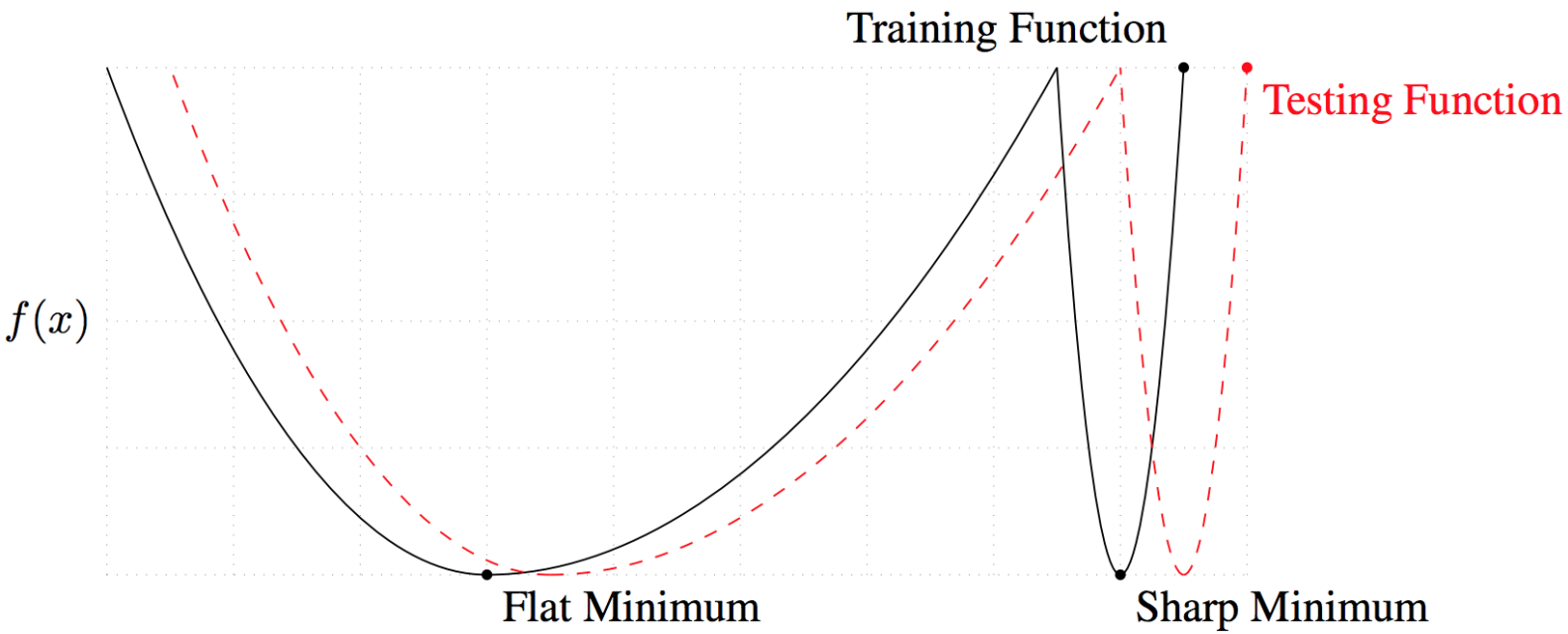
Understanding Loss Valleys for Practical Bayesian Deep Learning

Andrew Gordon Wilson

https://cims.nyu.edu/~andrewgw/
Courant Institute for Mathematical Sciences
Center for Data Science

Optimization, Big Data and Applications (OBA)
Veroli Summer School
July 7, 2022

Wide Optima Generalize Better



Keskar et. al, ICLR 2017.
On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima.

Bayesian integration will give *very* different predictions *in deep learning especially*!

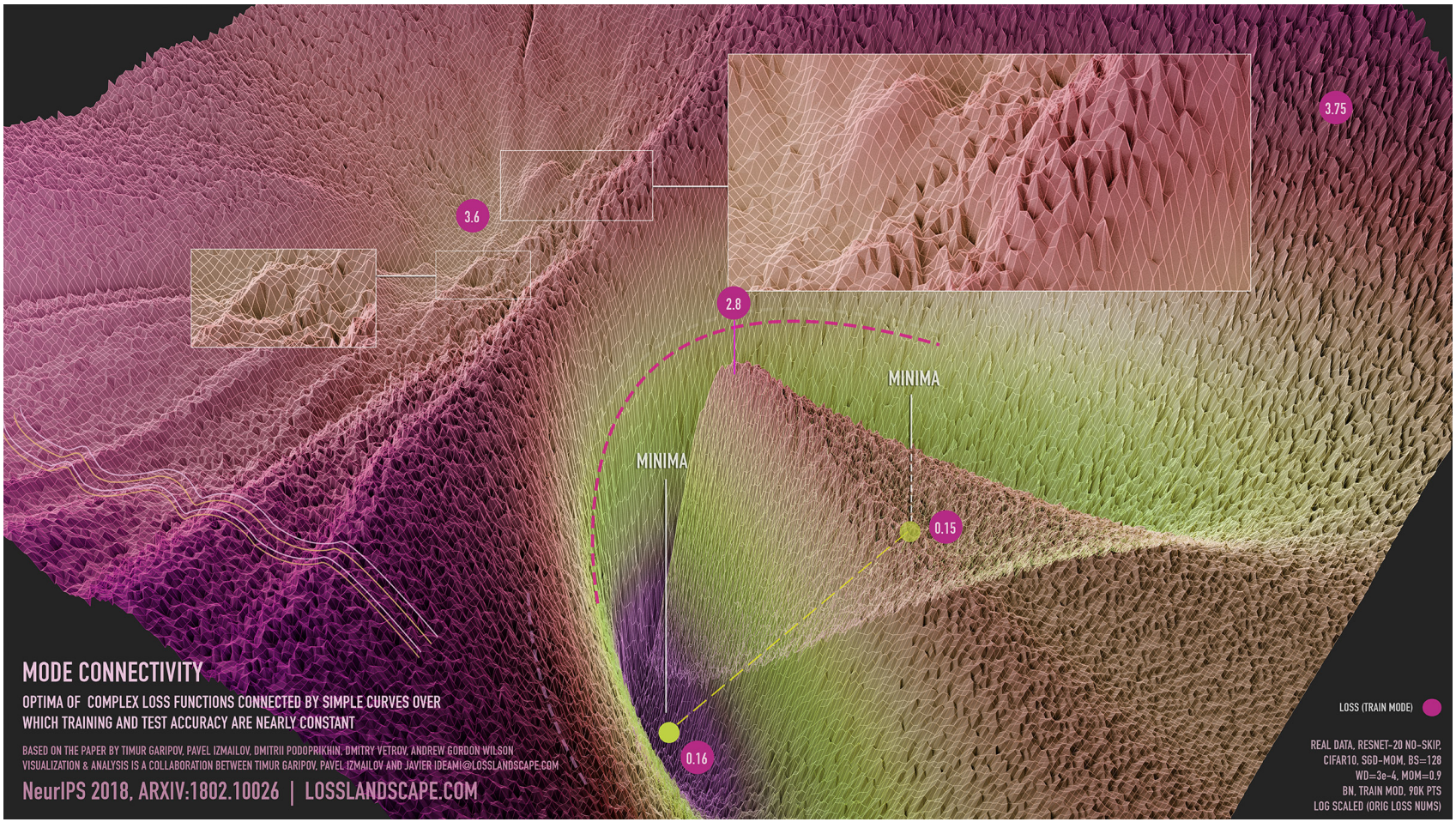
Understanding Loss Surfaces for BMA

Recall the *Bayesian model average* (BMA):

$$p(y|x_*, \mathcal{D}) = \int p(y|x_*, w)p(w|\mathcal{D})dw\,. \tag{1}$$

- The posterior $p(w|\mathcal{D})$ (or loss $\mathcal{L} = -\log p(w|\mathcal{D})$) for neural networks is extraordinarily complex, containing many complementary solutions, which is why BMA is *especially* significant in deep learning.
- Understanding the structure of neural network loss landscapes is crucial for better estimating the BMA.

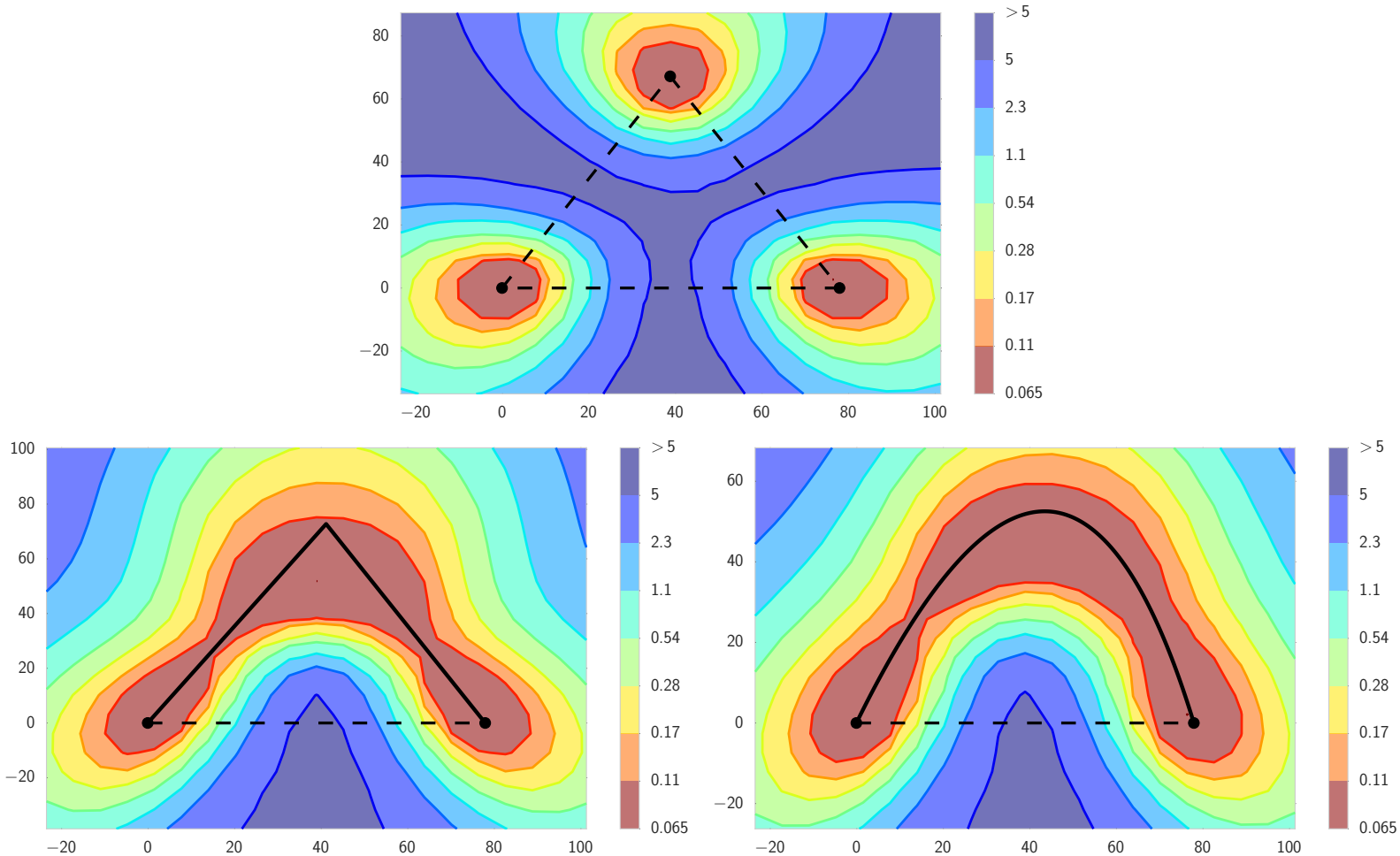
Mode Connectivity



Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs.
T. Garipov, P. Izmailov, D. Podoprikhin, D. Vetrov, A.G. Wilson. NeurIPS 2018.

Loss landscape figures in collaboration with Javier Ideami (losslandscape.com).

Mode Connectivity



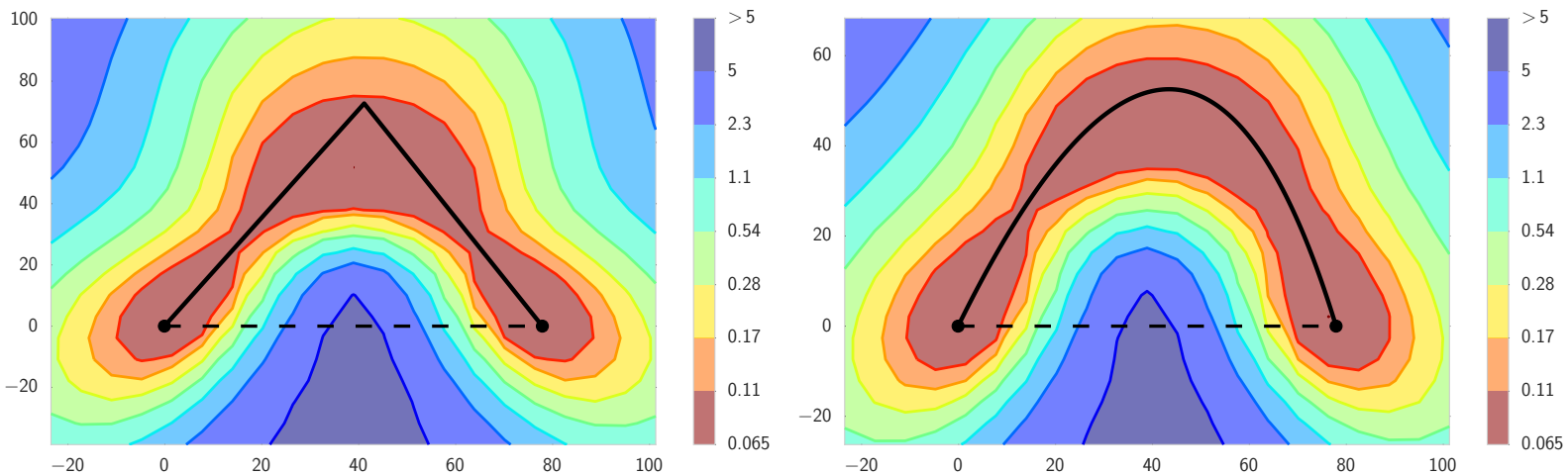
Example Parametrizations

Polygonal Chain:

$$\phi_\theta(t) = \begin{cases} 2\left(t\theta + (0.5-t)\hat{w}_1\right), & 0 \le t \le 0.5 \\ 2\left((t-0.5)\hat{w}_2 + (1-t)\theta\right), & 0.5 \le t \le 1. \end{cases} \tag{2}$$

Bezier Curve:

$$\phi_\theta(t) = (1-t)^2\hat{w}_1 + 2t(1-t)\theta + t^2\hat{w}_2, \quad 0 \le t \le 1. \tag{3}$$



Connection Procedure with Tractable Loss

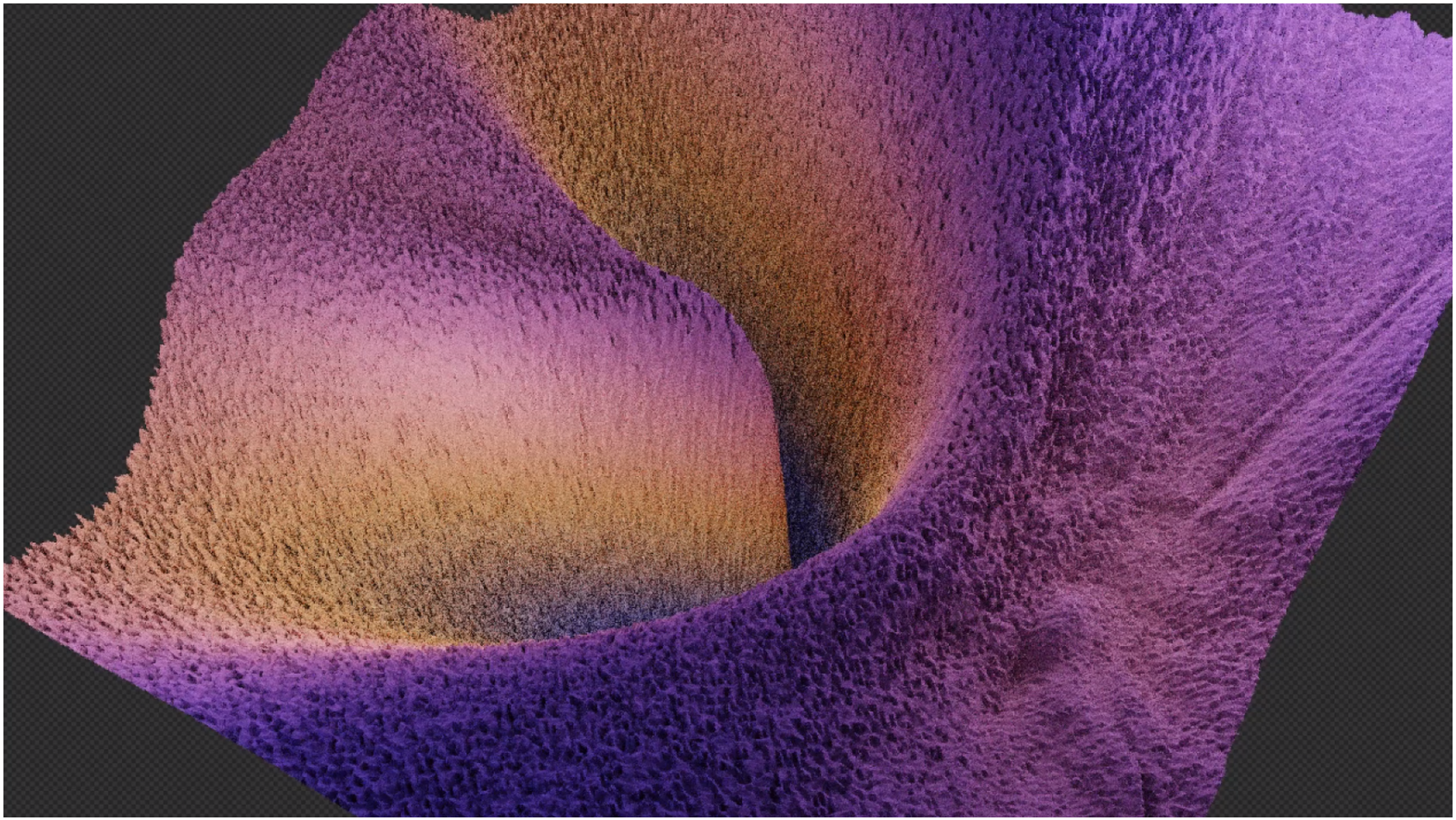
We propose the computationally tractable loss:

$$\ell(\theta) = \int_0^1 \mathcal{L}(\phi_\theta(t))dt = \mathbb{E}_{t\sim U(0,1)}\mathcal{L}(\phi_\theta(t)) \tag{4}$$

At each iteration we sample $t \in [0, 1]$ and make a gradient step for θ with respect to $\mathcal{L}(\phi_\theta(t))$:

