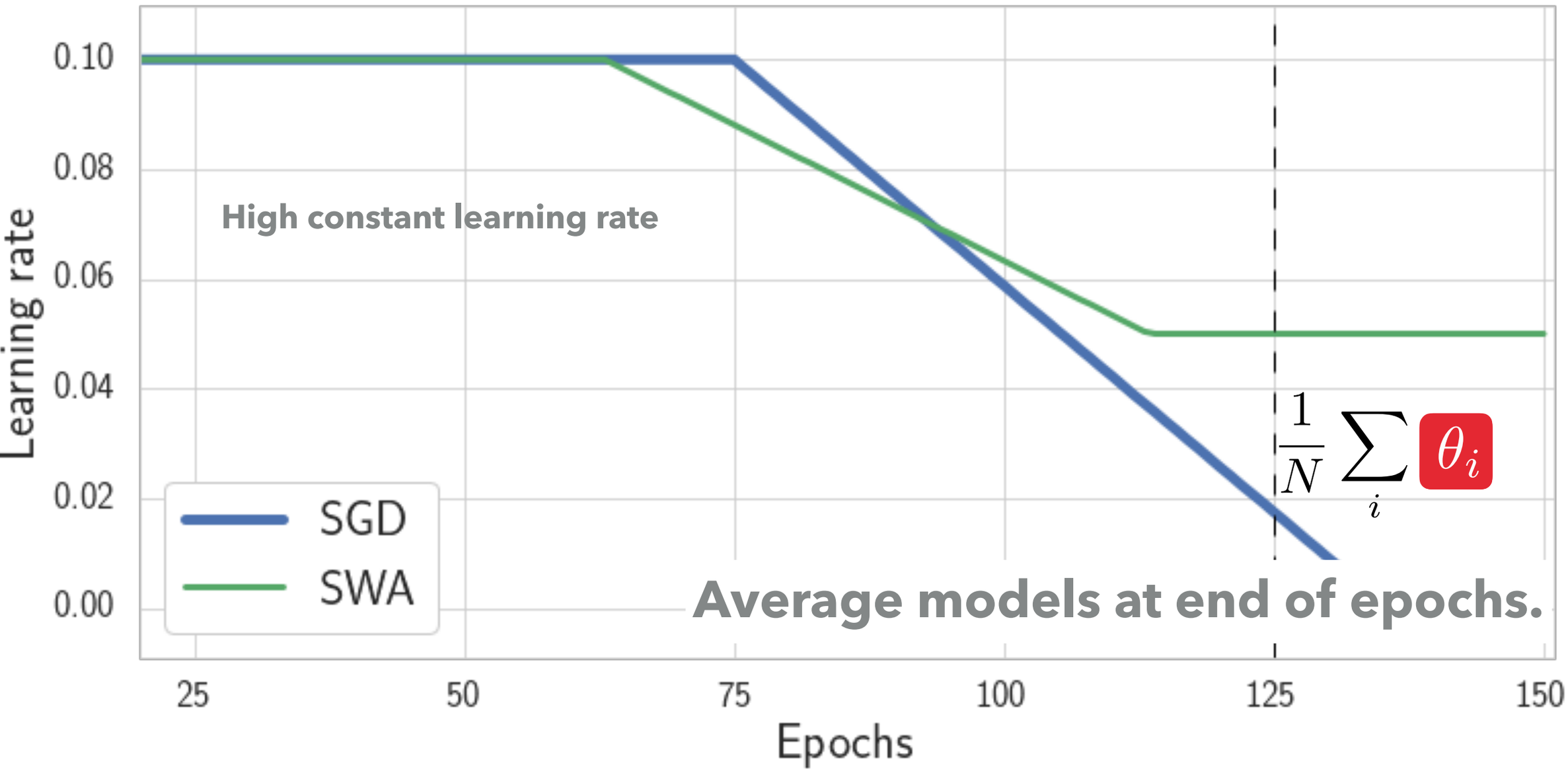
$$\psi(t_1, t_2) = \mathcal{L}(\theta_{\text{SWA}} + t_1 \cdot \frac{v_i}{\|v_i\|} + t_2 \cdot \frac{v_j}{\|v_j\|})$$

METHOD

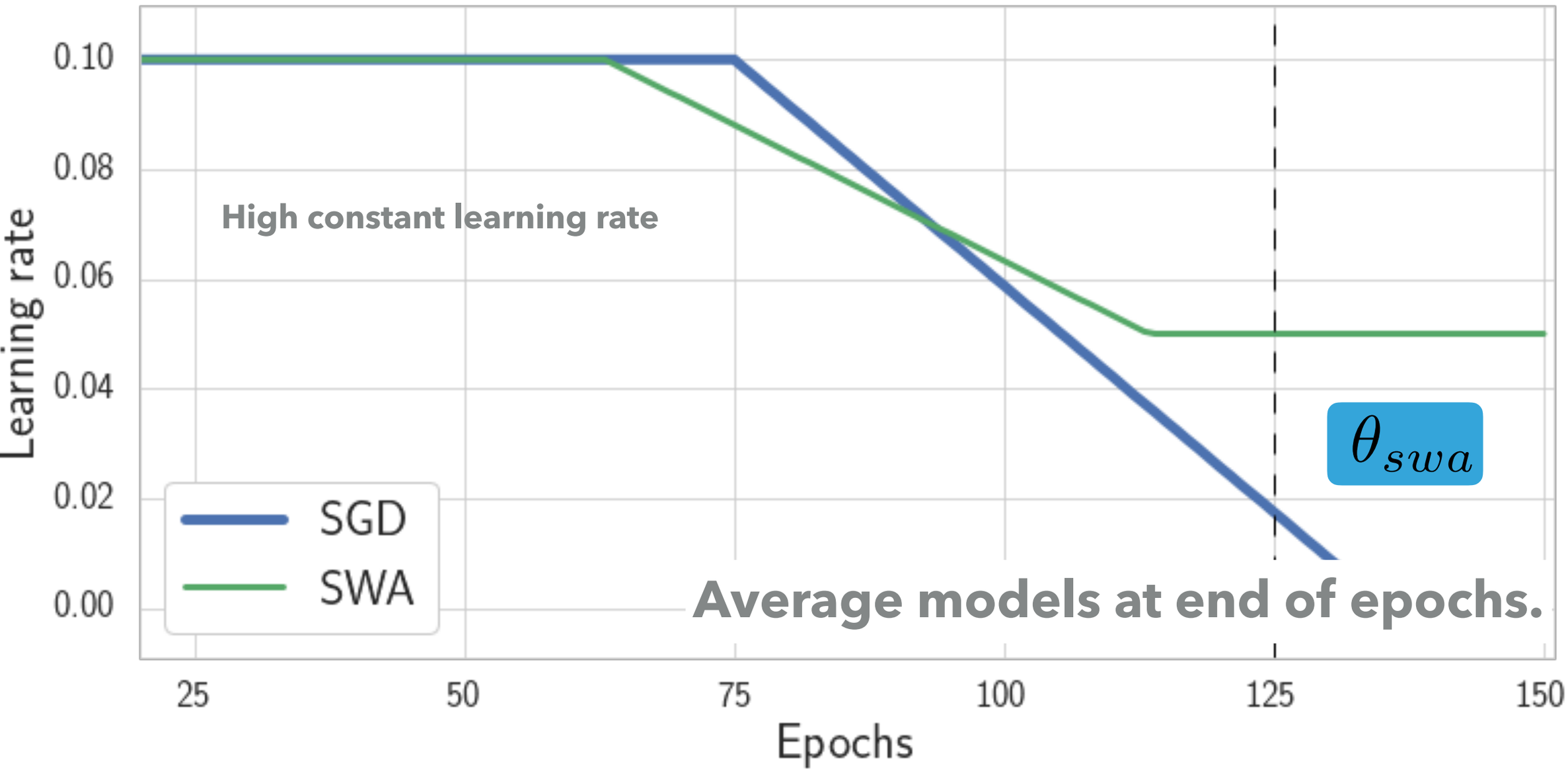
STOCHASTIC WEIGHT AVERAGING (IZMAILOV ET AL 2018)



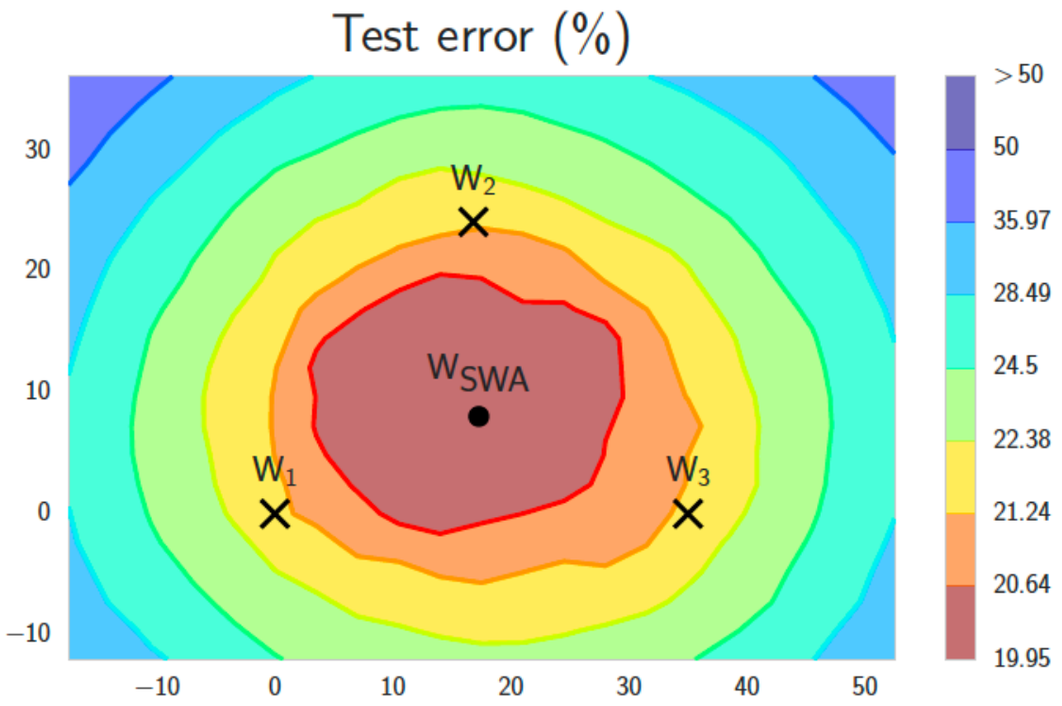
STOCHASTIC WEIGHT AVERAGING (IZMAILOV ET AL 2018)



STOCHASTIC WEIGHT AVERAGING (IZMAILOV ET AL 2018)



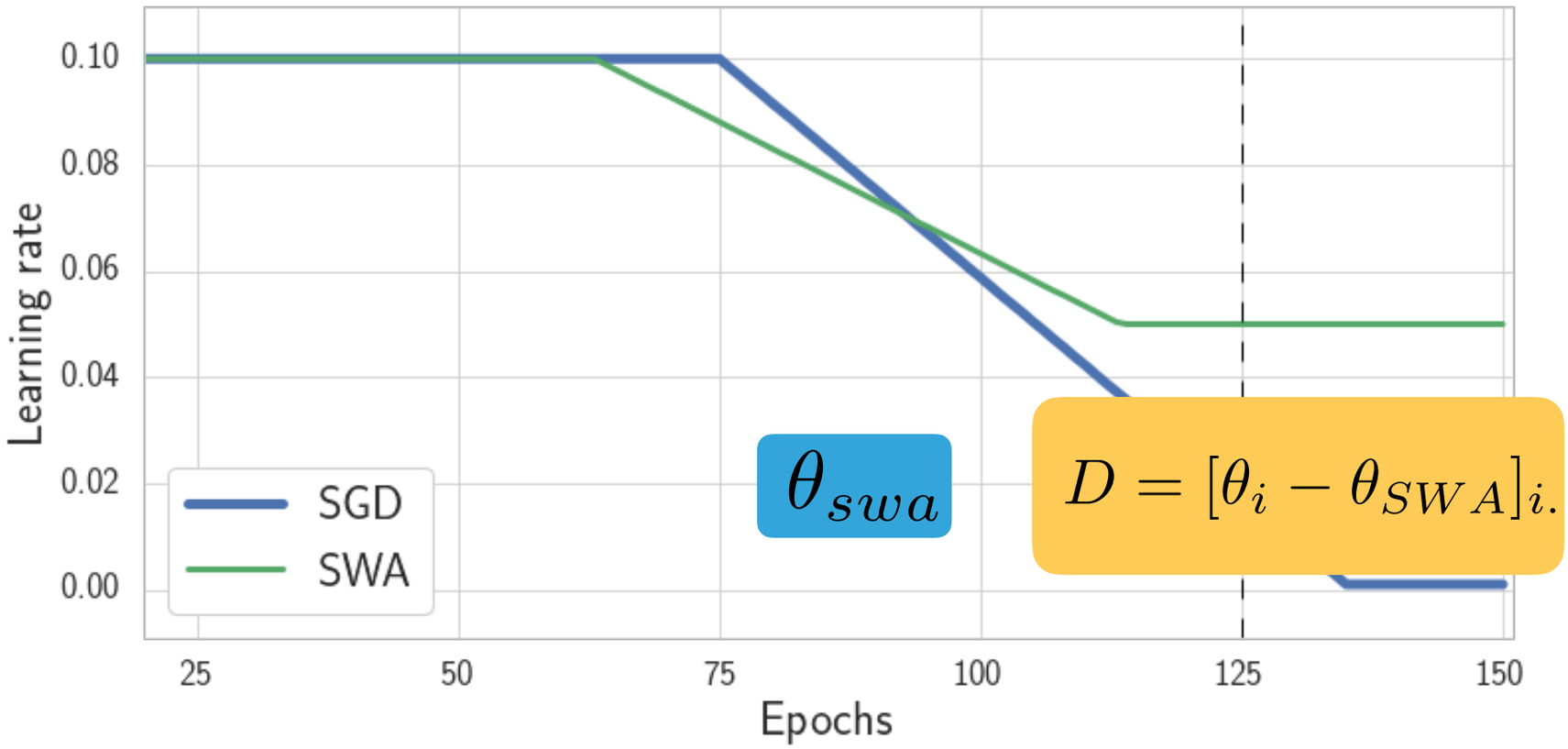
STOCHASTIC WEIGHT AVERAGING (IZMAILOV ET AL 2018)



Average models at ends of epochs.

