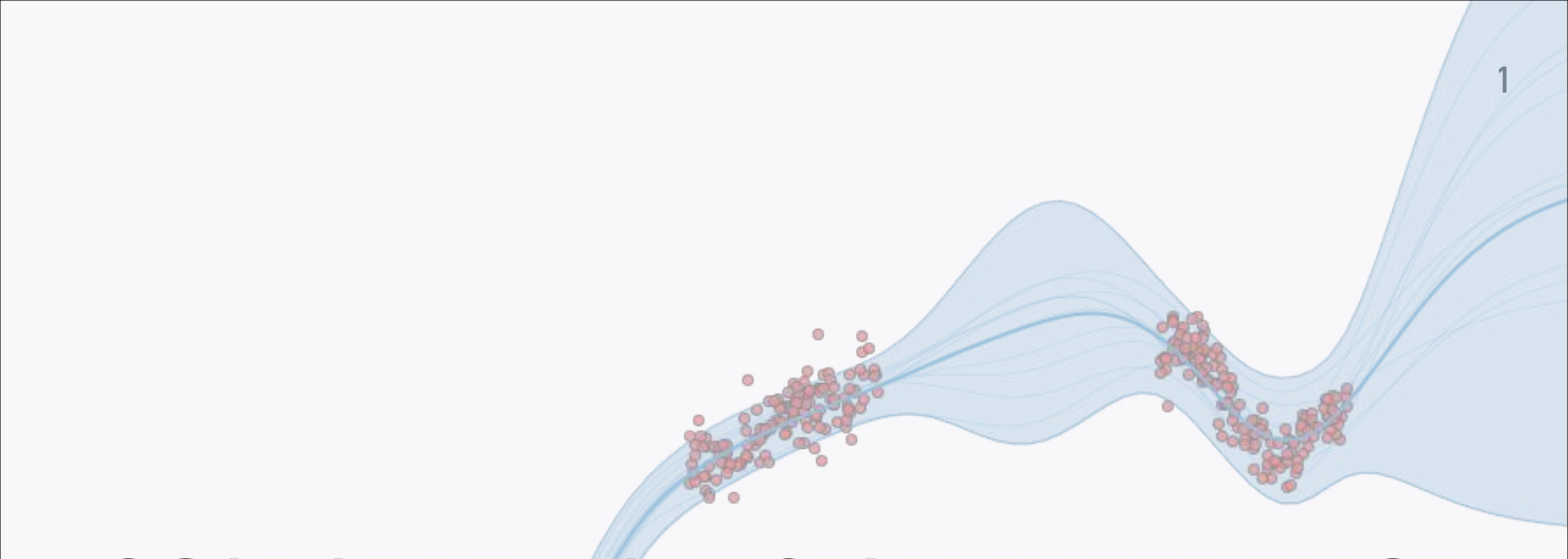


SCALABLE BAYESIAN INFERENCE IN LOW-DIMENSIONAL SUBSPACES

Polina Kirichenko Pavel Izmailov

New York University

PLAN

- **Subspace Inference for Bayesian Deep Learning**
  Pavel Izmailov, Wesley Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, Andrew Gordon Wilson
  UAI 2019

- **Projected BNNs: Avoiding weight-space pathologies by learning latent representations of neural network weights**
  Melanie F. Pradier, Weiwei Pan, Jiayu Yao, Soumya Ghosh, Finale Doshi-Velez
  NeurIPS 2018 BDL Workshop

WHY BAYESIAN INFERENCE?

- Combining models for better predictions 📊
- Uncertainty representation (crucial for decision making) 🤷‍♀️
- Interpretably incorporate prior knowledge and domain expertise 👩‍⚕️

WHY BAYESIAN INFERENCE?

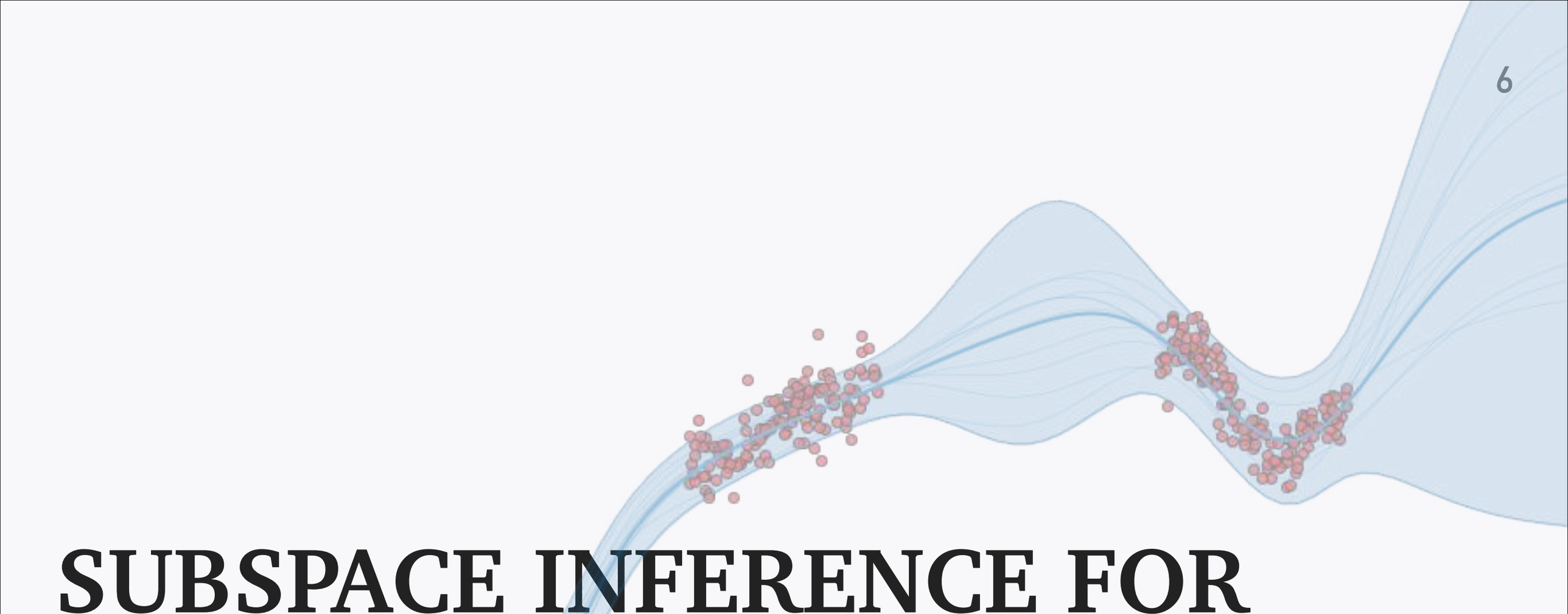
- Combining models for better predictions 📊
- Uncertainty representation (crucial for decision making) 🤷‍♀️
- Interpretably incorporate prior knowledge and domain expertise 👩‍⚕️

WHY NOT?

- Challenging for Deep NNs due to high dimensional weight spaces 😩

IDEA

- Learn low-dimensional representations z for model's weights w
- Perform inference over z
- Bayesian Model Averaging by sampling z, and transforming $z \rightarrow w$



SUBSPACE INFERENCE FOR BAYESIAN DEEP LEARNING

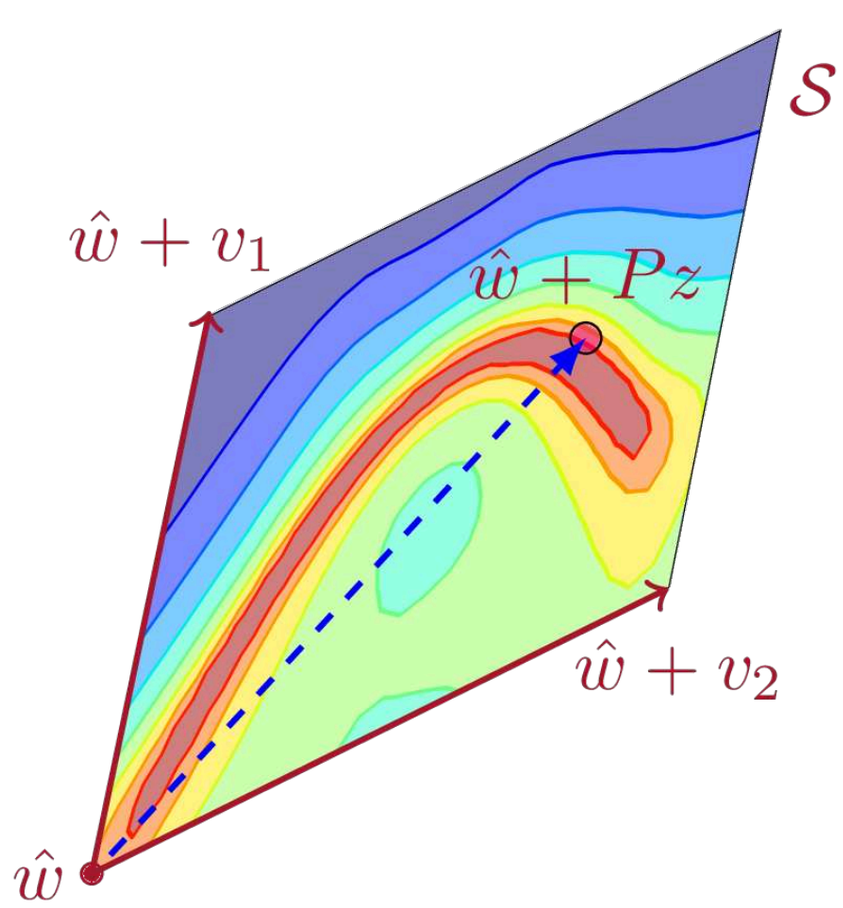
PAVEL IZMAILOV, WESLEY MADDOX, POLINA KIRICHENKO,
TIMUR GARIPOV, DMITRY VETROV, ANDREW GORDON WILSON

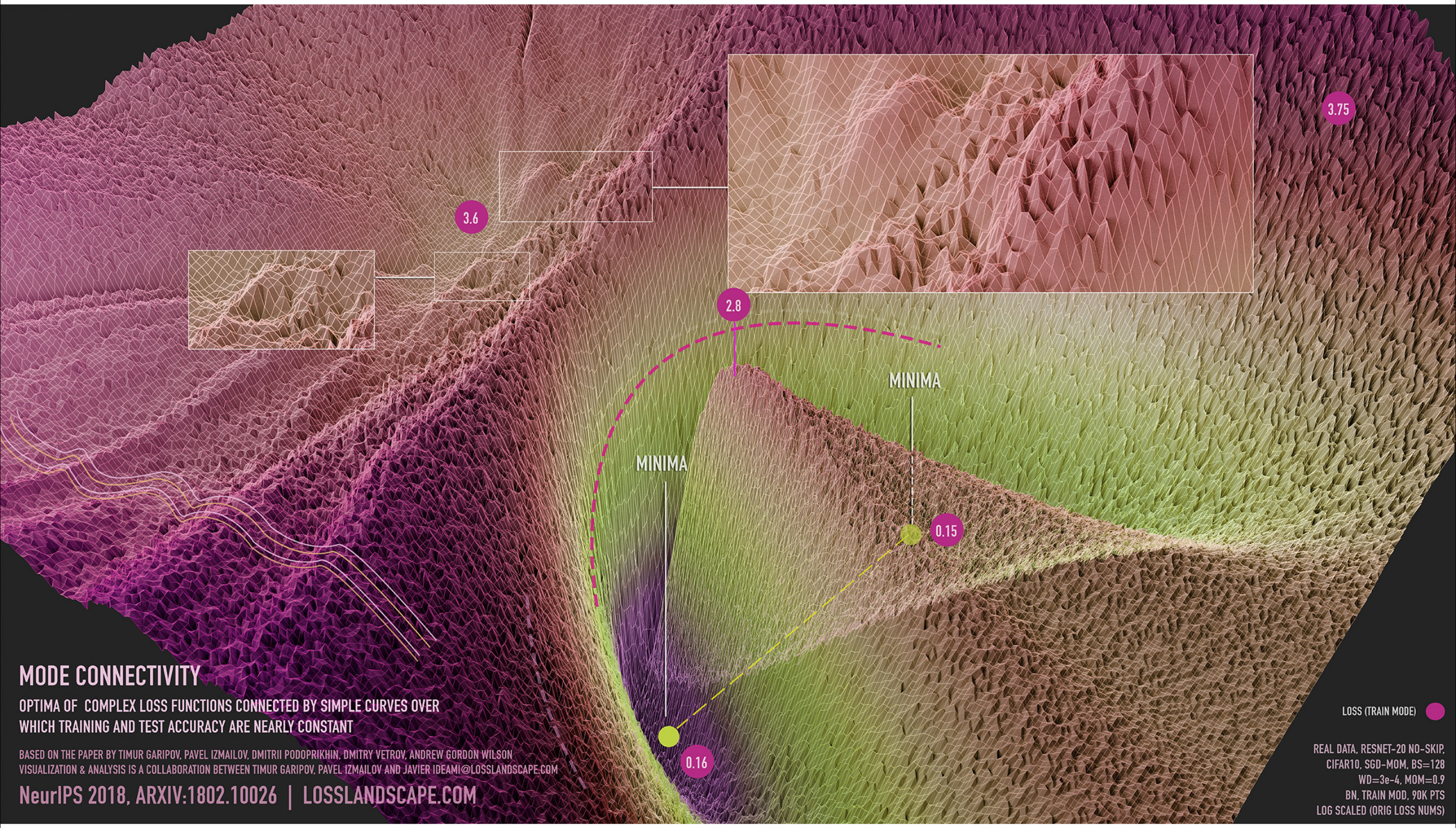
p(B|A)yesgroup.ru

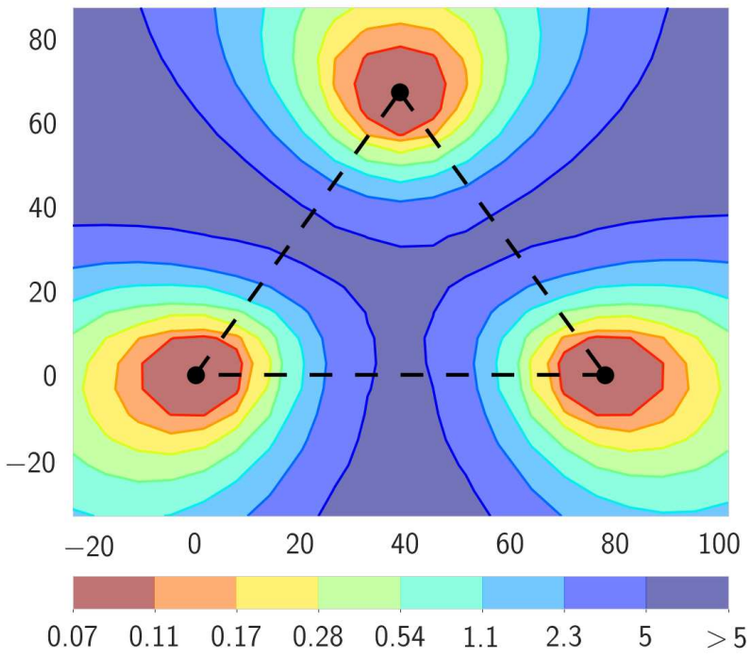
NYU

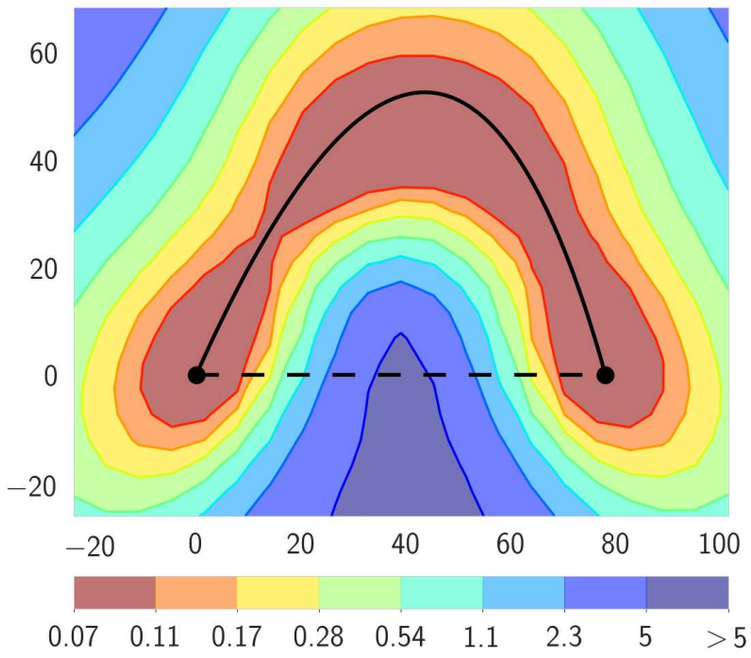
SUBSPACE INFERENCE

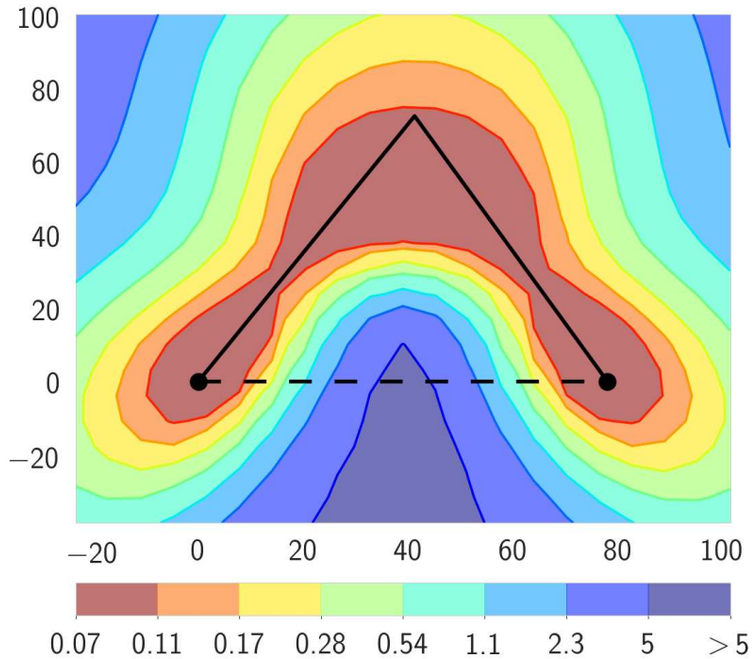
- Motivation: capture *geometry* of the posterior in *interesting* low-dimensional subspaces of the parameter space




3.6
3.75
2.8
MINIMA
MINIMA
0.15
0.16
MODE CONNECTIVITY
OPTIMA OF COMPLEX LOSS FUNCTIONS CONNECTED BY SIMPLE CURVES OVER
WHICH TRAINING AND TEST ACCURACY ARE NEARLY CONSTANT
BASED ON THE PAPER BY TIMUR GARIPOV, PAVEL IZMAILOV, DMITRII PODOPRIKHIN, DMITRY VETROV, ANDREW GORDON WILSON
VISUALIZATION & ANALYSIS IS A COLLABORATION BETWEEN TIMUR GARIPOV, PAVEL IZMAILOV AND JAVIER IDEAMI@LOSSLANDSCAPE.COM
NeurIPS 2018, ARXIV:1802.10026 | LOSSLANDSCAPE.COM
LOSS (TRAIN MODE)
REAL DATA, RESNET-20 NO-SKIP,
CIFAR10, SGD-MOM, BS=128
WD=3e-4, MOM=0.9
BN, TRAIN MOD, 90K PTS
LOG SCALED (ORIG LOSS NUMS)