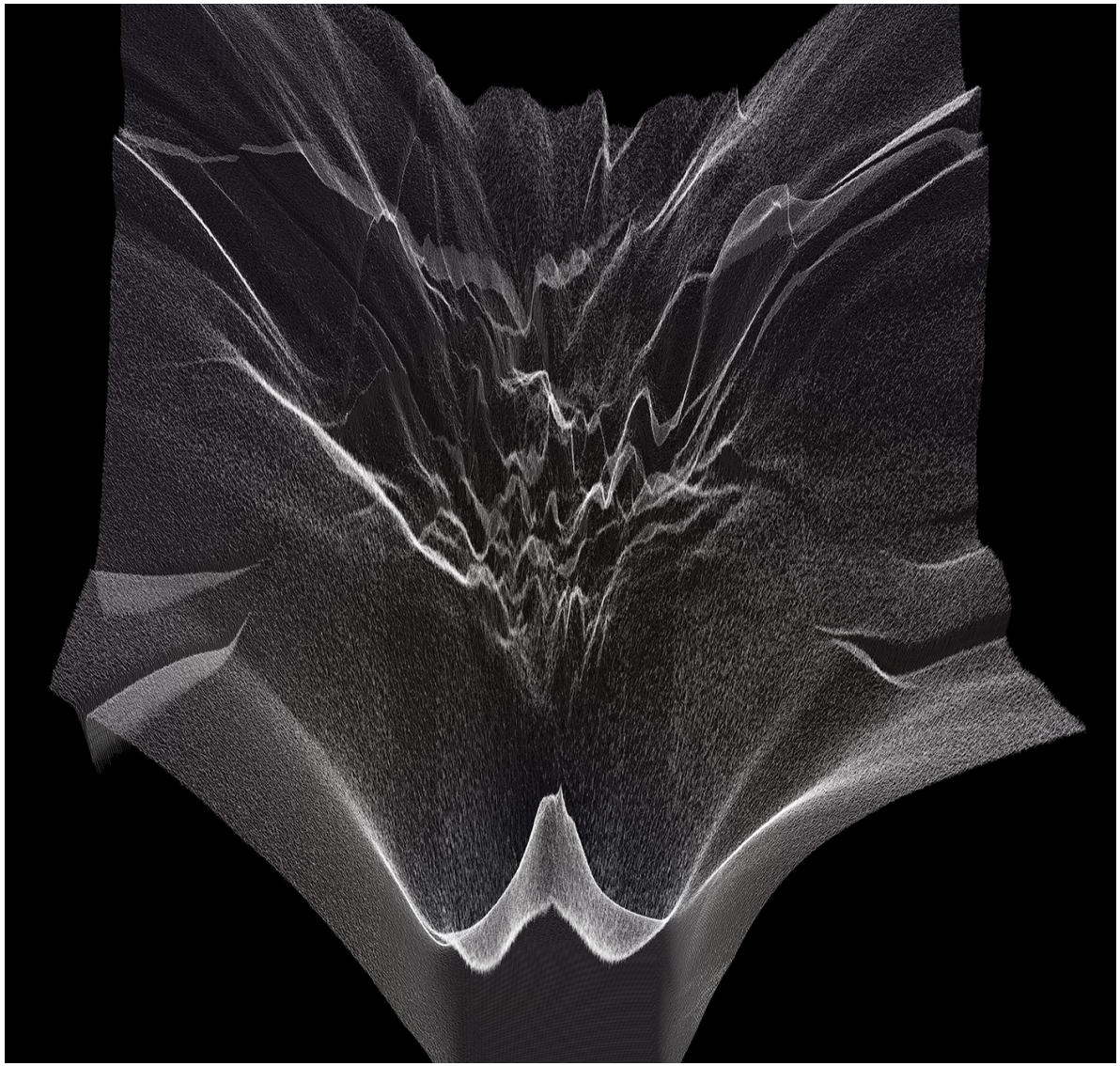
$$\mathbb{E}_{t\sim U(0,1)}\nabla_\theta\mathcal{L}(\phi_\theta(t)) = \nabla_\theta\mathbb{E}_{t\sim U(0,1)}\mathcal{L}(\phi_\theta(t)) = \nabla_\theta\ell(\theta).$$