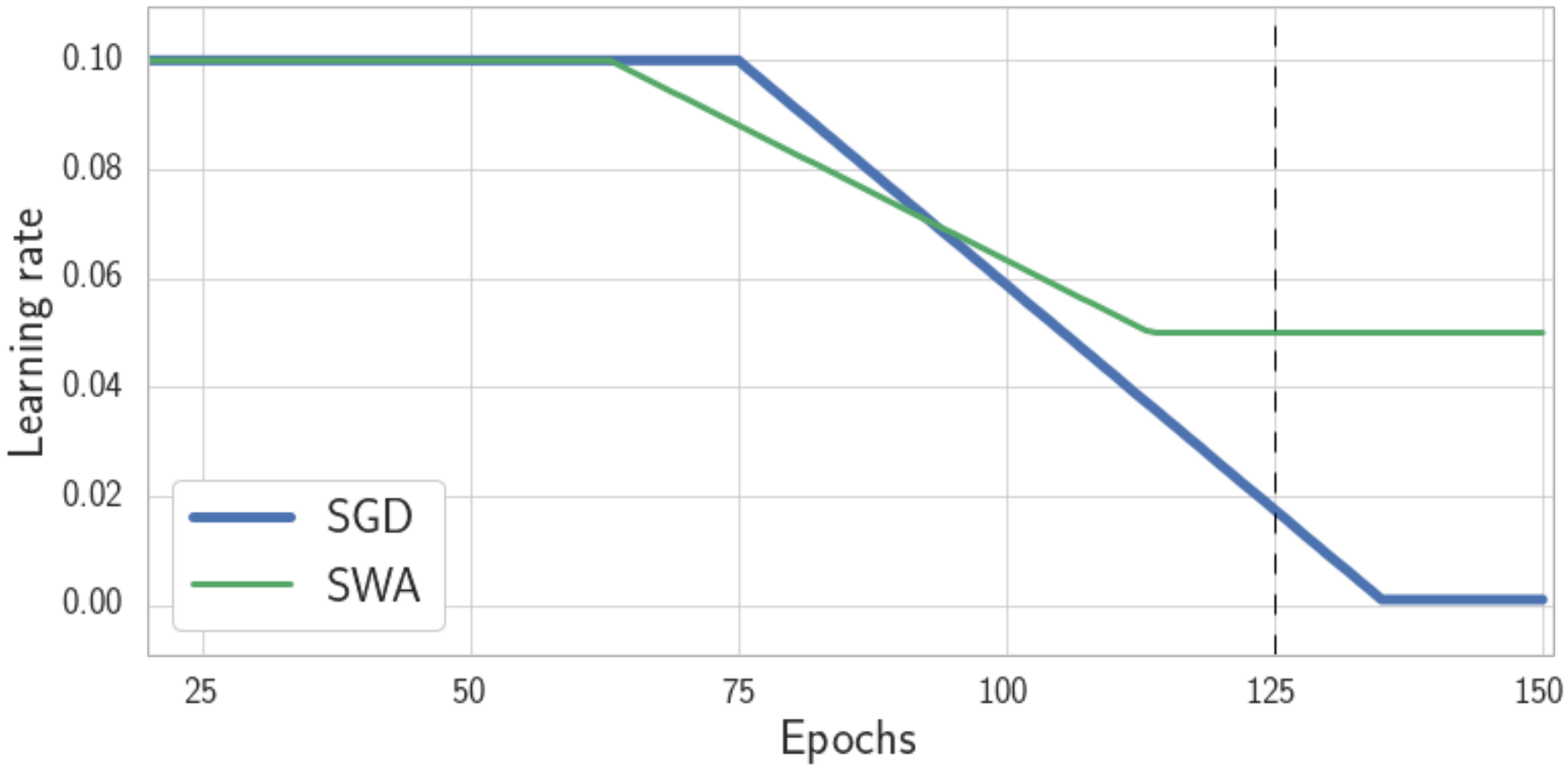
$$\theta_{swa}$$

SWAG (MADDOX ET AL., 2019+)



$$\theta_{SWA} \sim N(\theta_{SWA}, \frac{1}{2(K-1)} DD' + \frac{1}{2} diag(\Sigma_{diag}))$$

SWAG (MADDOX ET AL, 2019+)



θ_{swa}

Diagonal version: $\theta_{\text{swa,diag}} \sim N\left(\theta_{\text{swa}}, \sum_{i=1} \theta_i^2 - \theta_{\text{swa}}^2\right)$

See also Liu et al, 2018, UDL Workshop

SWAG

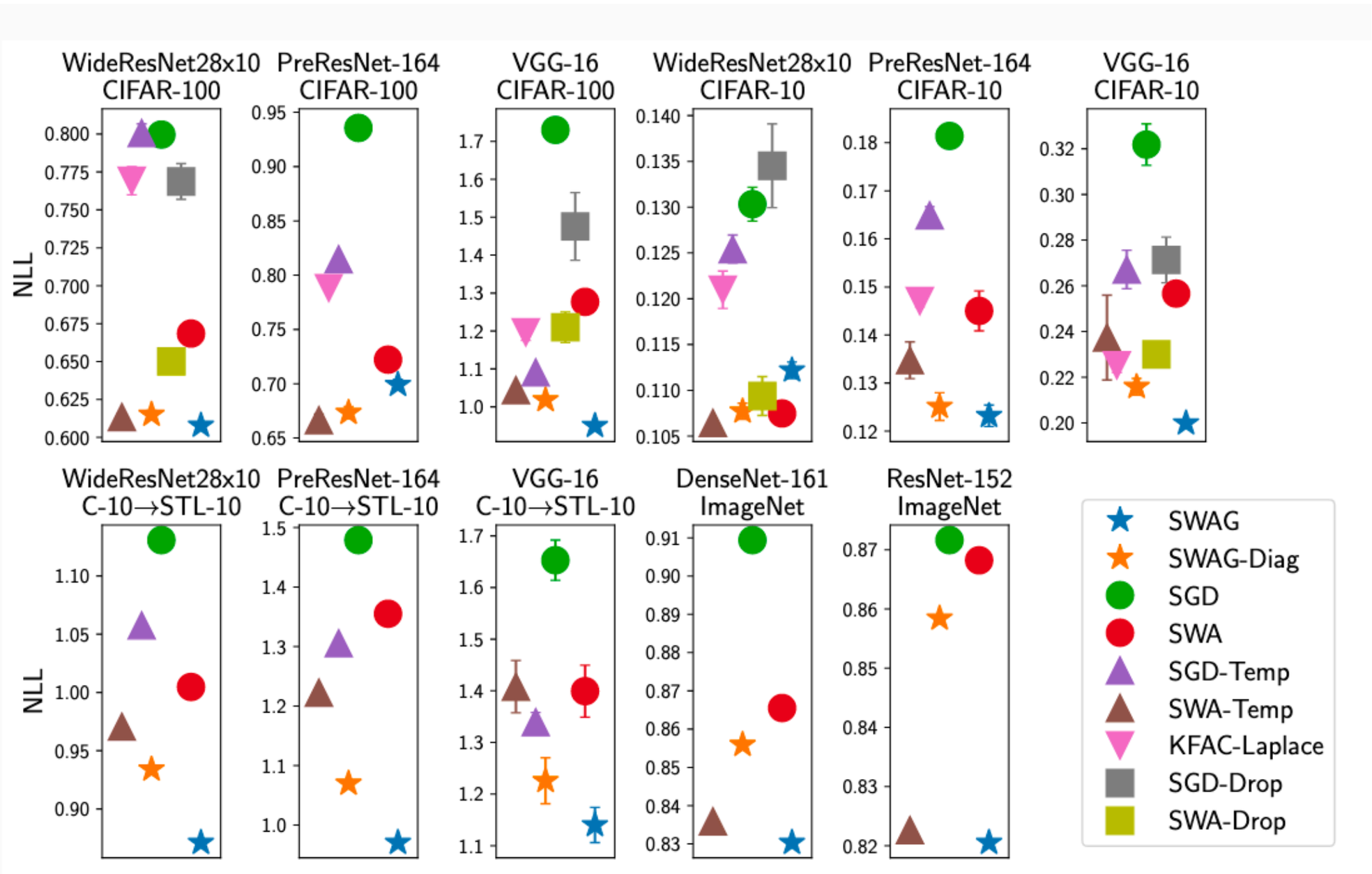
- Low rank + diagonal covariance approximation with last K SGD iterates

$$\theta_{SWA} \sim N(\theta_{SWA}, \frac{1}{2(K-1)} DD' + \frac{1}{2} diag(\Sigma_{diag}))$$

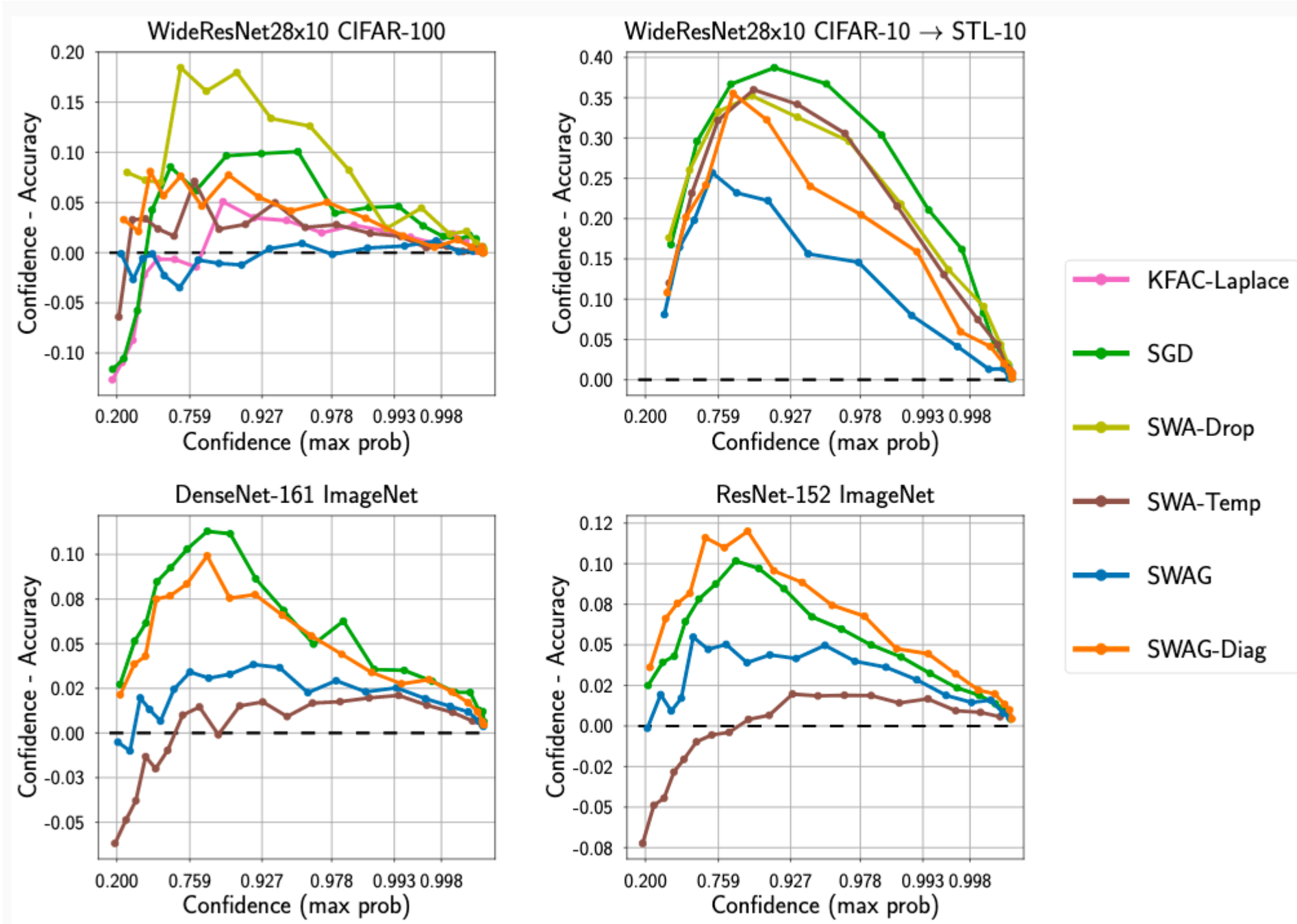
- Sampling + likelihoods (incl. marginal) straightforward