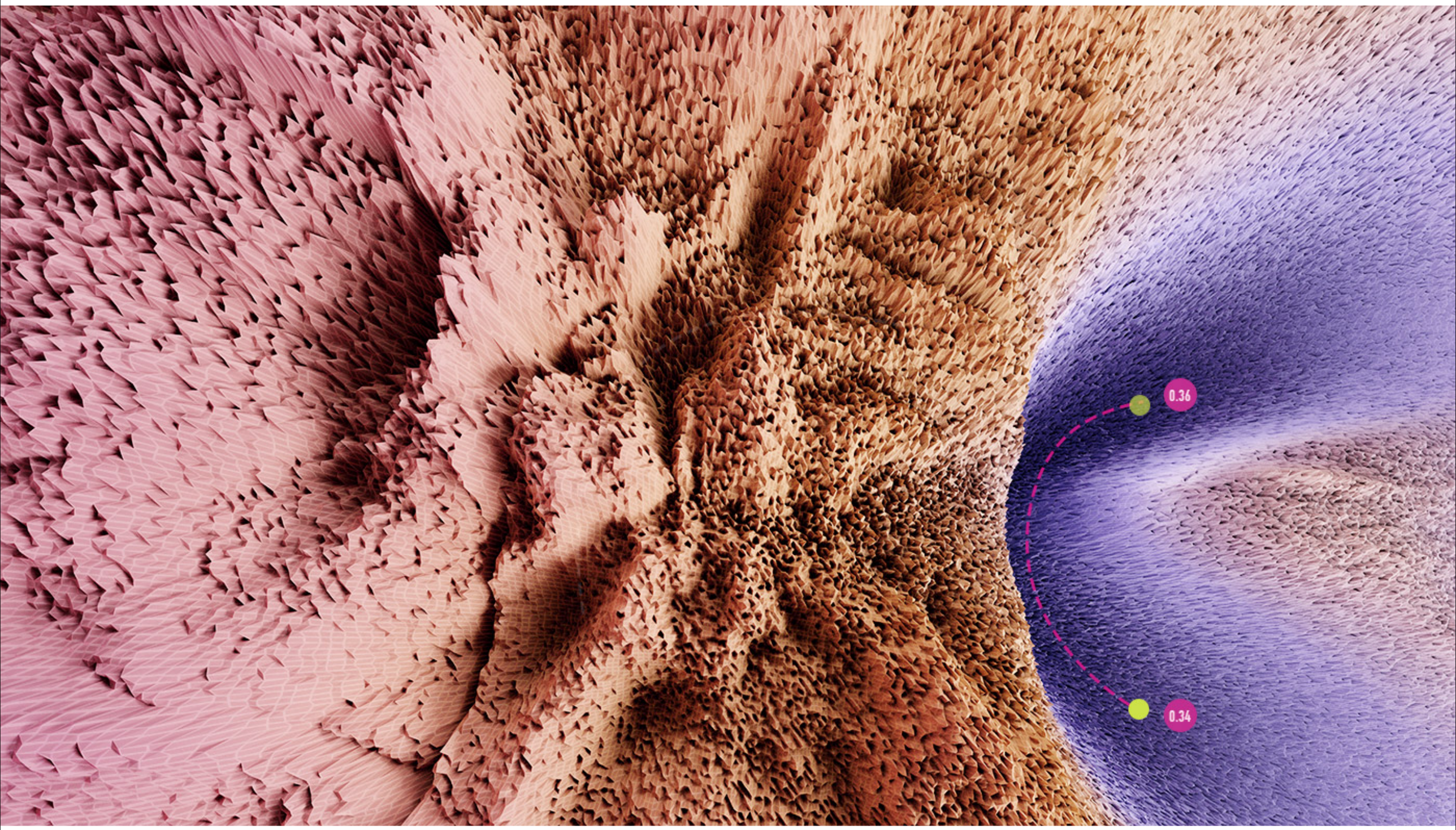


- Weights of pretrained networks: $\widehat{w}_1, \widehat{w}_2 \in \mathbb{R}^{|net|}$
- Define parametric curve: $\phi_\theta(\cdot) : [0, 1] \to \mathbb{R}^{|net|}$

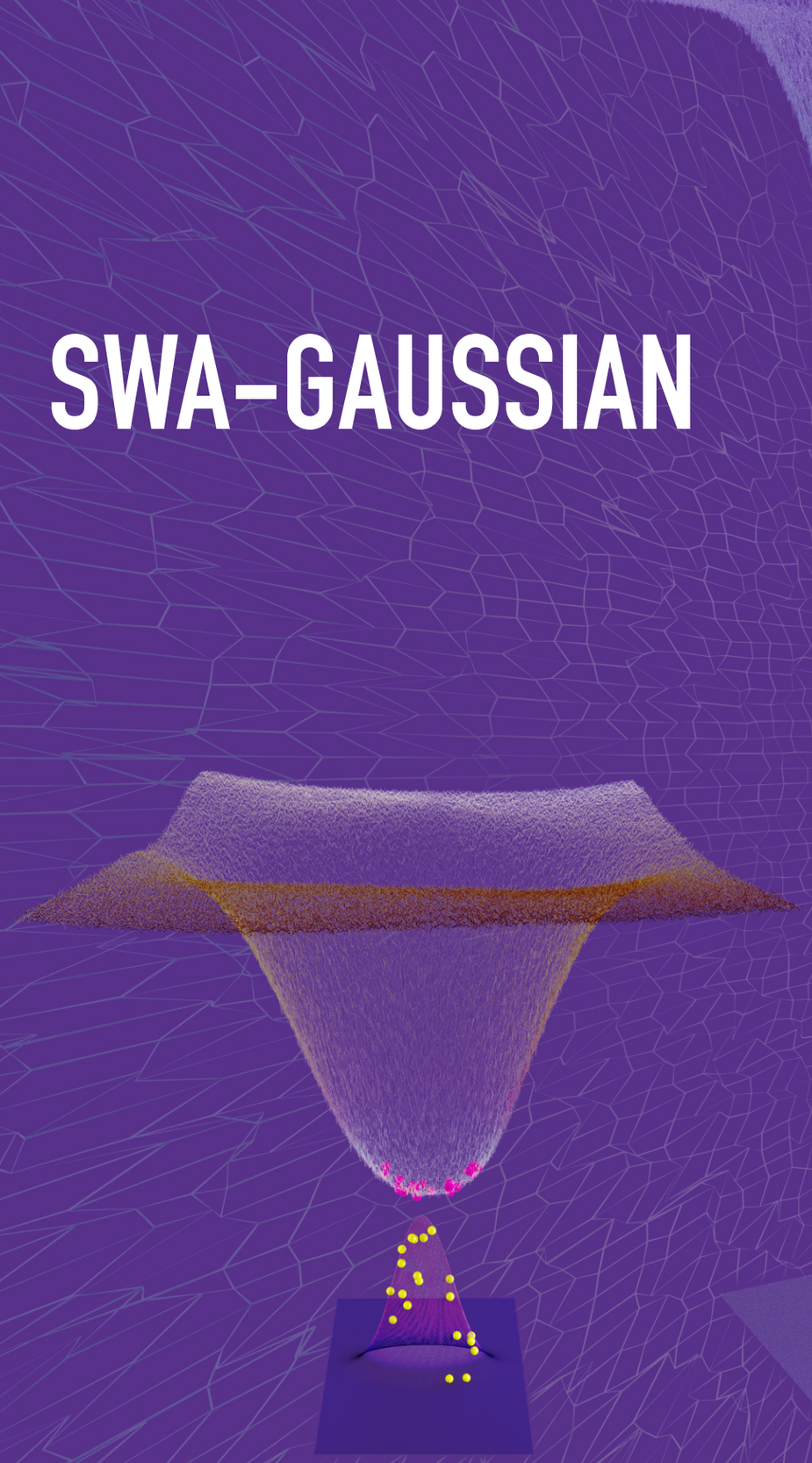
$$\phi_\theta(0) = \widehat{w}_1, \quad \phi_\theta(1) = \widehat{w}_2$$

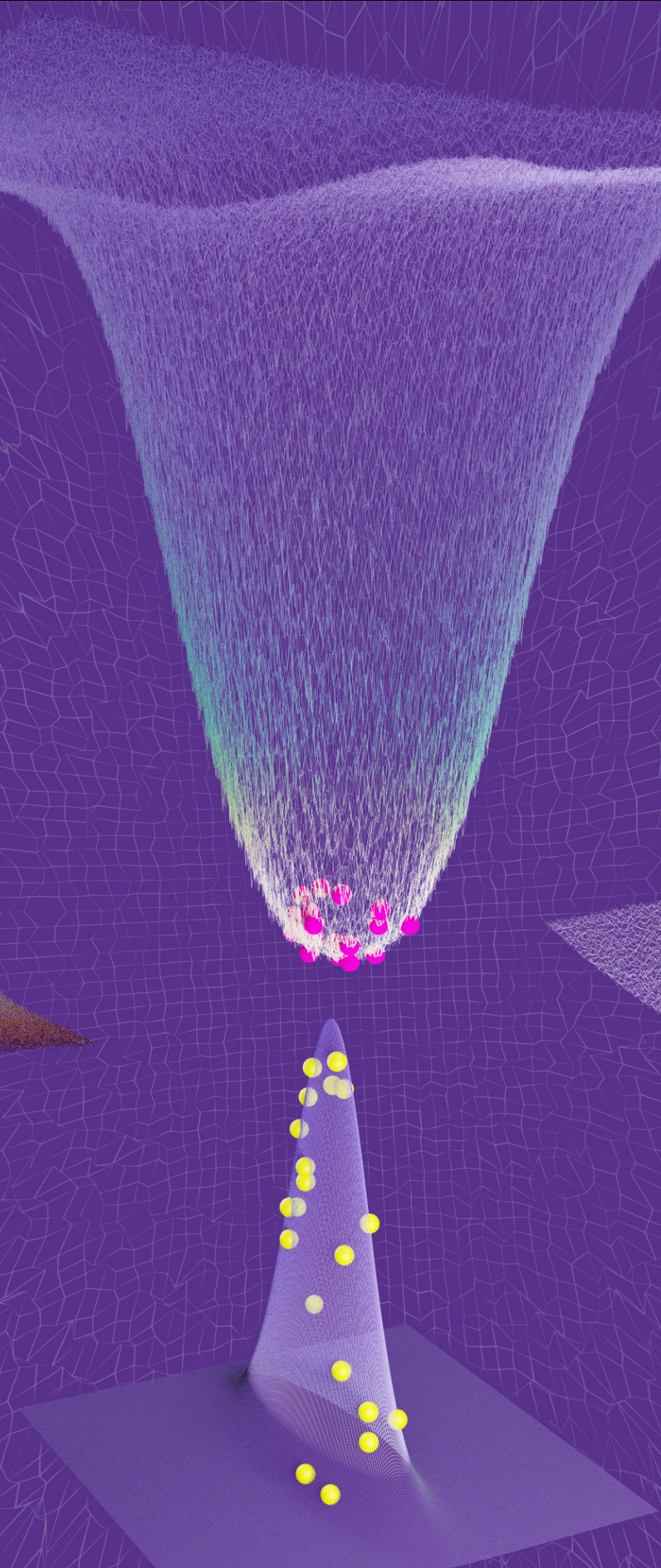
- DNN loss function: $\mathcal{L}(w)$
- Minimize averaged loss w.r.t. θ

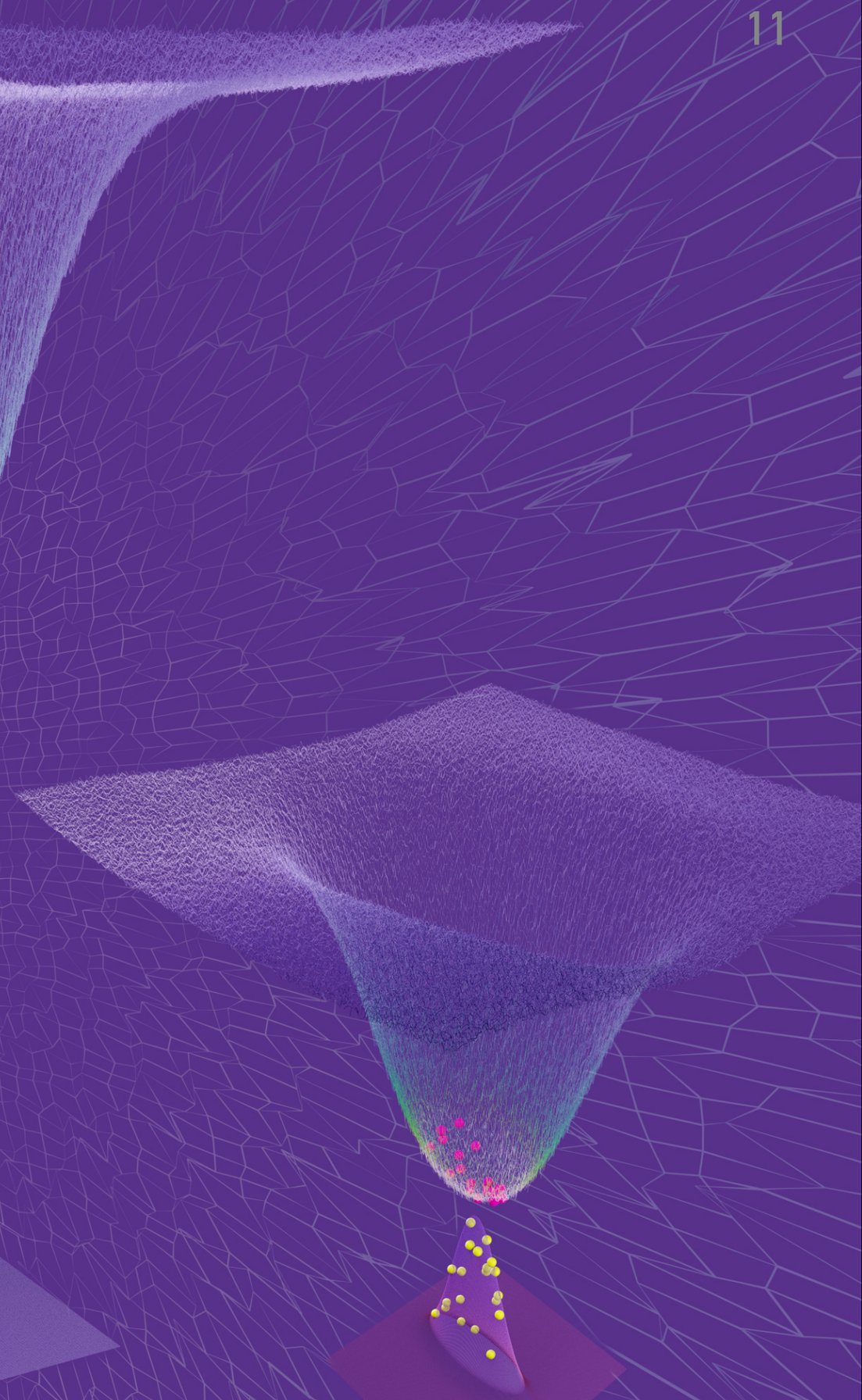
$$\underset{\theta}{\text{minimize}} \ \ell(\theta) = \int_0^1 \mathcal{L}(\phi_\theta(t))dt = \mathbb{E}_{t\sim U(0,1)}\mathcal{L}(\phi_\theta(t))$$