Mode Connectivity

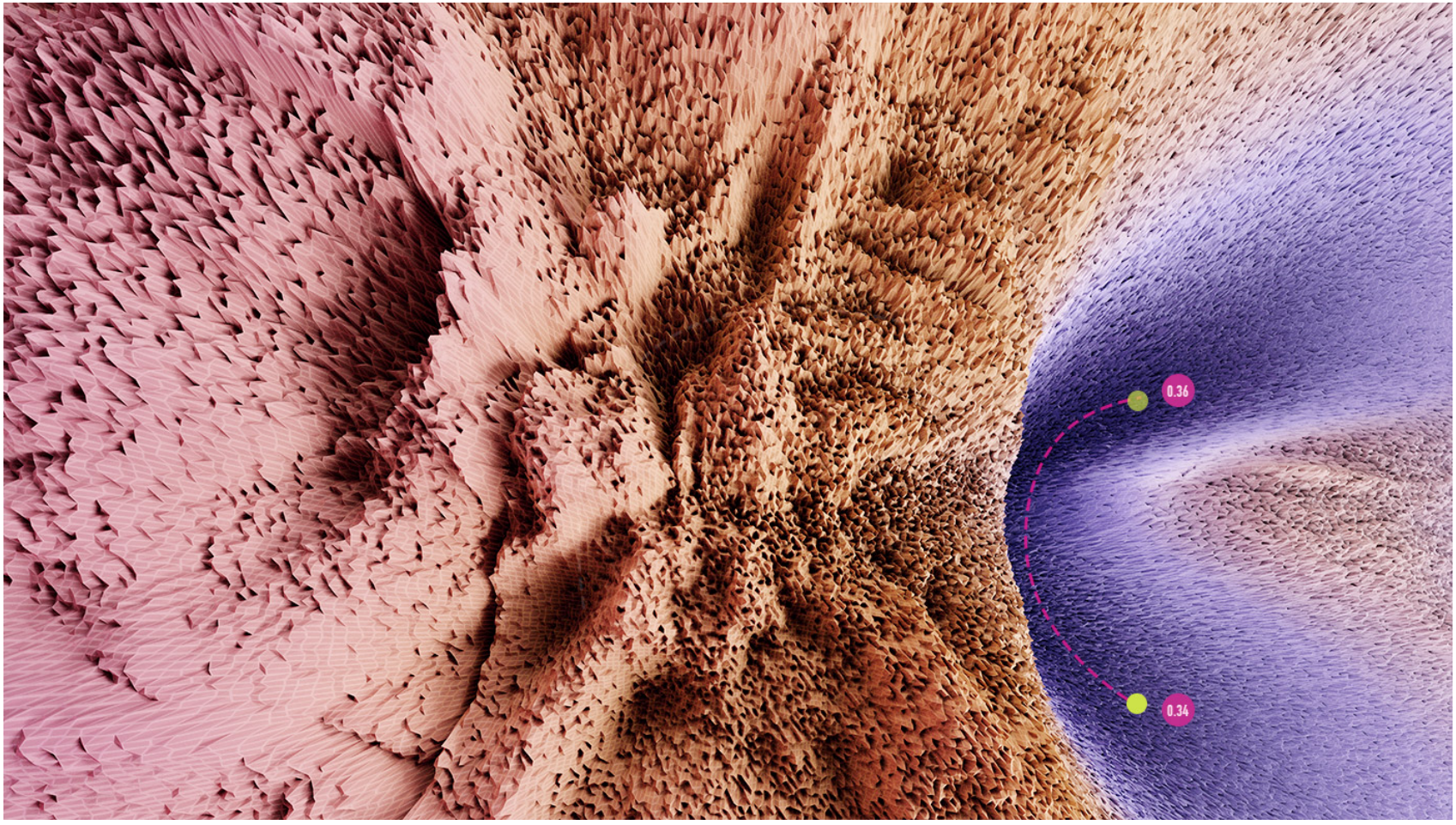


Mode Connectivity

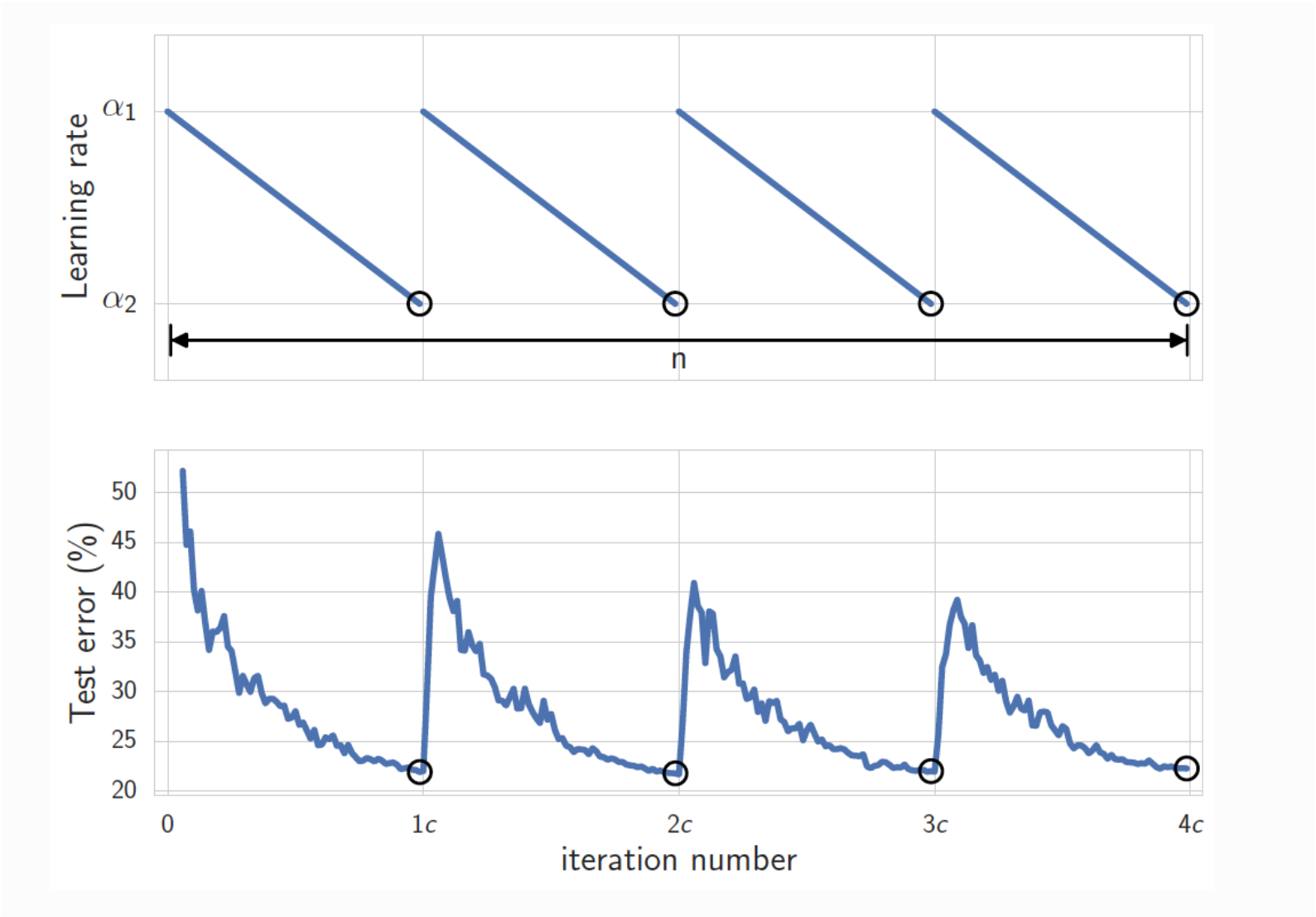
Mode Connectivity



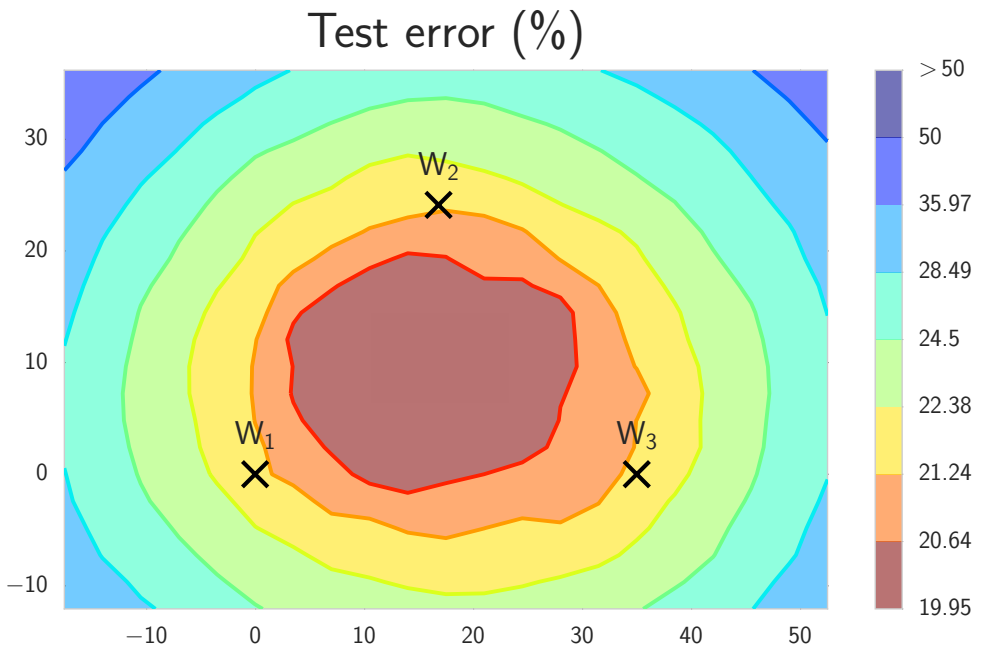
Mode Connectivity



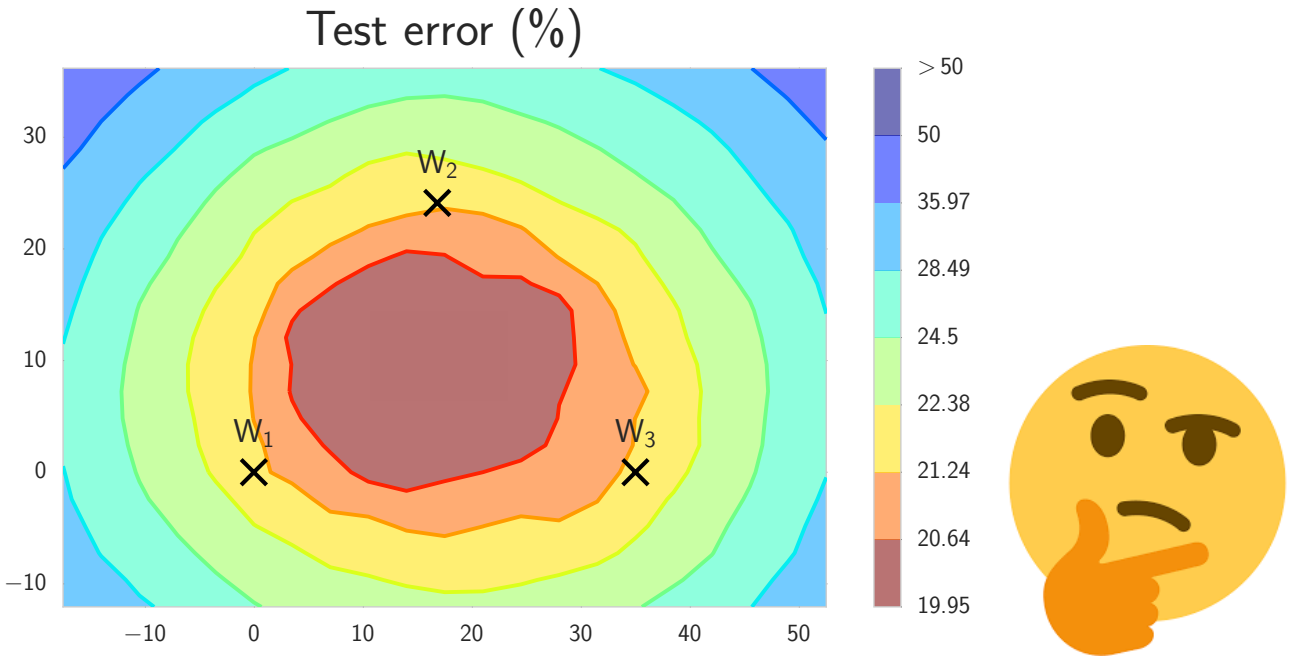
Cyclical Learning Rate Schedule



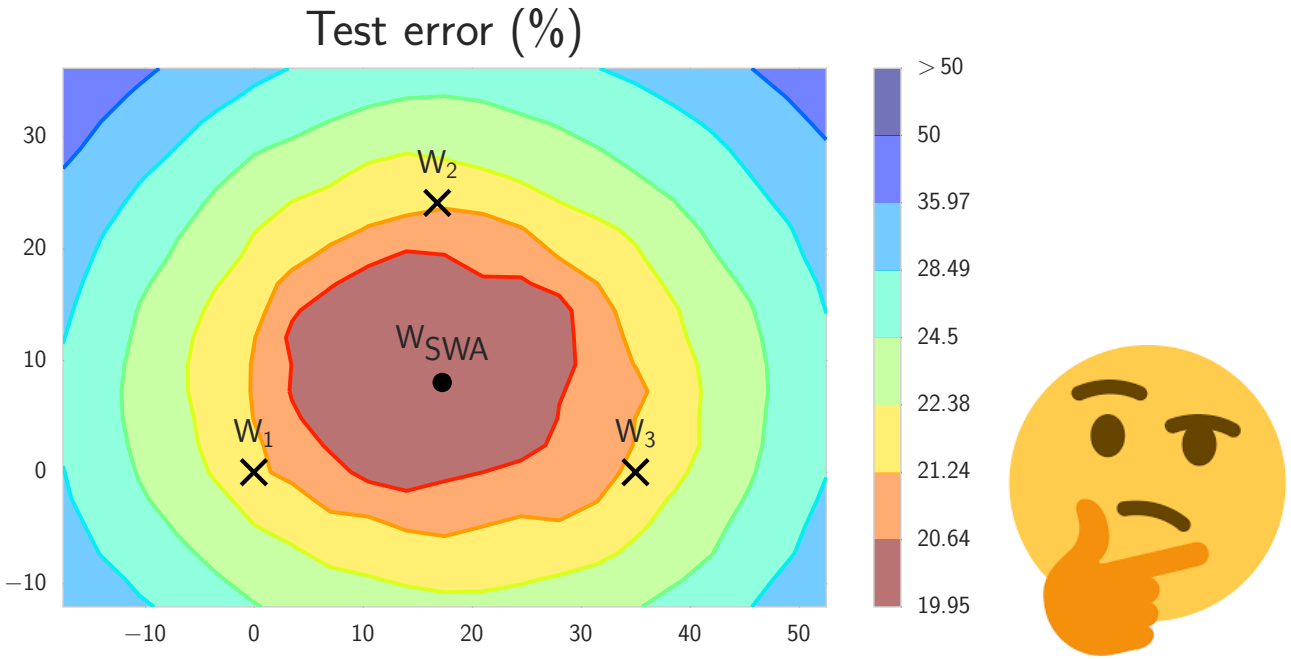
Trajectory of SGD



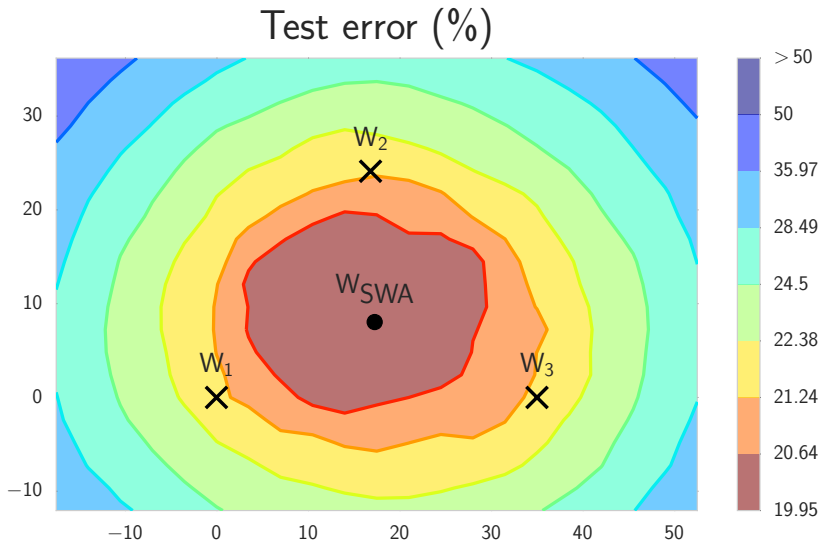
Trajectory of SGD

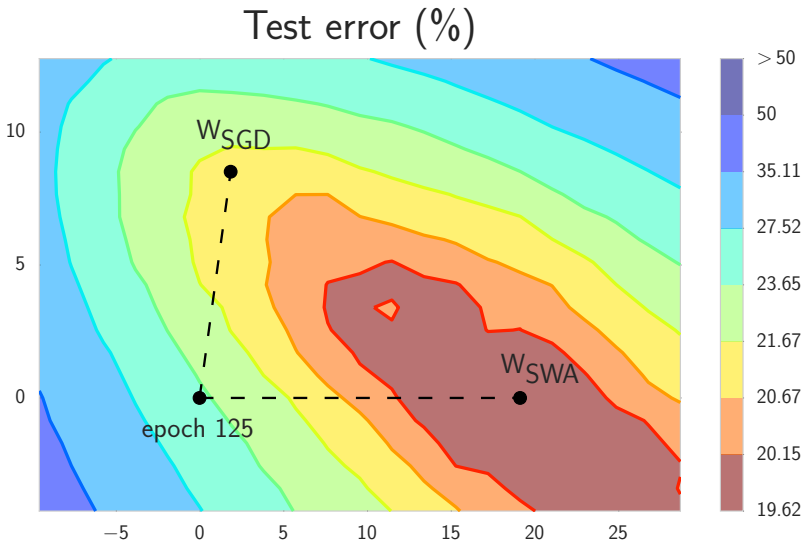


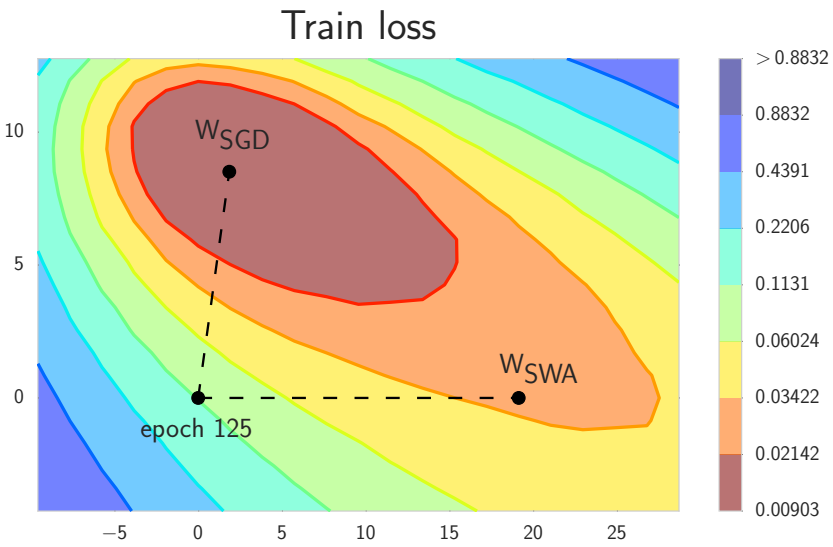
Trajectory of SGD



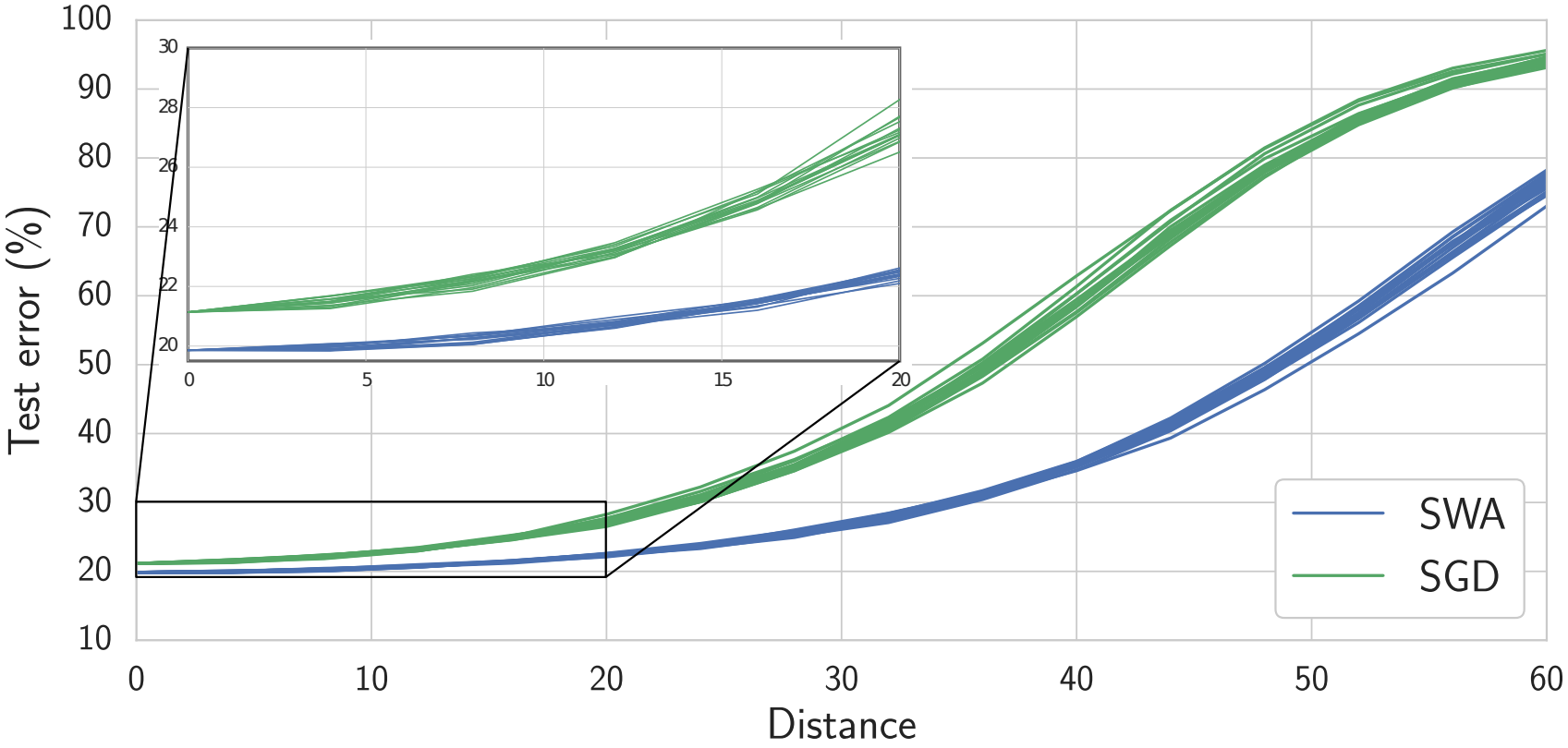
Trajectory of SGD



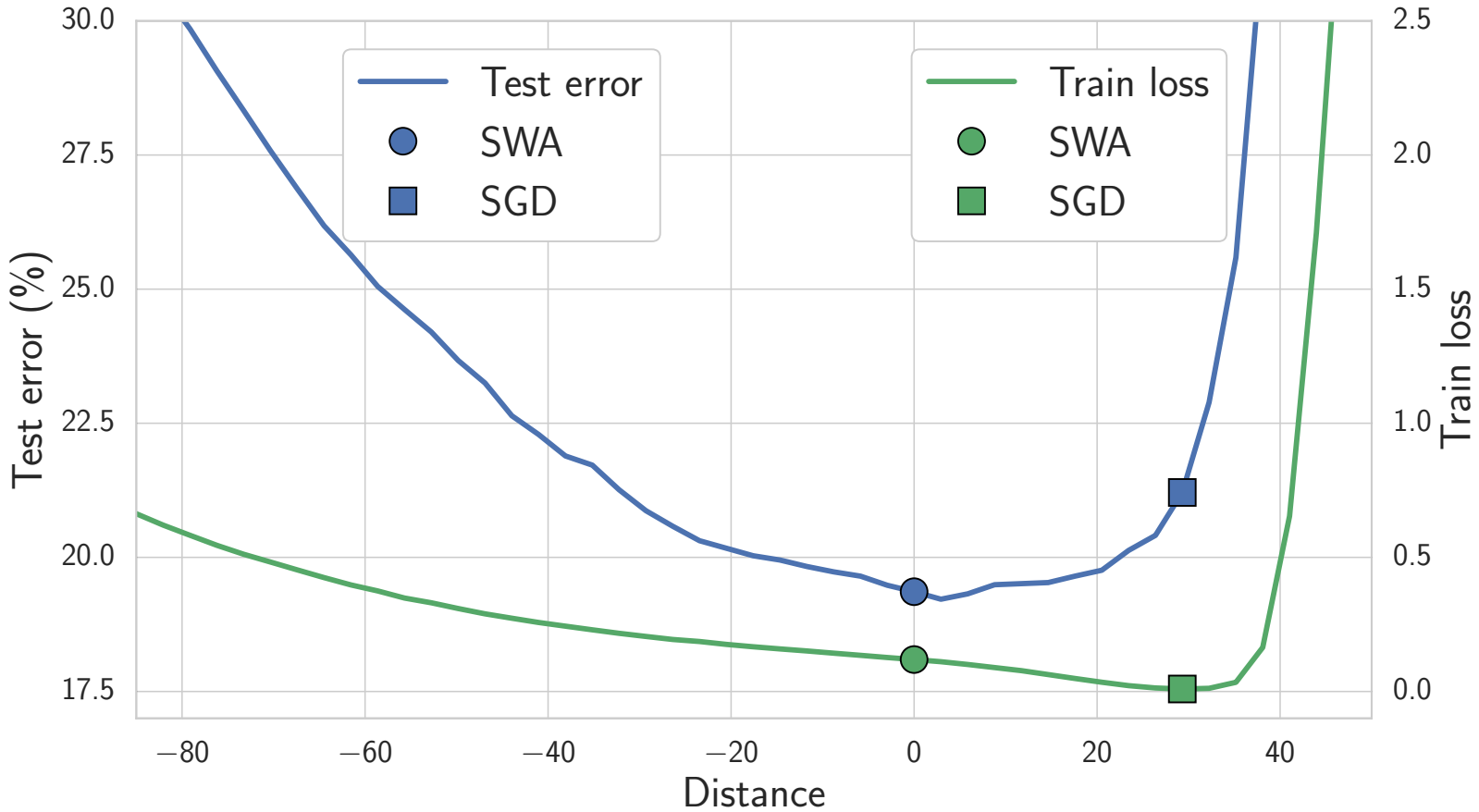




Following Random Paths



Path from w_{SWA} to w_{SGD}



Approximating an FGE Ensemble

Because the points sampled from an FGE ensemble take small steps in weight space *by design*, we can do a linearization analysis to show that

$$f(w_{\text{SWA}}) \approx \frac{1}{n} \sum f(w_i)$$

SWA Results, CIFAR

Table 1: Accuracies (%) of SWA, SGD and FGE methods on CIFAR-100 and CIFAR-10 datasets for different training budgets. Accuracies for FGE were taken from [Garipov et al., 2018].

			SWA		
DNN (Budget)	SGD	FGE (1 Budget)	1 Budget	1.25 Budgets	1.5 Budgets
		CIFAR-100			
VGG-16 (200)	72.55 ± 0.10	74.26	73.91 ± 0.12	74.17 ± 0.15	74.27 ± 0.25
ResNet-110 (150)	78.49 ± 0.36	79.84	79.77 ± 0.17	80.18 ± 0.23	80.35 ± 0.16
WRN-28-10 (200)	80.82 ± 0.23	82.27	81.46 ± 0.23	81.91 ± 0.27	82.15 ± 0.27
PyramidNet-272 (300)	83.41 ± 0.21	–	–	83.93 ± 0.18	84.16 ± 0.15
		CIFAR-10			
VGG-16 (200)	93.25 ± 0.16	93.52	93.59 ± 0.16	93.70 ± 0.22	93.64 ± 0.18
ResNet-110 (150)	95.28 ± 0.10	95.45	95.56 ± 0.11	95.77 ± 0.04	95.83 ± 0.03
WRN-28-10 (200)	96.18 ± 0.11	96.36	96.45 ± 0.11	96.64 ± 0.08	96.79 ± 0.05
ShakeShake-2x64d (1800)	96.93 ± 0.10	–	–	97.16 ± 0.10	97.12 ± 0.06

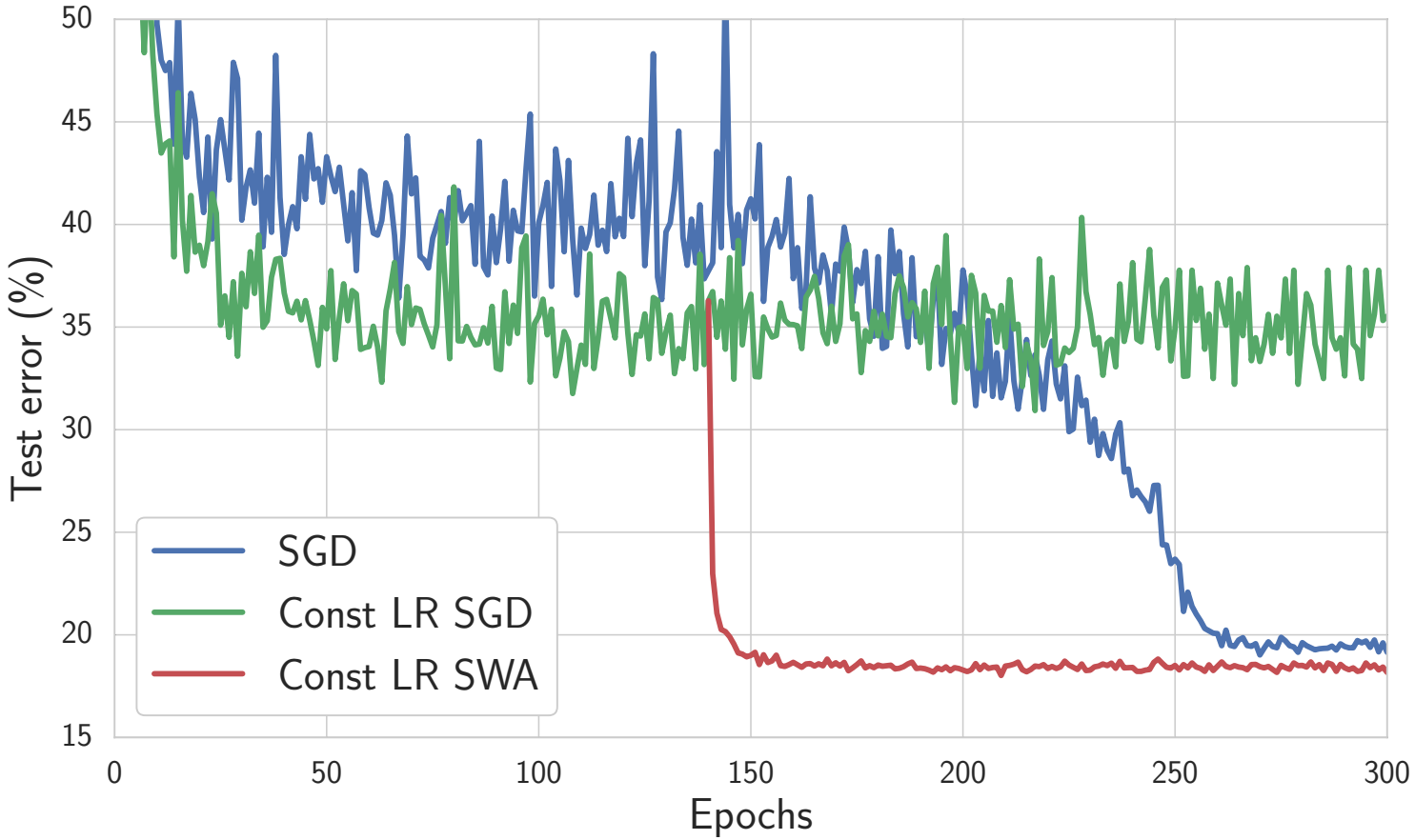
SWA Results, ImageNet (Top-1 Error Rate)

DNN	SGD	SWA	
		5 epochs	10 epochs
ResNet-50	76.15	76.83 ± 0.01	76.97 ± 0.05
ResNet-152	78.31	78.82 ± 0.01	78.94 ± 0.07
DenseNet-161	77.65	78.26 ± 0.09	78.44 ± 0.06

Orbiting Loss Valleys

SGD (with constant LR) moves around the surface of a low loss space of solutions.
Averaging the iterates finds a point centred in this space.

High Constant LR



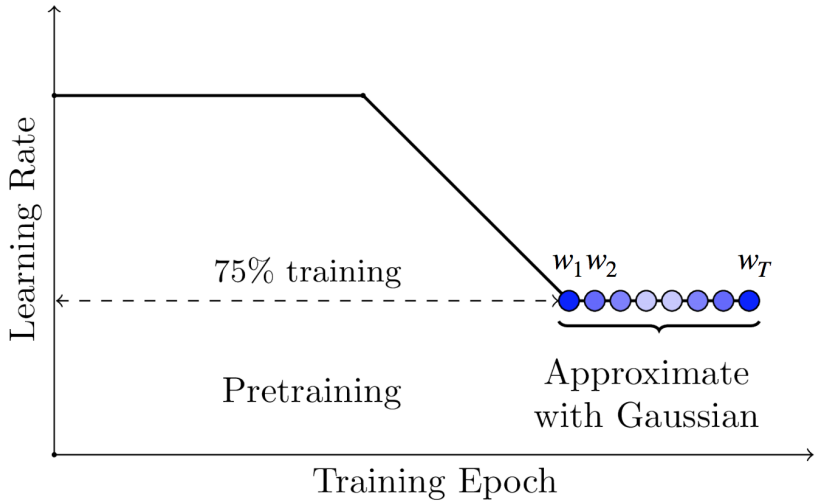
Side observation: Averaging bad models does not give good solutions. Averaging bad weights can give great solutions.