EXPERIMENTS

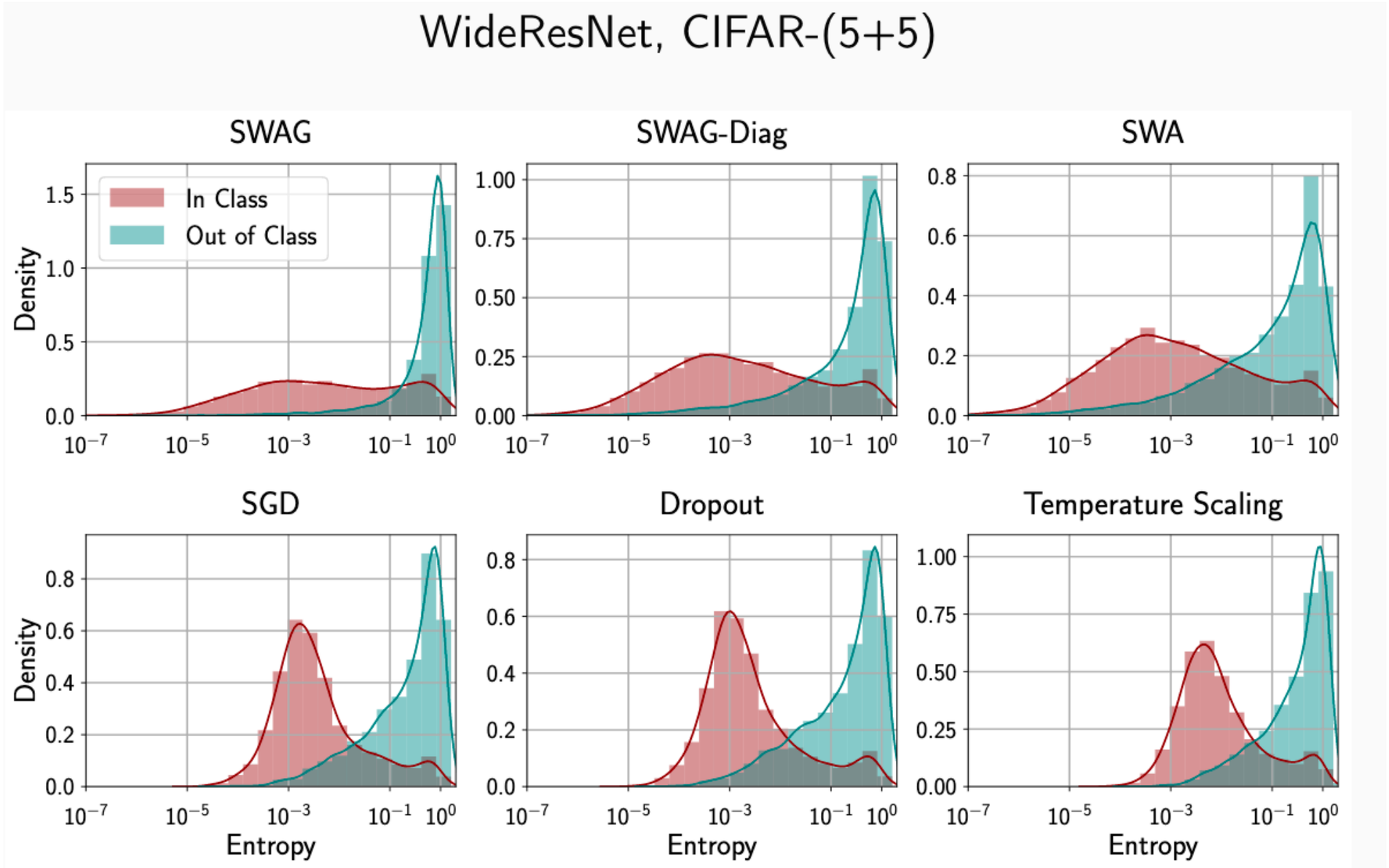
TEST NEGATIVE LOG LIKELIHOOD



CALIBRATION



OUT OF SAMPLE IMAGE DETECTION



ACCURACY (BAYESIAN MODEL AVERAGING)

Dataset	Model	SGD	SWA	SWAG-Diag	SWAG	KFAC-Laplace	SWA-Dropout	SWA-Temp
CIFAR-10	VGG-16	93.17	93.61	**93.66**	93.60	92.65	93.23	93.61
CIFAR-10	PreResNet-164	95.49	96.09	96.03	96.03	95.49	**96.18**	96.09
CIFAR-10	WideResNet28x10	96.41	**96.46**	96.41	96.32	96.17	96.39	96.46
CIFAR-100	VGG-16	73.15	74.30	74.68	**74.77**	72.38	72.50	74.30
CIFAR-100	PreResNet-164	78.50	**80.19**	80.18	79.90	78.51		80.19
CIFAR-100	WideResNet28x10	80.76	82.40	**82.40**	82.23	80.94	82.30	82.40
ImageNet	DenseNet-161	77.79	**78.60**	78.59	78.59			78.60
ImageNet	ResNet-152	78.39	78.92	78.96	**79.08**			78.92
CIFAR10 $\rightarrow$ STL10	VGG-16	**72.42**	71.92	72.09	72.19		71.45	71.92
CIFAR10 $\rightarrow$ STL10	PreResNet-164	75.56	**76.02**	75.95	75.88			76.02
CIFAR10 $\rightarrow$ STL10	WideResNet28x10	76.75	**77.50**	77.26	77.09		76.91	77.50

ACCURACY (BAYESIAN MODEL AVERAGING): NLP TASKS

Table 5: Validation and Test perplexities for NT-ASGD, SWA and SWAG on Penn Treebank and WikiText-2 datasets.

Method	PTB val	PTB test	WikiText-2 val	WikiText-2 test
NT-ASGD	61.2	58.8	68.7	65.6
SWA	59.1	56.7	68.1	65.0
SWAG	**58.6**	**56.26**	**67.2**	**64.1**

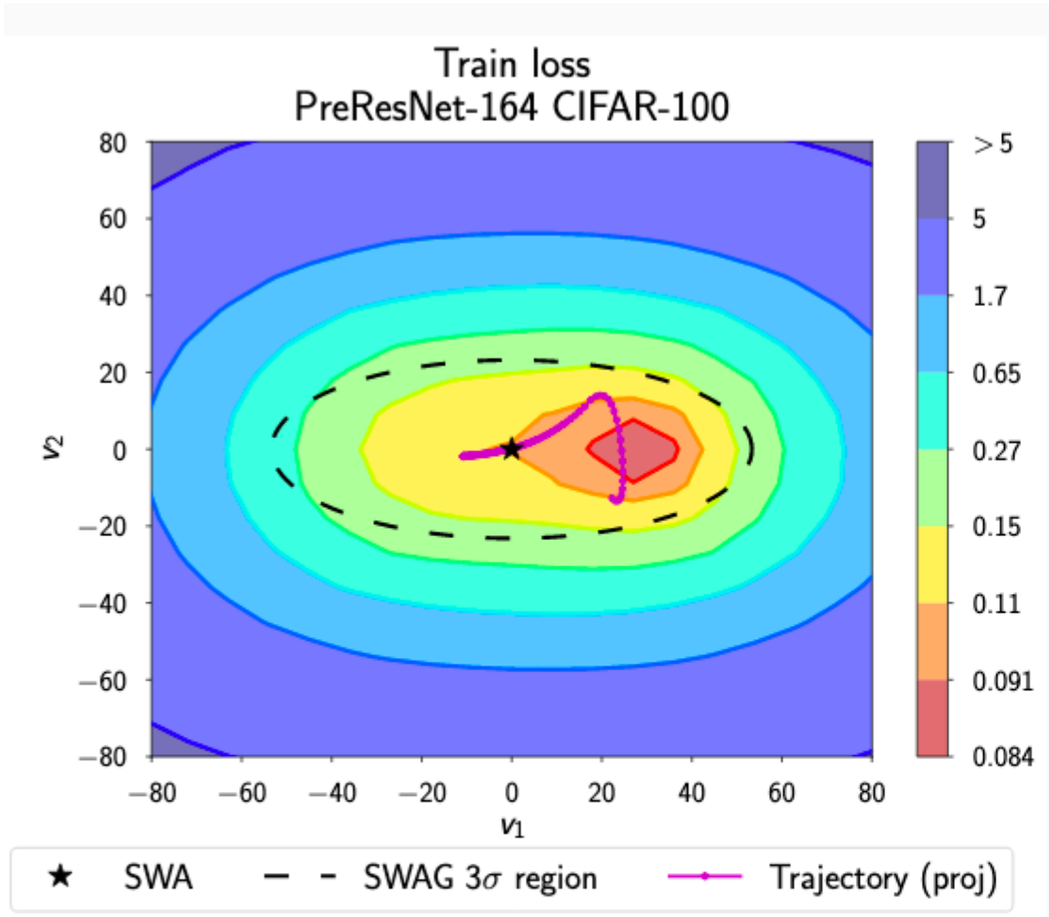
FUTURE WORK

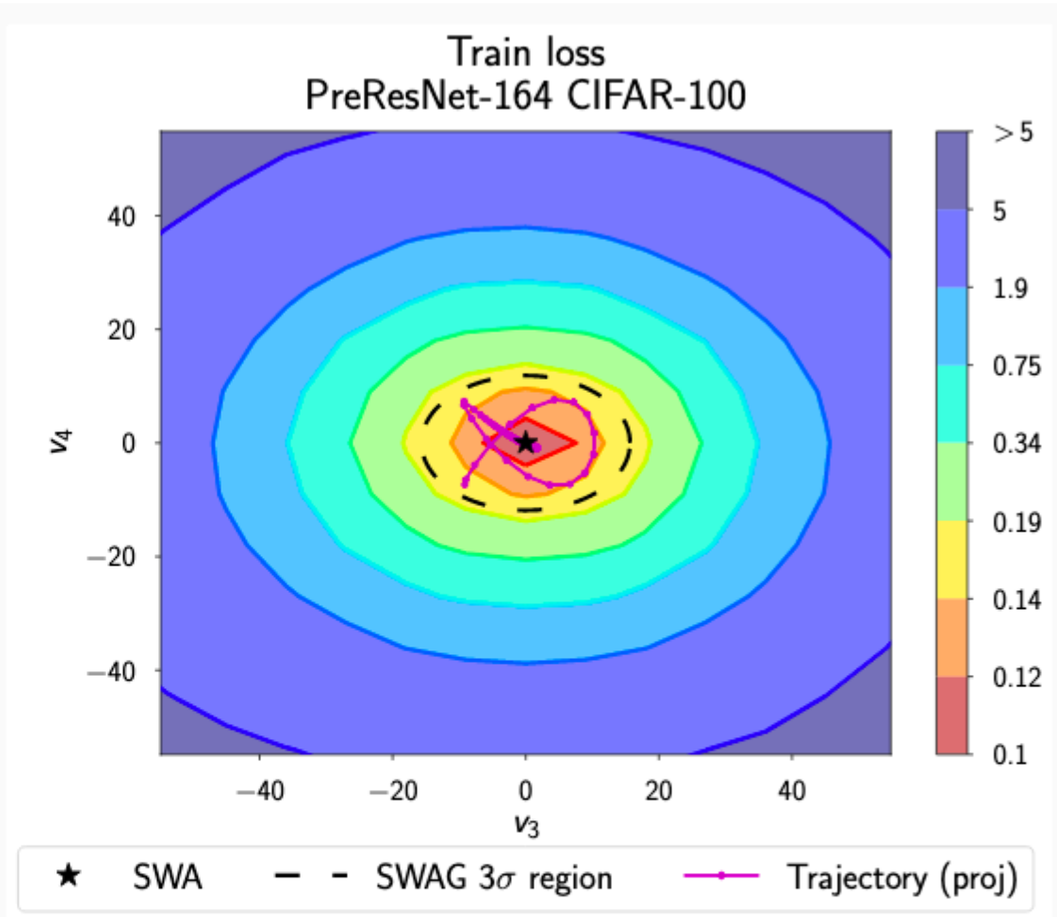
- Downstream tasks:
  - Bayesian optimization
  - Image segmentation
- Theoretical work still to be done:
  - Quantify the shape of the posterior + its approximation error
  - Connection to frequentist methods (Chen et al, JASA, 2019)

SUBSPACE INFERENCE FOR BAYESIAN DEEP LEARNING

EMPIRICAL MOTIVATION

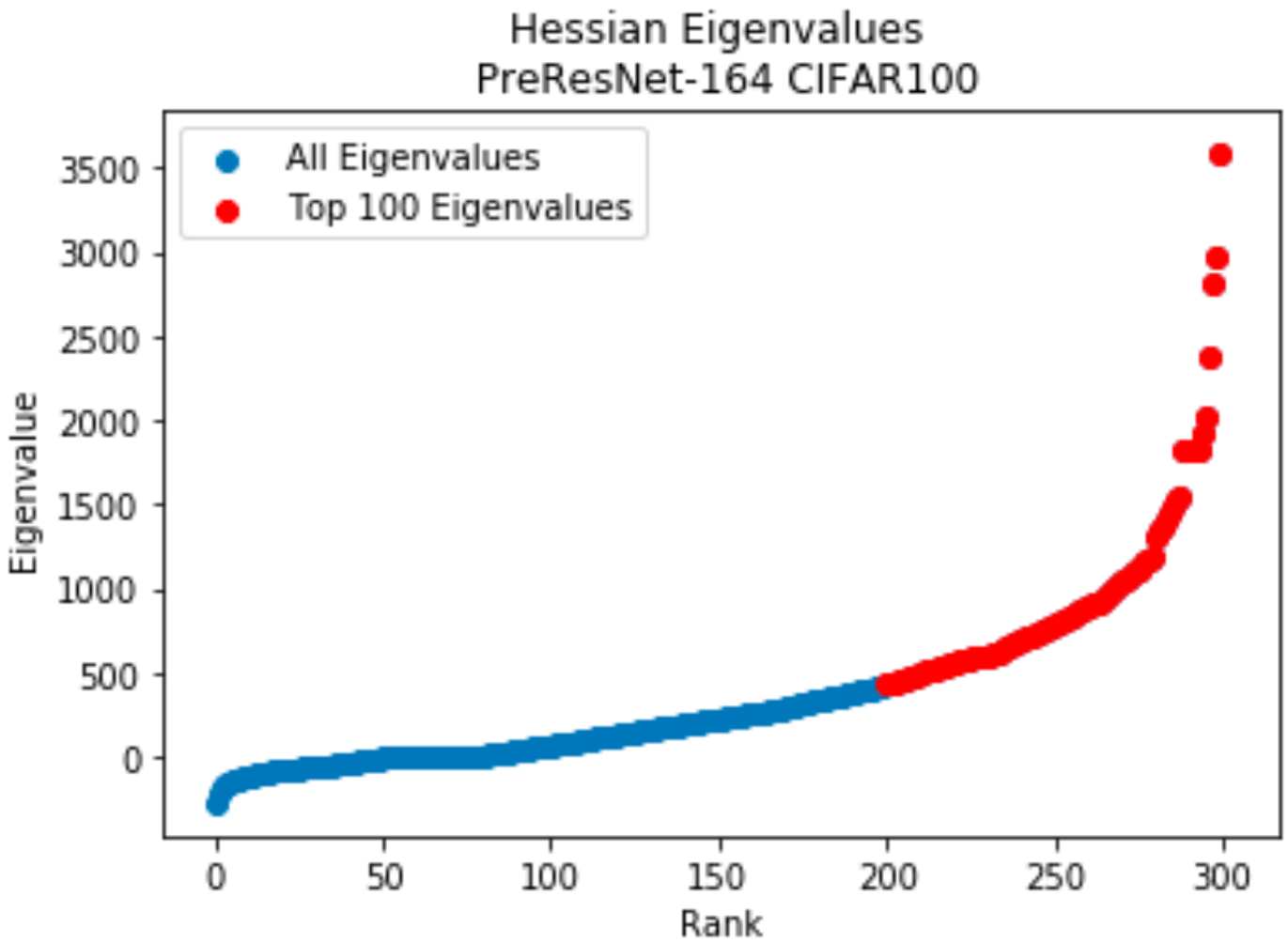
2D surface along eigenvectors of approx. posterior