0.36
0.34

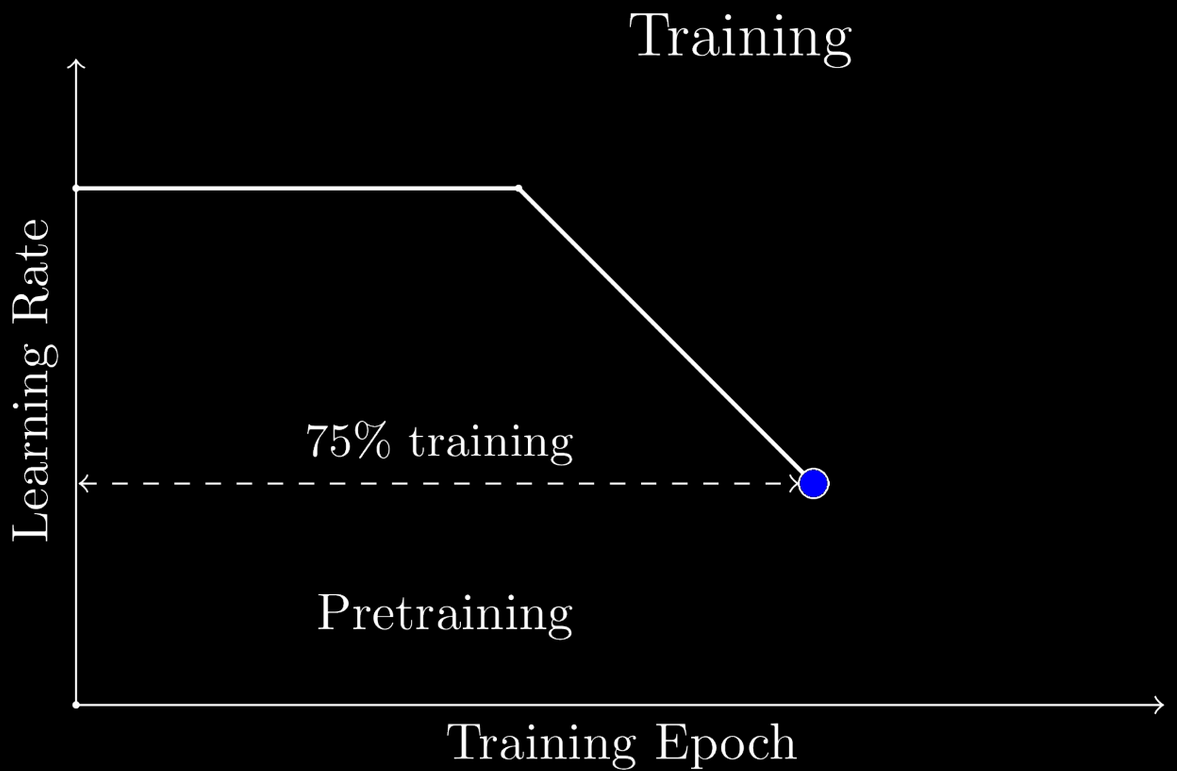
SWA-GAUSSIAN

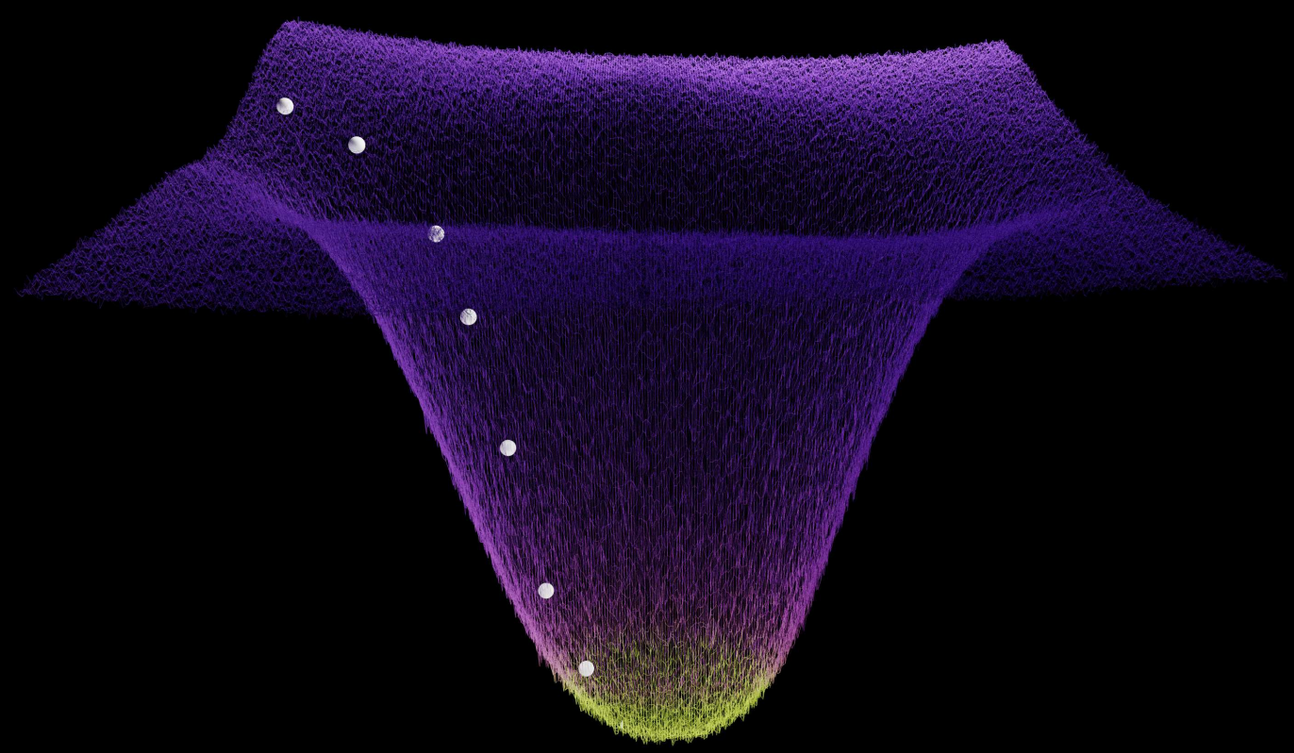






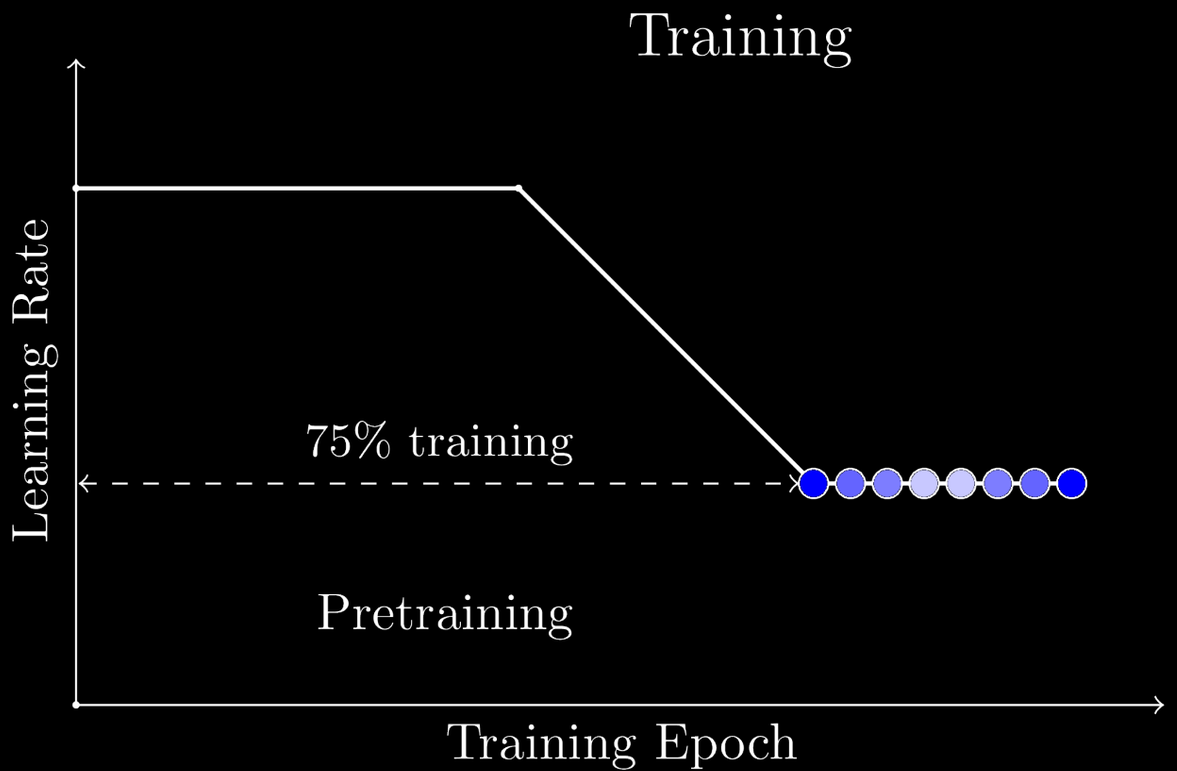
SWAG

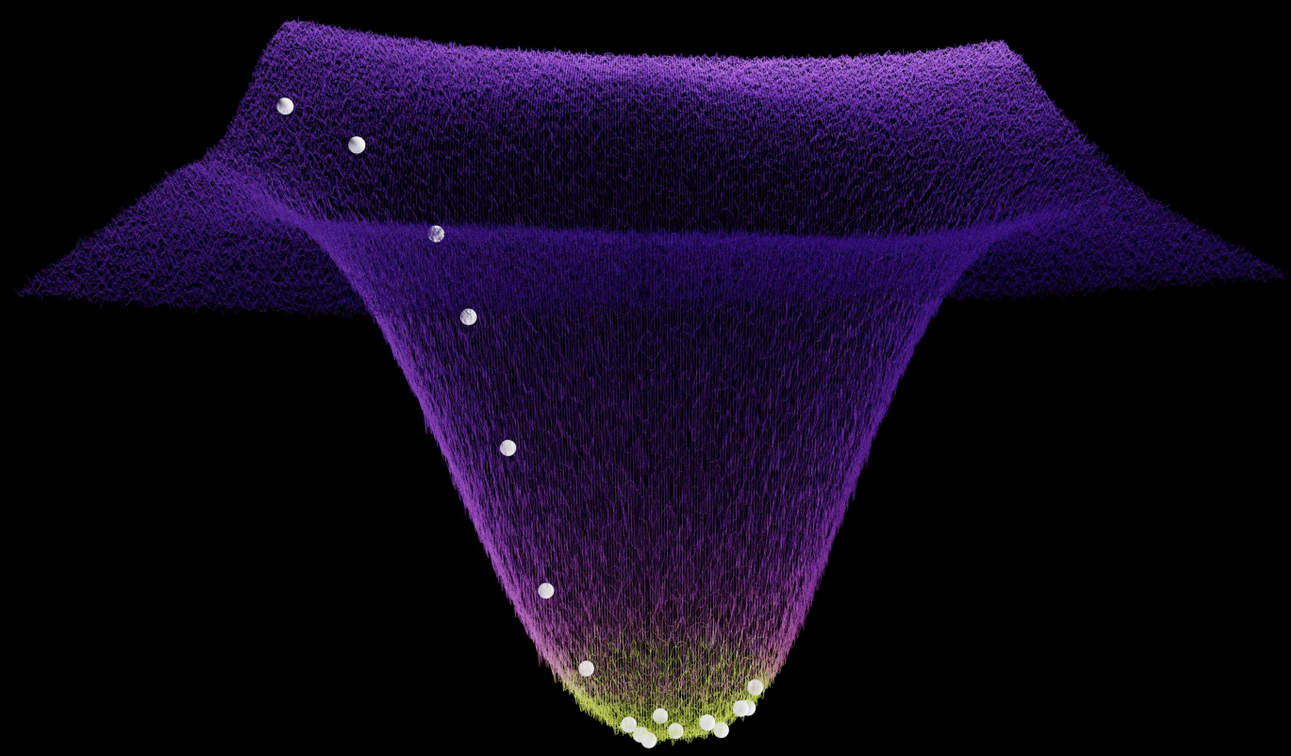




Visualization created in collaboration with Javier Ideami (https://losslandscape.com)

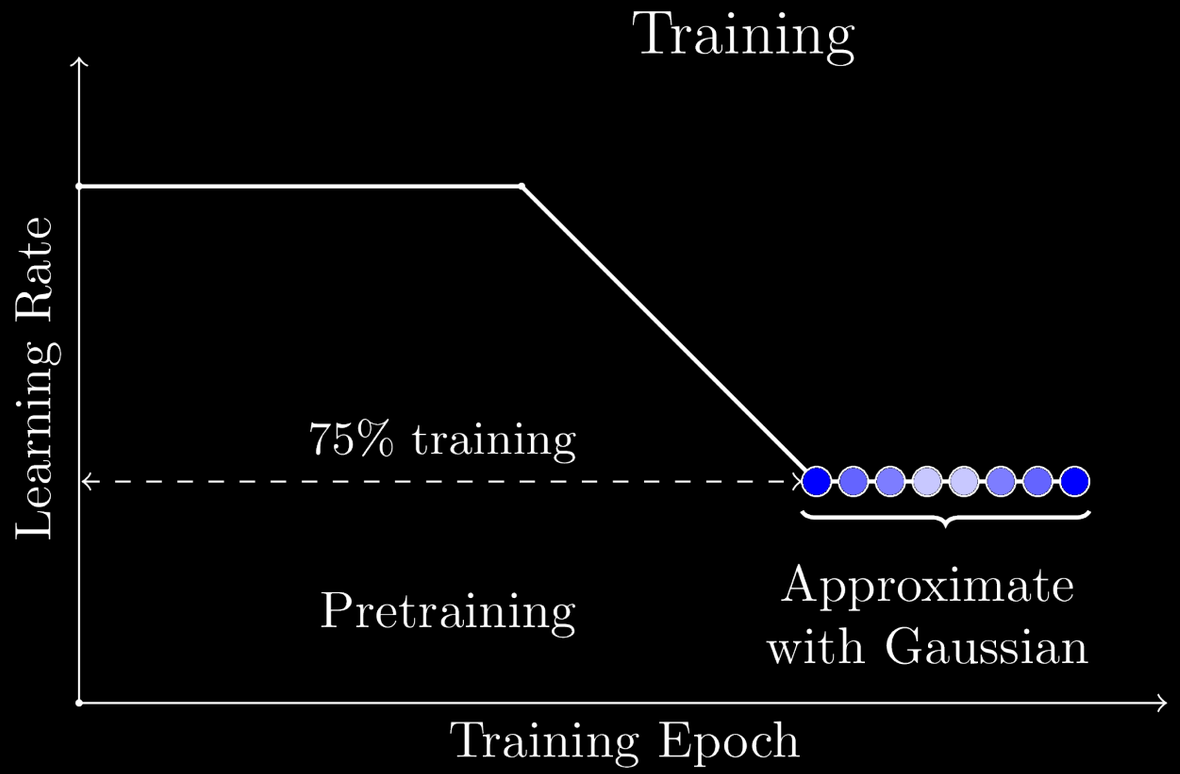
SWAG

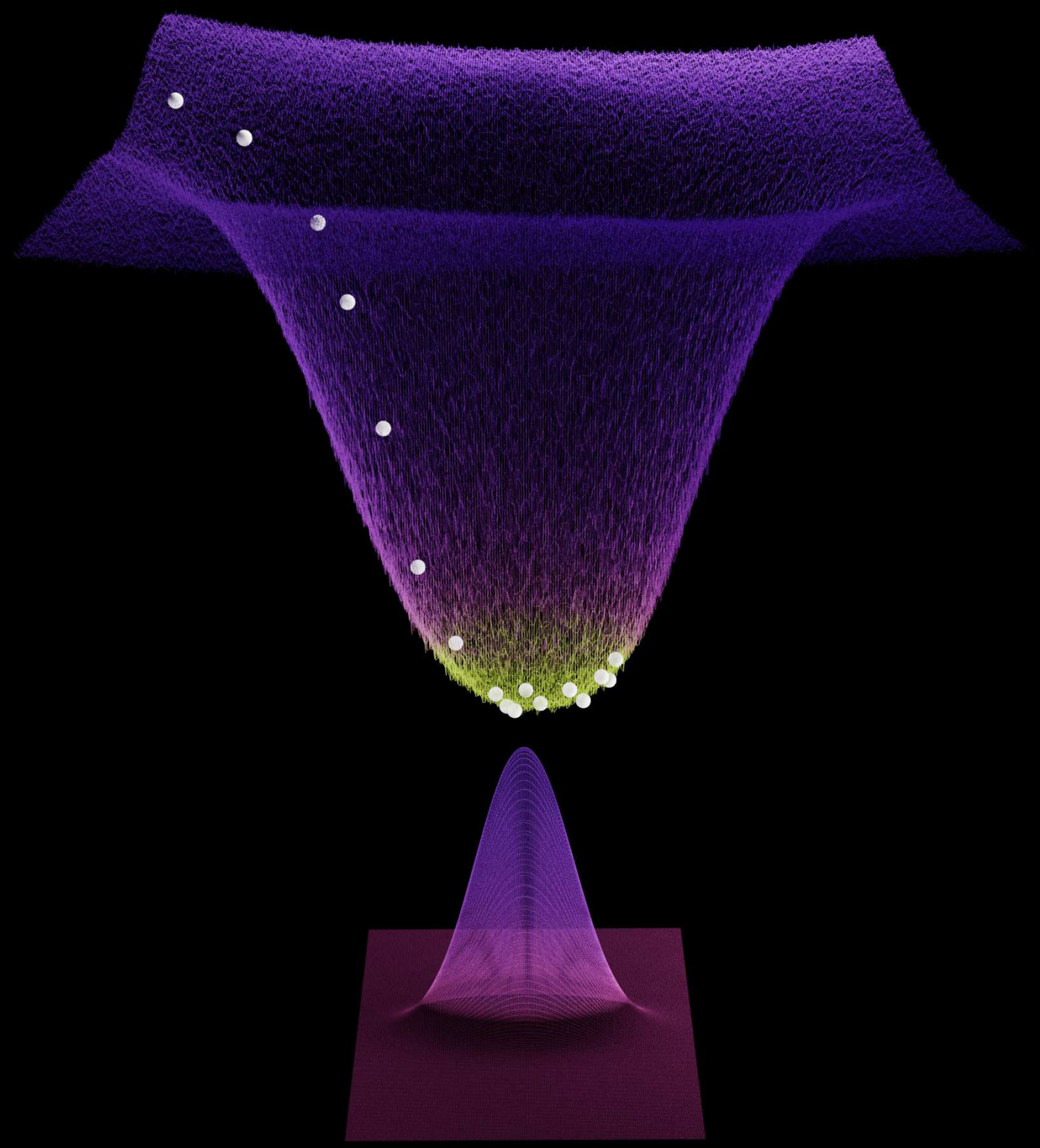




Visualization created in collaboration with Javier Ideami (https://losslandscape.com)

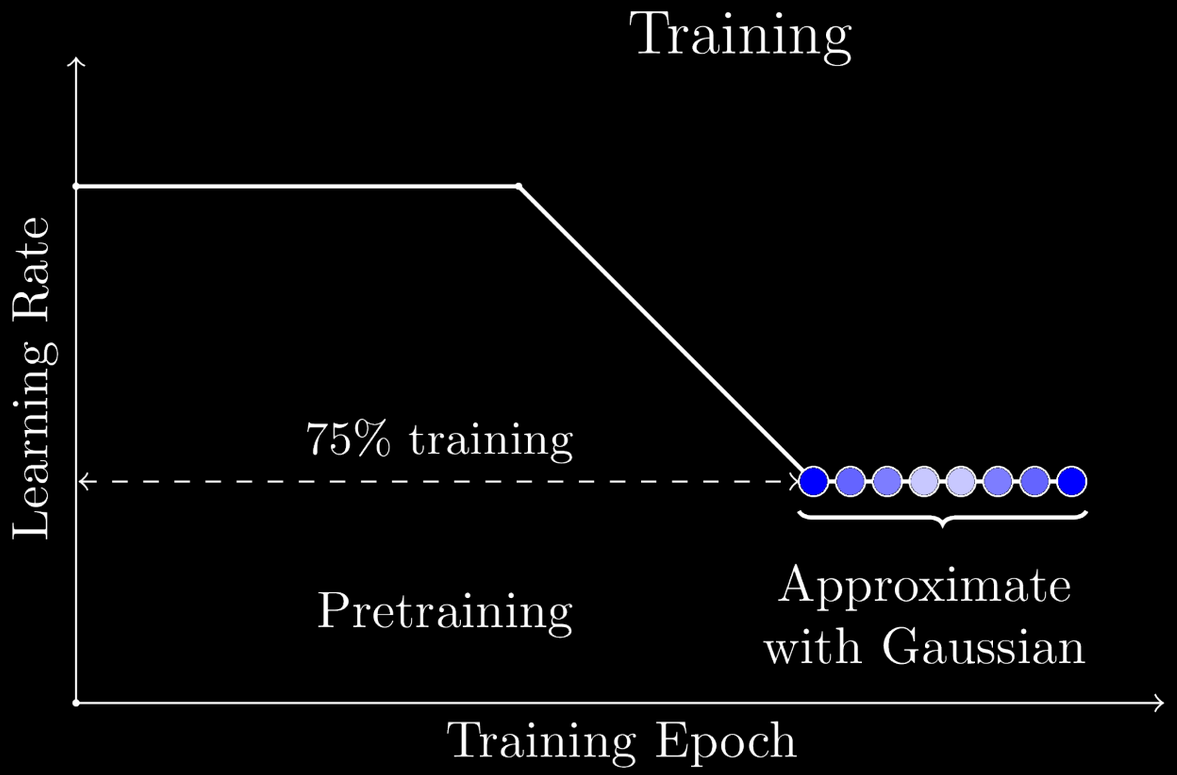
SWAG

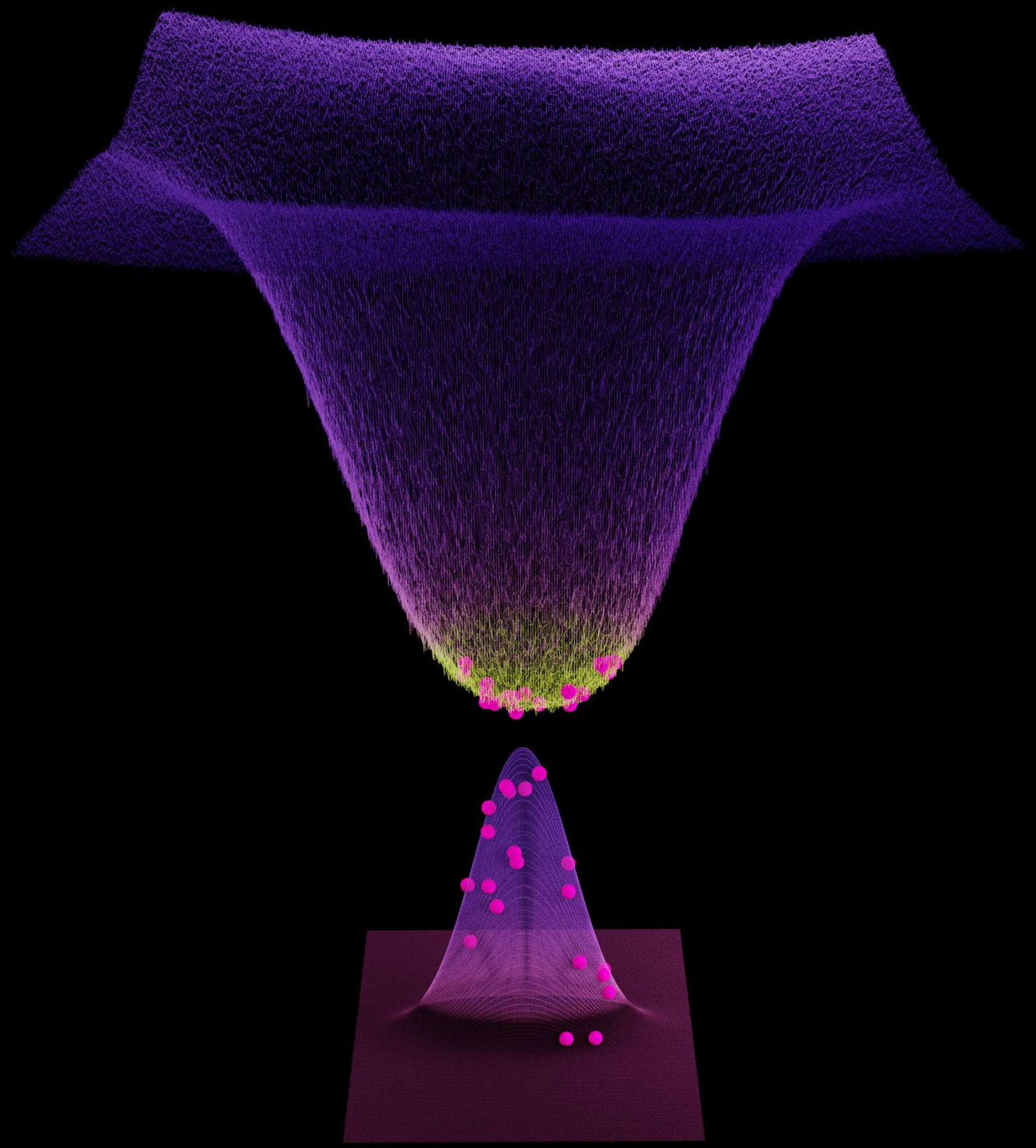




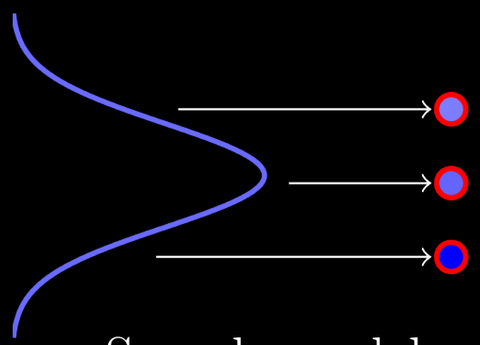
Visualization created in collaboration with Javier Ideami (https://losslandscape.com)