Use SGD Trajectory for BMA

Can we recycle geometric information in the SGD trajectory for scalable posterior approximations, centred on flat regions of the loss?

Uncertainty Representation with SWAG

1. Leverage theory that shows SGD with a constant learning rate is approximately sampling from a Gaussian distribution.
2. Compute first *two* moments of SGD trajectory (SWA computes just the first).
3. Use these moments to construct a Gaussian approximation in weight space.
4. Sample from this Gaussian distribution, pass samples through predictive distribution, and form a Bayesian model average.

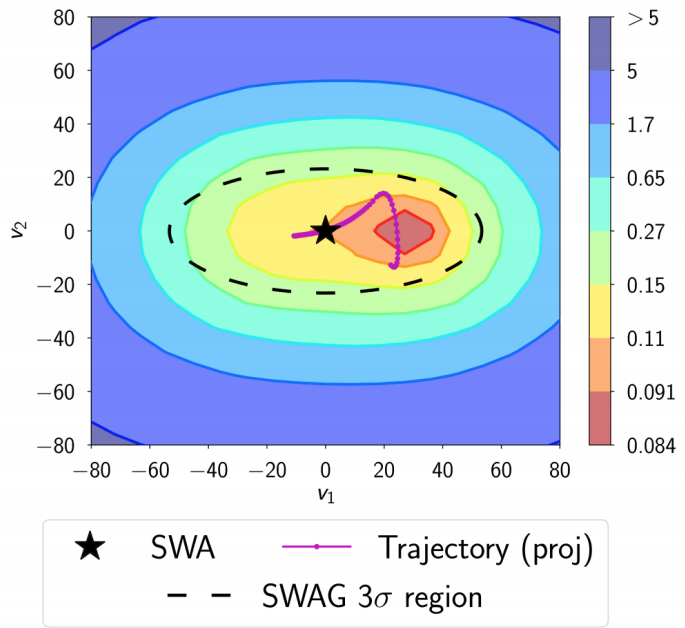


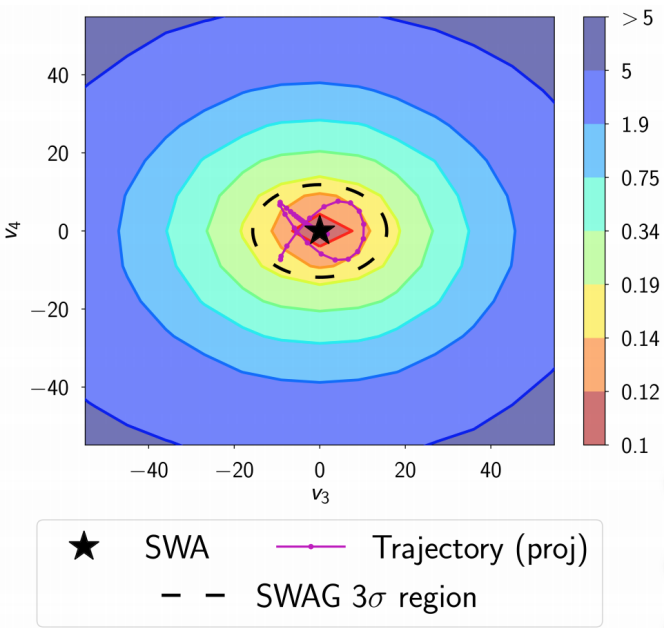
$$p(y_*|\mathcal{D}) \approx \frac{1}{J}\sum_{j=1}^{J} p(y_*|w_j)\,, \quad w_j \sim q(w|\mathcal{D})\,, \quad q(w|\mathcal{D}) = \mathcal{N}(\bar{w}, K)$$

$$\bar{w} = \frac{1}{T}\sum_t w_t\,, \quad K = \frac{1}{2}\left(\frac{1}{T-1}\sum_t (w_t - \bar{w})(w_t - \bar{w})^{\mathrm{T}} + \frac{1}{T-1}\sum_t diag(w_i - \bar{w})^2\right)$$

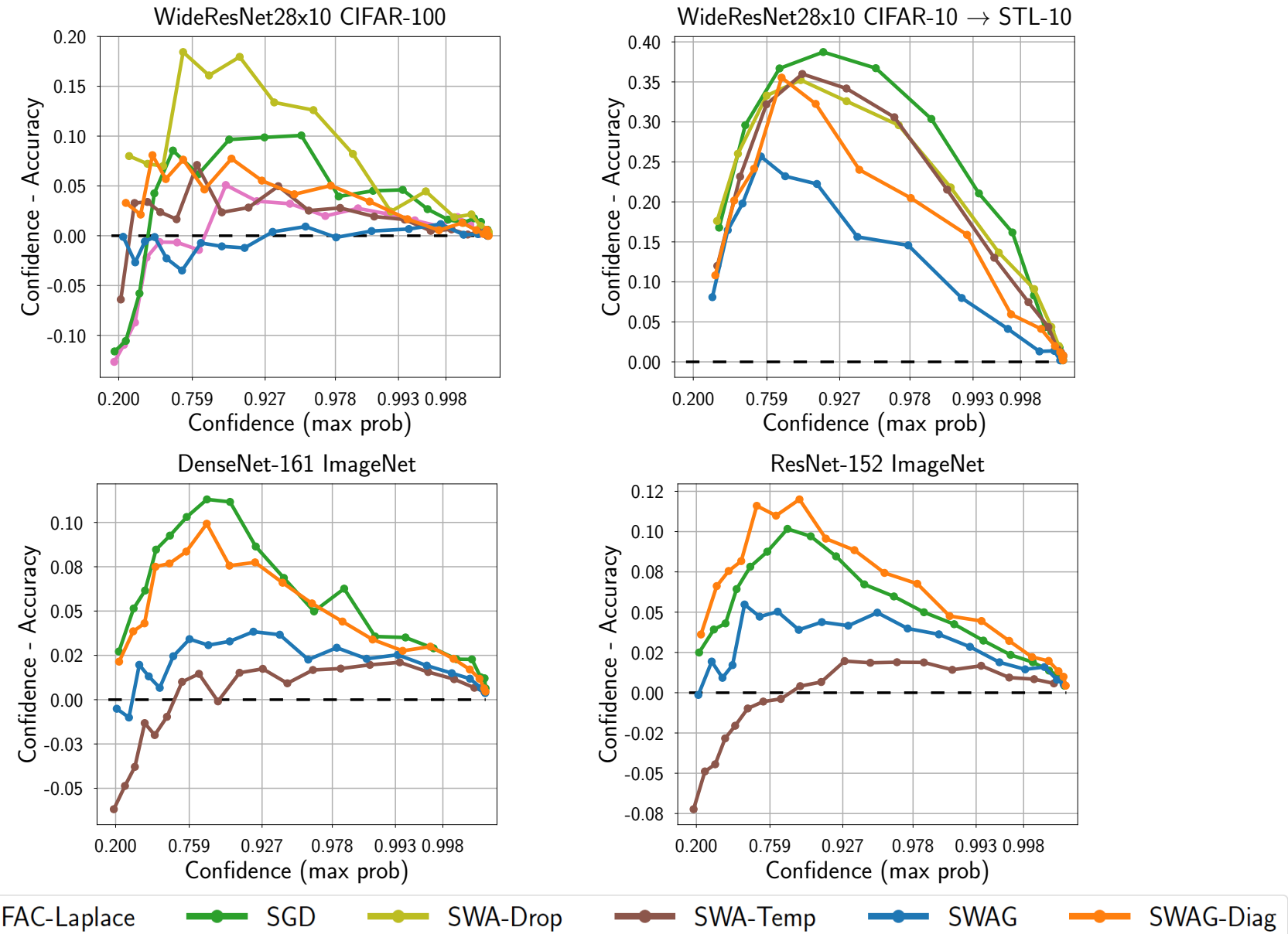
SWAG: *A Simple Baseline for Bayesian Uncertainty in Deep Learning*. Maddox et. al, NeurIPS 2019.
SWA: *Averaging Weights Leads to Wider Optima and Better Generalization*. Izmailov et. al, UAI 2018.

Trajectory in PCA Subspace

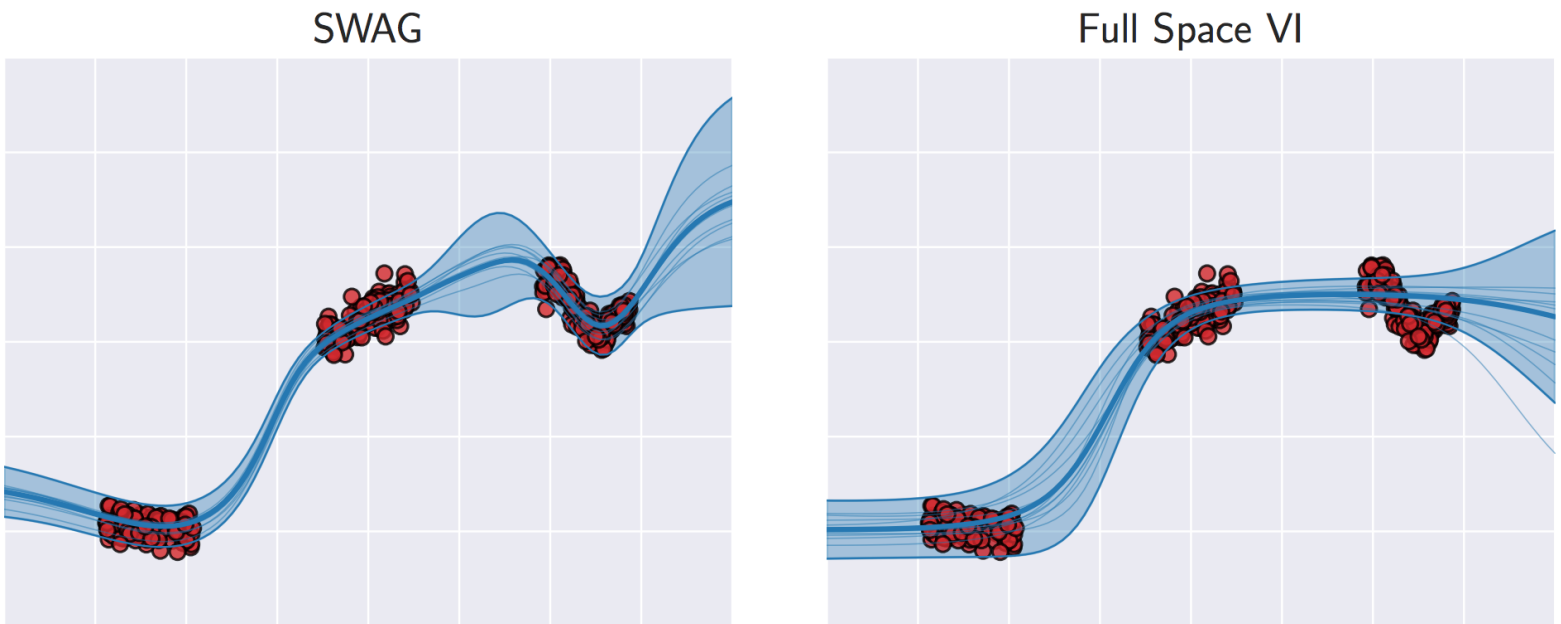




Uncertainty Calibration



SWAG Regression Uncertainty



Subspace Inference for Bayesian Deep Learning

A modular approach:

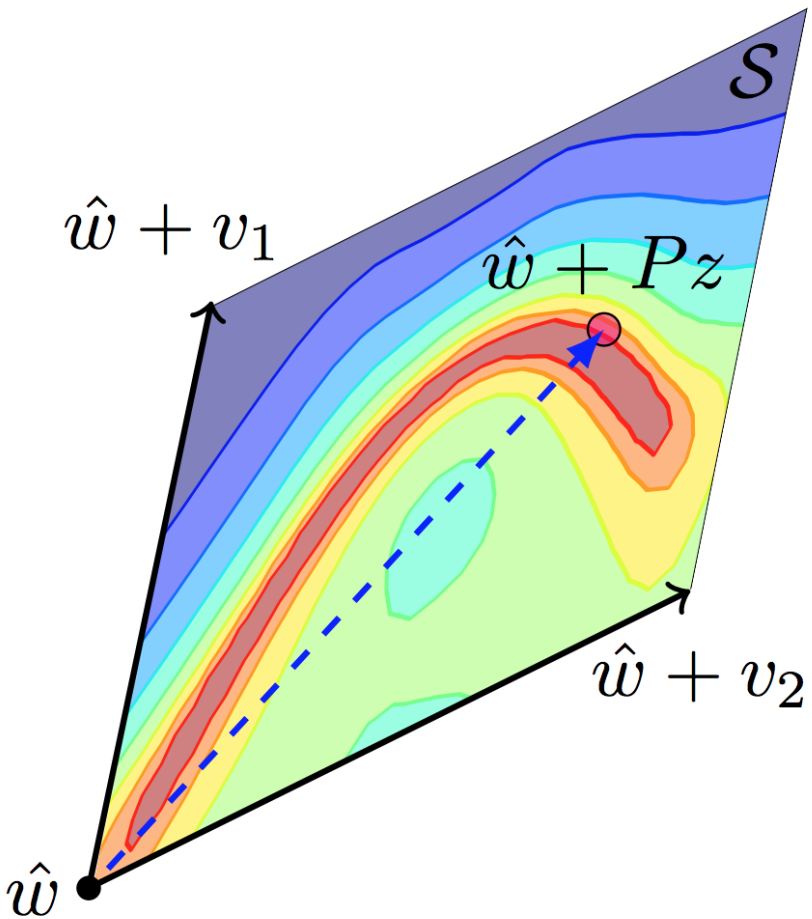
- Construct a subspace of a network with a high dimensional parameter space
- Perform inference directly in the subspace
- Sample from approximate posterior for Bayesian model averaging

We can approximate the posterior of a WideResNet with 36 million parameters in a 5D subspace and achieve state-of-the-art results!

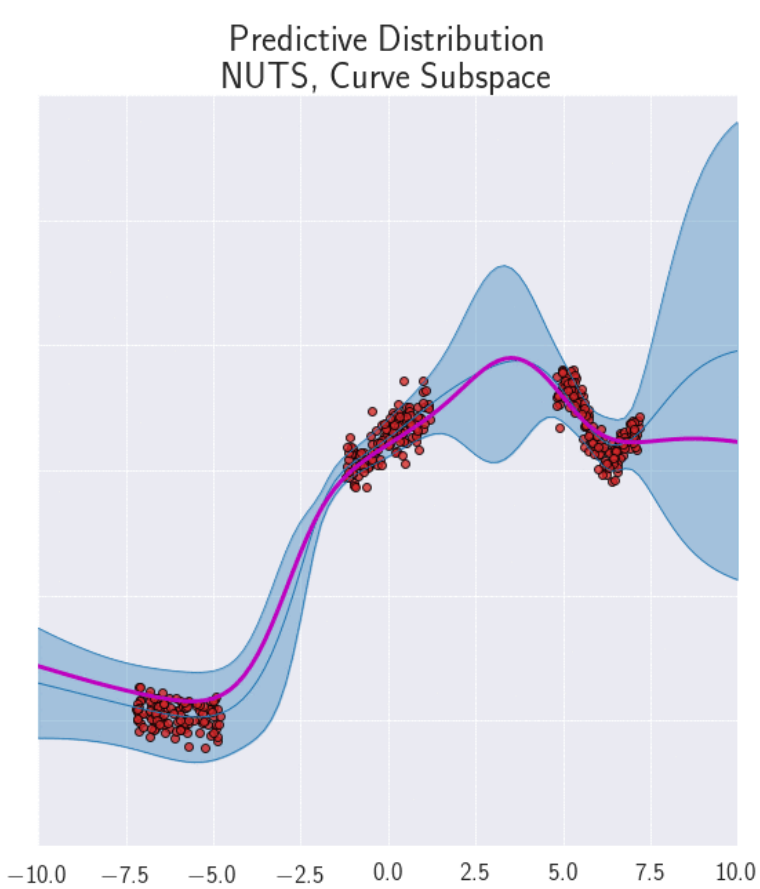
Subspace Inference for Bayesian Deep Learning.
P. Izmailov, W. Maddox, P. Kirichenko, T. Garipov, D. Vetrov, A.G. Wilson
UAI 2018

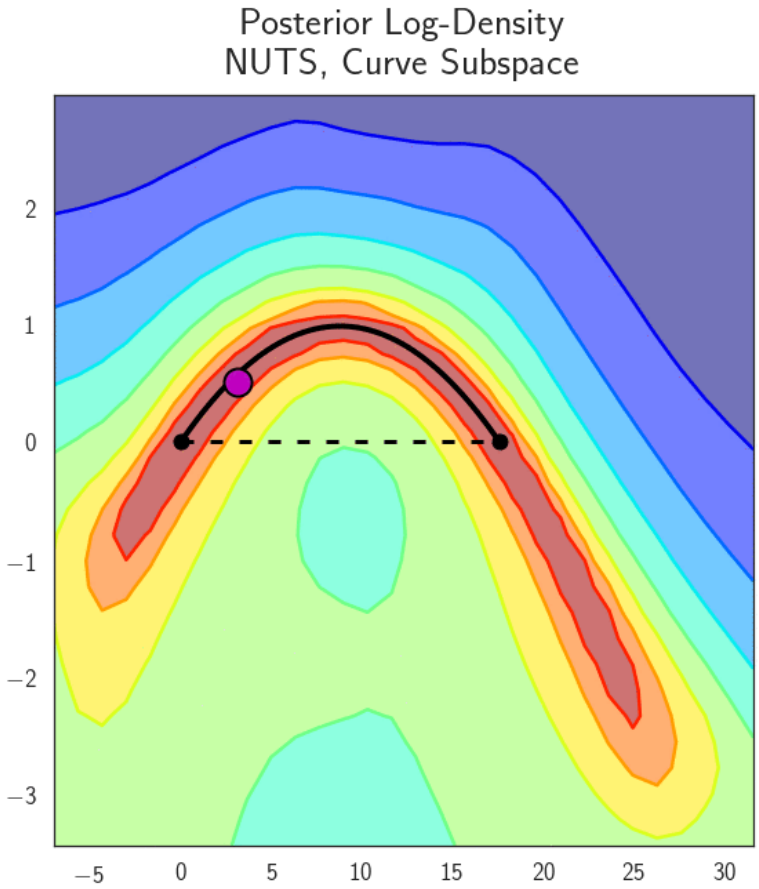
Subspace Construction

- Choose shift $\hat{w}$ and basis vectors $\{d_1, \ldots, d_k\}$.
- Define subspace $S = \{w | w = \hat{w} + z_1 d_1 + z_k d_k\}$.
- Likelihood $p(\mathcal{D}|z) = p_M(\mathcal{D}|w = \hat{w} + Pz)$.
- Posterior inference $p(z|\mathcal{D}) \propto p(\mathcal{D}|z)p(z)$.