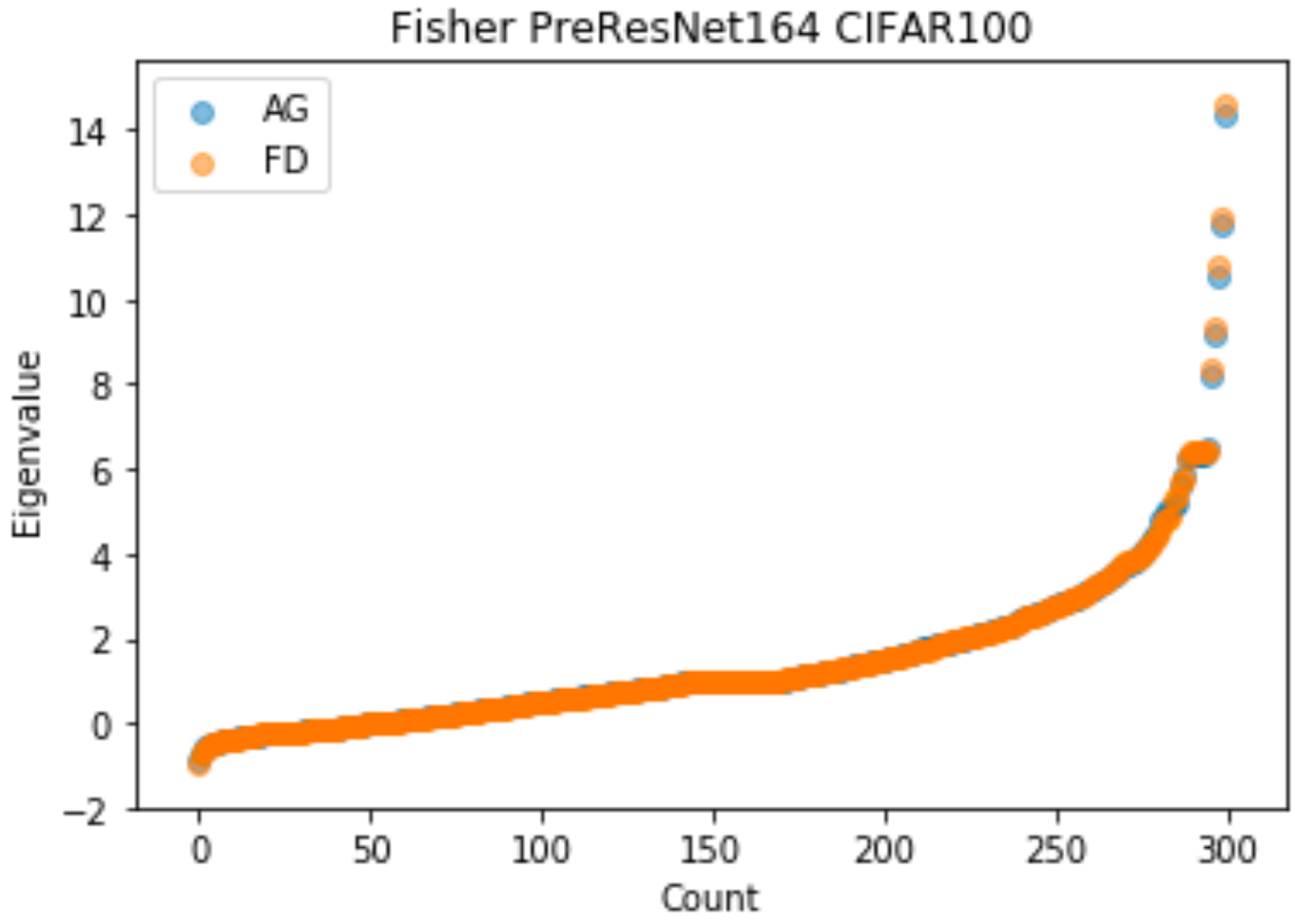
$$\psi(t_1, t_2) = \mathcal{L}(\theta_{\mathsf{SWA}} + t_1 \cdot \frac{v_i}{\|v_i\|} + t_2 \cdot \frac{v_j}{\|v_j\|})$$

EMPIRICAL MOTIVATION: EIGENVALUES OF THE HESSIAN



- Max: 3580, Min: -**272.** Using Lanczos (GPyTorch)

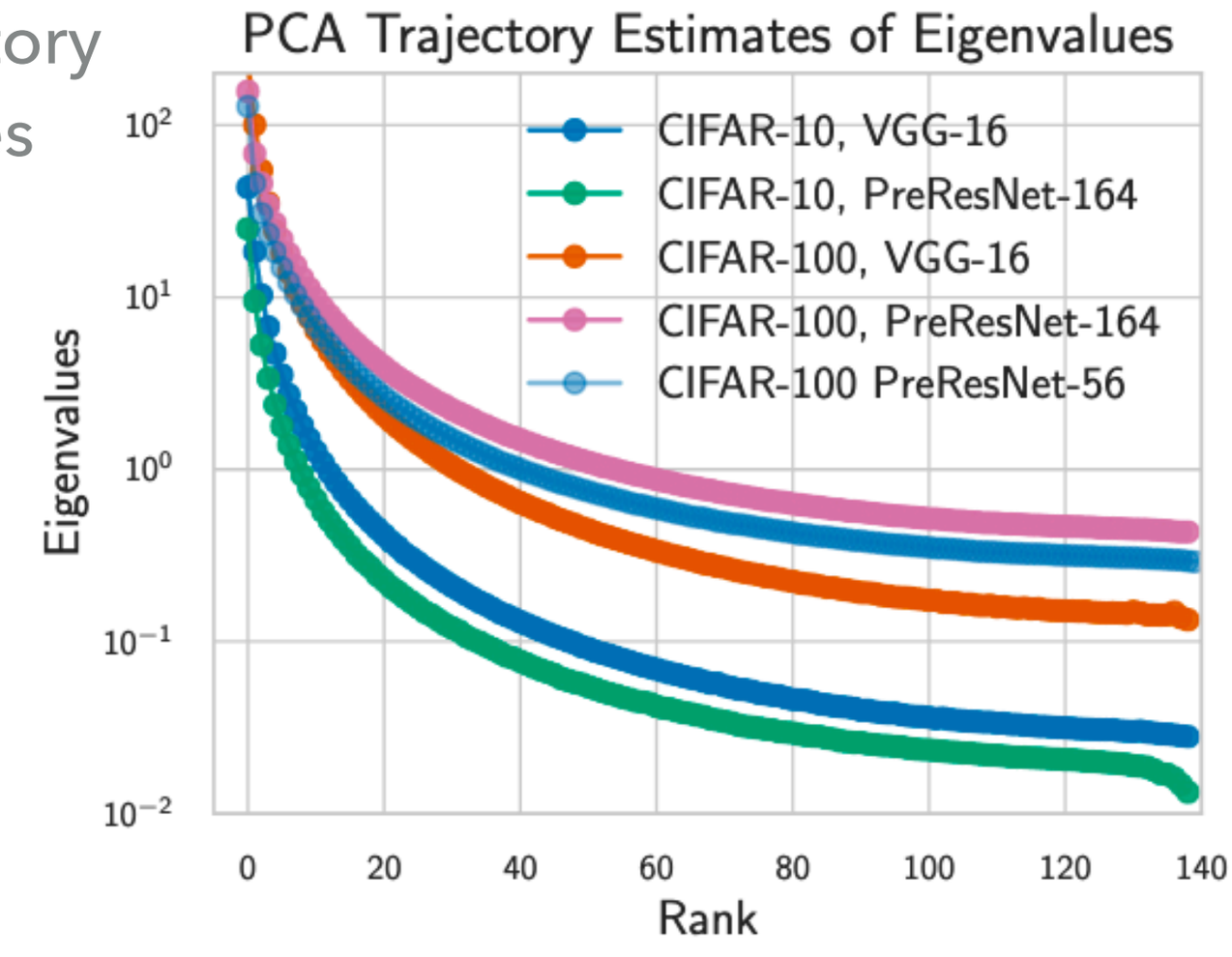
EMPIRICAL MOTIVATION: EIGENVALUES OF THE FISHER



- Max:14, Min: -1 😬
- Using Lanczos (GPyTorch)

EMPIRICAL MOTIVATION: SGD TRAJECTORY

- The eigenvalues of the SGD trajectory decays very rapidly
  - Can summarize the trajectory with very low rank matrices



SUBSPACE INFERENCE

A modular approach:

- Design subspace
- Approximate posterior over parameters in the subspace
- Sample from approximate posterior for Bayesian model averaging

SUBSPACE INFERENCE

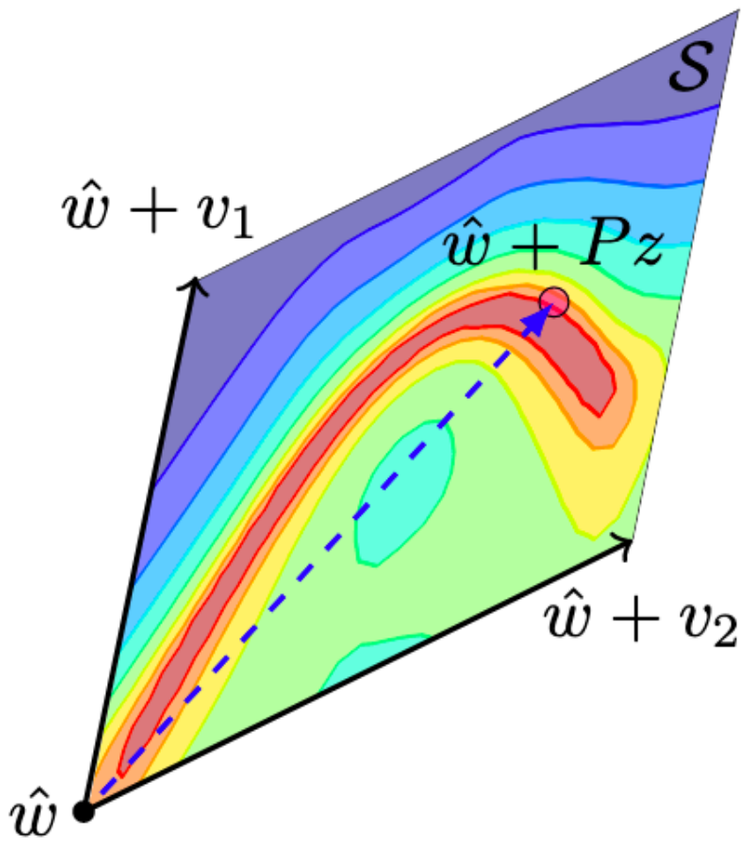
A modular approach:

- Design subspace
- Approximate posterior over parameters in the subspace
- Sample from approximate posterior for Bayesian model averaging

We can approximate posterior of 36 million dimensional WideResNet in 5D subspace and get state-of-the-art results!