SWAG



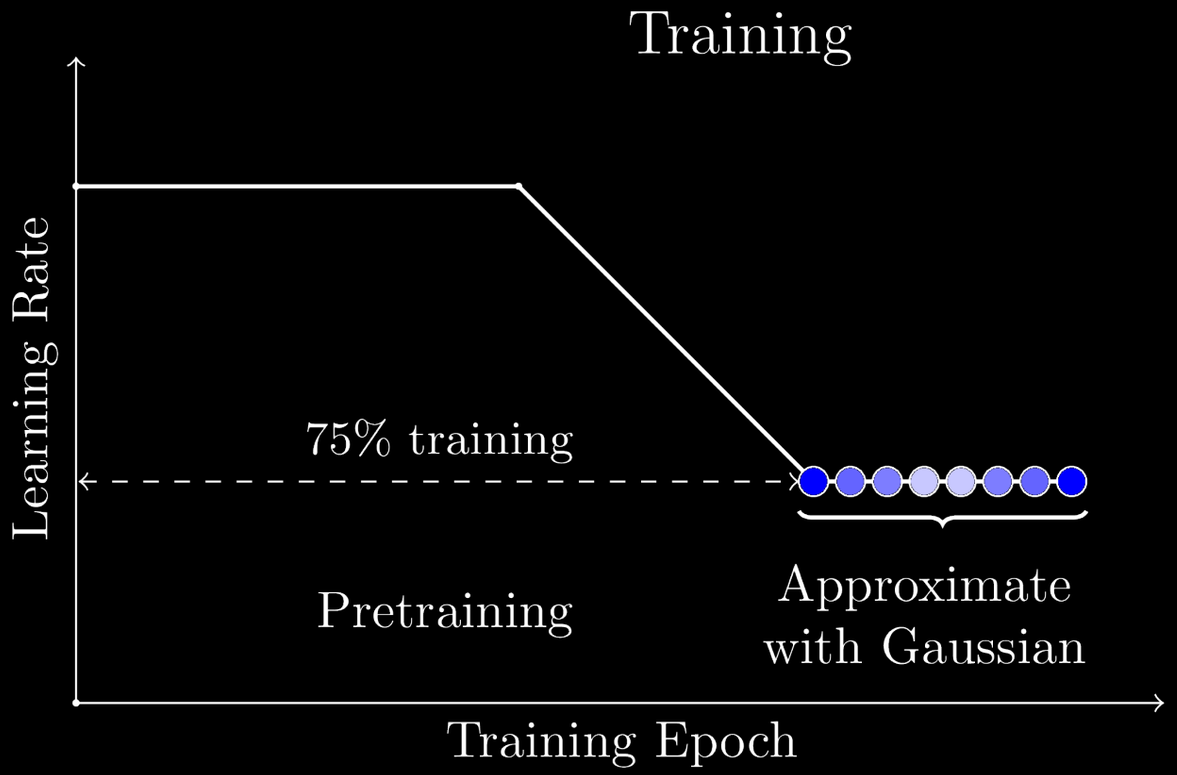


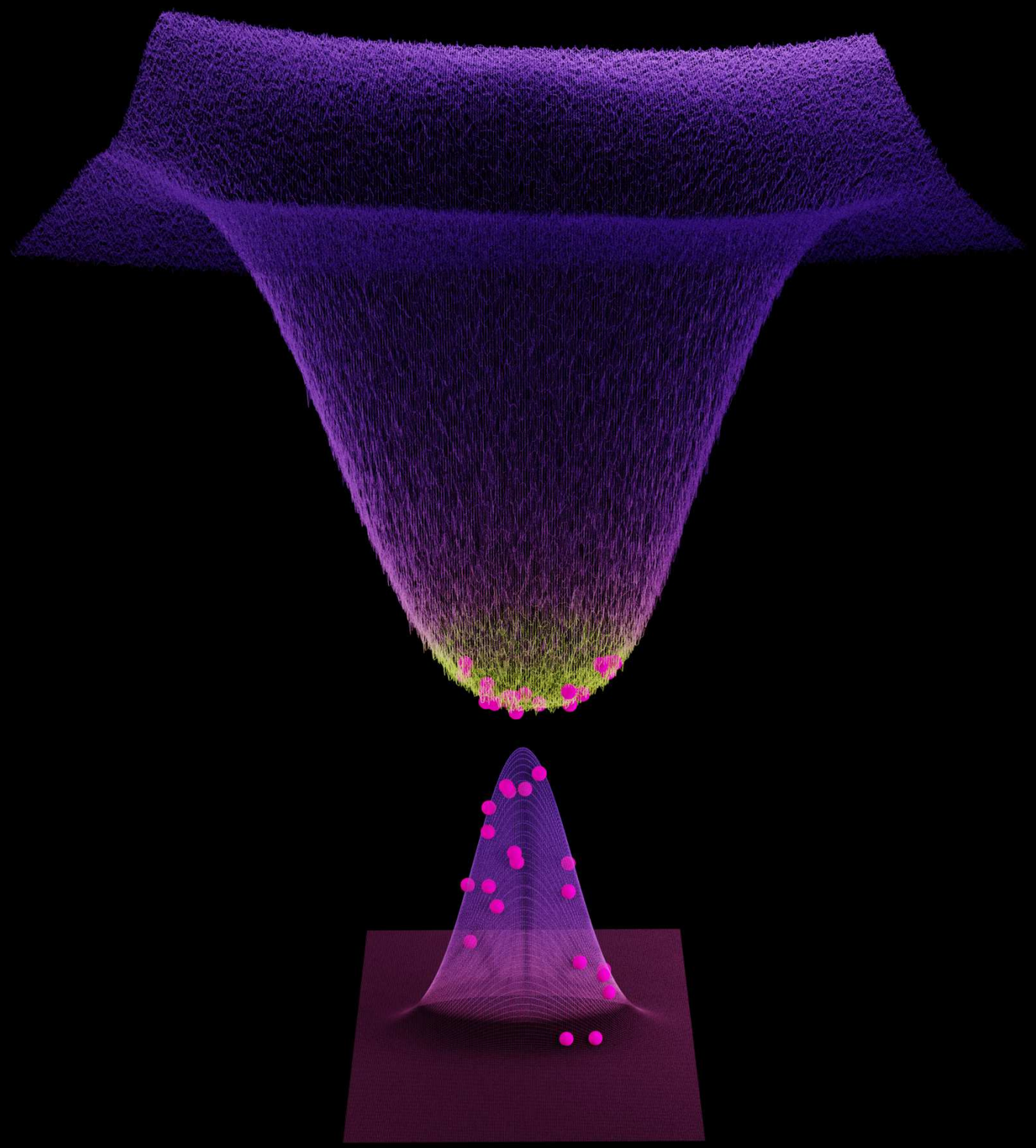
Evaluation

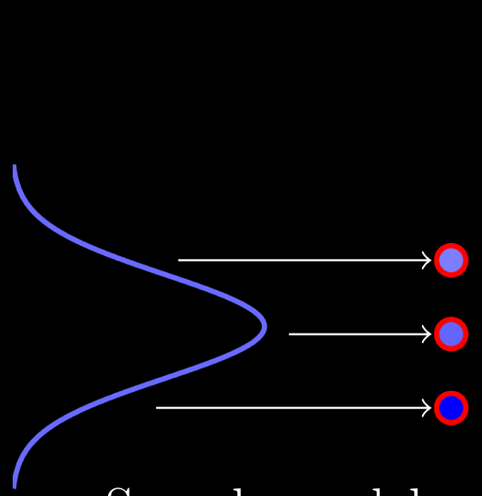


Visualization created in collaboration with Javier Ideami (https://losslandscape.com)

SWAG

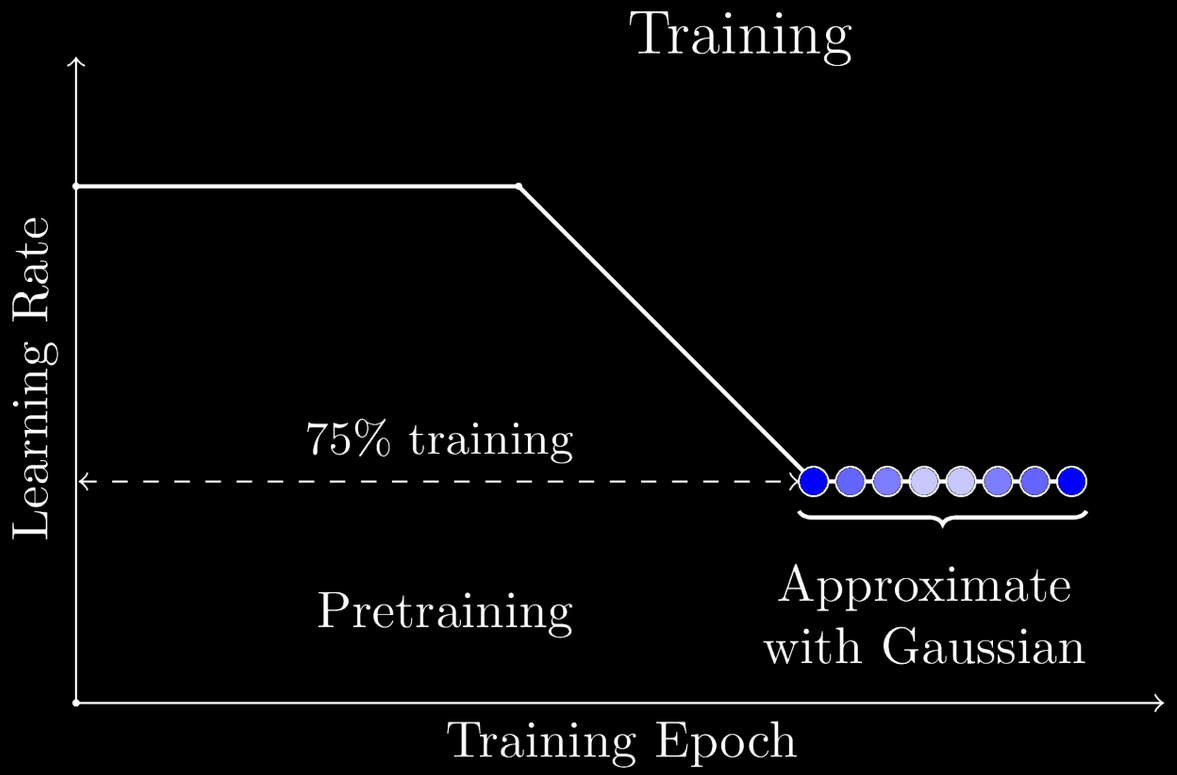


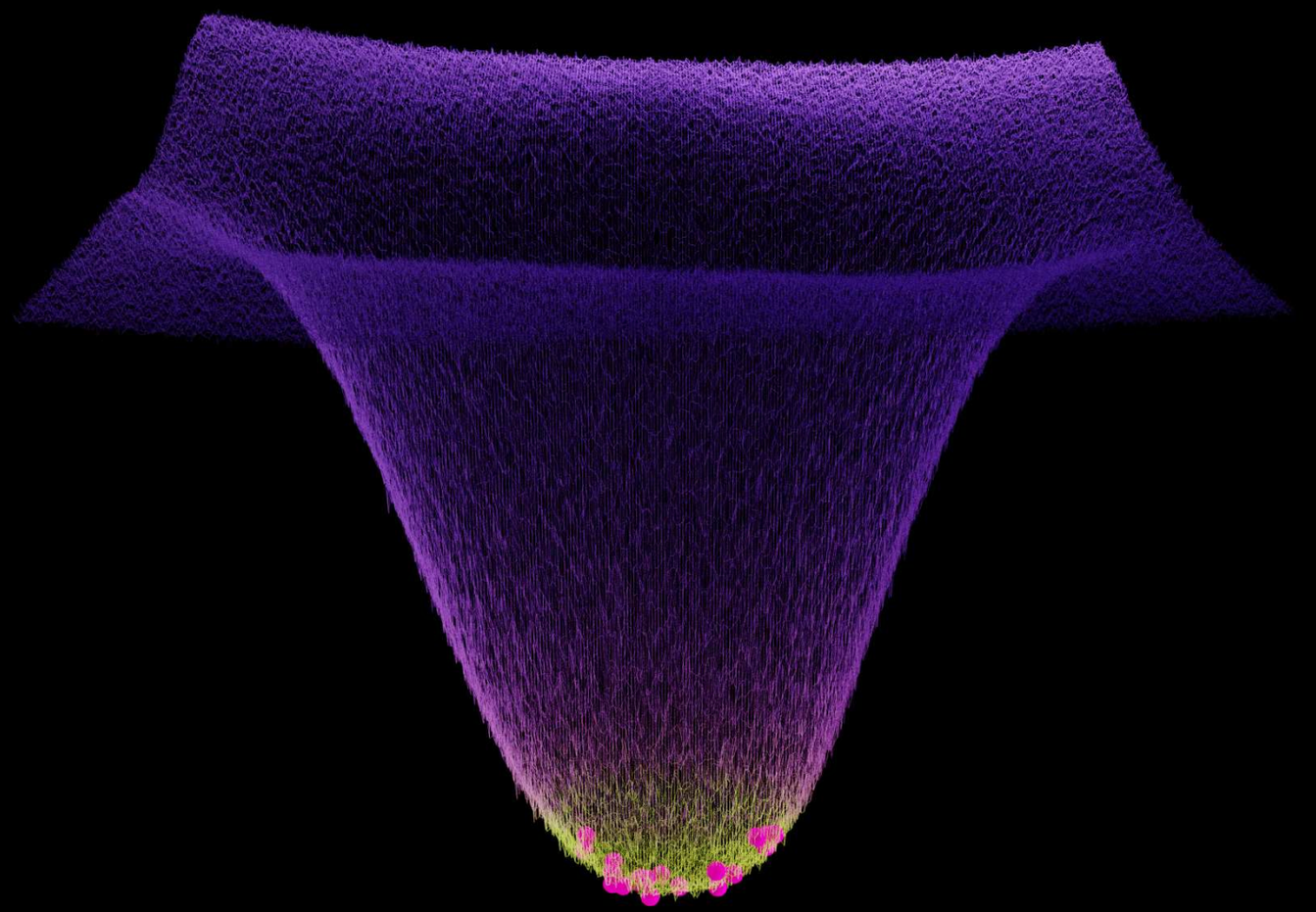


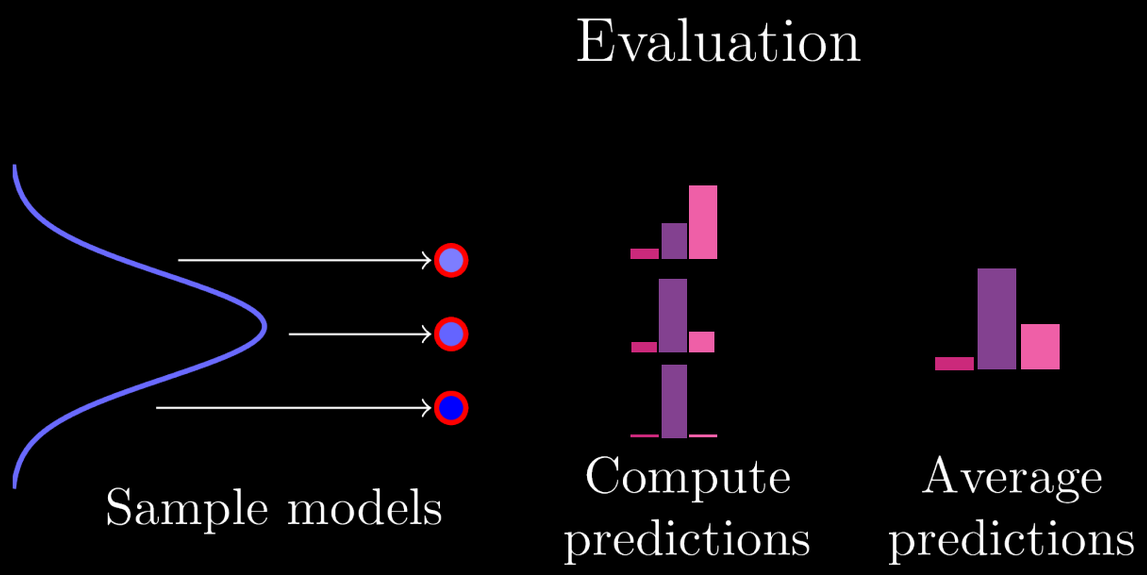


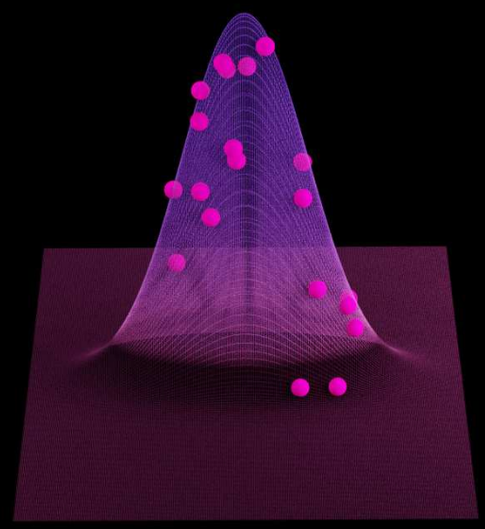
Visualization created in collaboration with Javier Ideami (https://losslandscape.com)

SWAG









Visualization created in collaboration with Javier Ideami (https://losslandscape.com)