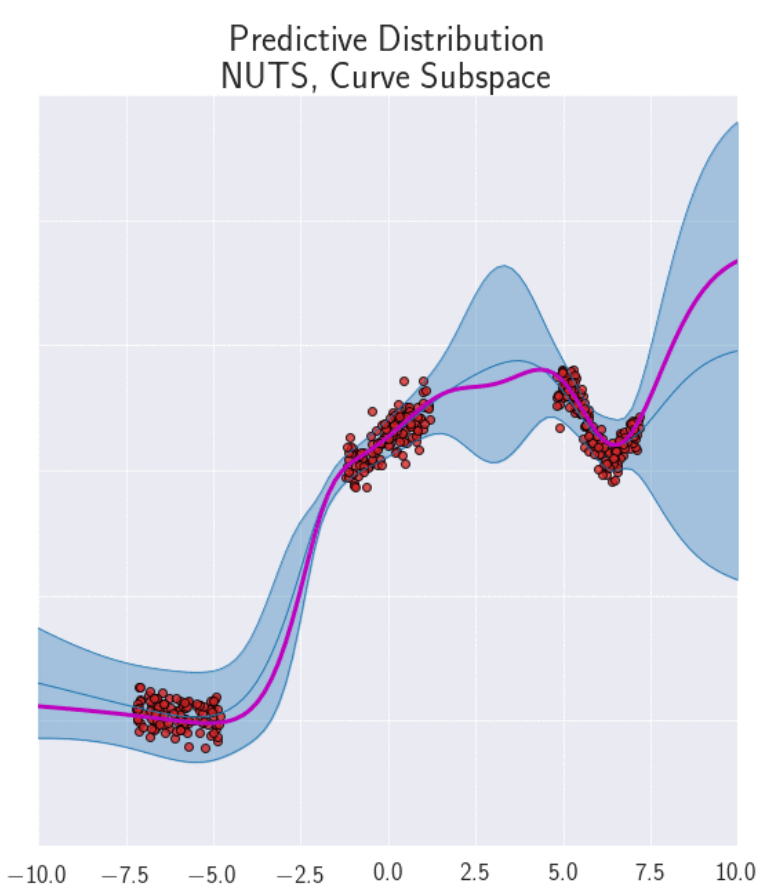


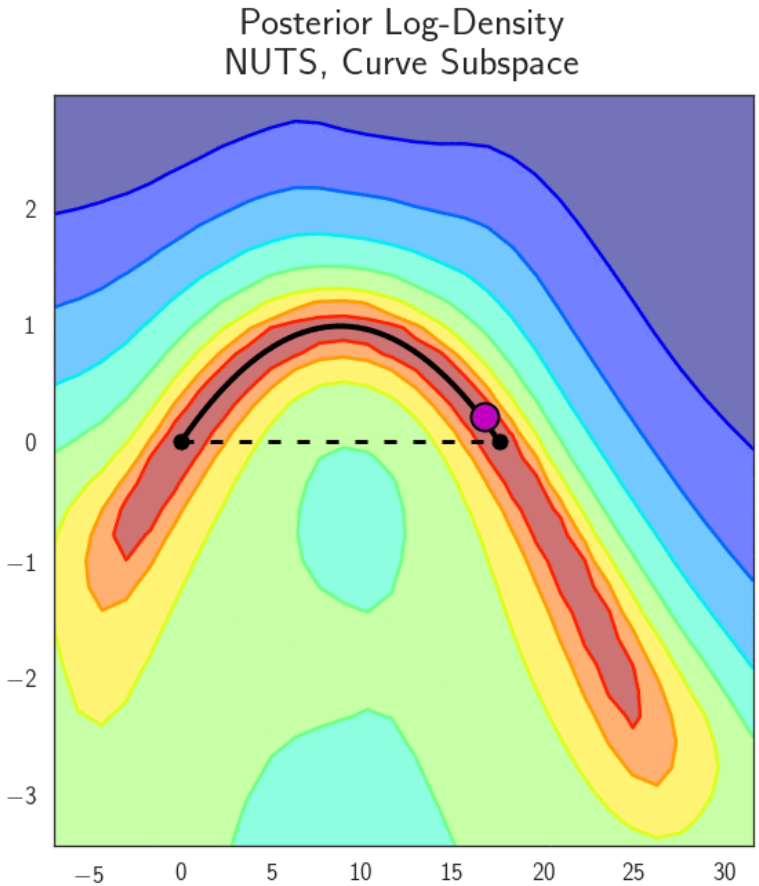
Curve Subspace Traversal



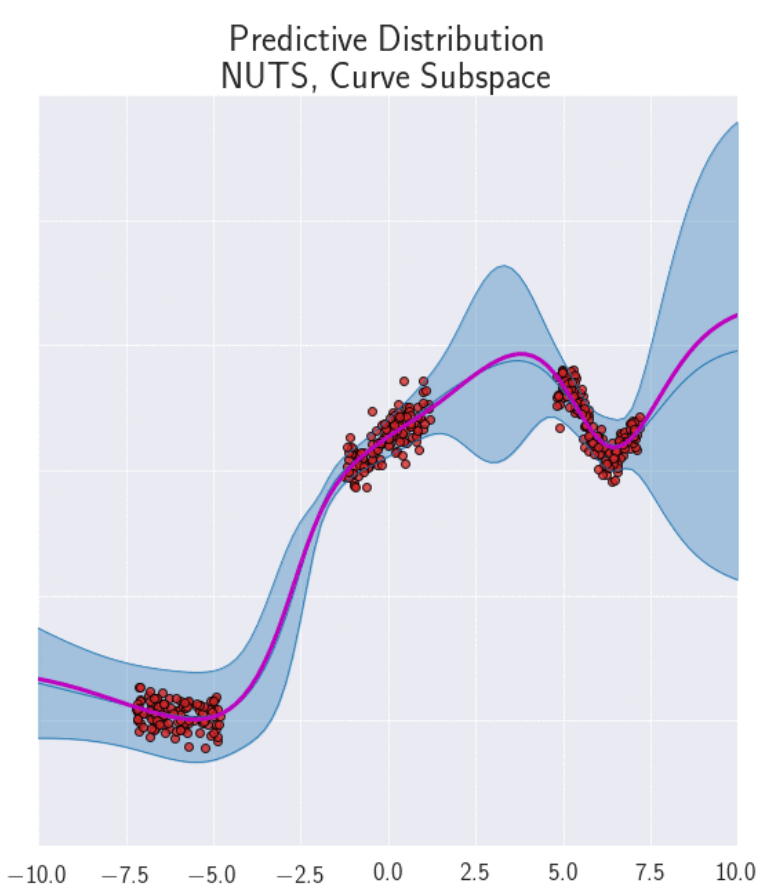


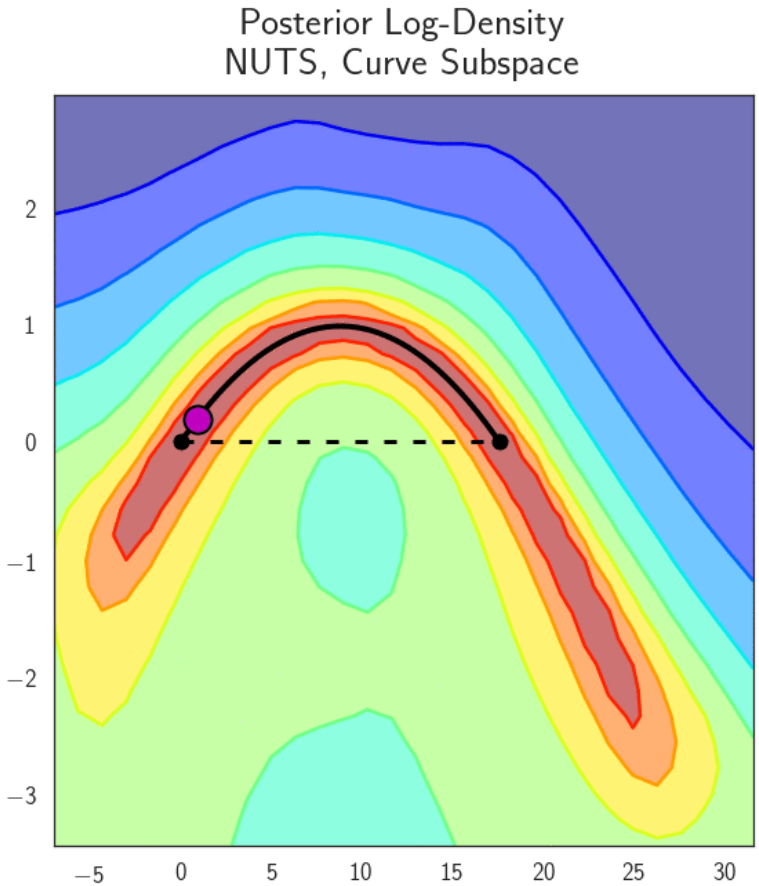
Curve Subspace Traversal



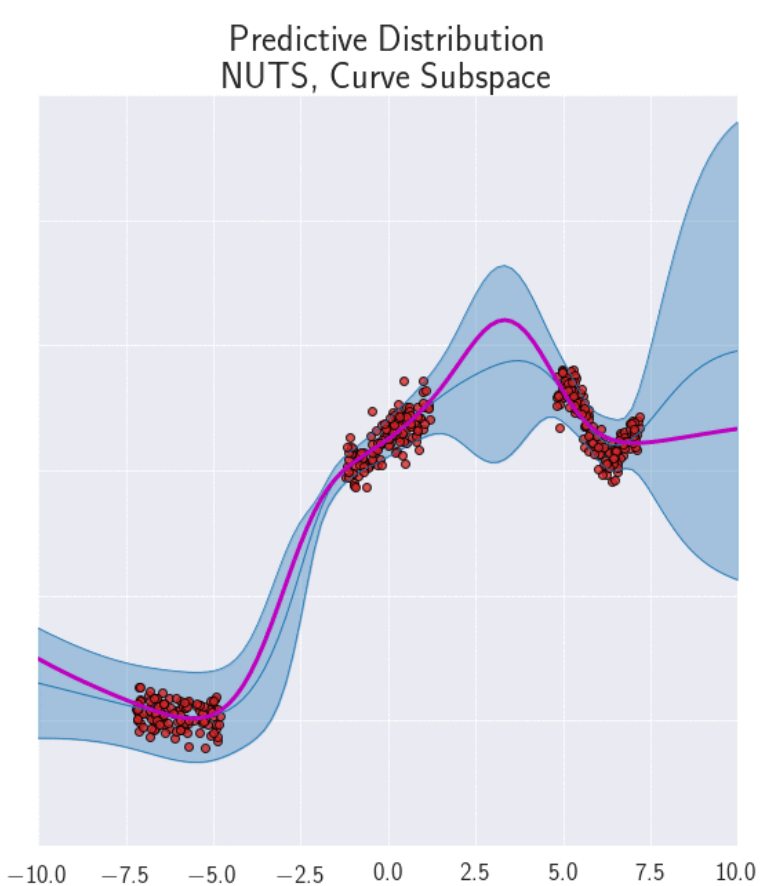


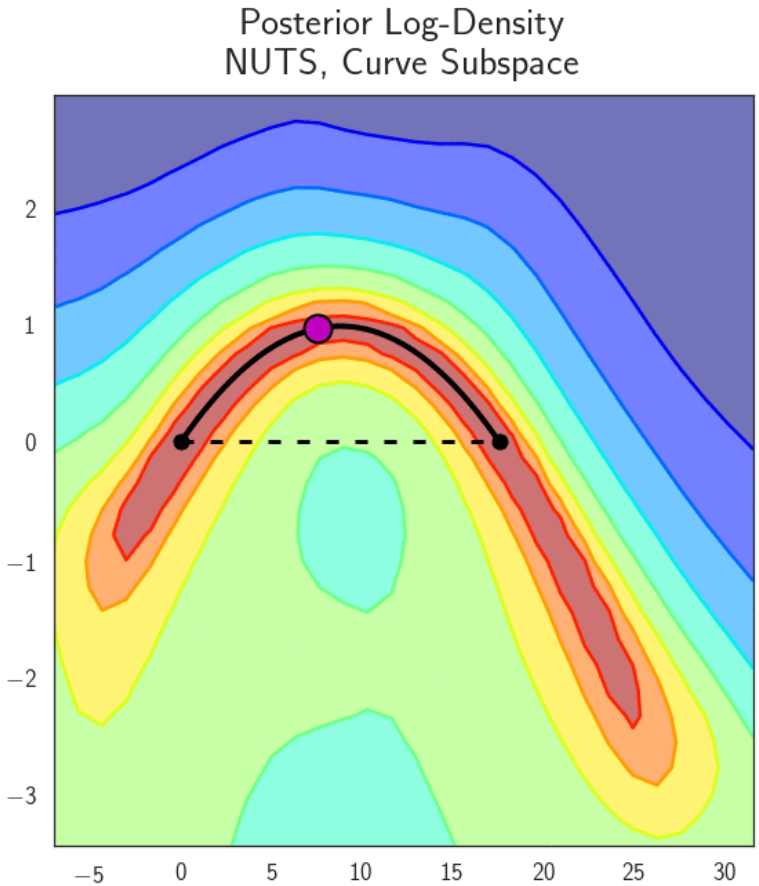
Curve Subspace Traversal



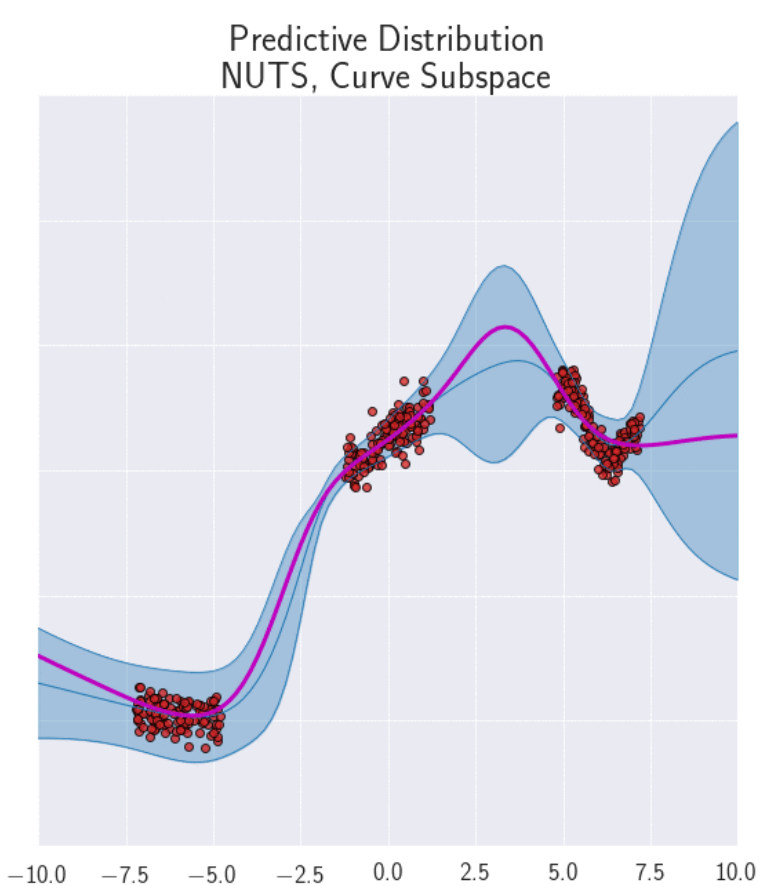


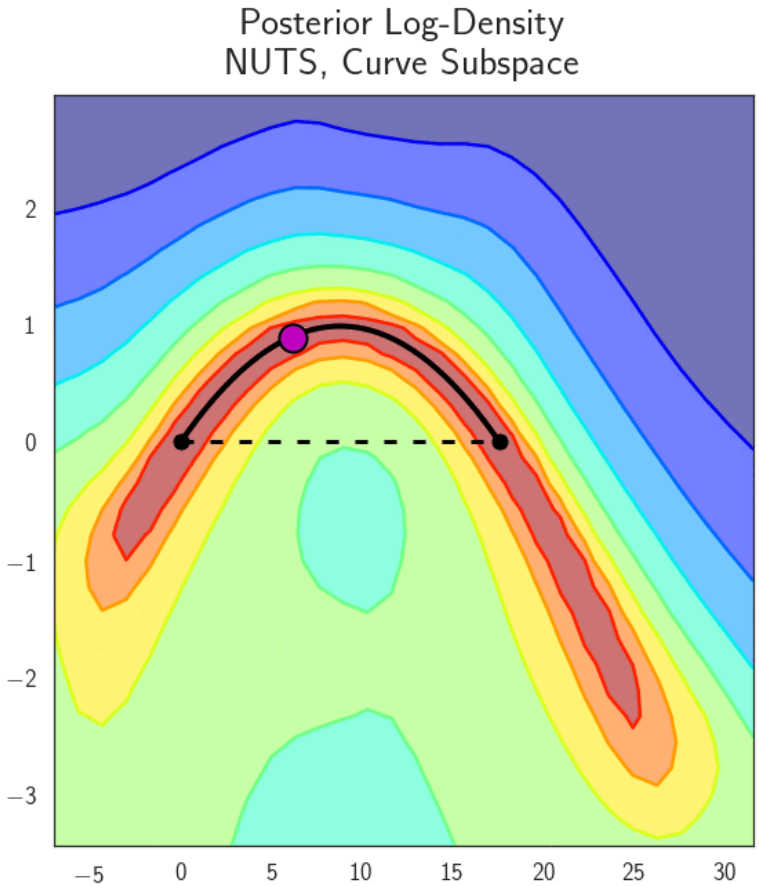
Curve Subspace Traversal



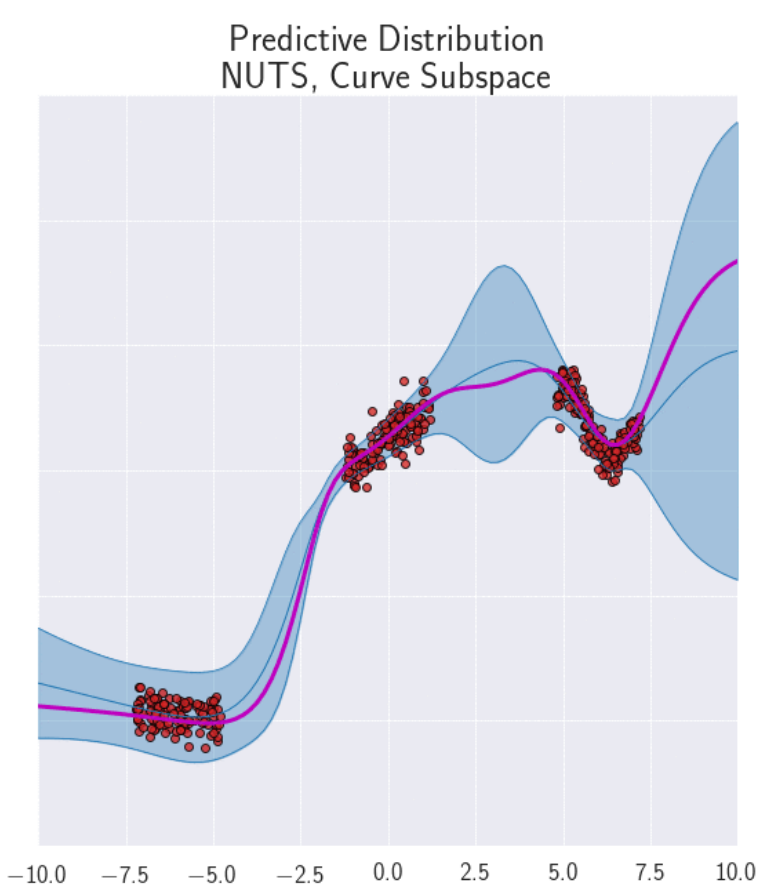


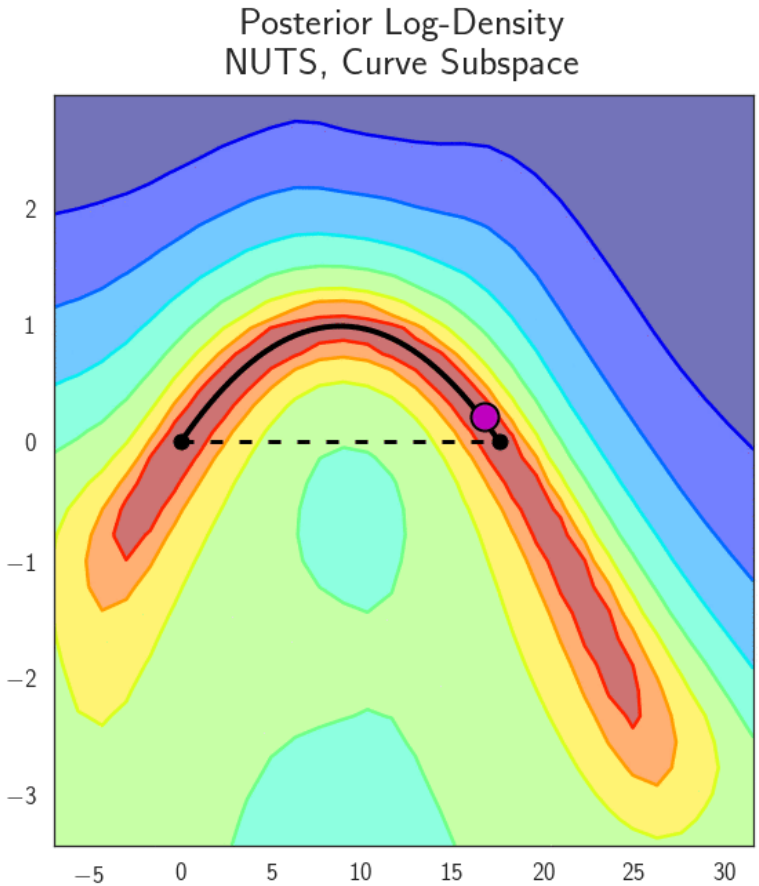
Curve Subspace Traversal



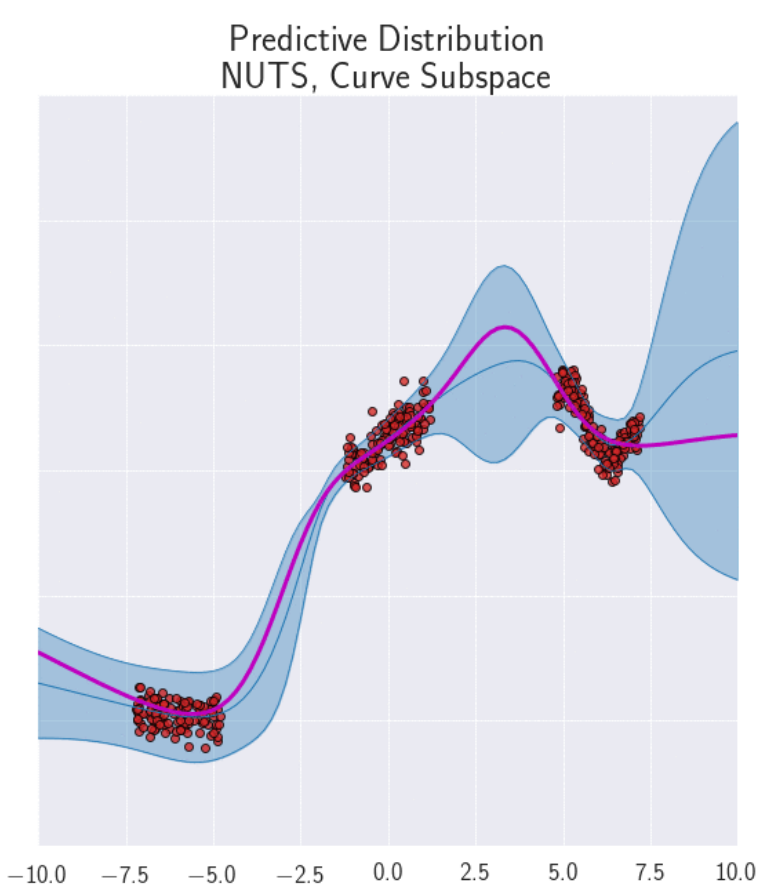


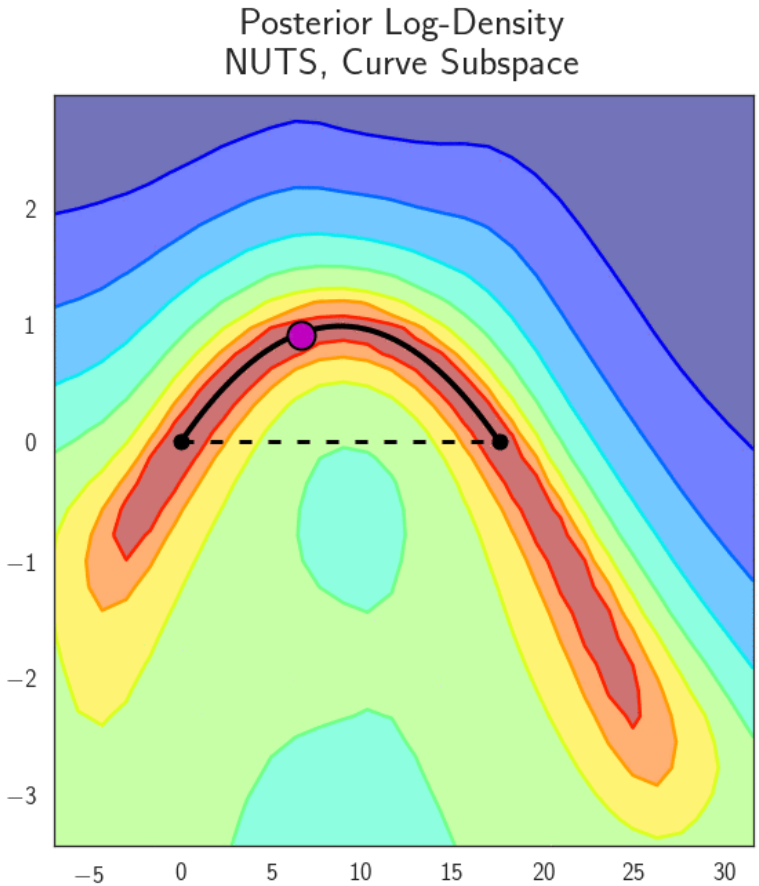
Curve Subspace Traversal



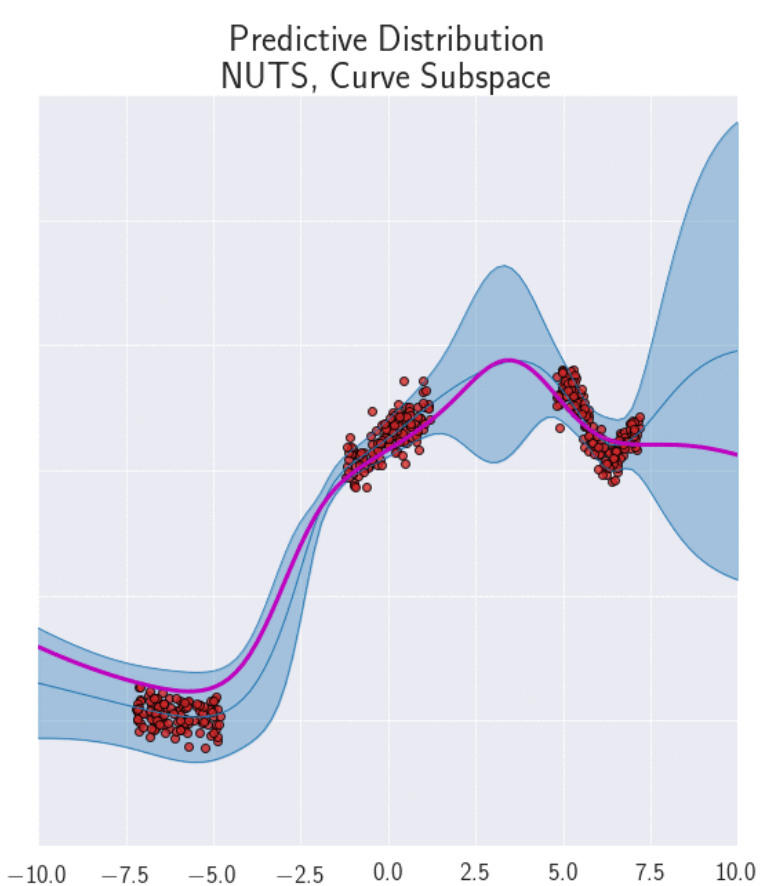


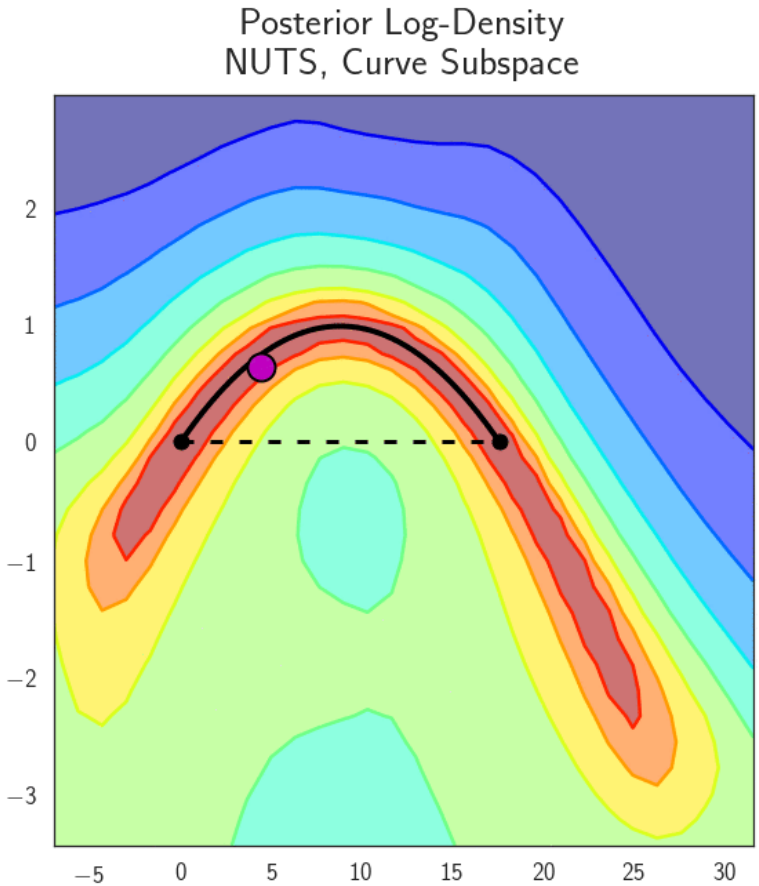
Curve Subspace Traversal



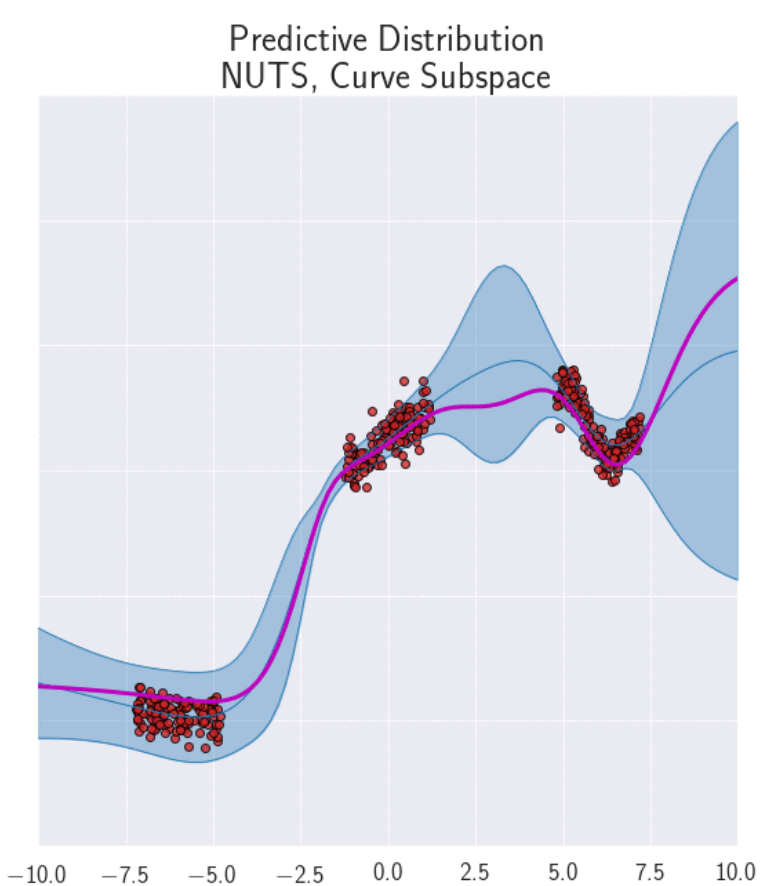