SUBSPACE

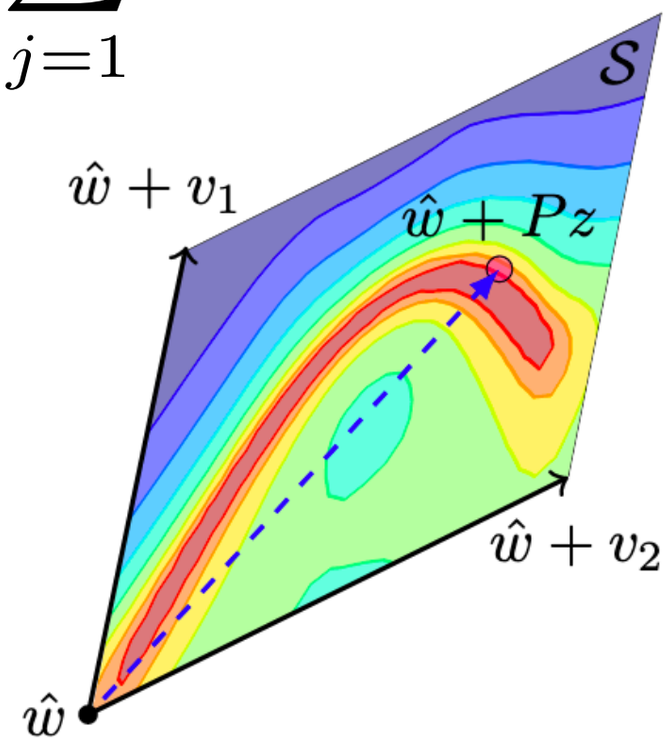
- Choose shift $\hat{w}$ and basis vectors $\{d_1, \ldots, d_K\}$
- Define subspace $S = \{w | w = \hat{w} + \underbrace{d_1 z_1 + \cdots + d_K z_k}_{Pz}\}$
- Likelihood $p(\mathcal{D}|z) = p_{\mathcal{M}}(\mathcal{D}|w = \hat{w} + Pz)$



INFERENCE

- Approximate inference over parameters t
  - MCMC, Variational Inference, Normalizing Flows, …
- Bayesian model averaging at test time:

$$p(\mathcal{D}^*|\mathcal{D}) = \frac{1}{J}\sum_{j=1}^{J} p_{\mathcal{M}}(\mathcal{D}|\tilde{w} = \hat{w} + P\tilde{z}_j), \quad \tilde{z}_j \sim q(\tilde{z}|\mathcal{D})$$



TEMPERING POSTERIOR

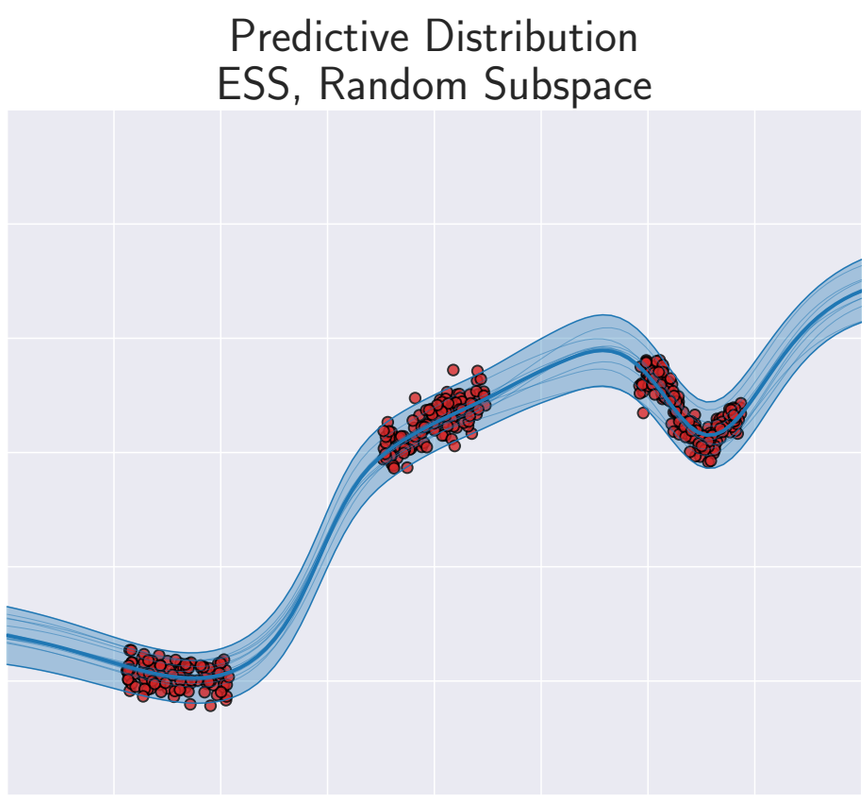
- In the subspace model *# parameters << # data points*
  - *~5-10 parameters, ~50K data points*
- Posterior over t is extremely concentrated
- To address this issue, we utilize the tempered posterior:

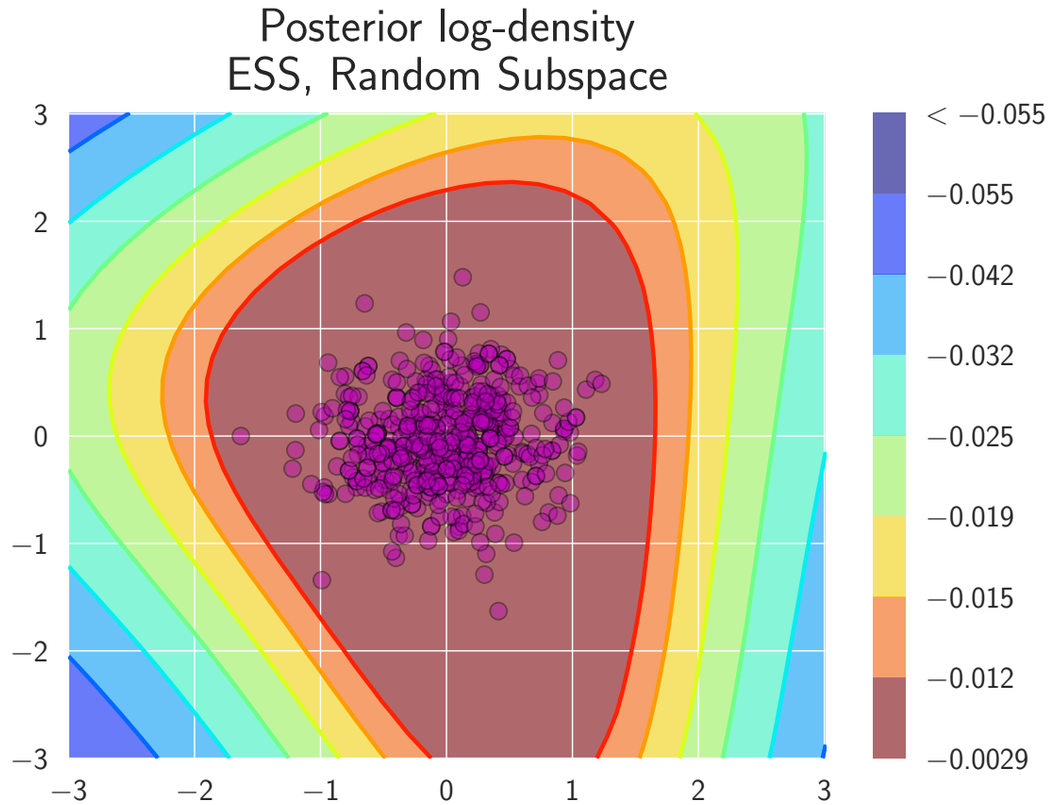
$$p_T(z \mid D) \propto \underbrace{p(D \mid z)^{1/T}}_{\text{likelihood}} \underbrace{p(t)}_{\text{prior}}$$

- *T* can be learned by cross-validation
- Heuristic: $T = \dfrac{\text{\# data points}}{\text{\# parameters}}$

RANDOM SUBSPACE

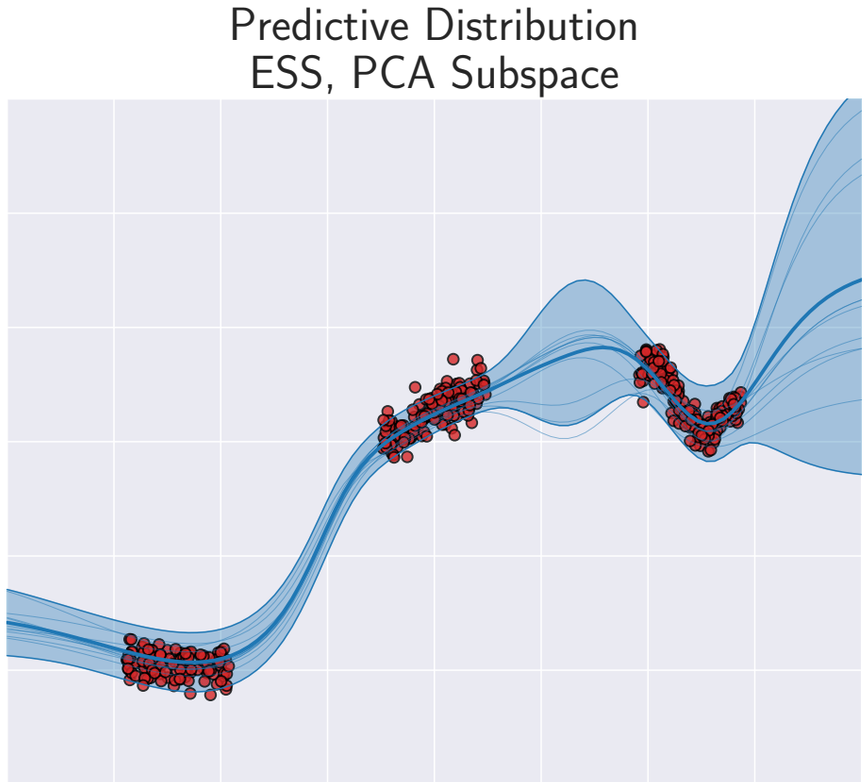
- Directions $d_1, \ldots, d_K \sim N(0, I_p)$
- Use pre-trained solution as shift $\hat{w}$
- Subspace $S = \{w \,|\, w = \hat{w} + Pz\}$

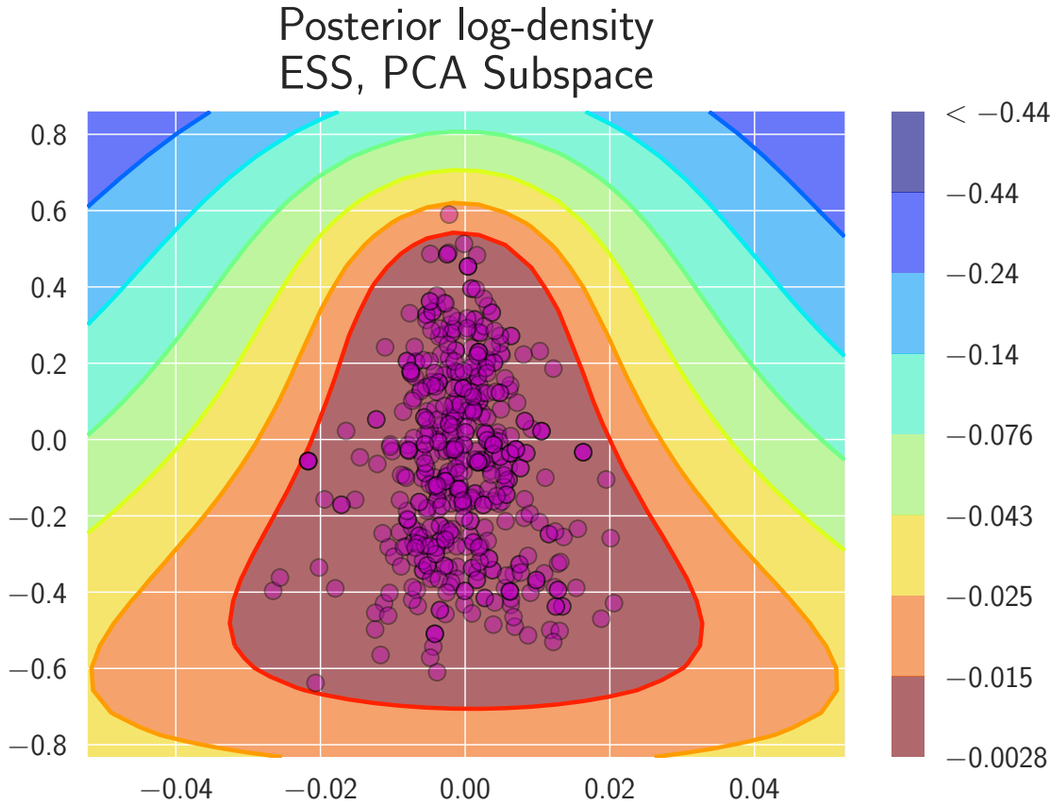




PCA OF THE SGD TRAJECTORY

- Run SGD with high constant learning rate from a pre-trained solution
- Collect snapshots of weights w_i
- Use SWA solution as shift $\hat{w} = \frac{1}{T}\sum_i w_i$
- $\{d_1, \ldots, d_K\}$ – first K PCA components of vectors $\hat{w} - w_i$