SWAG

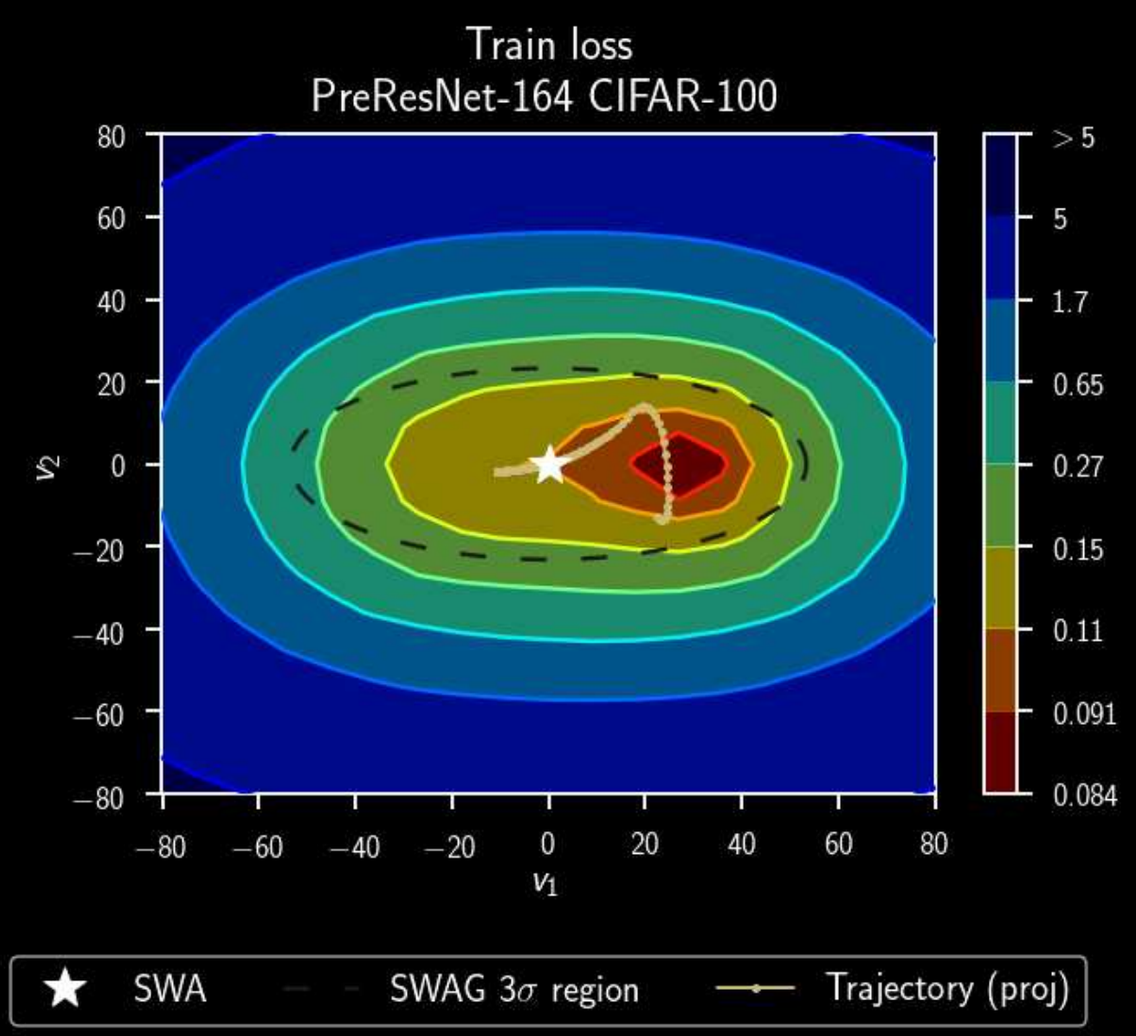
$$N(\mu_{swag}, \Sigma_{swag})$$

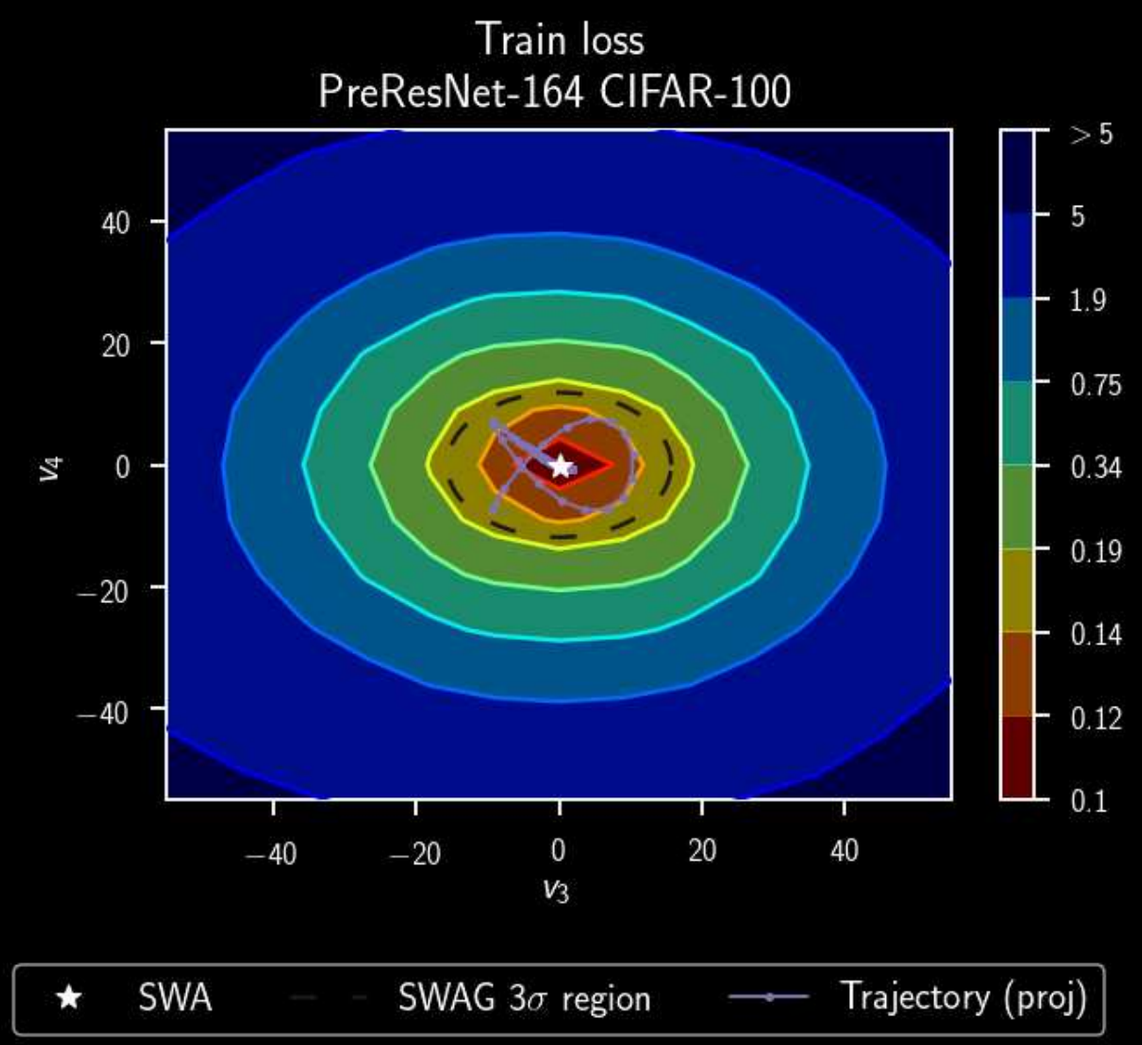
$$\mu_{swag} = w_{swa} = \tilde{w} = \frac{1}{T}\sum_i w_i$$

$$\Sigma_{swag} = \frac{1}{w}(\Sigma_{diag} + \Sigma_{low-rank})$$

$$\Sigma_{diag} = \frac{1}{T-1}\sum_i diag(w_i - \tilde{w})^2$$

$$\Sigma_{low-rank} = \frac{1}{T-1}\sum_i (w_i - \tilde{w})(w_i - \tilde{w})^T$$





SWAG captures local geometry of the posterior

SUBSPACE INFERENCE

A modular approach:

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

SUBSPACE INFERENCE

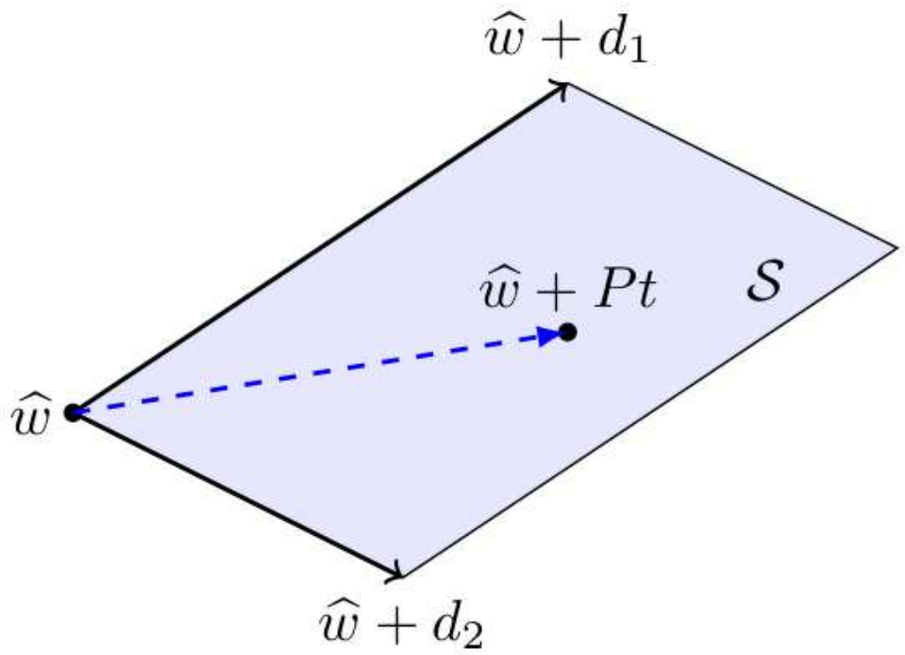
A modular approach:

- 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

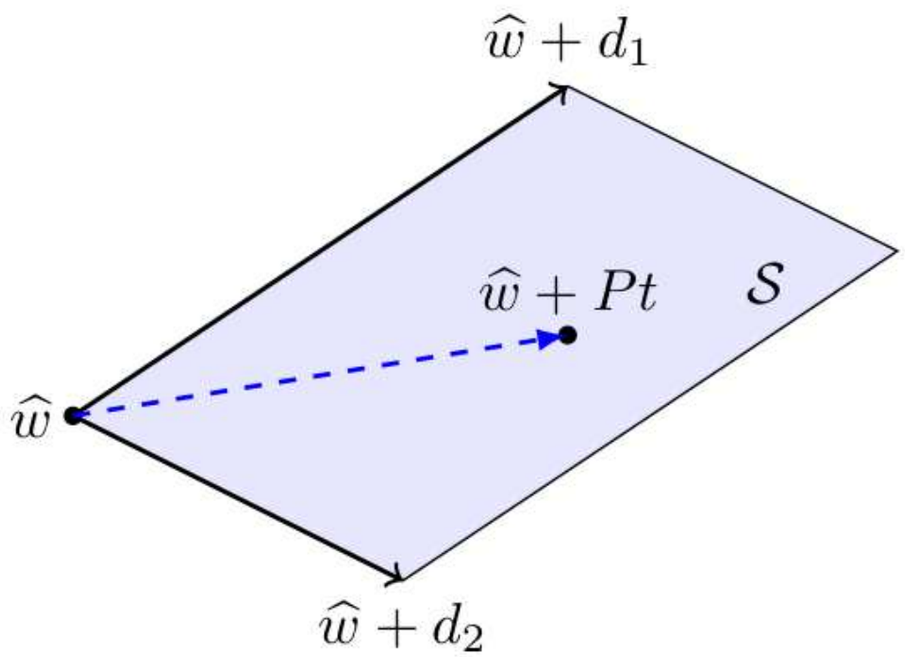
- Choose shift $\hat{w}$ and basis vectors $\{d_1, \ldots, d_K\}$
- Define subspace $S = \{w \,|\, w = \hat{w} + \underbrace{t_1 d_1 + \ldots + t_k d_K}_{Pt}\}$
- Likelihood $p(D \,|\, t) = p_M(D \,|\, w = \hat{w} + Pt)$.



INFERENCE

- Approximate inference over parameters t
- Bayesian model averaging at test time:

$$p(D^* \mid D) = \frac{1}{J} \sum_{i=1}^{J} p_M(D^* \mid \tilde{w} = \hat{w} + P\tilde{t}_i), \quad \tilde{t}_i \sim q(t \mid D)$$



INFERENCE

In the subspace, we are able to apply inference methods which struggle in the full parameter space.

- Variational Inference
  - Mean Field Gaussian family
  - RealNVP family
- Monte Carlo
  - No-U-Turn Sampler
  - Elliptical Slice Sampling

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:

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

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

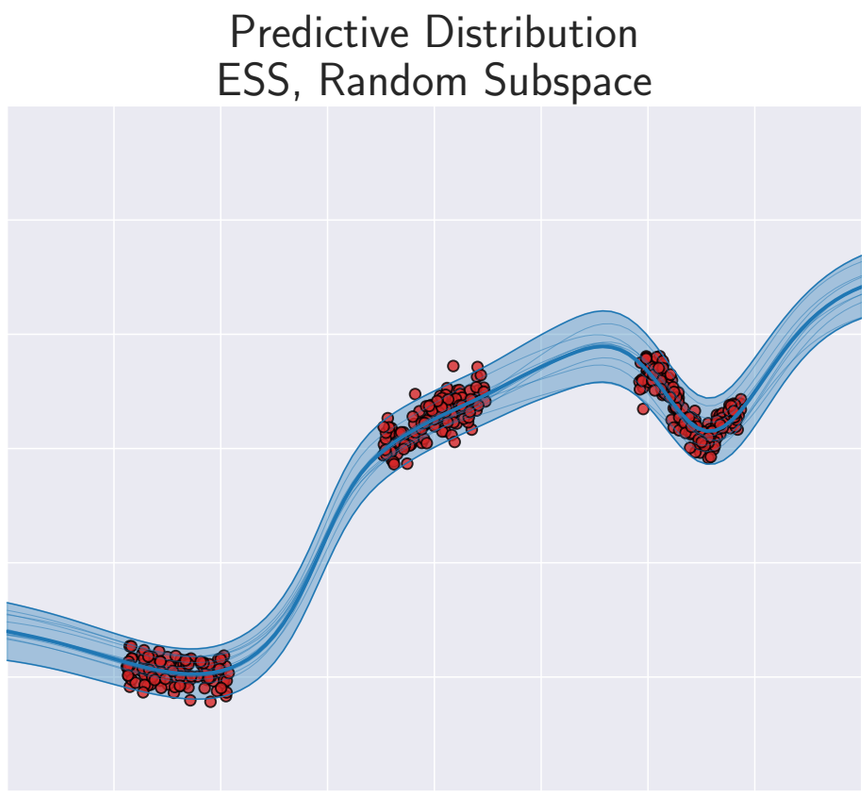
SUBSPACE CHOICE

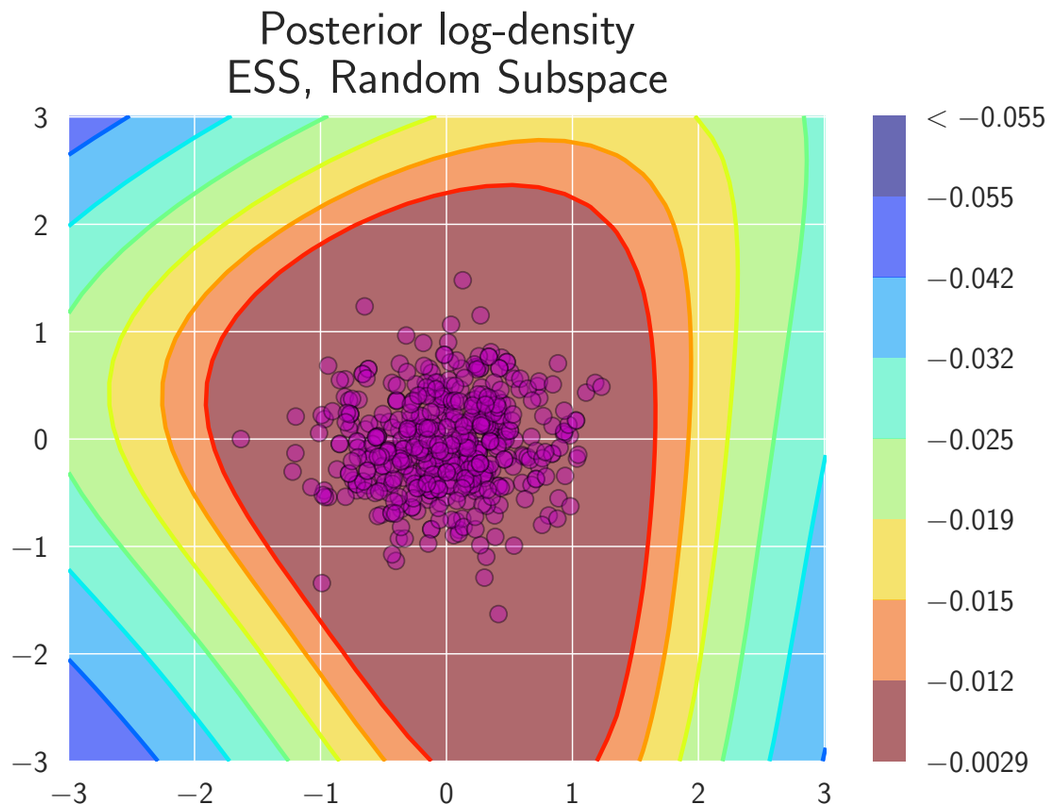
We want a subspace that

- Contains **diverse** models
- **Cheap** to construct

RANDOM SUBSPACE

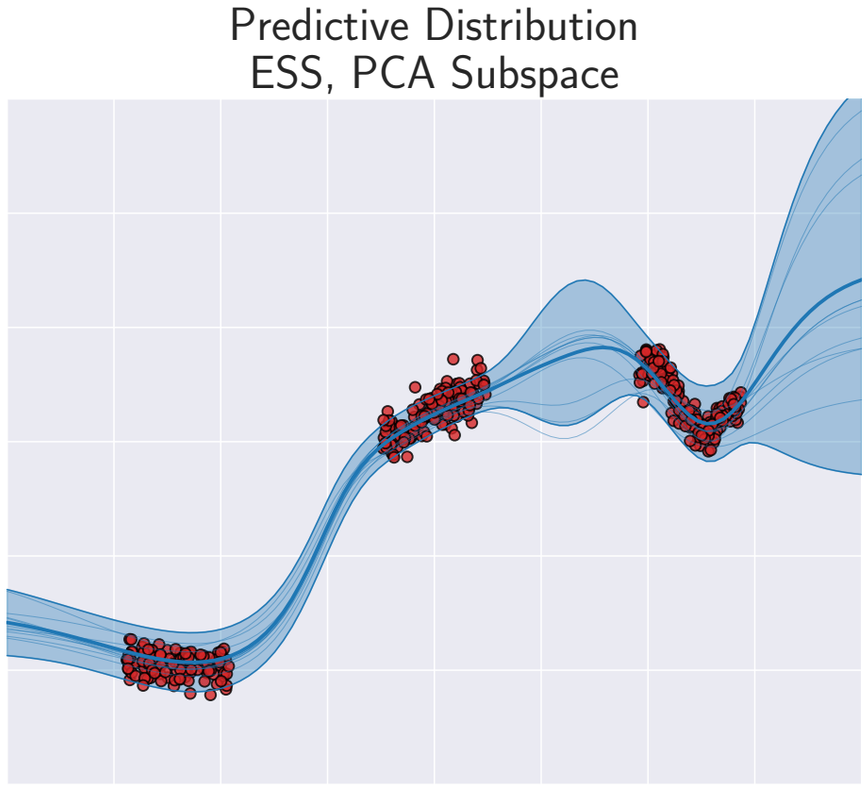
- Directions $d_1, \ldots, d_K \sim N(0, I_p)$
- Use pre-trained solution as shift $\hat{w}$
- Subspace $S = \{w \,|\, w = \hat{w} + Pt\}$