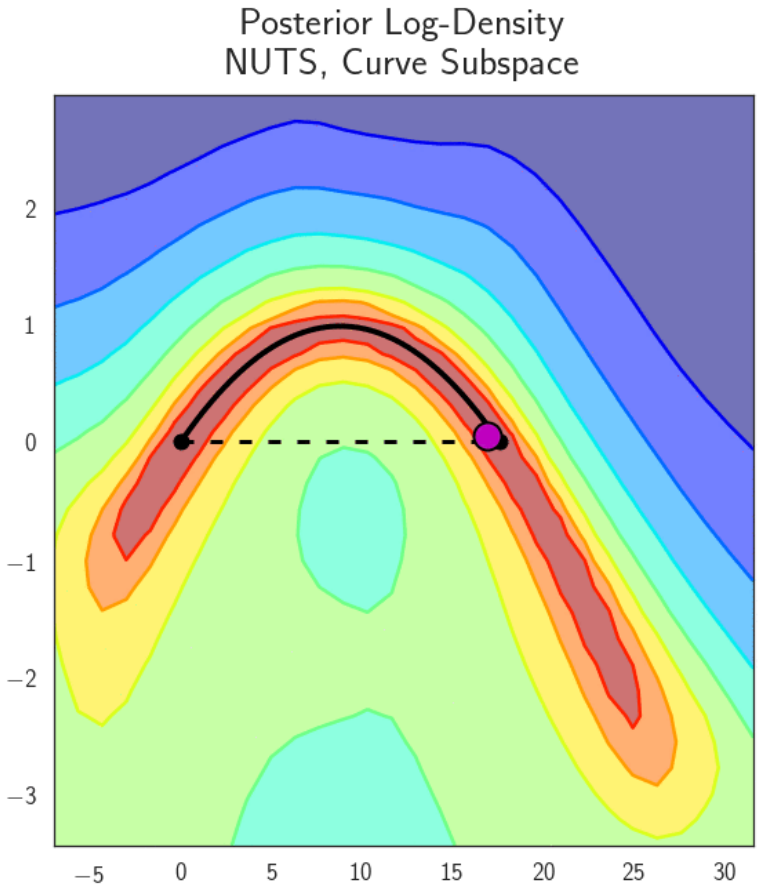
Curve Subspace Traversal



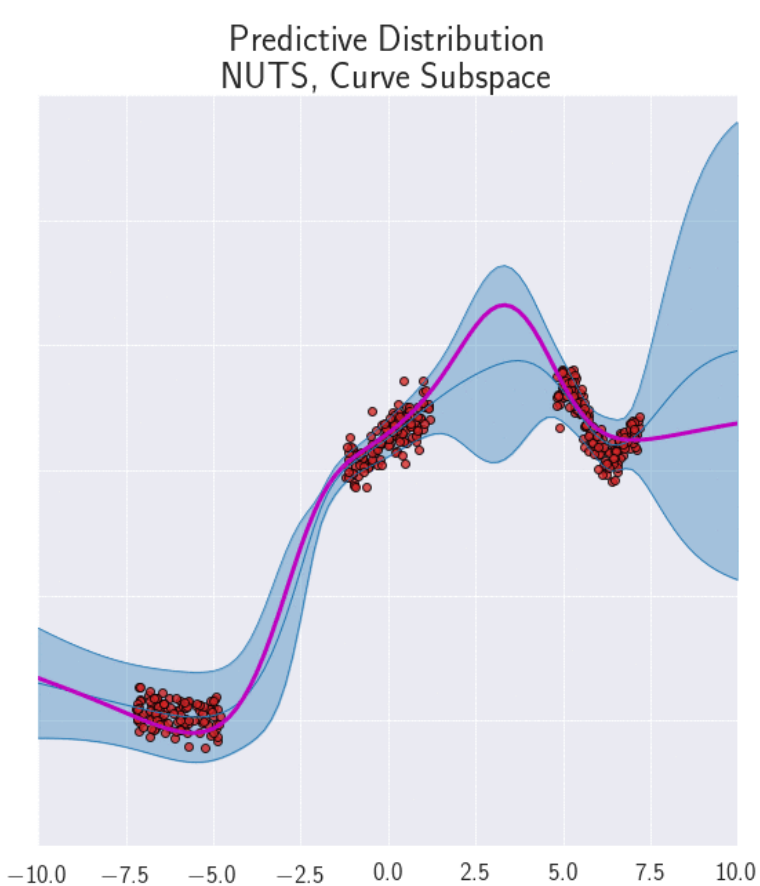


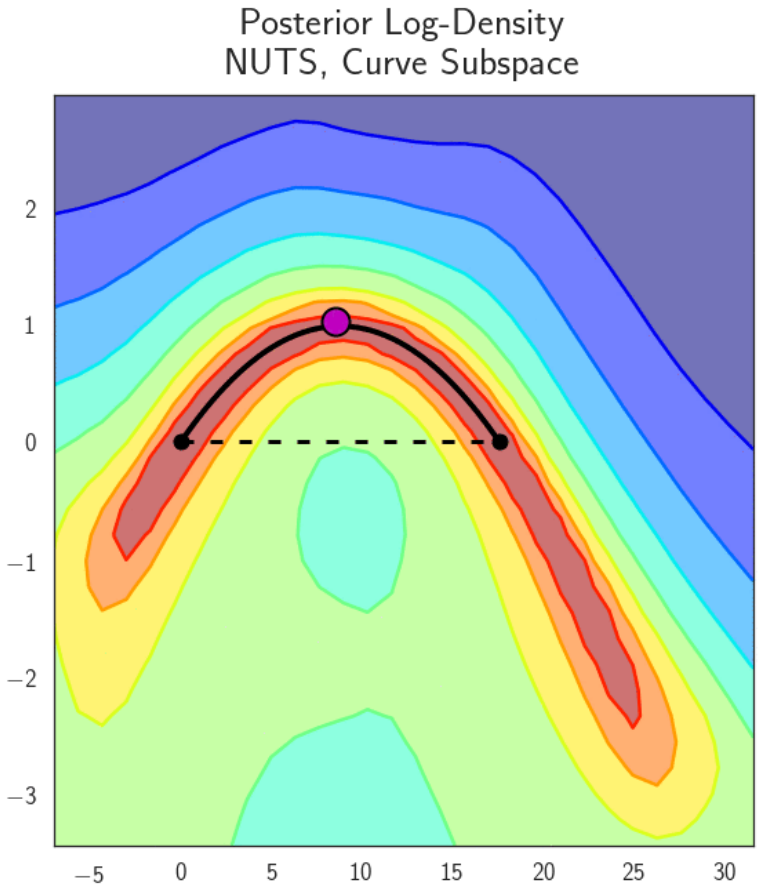
Curve Subspace Traversal



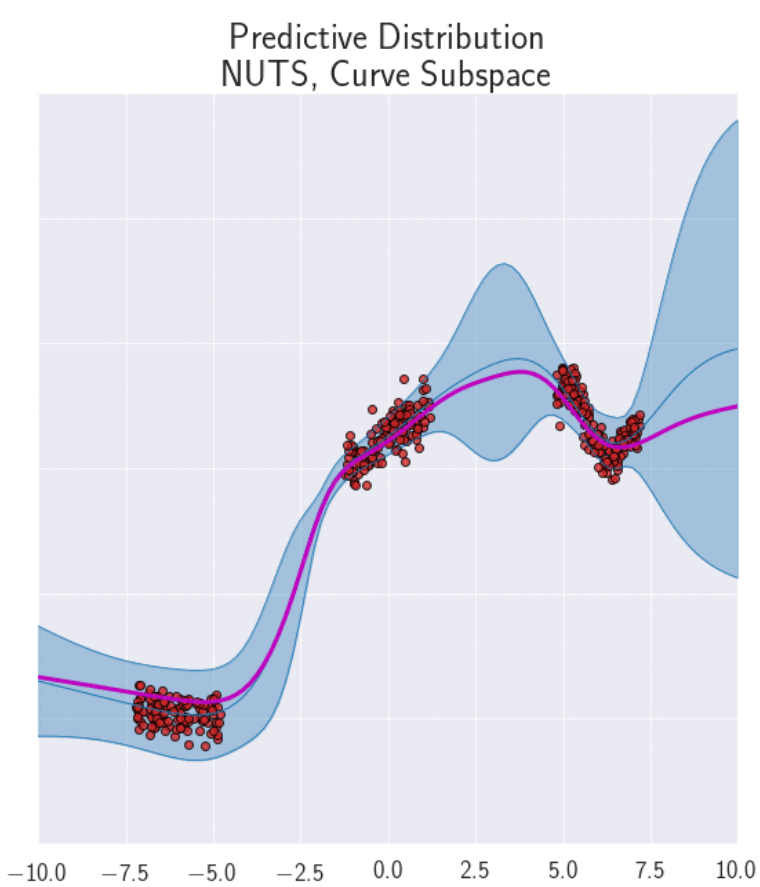


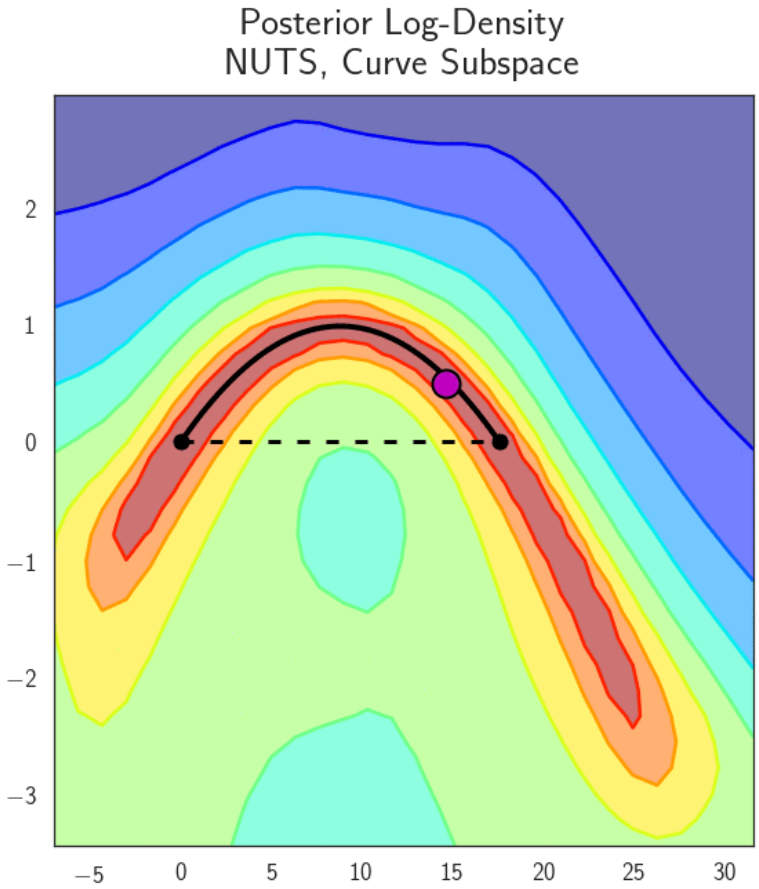
Curve Subspace Traversal



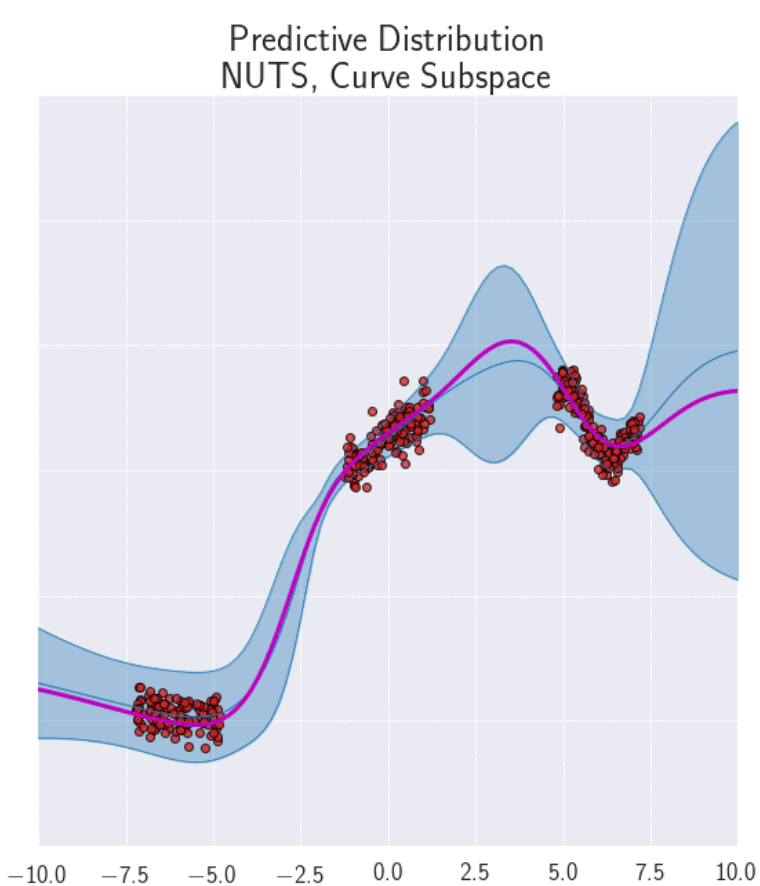


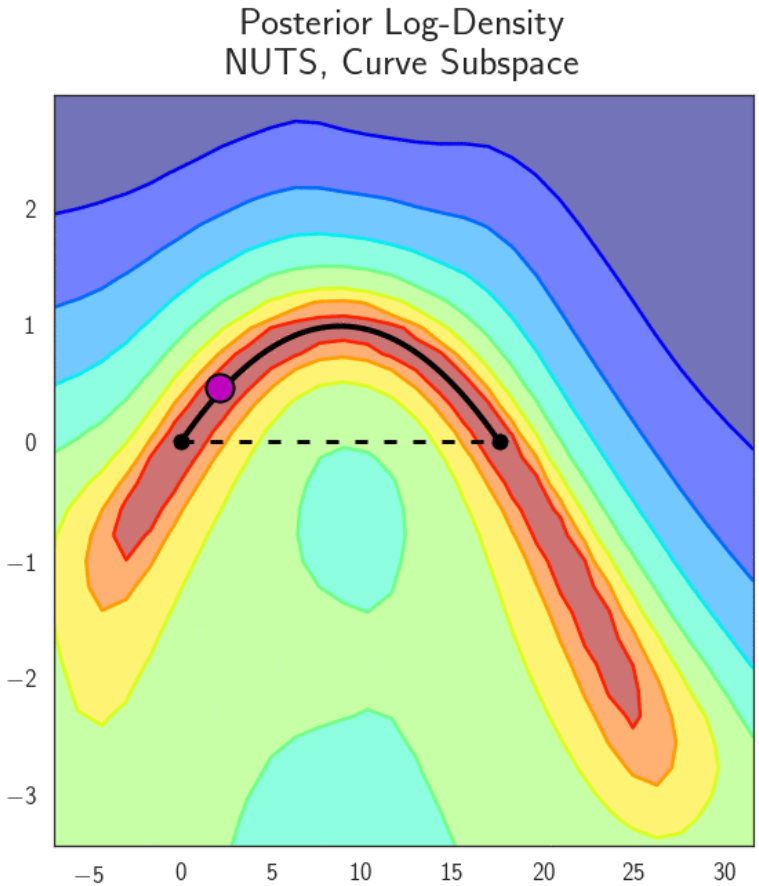
Curve Subspace Traversal



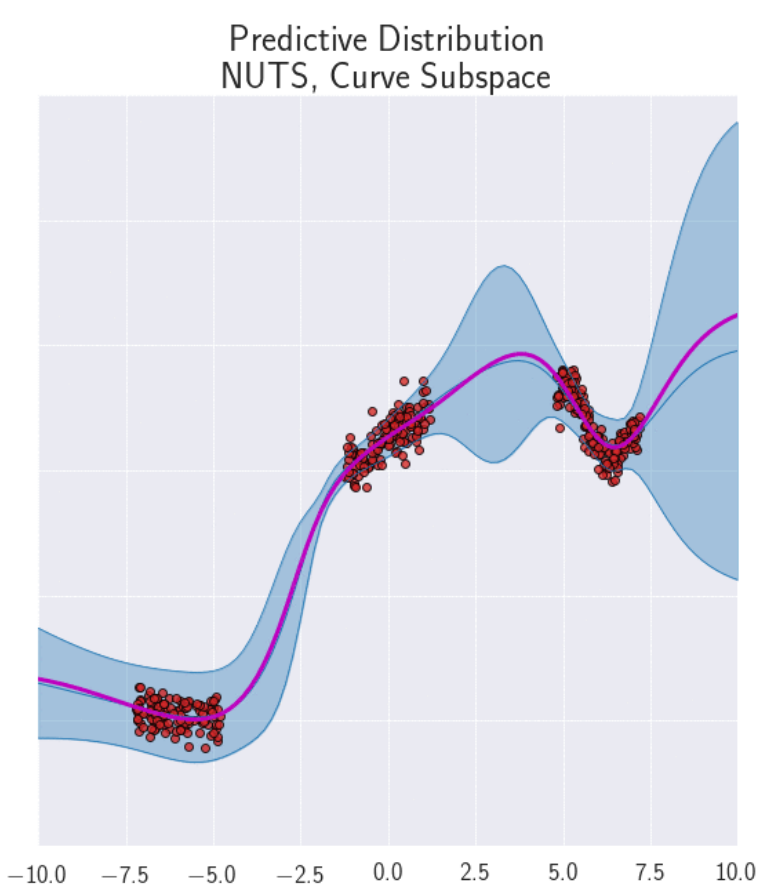


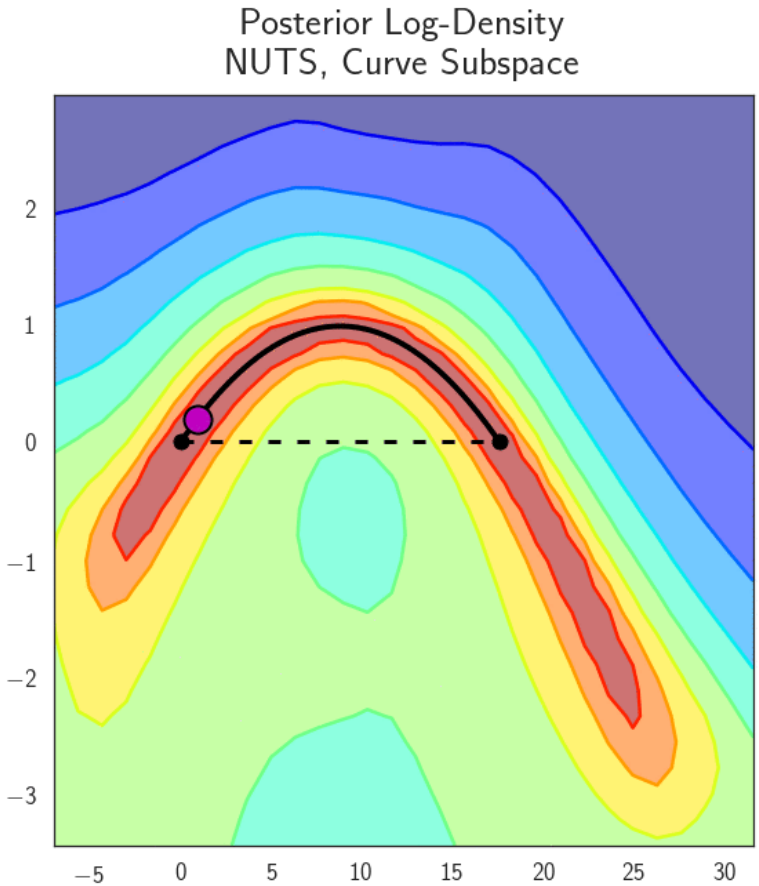
Curve Subspace Traversal



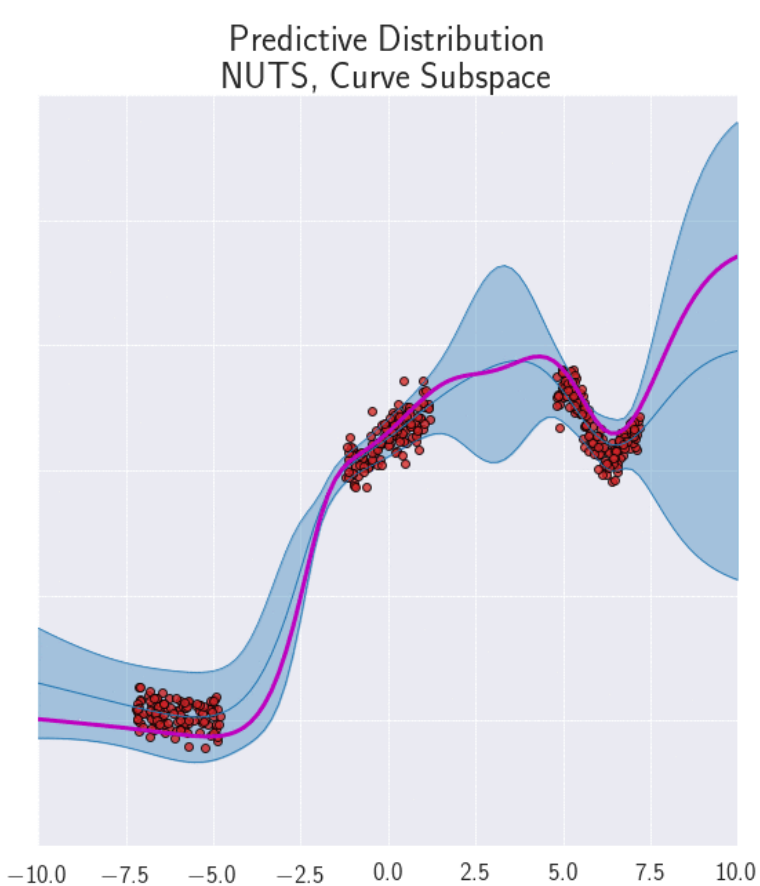


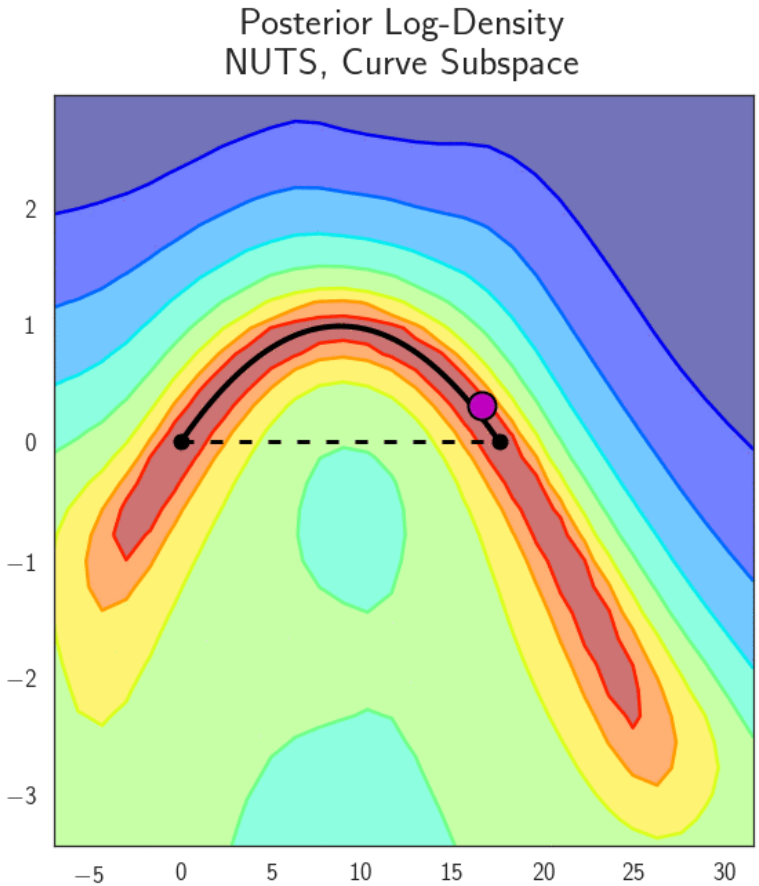
Curve Subspace Traversal



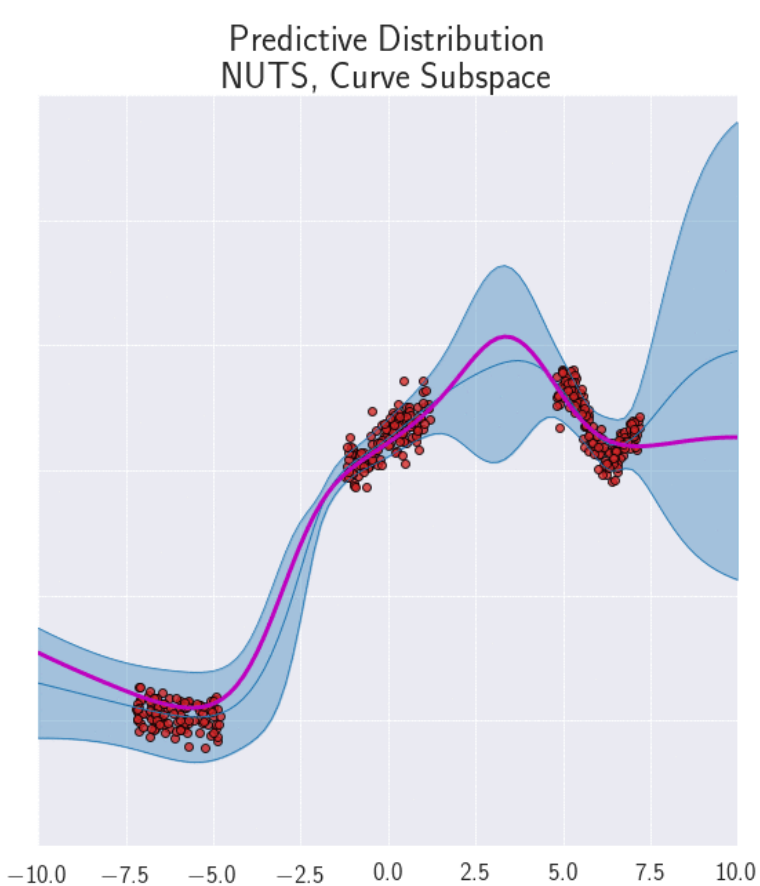


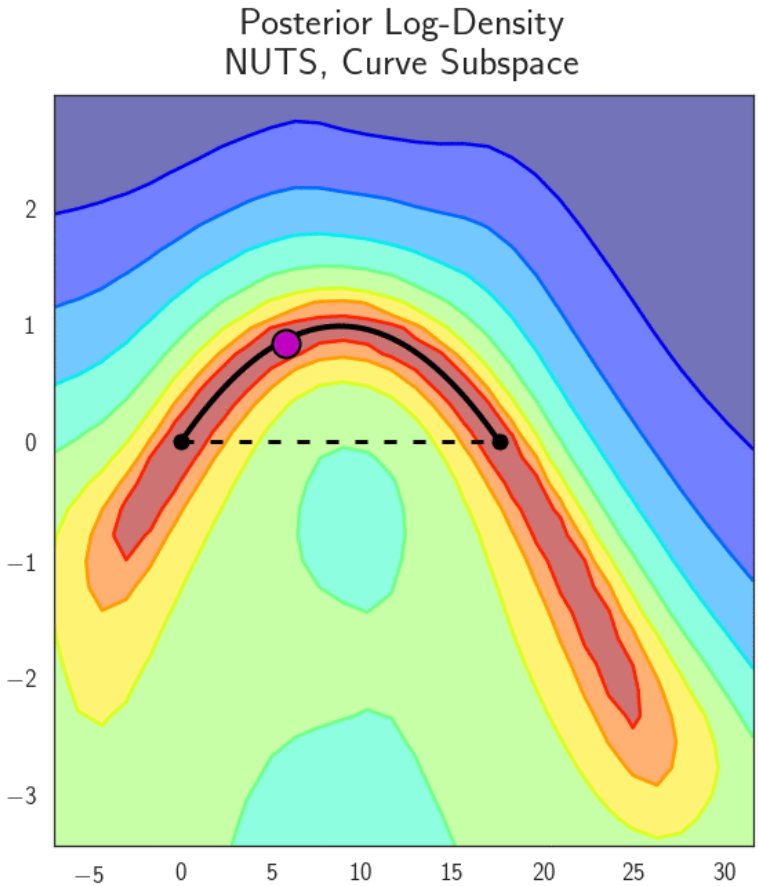
Curve Subspace Traversal



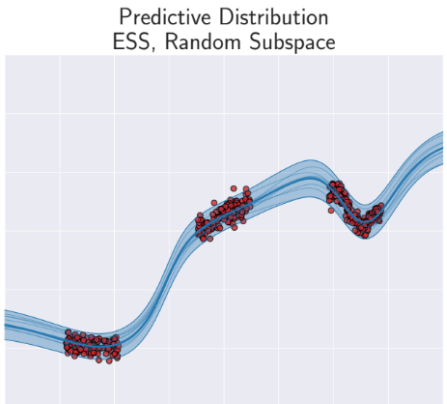


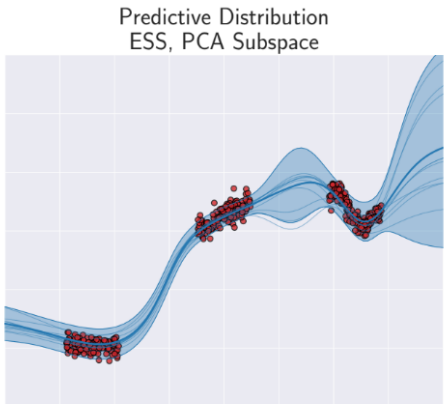