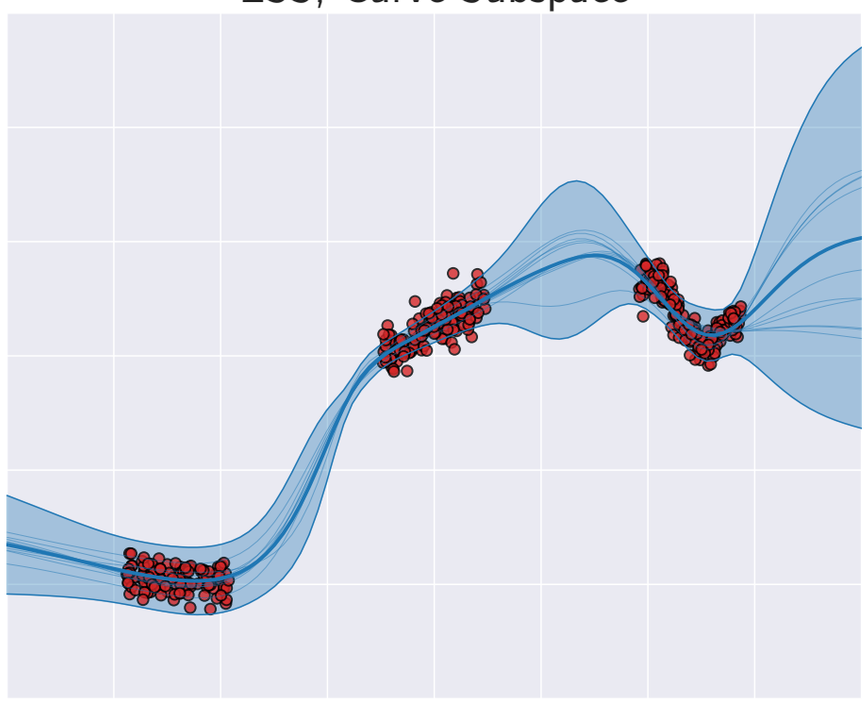


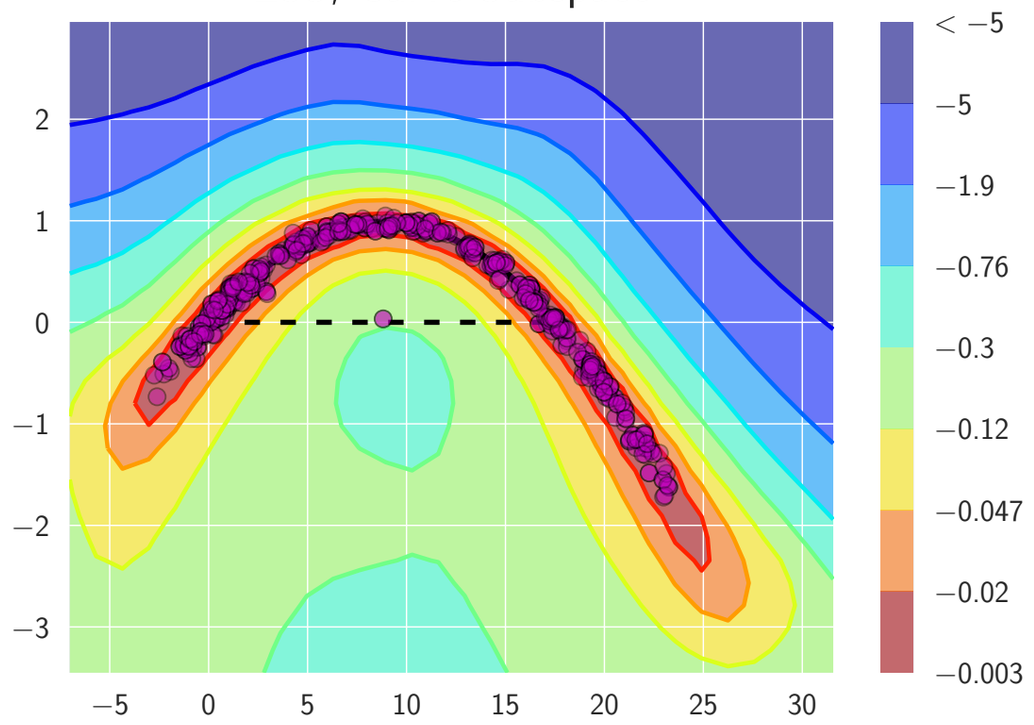
CURVE SUBSPACE

- Garipov et al. 2018 proposed a method to find 2D subspaces containing a path of low loss between weights of two independently trained neural networks

$$\arg\min_{\theta} \mathbb{E}_{t\sim U(0,1)}(\mathcal{L}(\phi_\theta(t)))$$

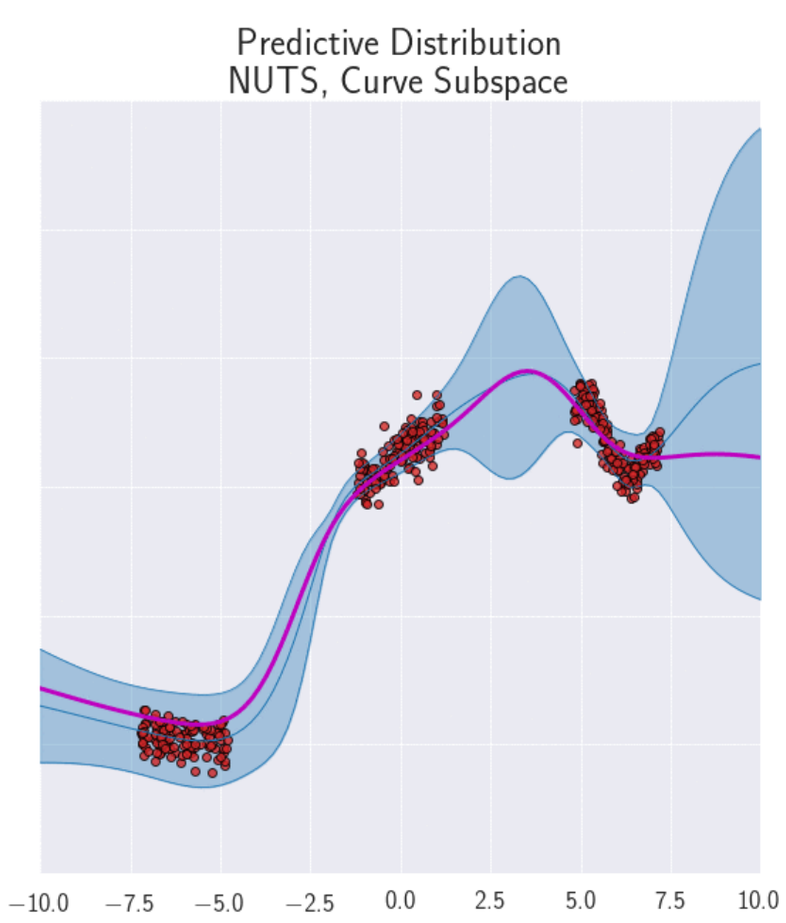
$$\phi_\theta(t) = (1-t)^2\hat{w}_1 + 2t(1-t)\theta + t^2\hat{w}_2^2$$

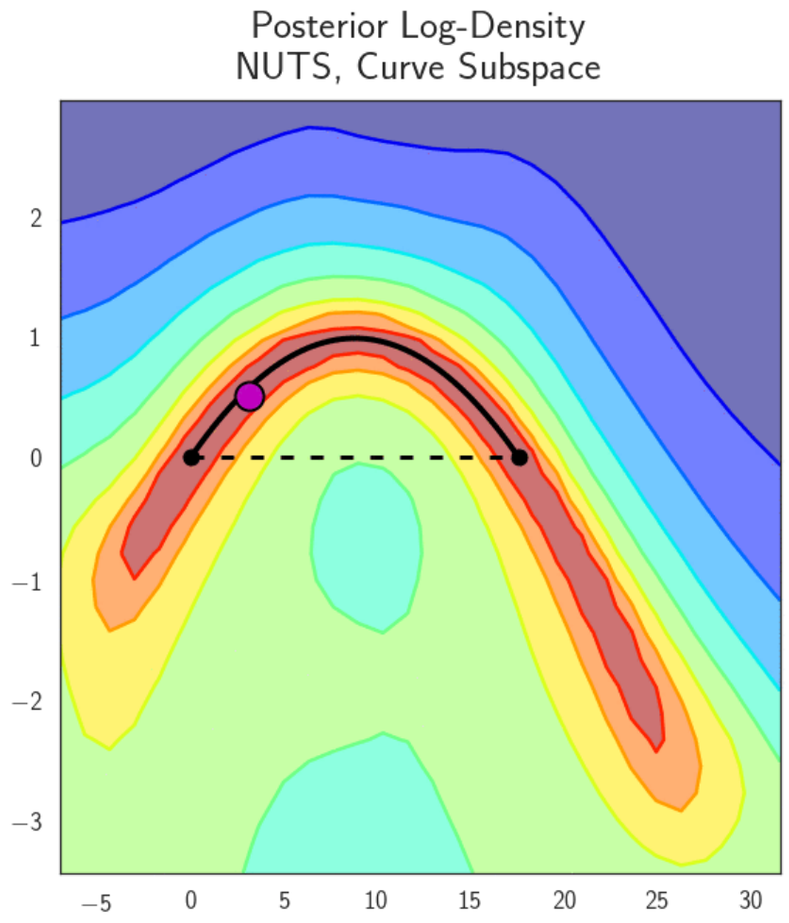




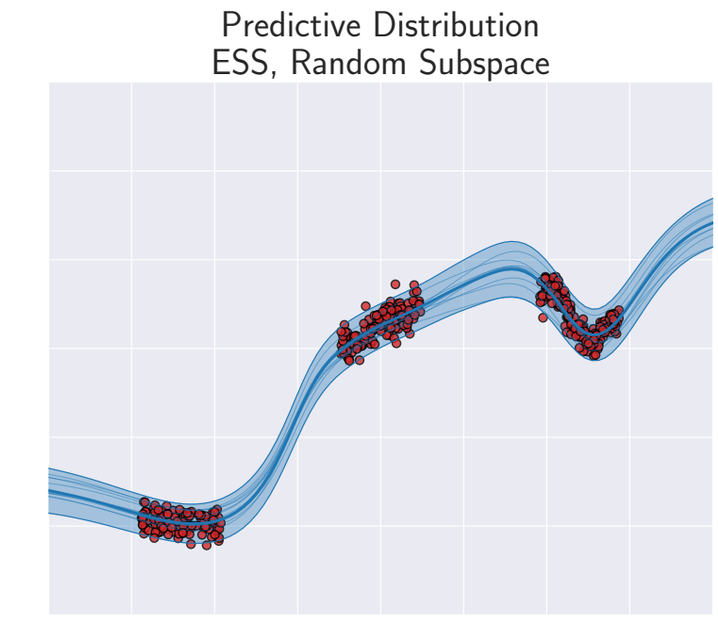
CURVE SUBSPACE

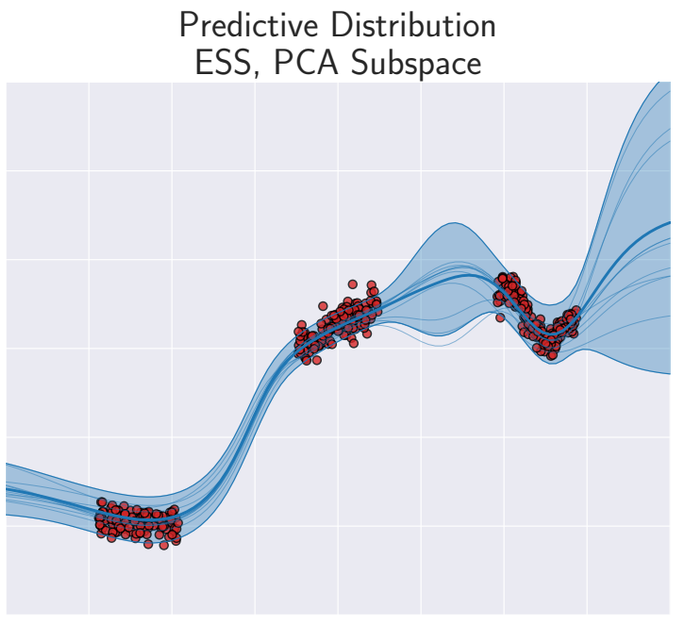
- Garipov et al. 2018 proposed a method to find 2D subspaces containing a path of low loss between weights of two independently trained neural networks