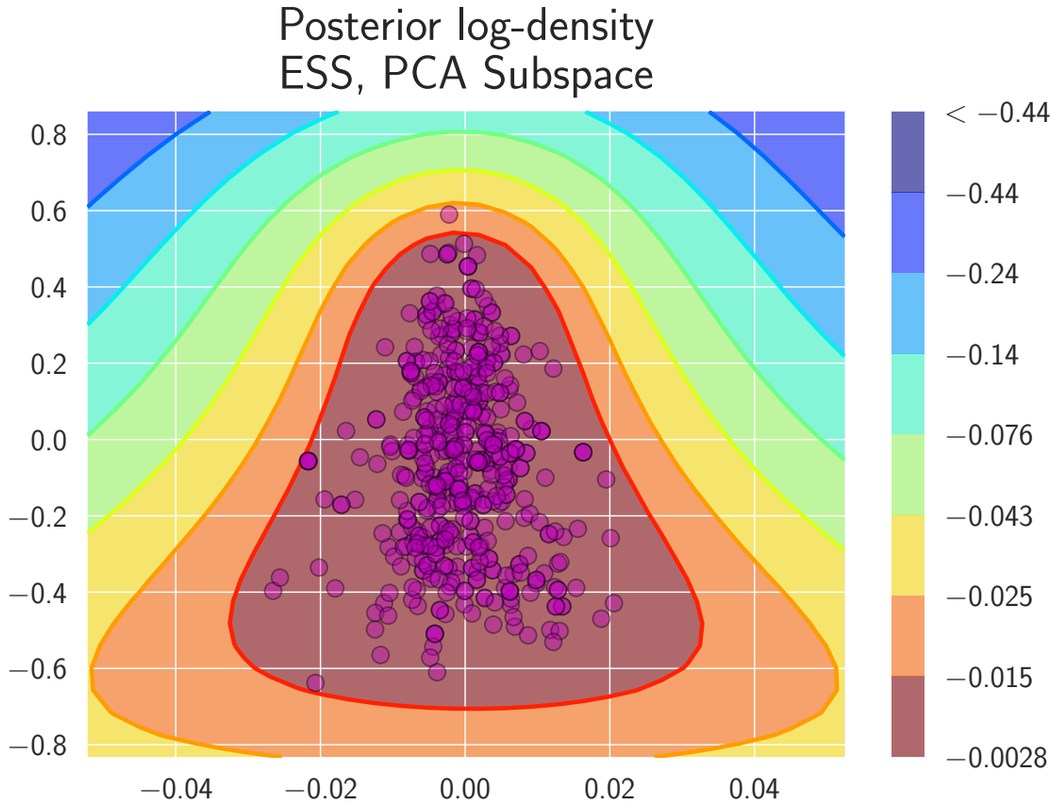




PCA OF THE SGD TRAJECTORY

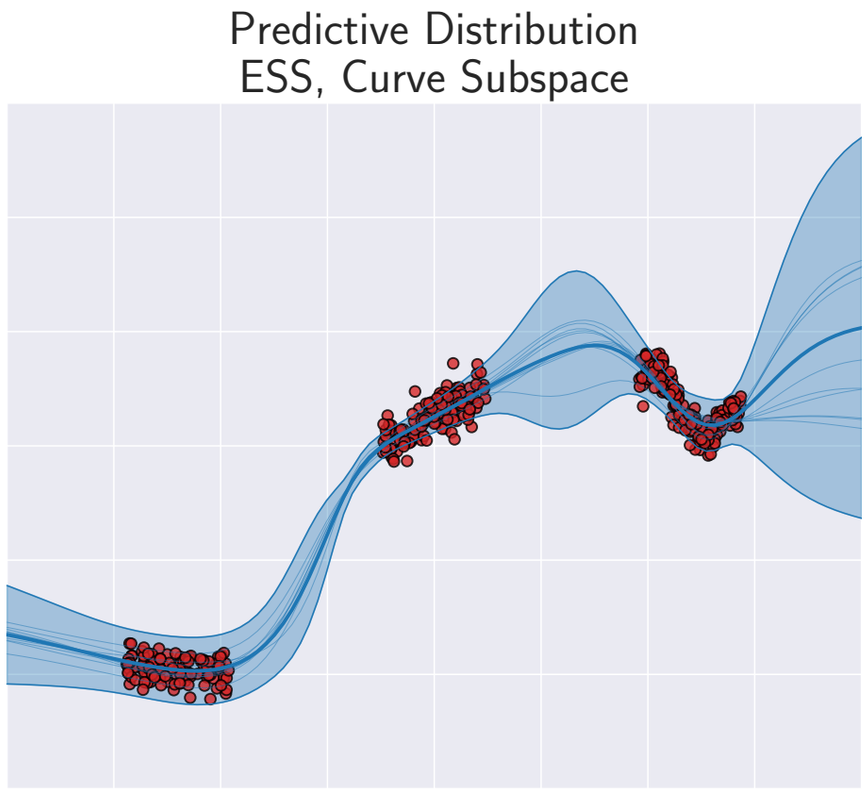
- Run SGD with high constant learning rate from a pre-trained solution
- Collect snapshots of weights w_i
- Use SWA solution as shift $\hat{w} = \frac{1}{T}\sum_i w_i$
- $\{d_1, \ldots, d_K\}$ – first K PCA components of vectors $\hat{w} - w_i$

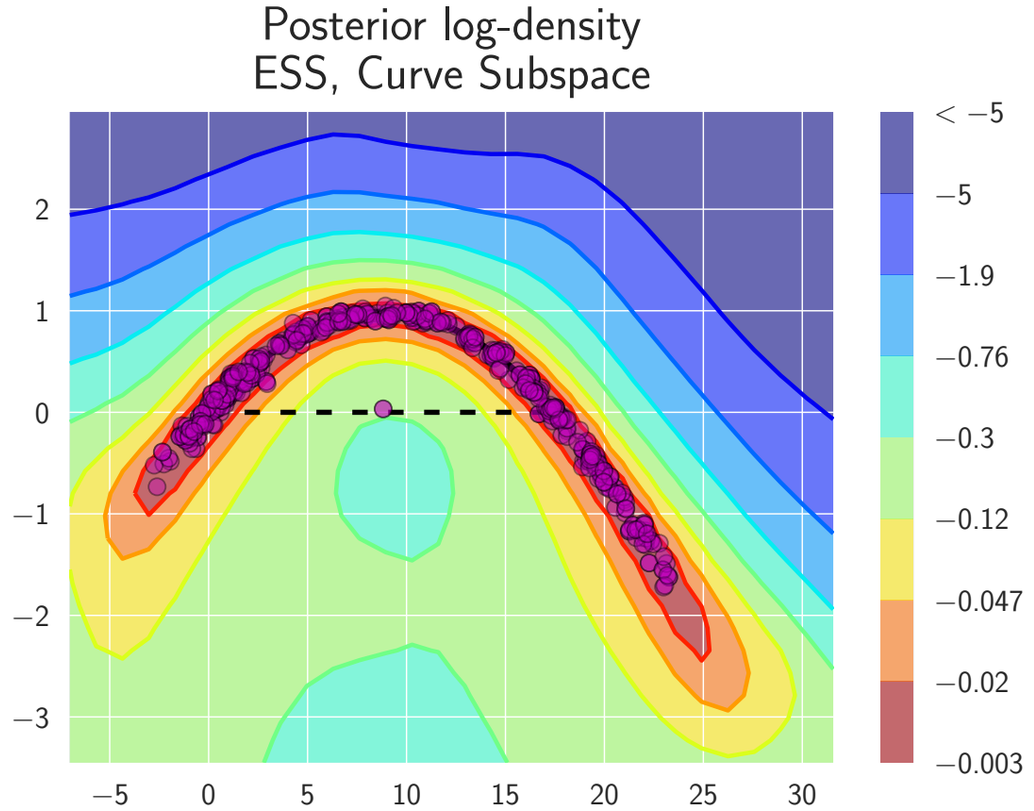




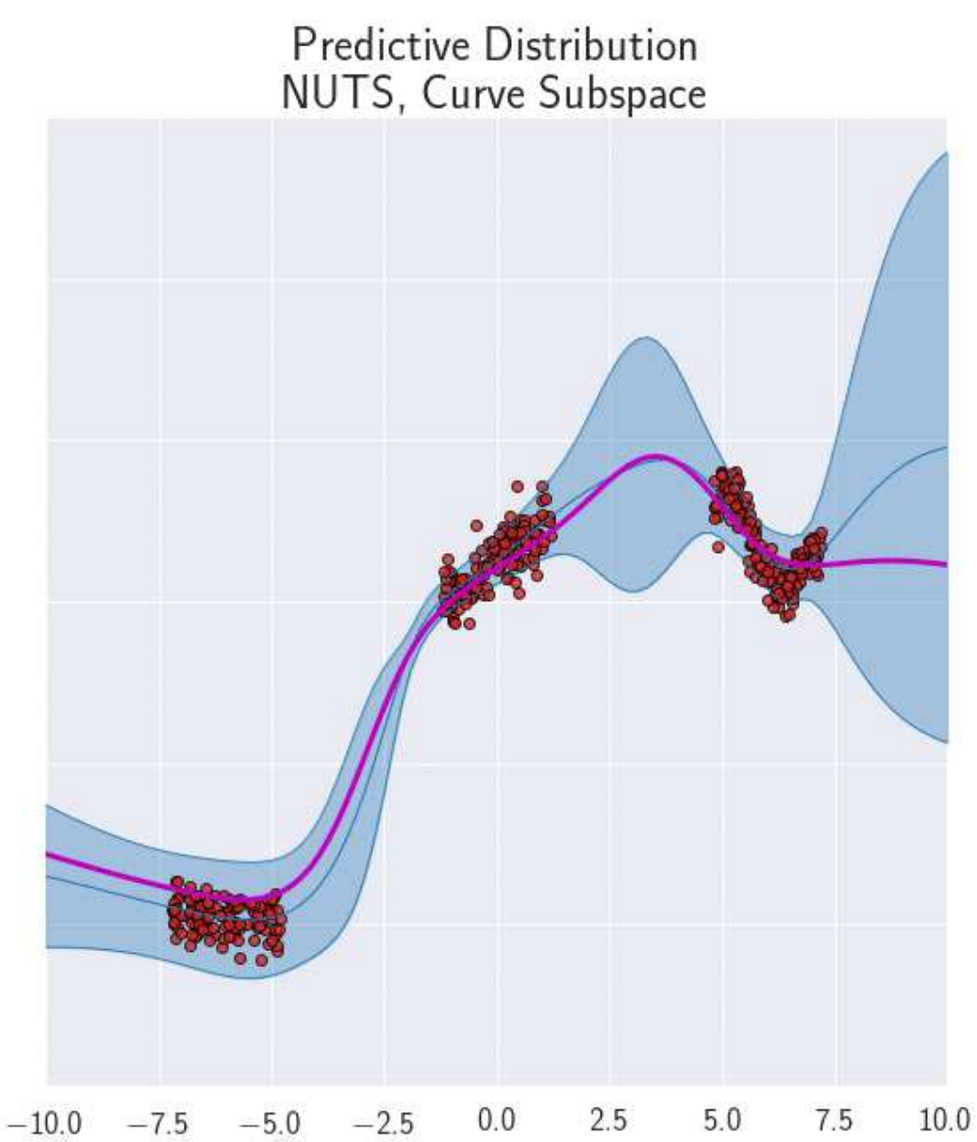
CURVE SUBSPACE

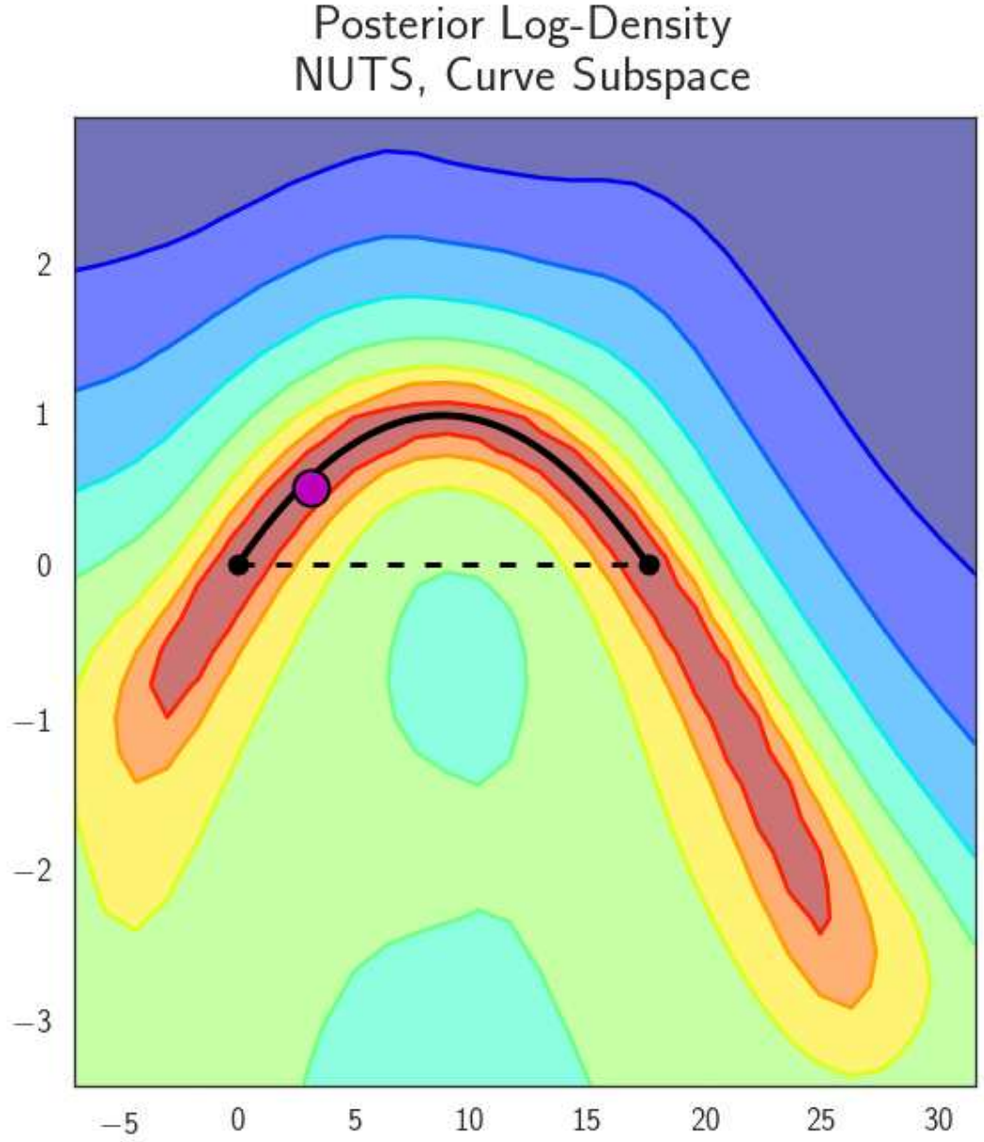
- Garipov et al. 2018 proposed a method to find 2D subspaces containing a path of low loss between weights of two independently trained neural networks