Curve Subspace Traversal

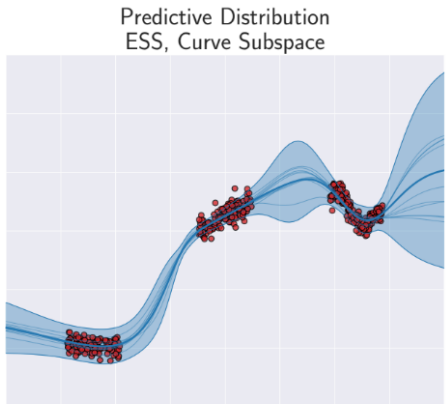


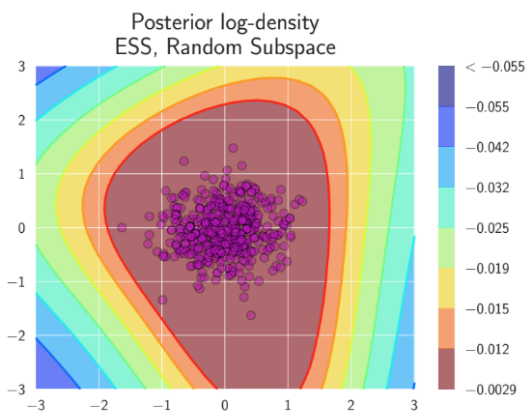


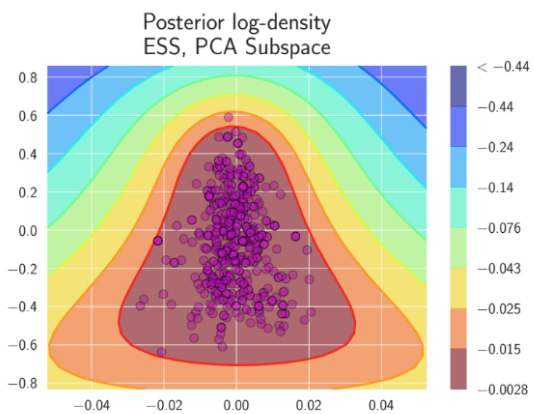
Subspace Comparison (Regression)

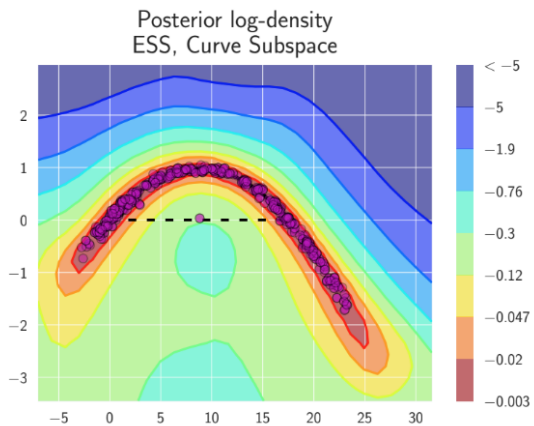






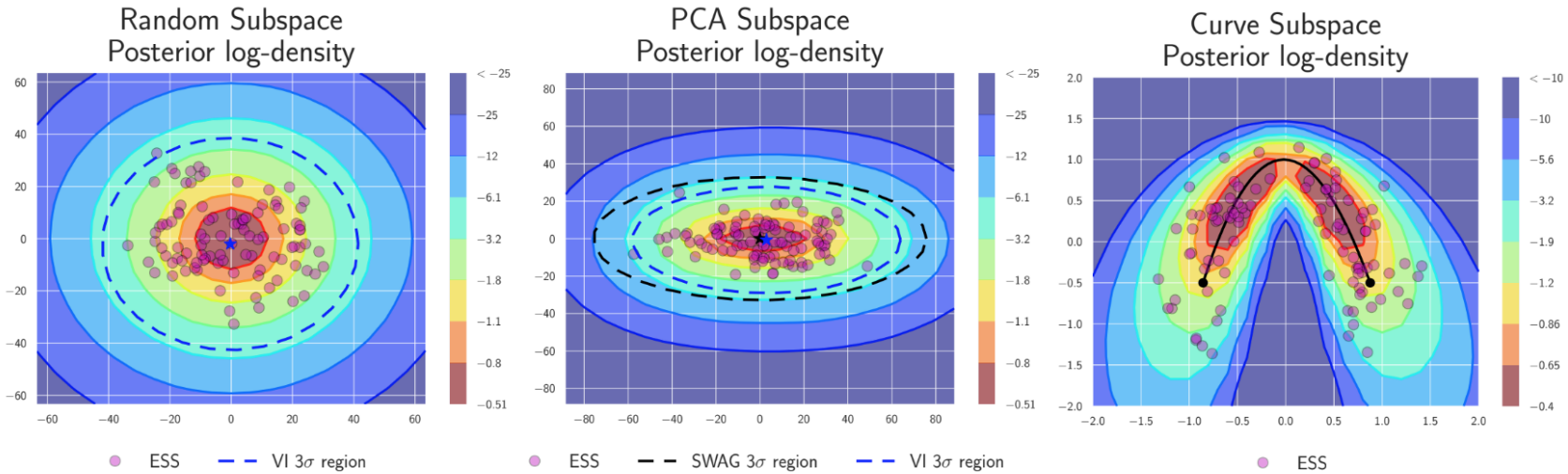






Subspace Comparison (Classification)

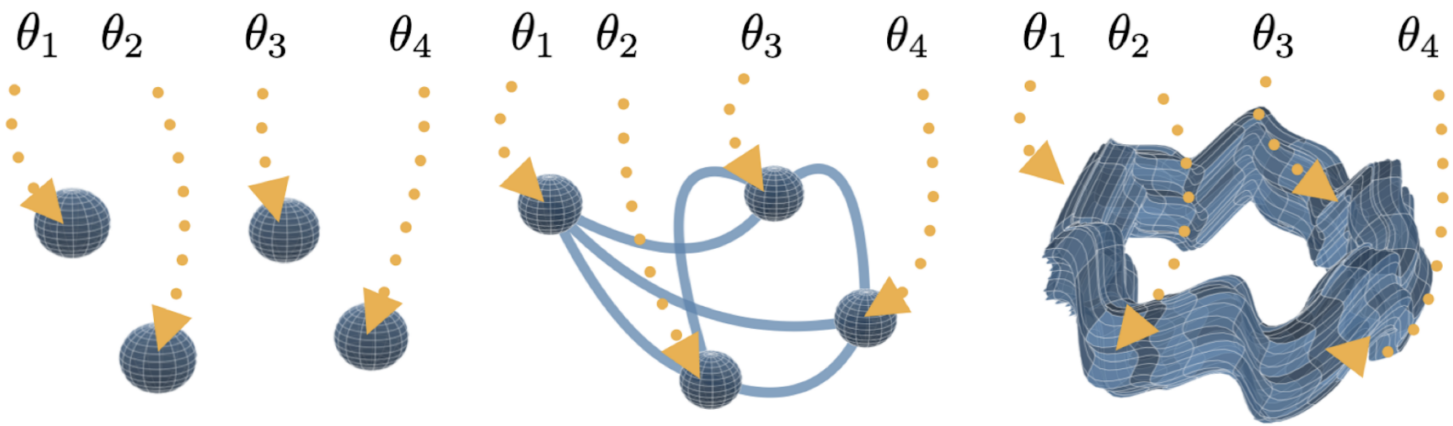
Accuracy and NLL on CIFAR-100



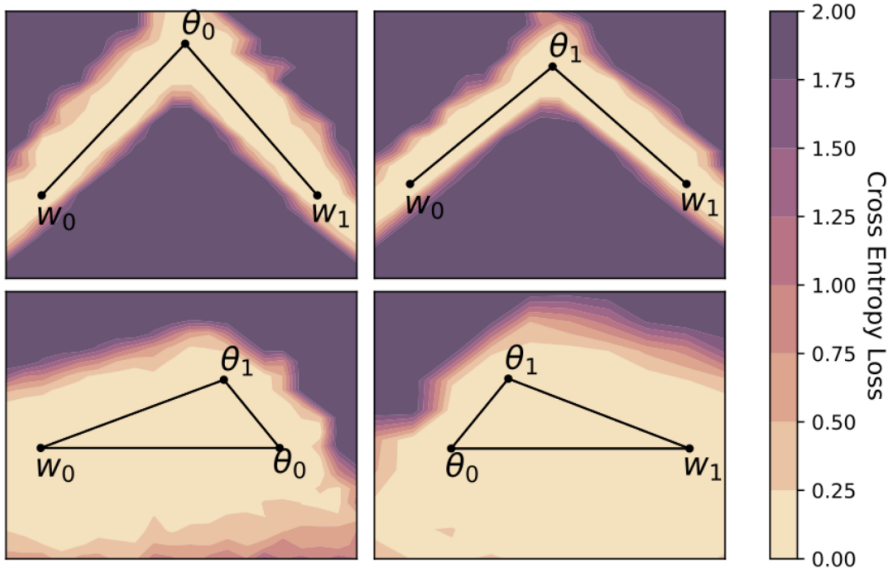
	SGD	Random	PCA	Curve
NLL	0.946 ± 0.001	0.686 ± 0.005	0.665 ± 0.004	0.646
Accuracy (%)	78.50 ± 0.32	80.17 ±0.03	80.54 ± 0.13	81.28

Bayesian methods also lead to better point predictions in deep learning!

Loss Surfaces: An *Evolution* of Understanding

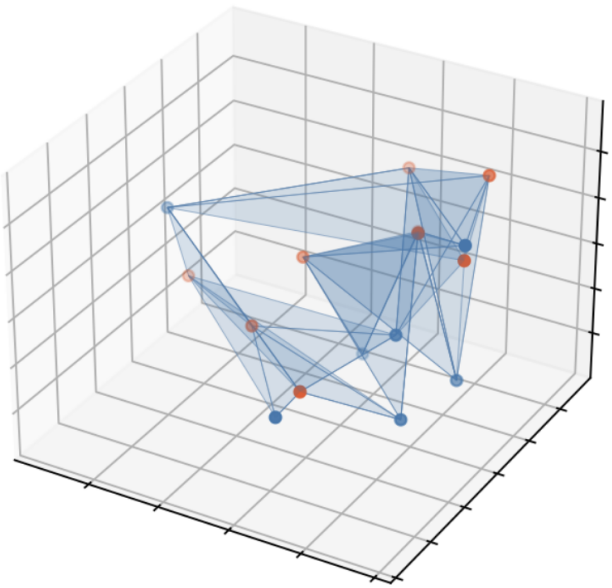


Mode Connecting Simplexes



- Form *simplicial complexes* in parameter space