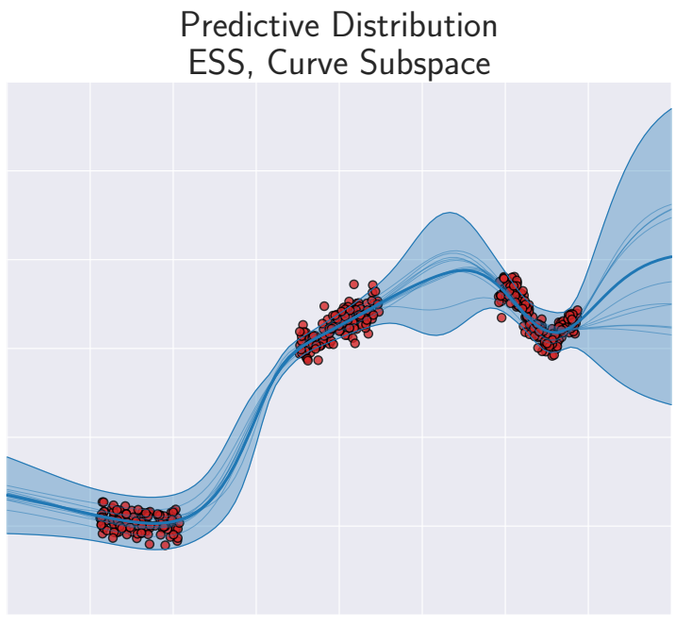


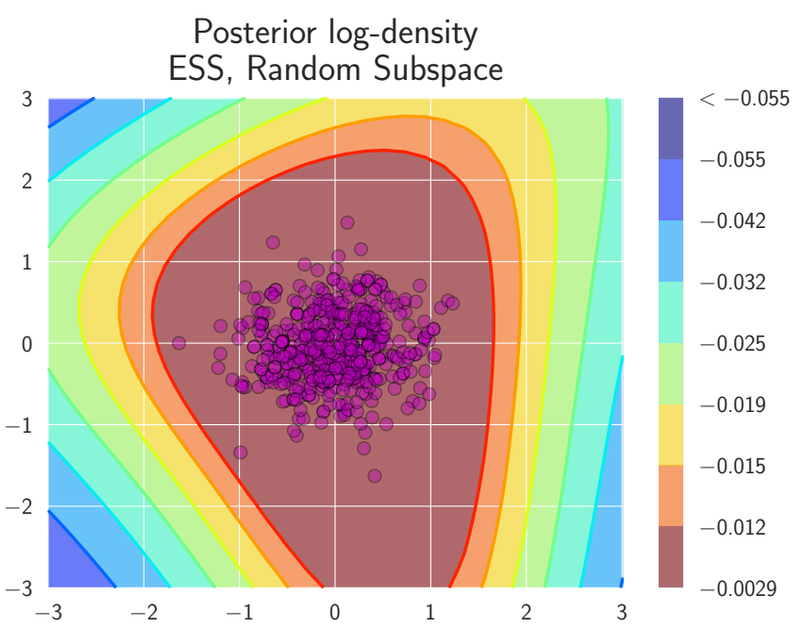


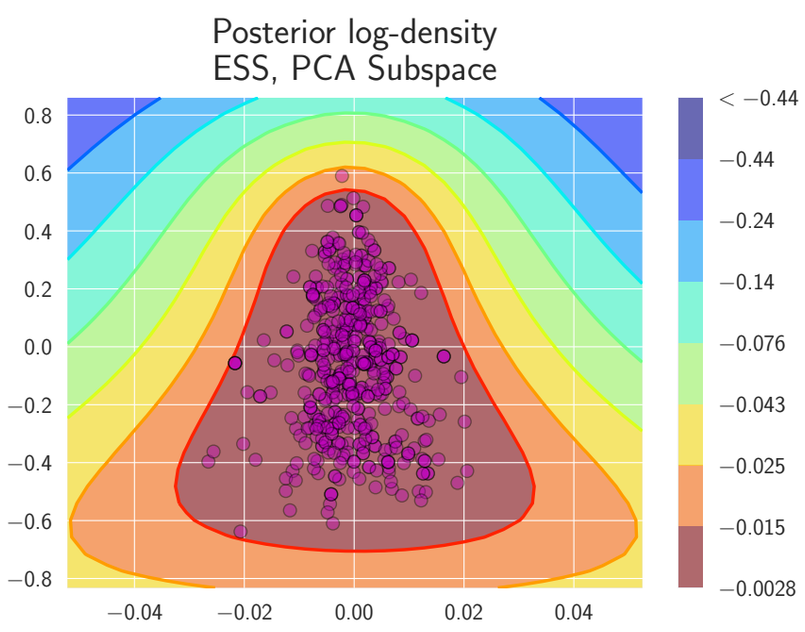
SUBSPACE COMPARISON

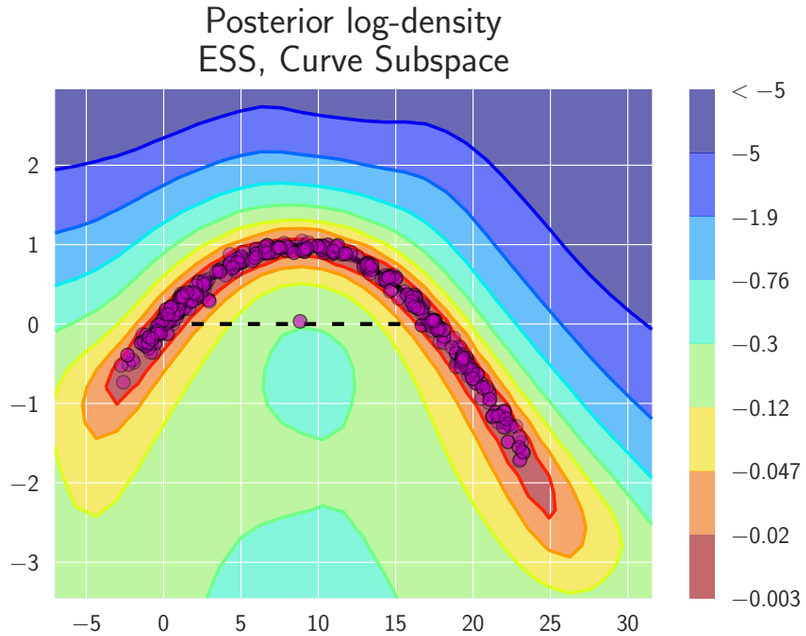




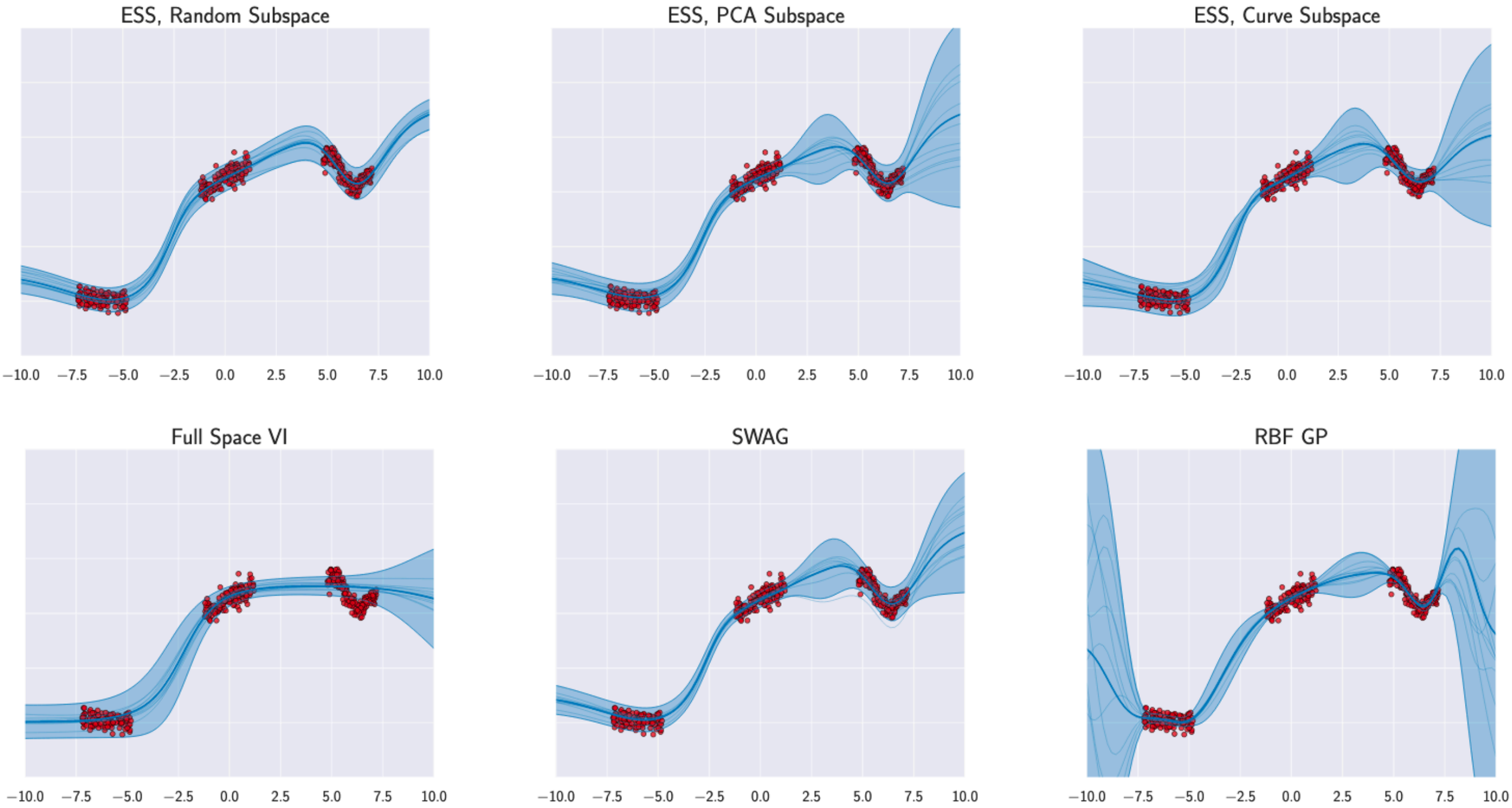








QUALITATIVE COMPARISONS: REGRESSION



QUANTITATIVE COMPARISONS: REGRESSION

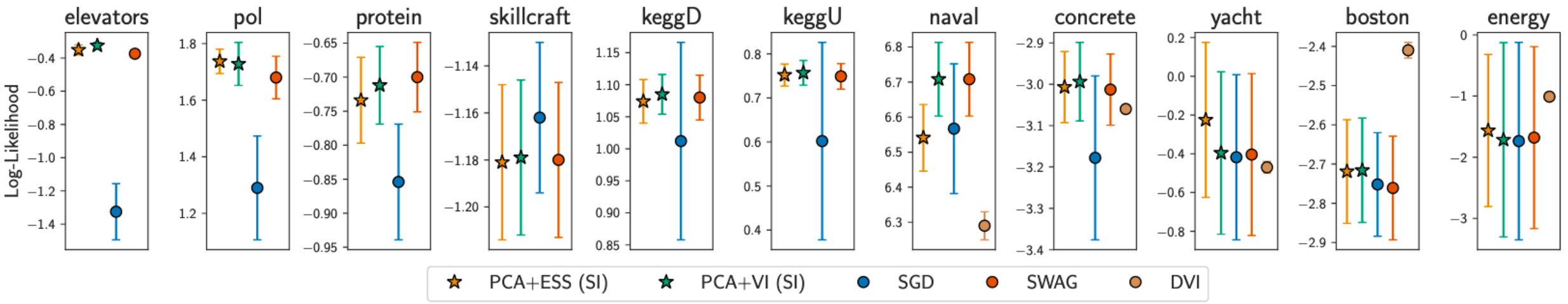
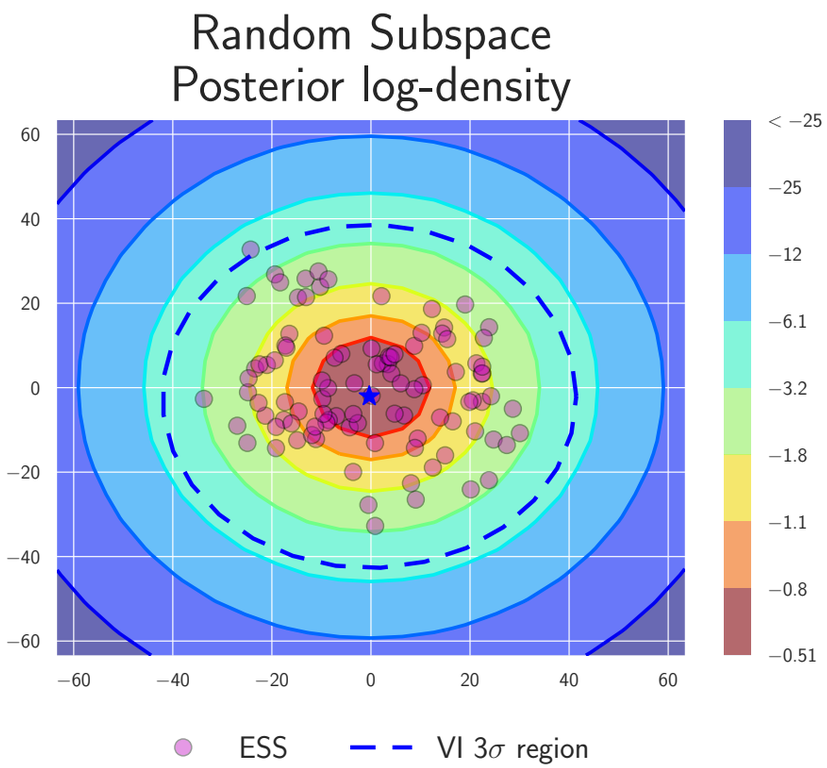
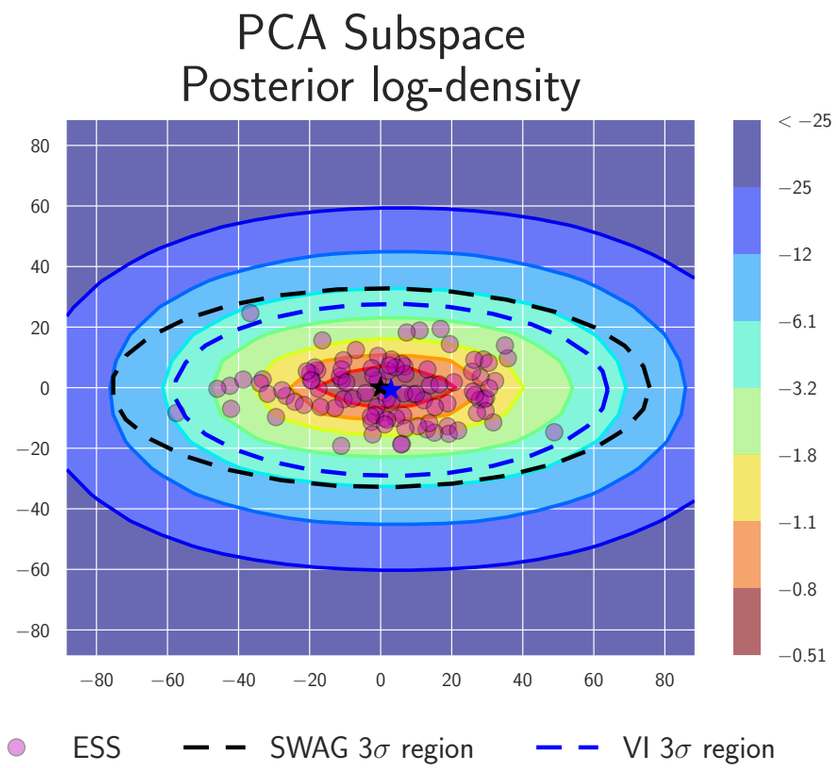
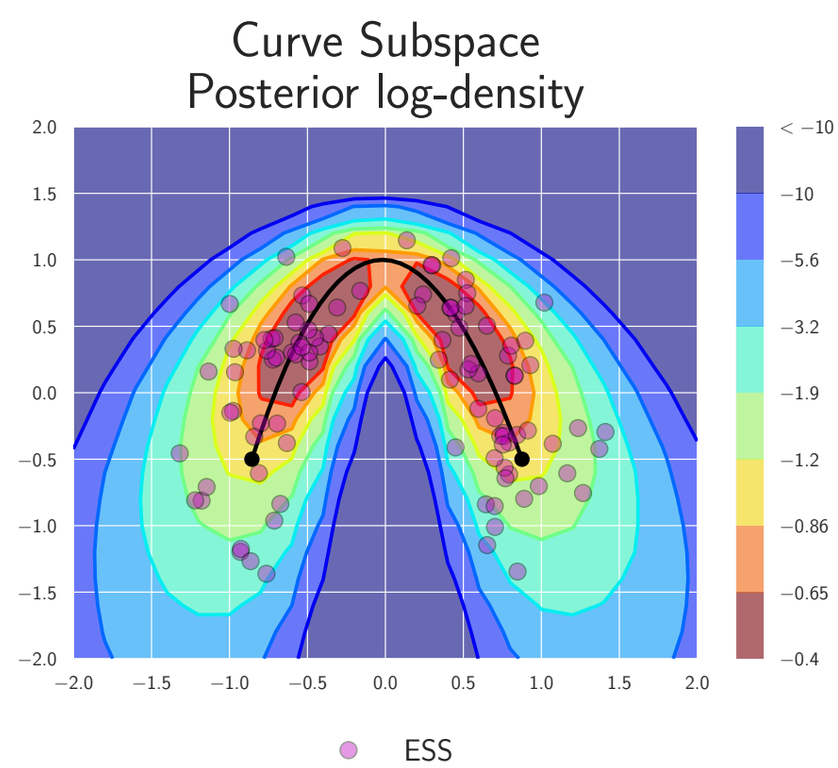


Figure 4: Test log-likelihoods for subspace inference and baselines on UCI regression datasets. Subspace inference (SI) with PCA achieves as good or better test log-likelihoods compared to SGD and SWAG, and is competitive with DVI (which does not immediately scale to larger problems or networks). We report mean over 20 random splits of the data $\pm$ one standard deviation. For consistency with the literature, we report normalized log-likelihoods on large datasets (elevators, pol, protein, skillcraft, keggD, keggU; see Section 5.2.1) and unnormalized log-likelihoods on small datasets (naval, concrete, yacht, boston, energy; see Section 5.2.2).