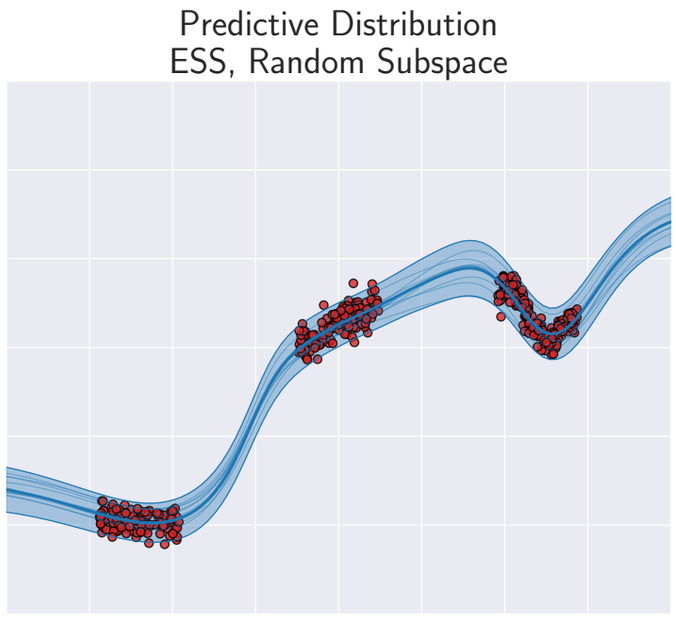


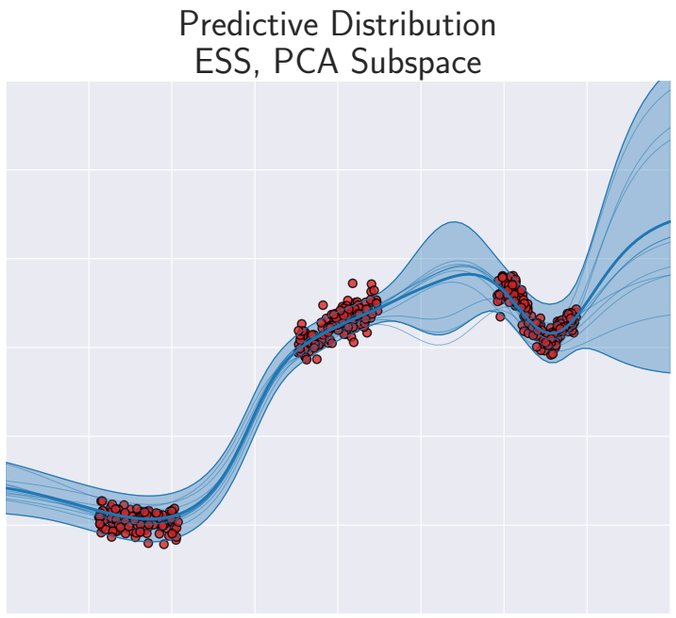
SUBSPACE INFERENCE ILLUSTRATION

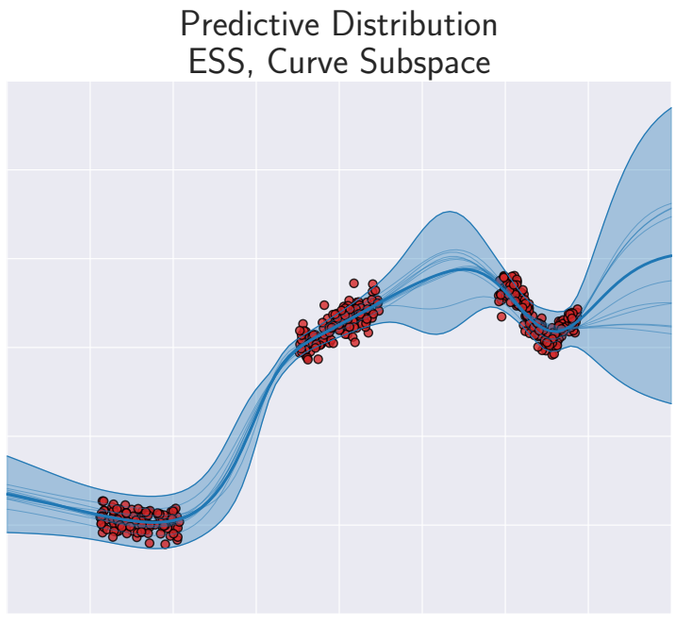


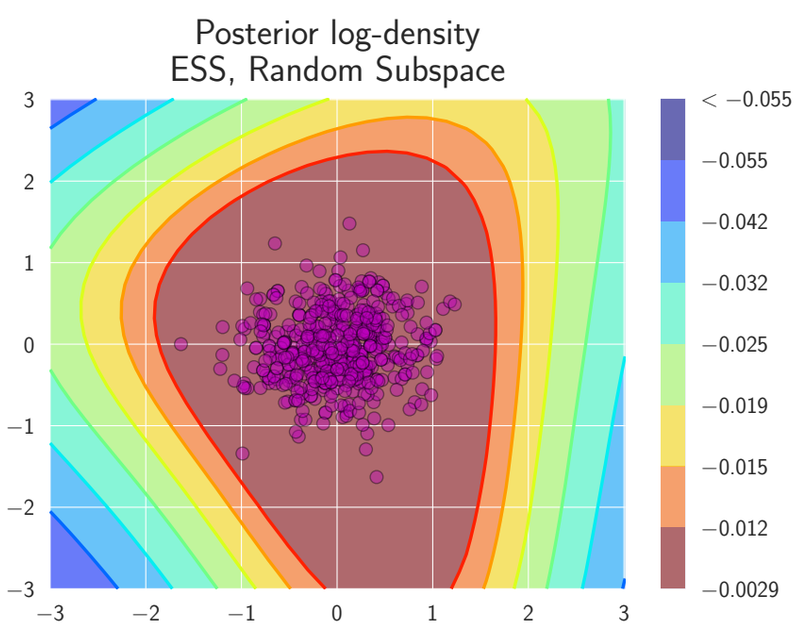


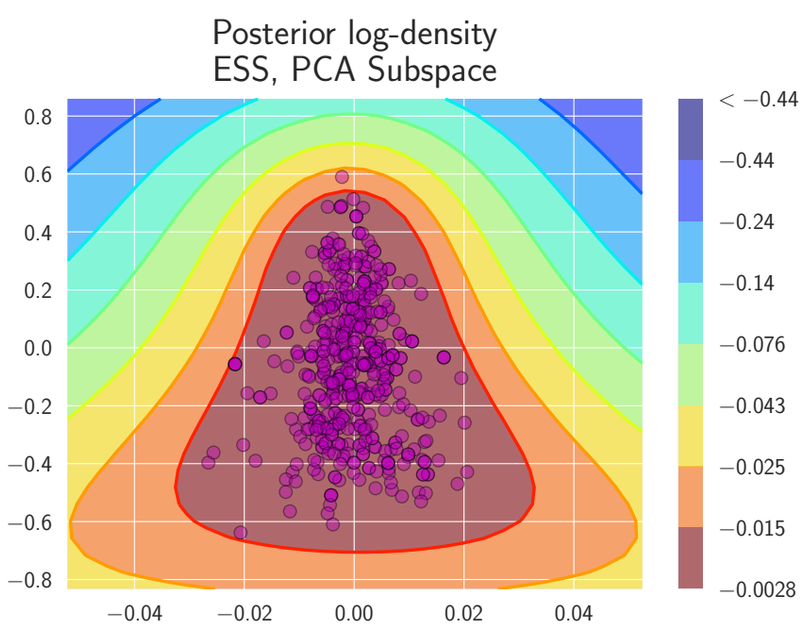
SUBSPACE COMPARISON

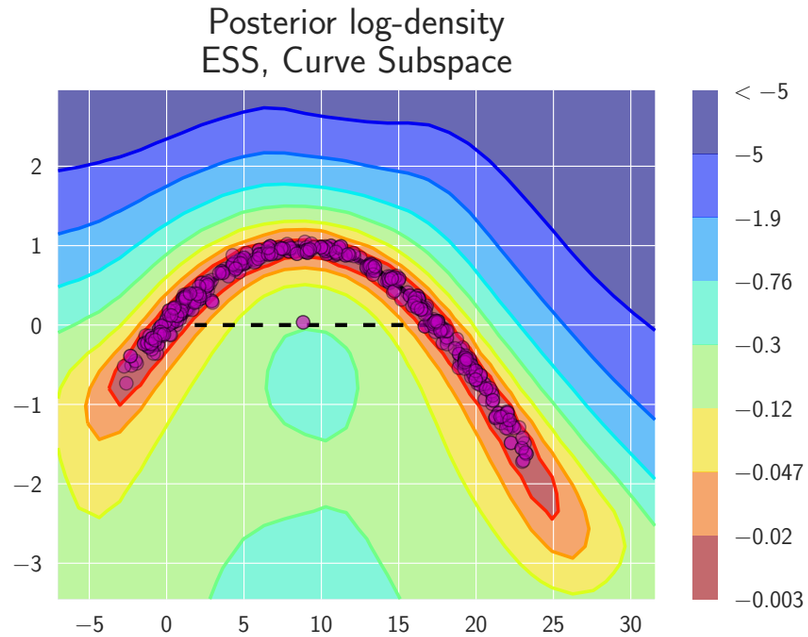




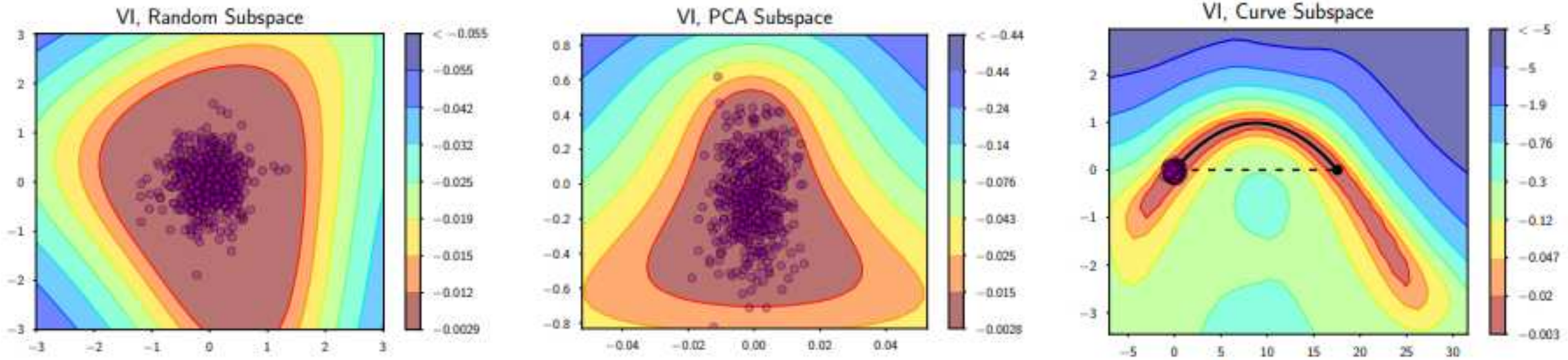


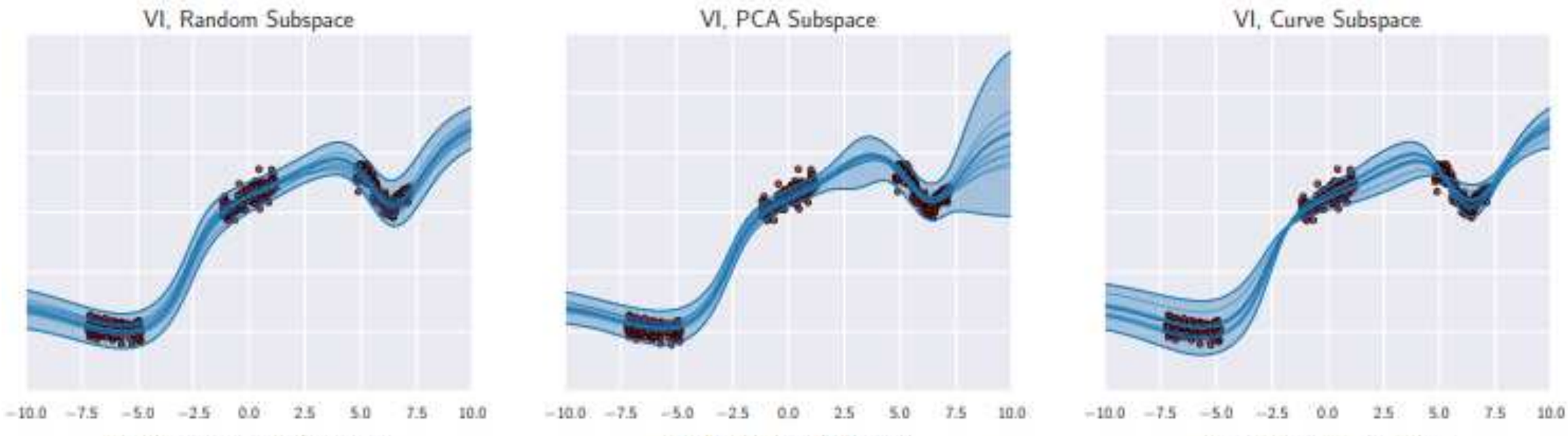




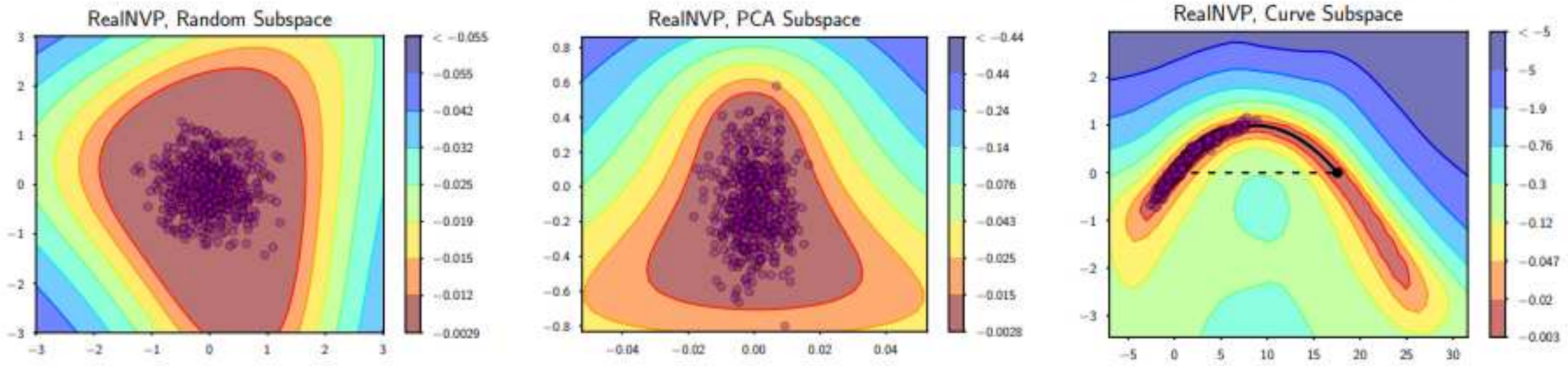


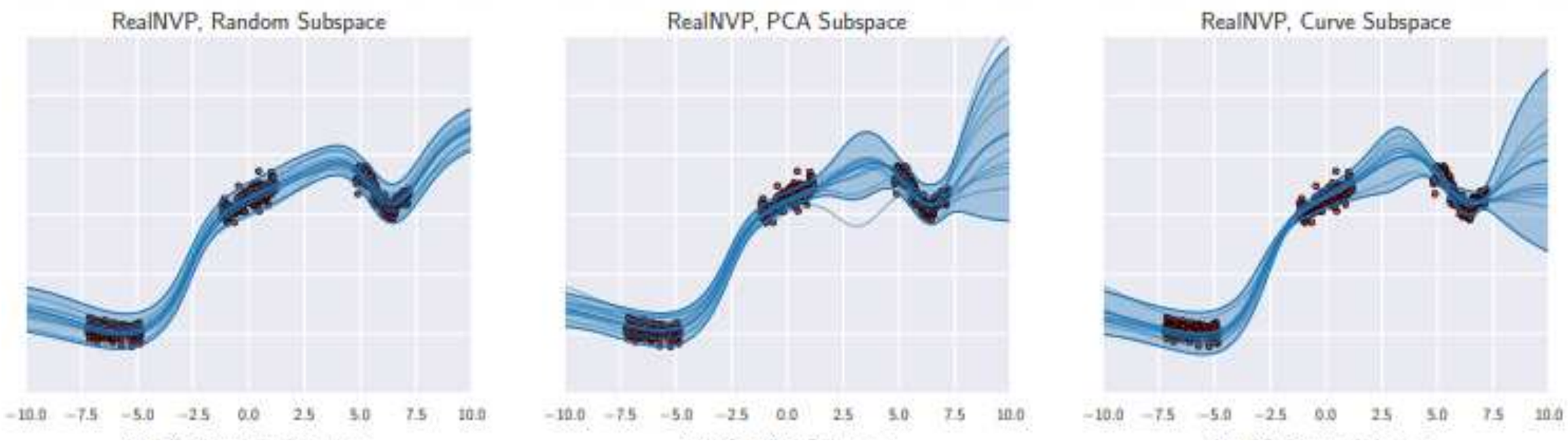
INFERENCE METHOD COMPARISON



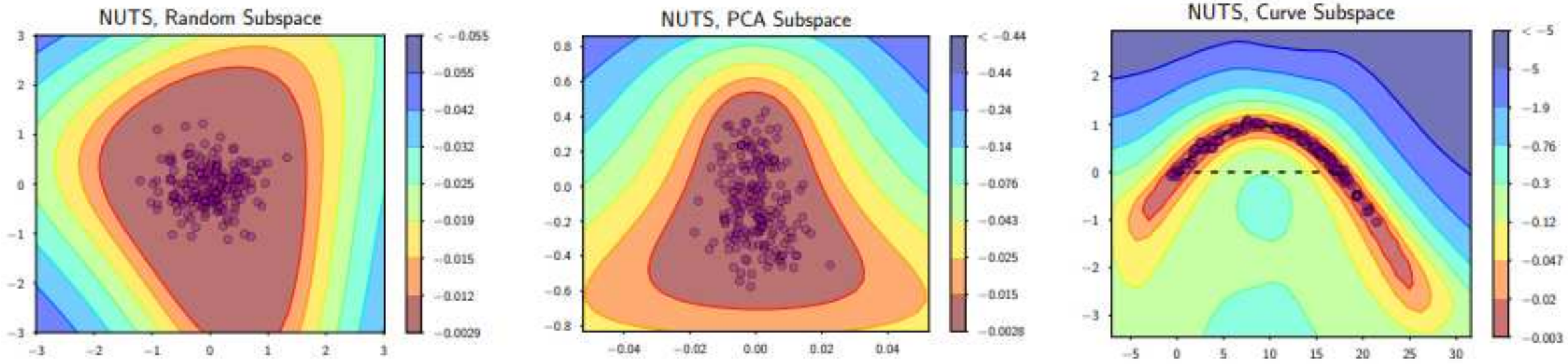


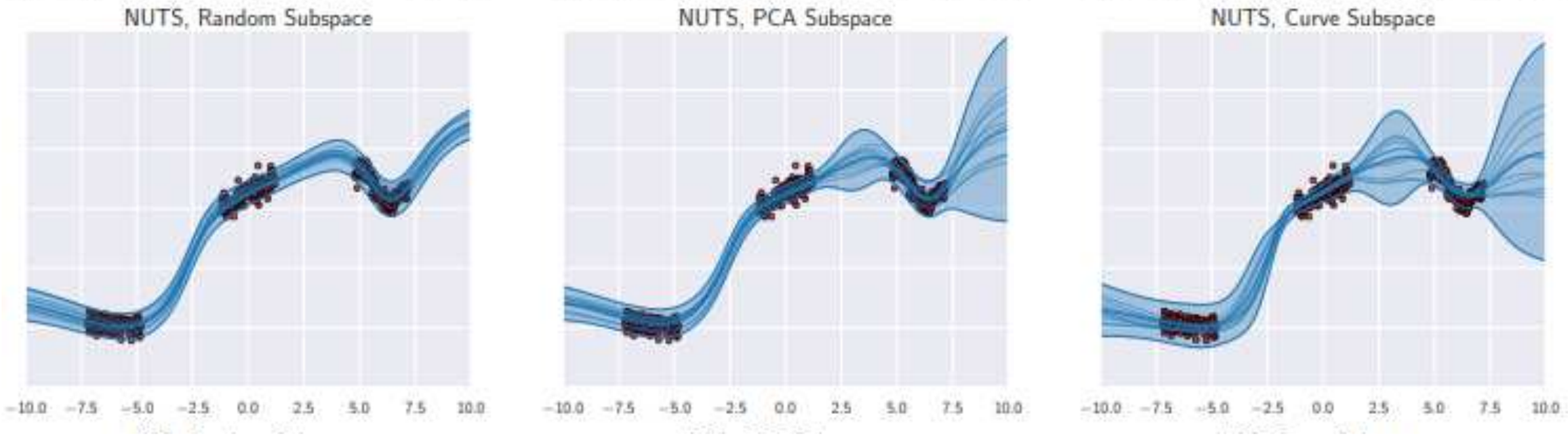
INFERENCE METHOD COMPARISON



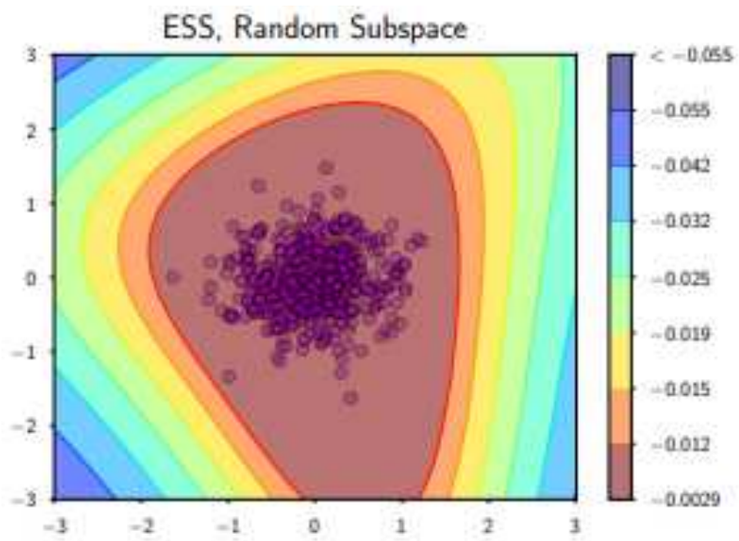


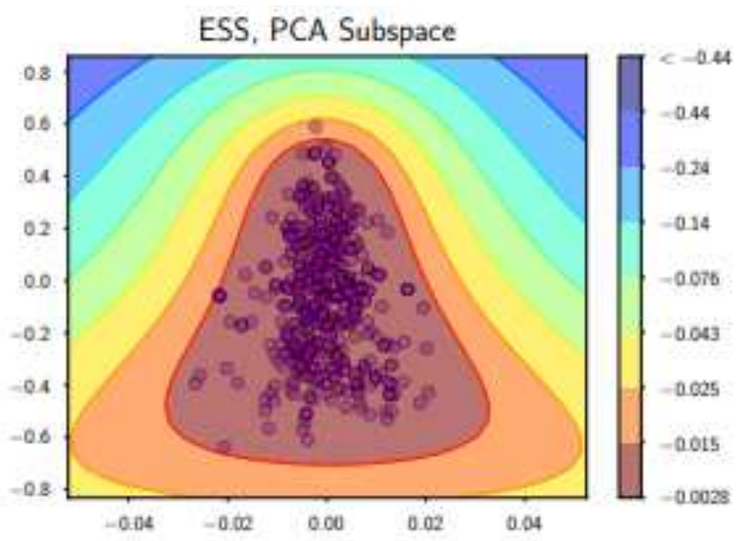
INFERENCE METHOD COMPARISON

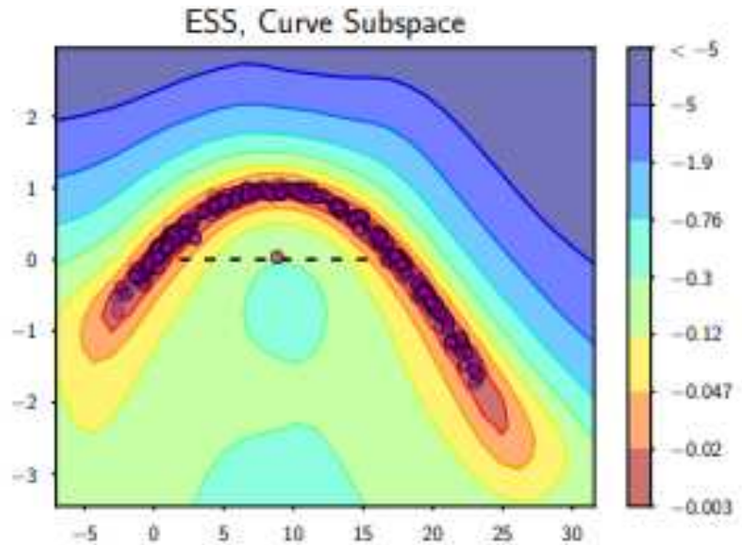


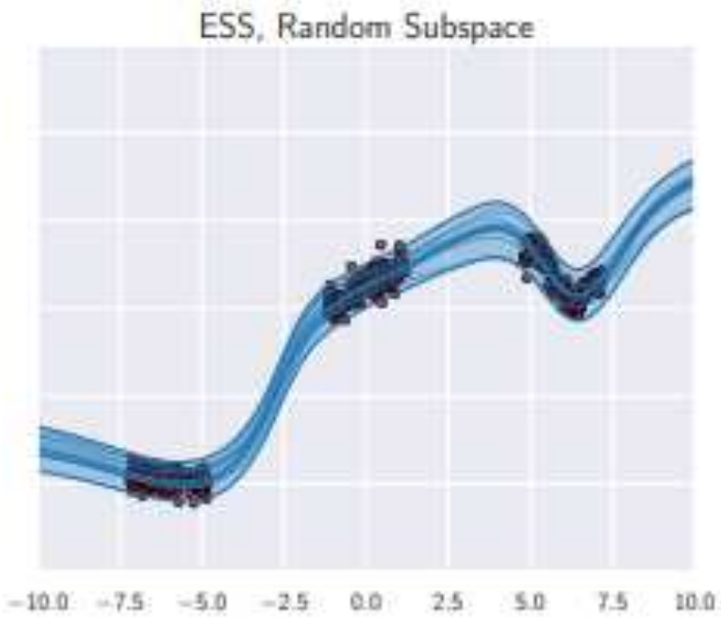


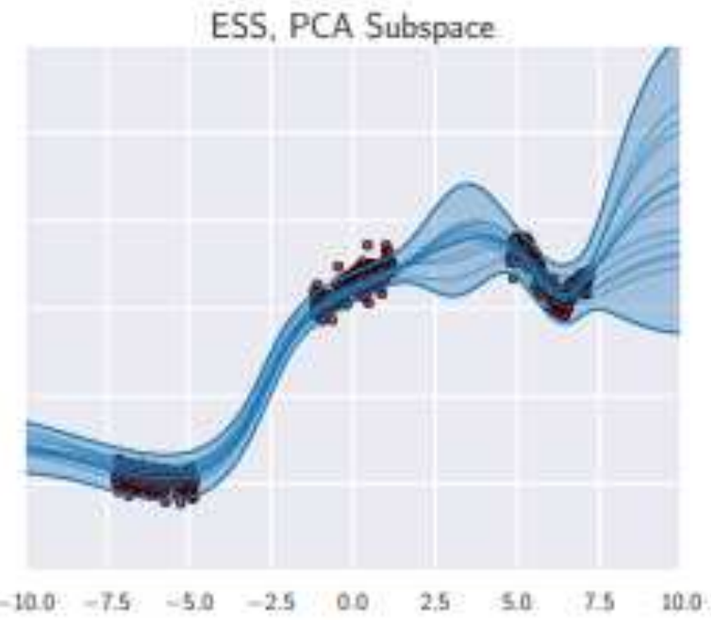
INFERENCE METHOD COMPARISON

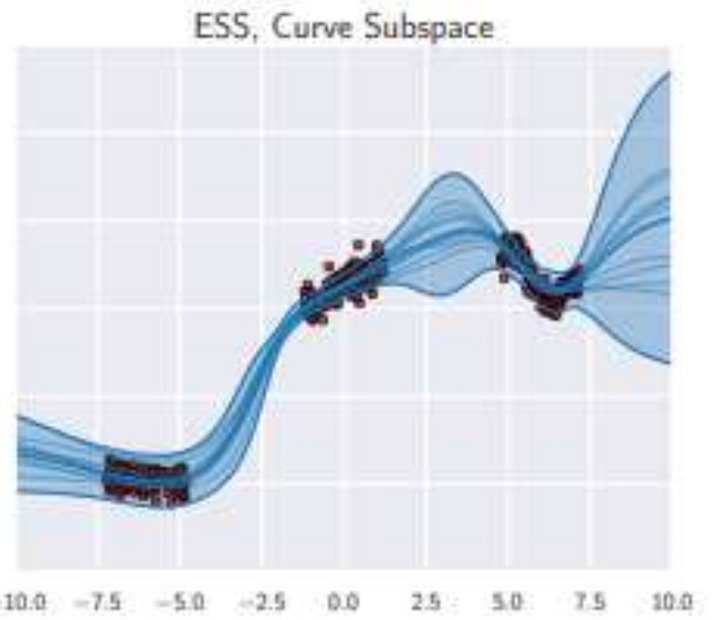




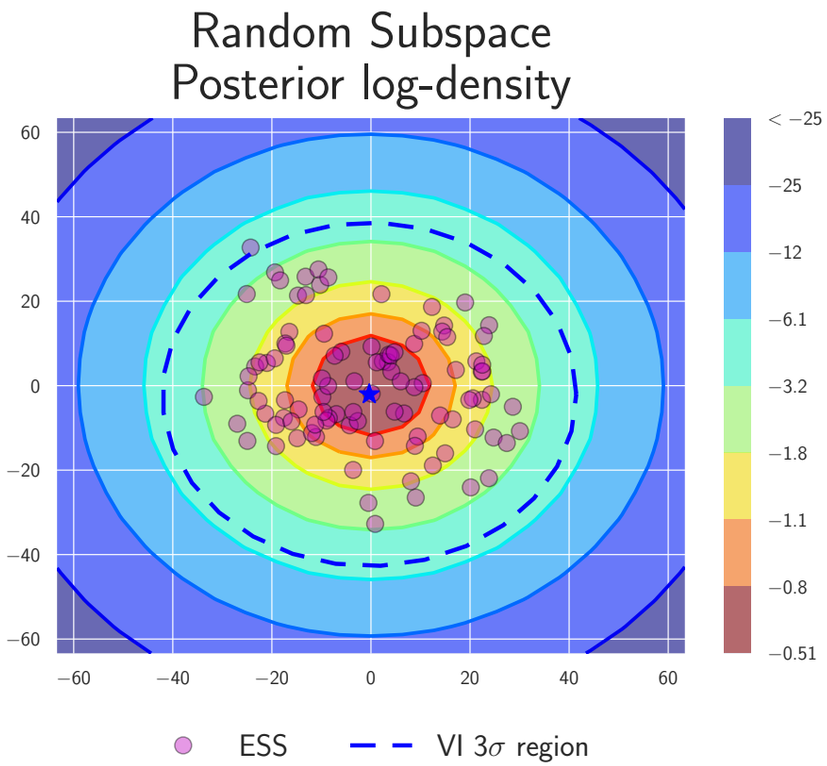


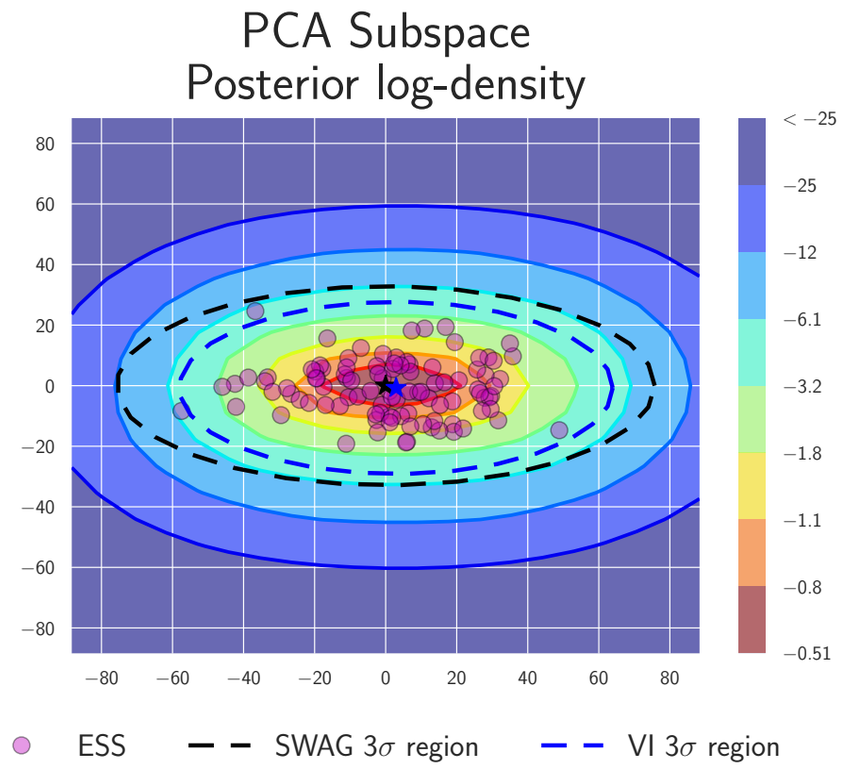


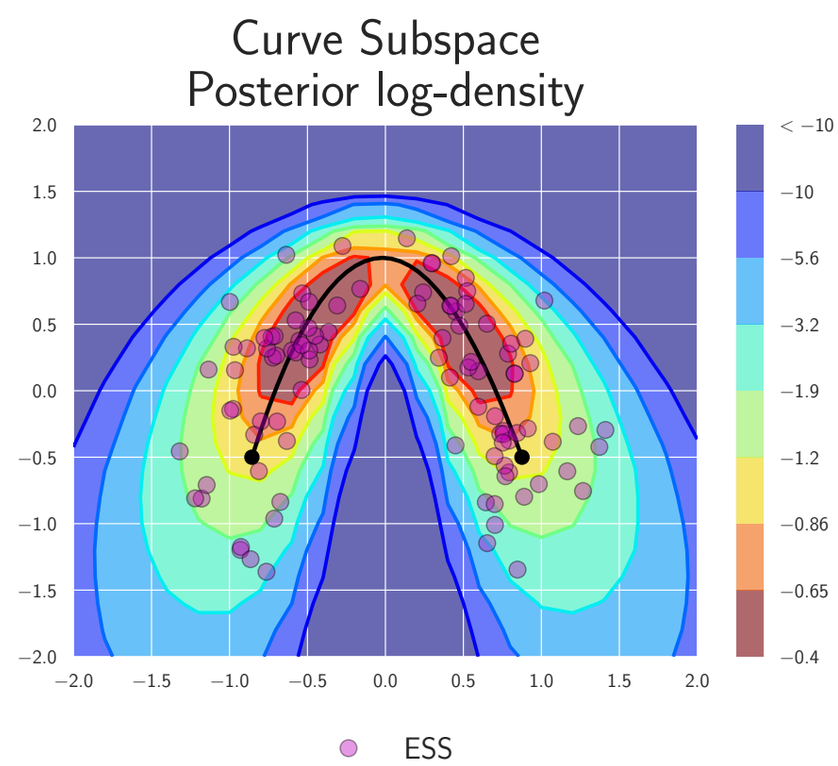




SUBSPACE COMPARISON ON PRERESNET-164, CIFAR-100