- Add multiple connecting points such within each simplex of points we have low loss
- Here we have two modes (w_1, w_2) connected through two shared connecting points (θ_1, θ_2).

Building the Simplex



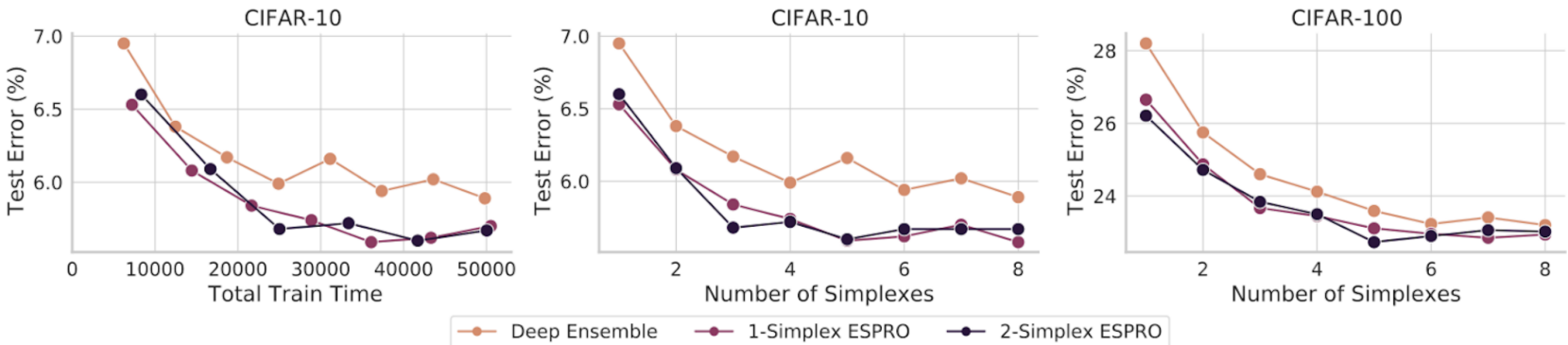
Finding connecting points is easy!

$$\mathcal{L}(\mathcal{K}) = \frac{1}{M} \sum_{\phi_m \sim \mathcal{K}} \mathcal{L}(\mathcal{D}, \phi_m) - \lambda_j \log V(\mathcal{K}) \qquad (5)$$

- First term: Loss over the simplicial complex, where ϕ_m are sampled uniformly from simplicial complex $\mathcal{K}$.
- Second term: Maximize the volume of the simplicial complex.
- *Simplicial Pointwise Random Optimization (SPRO).*

Loss Surface Simplexes for Mode Connecting Volumes and Fast Ensembling
G. Benton, W. Maddox, S. Lotfi, A.G. Wilson
ICML 2021

Empirical Results



Empirical Results