SUBSPACE COMPARISON ON PRERESNET-164, CIFAR-100

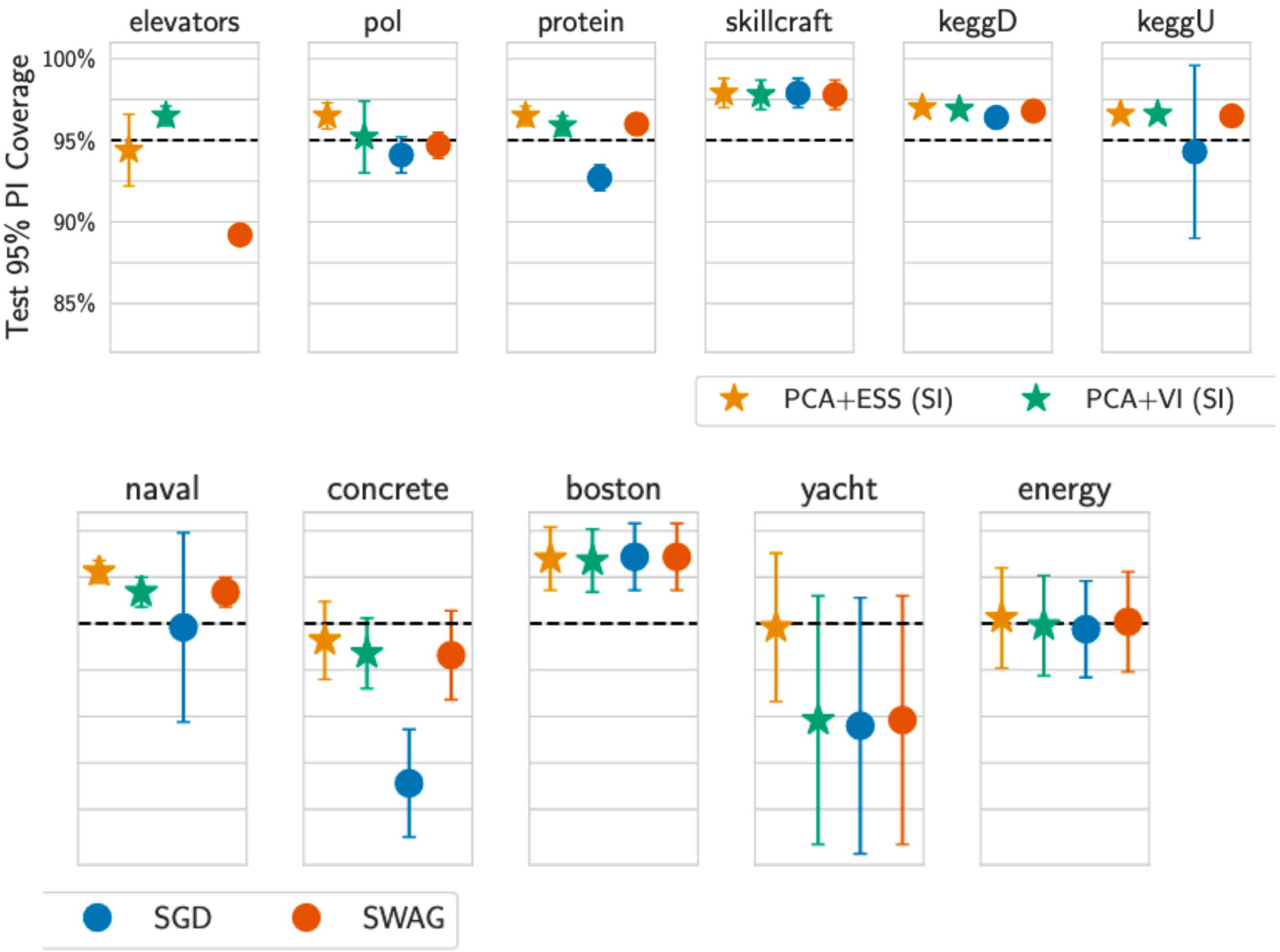






	SGD	Random	PCA	Curve
NLL	0.946 ± 0.001	0.686 ± 0.005	0.665 ± 0.004	0.646
Accuracy (%)	78.50 ± 0.32	80.17 ±0.03	80.54 ± 0.13	81.28

CALIBRATION STUDY: REGRESSION



TAKEAWAYS

- We can apply standard approximate inference methods in subspaces of parameter space
- More diverse subspaces => better performance:
  Curve Subspace > **PCA Subspace** > Random Subspace
- Subspace Inference in the PCA subspace is competitive with SWAG (Maddox et al., 2019), MC-Dropout (Gal & Ghahramani, 2016) and Temperature Scaling (Guo et al., 2017) on image classification and UCI regression

FUTURE WORK

- Relate to kernels and kernel approximation?
- Theoretical work still to be done:
  - Quantify the amount of tempering necessary
  - Choose dimensionality of subspace
  - Interpret PCA subspace as empirical bayes?

QUESTIONS?

Thanks

STRUCTURE

- Based off of:
  - *"A Simple Baseline for Bayesian Uncertainty in Deep Learning,"* **Maddox**, Garipov, Izmailov, Vetrov, Wilson. https://arxiv.org/abs/1902.02476. 2019
    - Code: https://github.com/wjmaddox/swa_gaussian
  - *"Subspace Inference for Bayesian Deep Learning,"* Izmailov, **Maddox**, Kirichenko, Garipov, Vetrov, Wilson. https://arxiv.org/abs/1907.07504. UAI, 2019
    - Code: https://github.com/wjmaddox/drbayes

REFERENCES

- P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. In UAI , 2018.
- Blundell, Charles, et al. "Weight Uncertainty in Neural Network." *International Conference on Machine Learning*. 2015.
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, August). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1321-1330). JMLR. org.
- Ritter, Hippolyt, Aleksandar Botev, and David Barber. "A scalable laplace approximation for neural networks." ICLR. 2018.
- Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." *international conference on machine learning*. 2016.
- Chen, Tianqi, Emily Fox, and Carlos Guestrin. "Stochastic gradient hamiltonian monte carlo." *International Conference on Machine Learning*. 2014.
- B. T. Polyak and A. B. Juditsky. Acceleration on Stochastic Approximation by Averaging. SIAM Journal on Control and Optimization , 30(4):838–855, July 1992.
- D. Ruppert. Efficient Estimators from a Slowly Convergent Robbins-Munro Process. Technical Report 781, Cornell University, School of Operations Research and Industrial Engineering, 1988.

REFERENCES

- X. Chen, J. D. Lee, X. T. Tong, and Y. Zhang. Statistical Inference for Model Parameters in Stochastic Gradient Descent. arXiv: 1610.08637, JASA, 2019.
- Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., & Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems* (pp. 8789-8798).
- Mandt, S., Hoffman, M. D., & Blei, D. M. (2017). Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, *18*(1), 4873-4907.
- J. Liu, S. Tripathi, U. Kurup, M. Shah, Make (Nearly) Every Neural Network Better: Generating Neural Network Ensembles by Weight Parameter Resampling. In UAI Workshop on Uncertainty in Deep Learning, 2018.
- Murray, I., Prescott Adams, R., & MacKay, D. J. (2010). Elliptical slice sampling. AISTATS.

EXPECTED CALIBRATION ERROR

$$E_{\hat{P}}\Big[\Big|P(\hat{Y}=Y|\hat{P}=p)-p\Big|\Big]$$

$$ECE=\sum_{m=1}^{M}\frac{|B_m|}{n}\Big|acc(B_m)-conf(B_m)\Big|$$

From Naeini et al 2015, also Guo et al ICML 2017 “On Calibration of Modern Neural Networks”

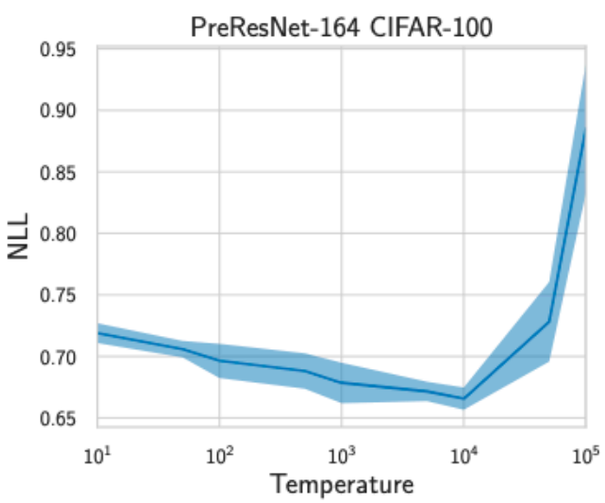
ASYMPTOTIC MOTIVATION OF SWAG

- Polyak-Ruppert Averaging (Ruppert 1988; Polyak & Juditsky, 1992)
  - Average the iterates of SGD
  - Asymptotic distribution (around stationary point):

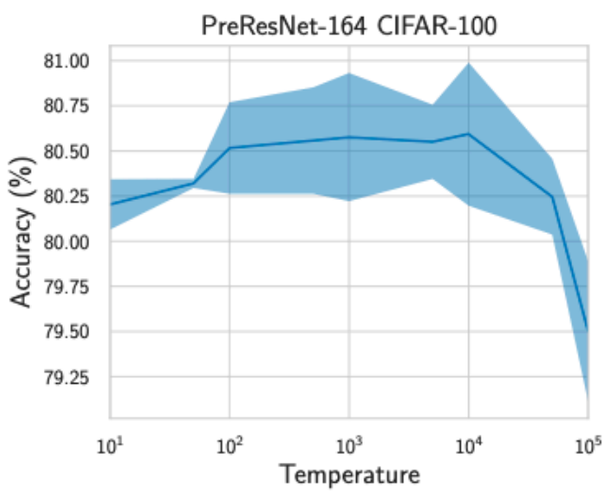
$$\frac{1}{T}\sum_{i=1}^{T}\theta_i \approx N(\theta, H(\theta)^{-1}SH(\theta)^{-1})$$

- Laplace approximation uses Gaussian around MAP with covariance H(\theta)

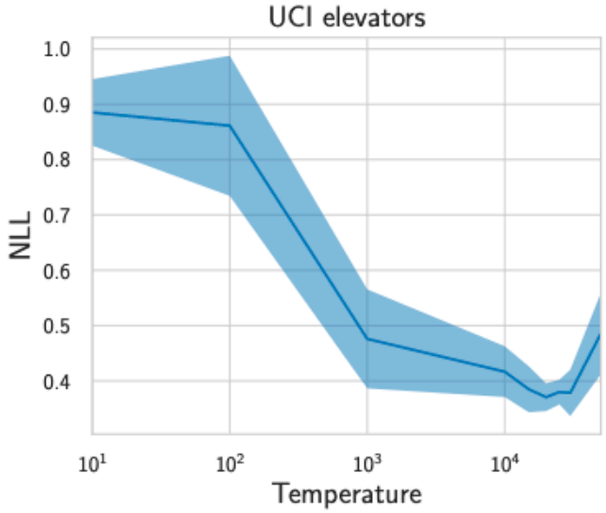
SI ABLATION STUDY: TEMPERATURE



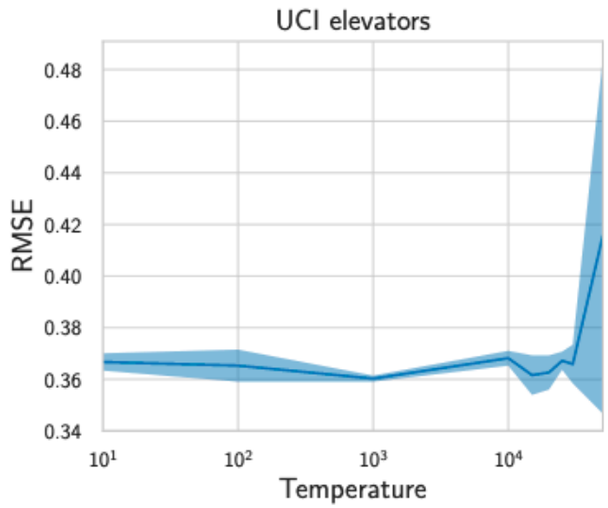
(a)



(b)



(c)



(d)