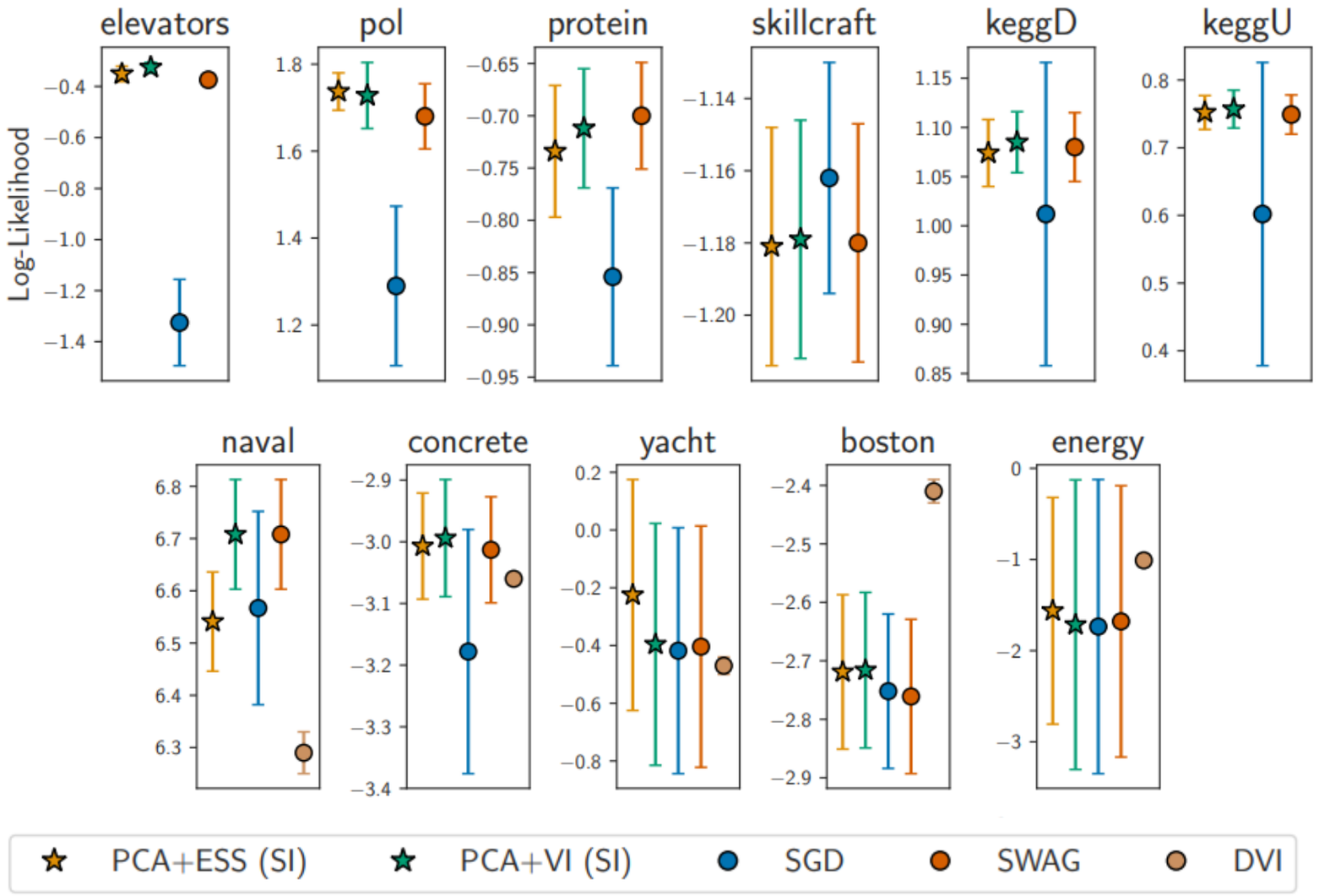






	SGD	Random	PCA	Curve
NLL	0.946 ± 0.001	0.686 ± 0.005	0.665 ± 0.004	0.646
Accuracy (%)	78.50 ± 0.32	80.17 ±0.03	80.54 ± 0.13	81.28

UCI EXPERIMENTS



SUBSPACE INFERENCE 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

PROJECTED BNNS: AVOIDING WEIGHT-SPACE PATHOLOGIES BY LEARNING LATENT REPRESENTATIONS OF NEURAL NETWORKS WEIGHTS

Melanie F. Pradier Weiwei Pan Jiayu Yao
Soumya Ghosh Finale Doshi-velez

Harvard University

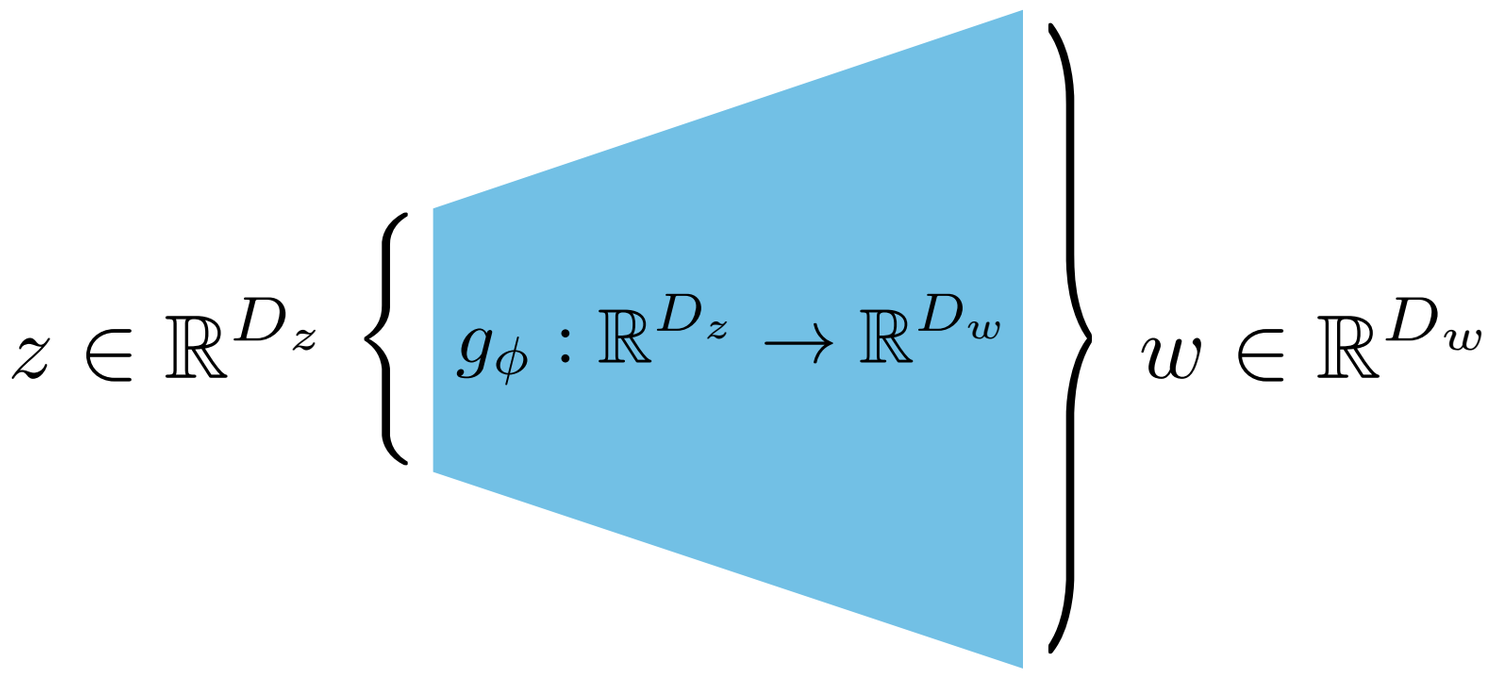
PROJECTED BNN: DIFFERENCES WITH SUBSPACE INFERENCE

- Non-linear subspaces
- Learn uncertainty over subspace
- VI for learning approximate posterior optimizing ELBO

GENERATIVE MODEL

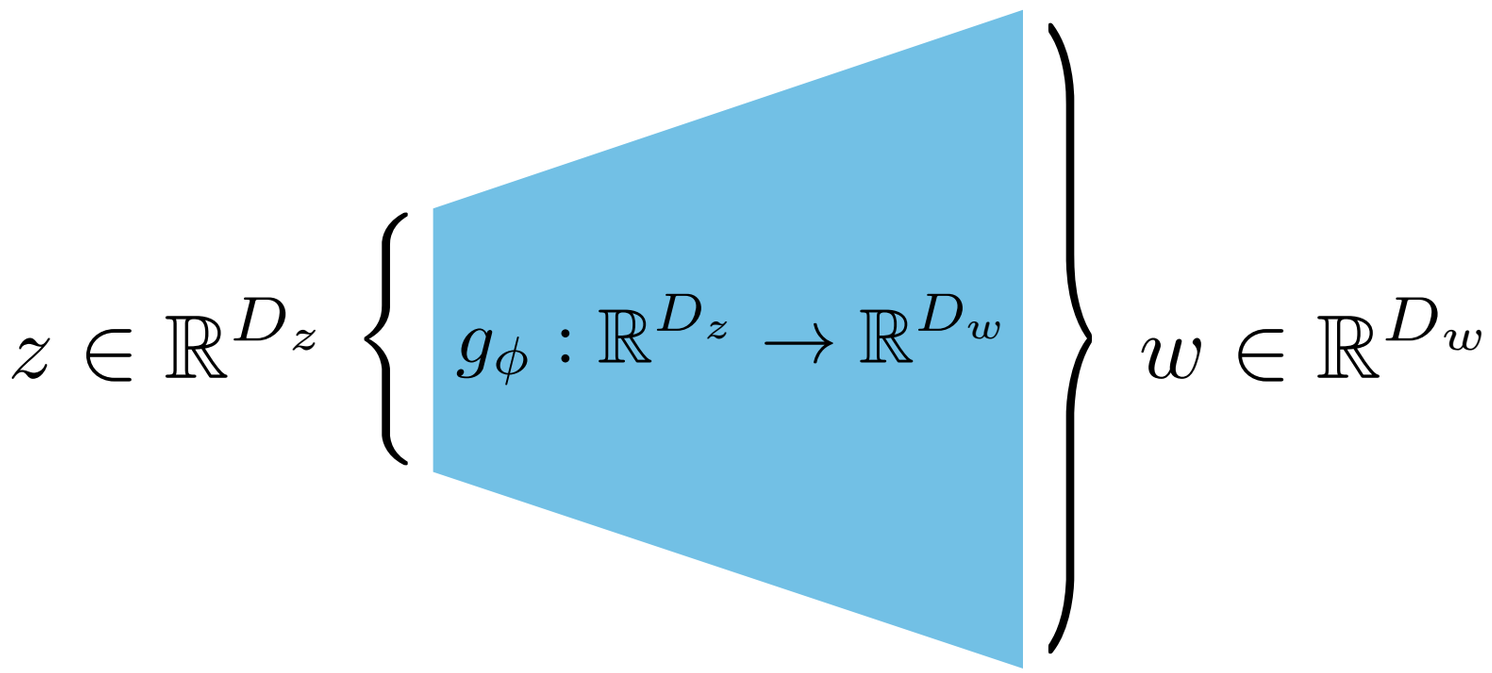
Neural network weights are generated from a low-dimensional manifold:

$$z \sim p(z),\ \ \phi \sim p(\phi),\ \ w = g_\phi(z),\ \ y \sim \mathcal{N}(f_w(x), \sigma_y^2)$$



GENERATIVE MODEL

Our objective is to infer the joint posterior distribution: $p(z, \phi | y, x)$



INFERENCE

Assume mean-field posterior approximation $q_\lambda(z, \phi) = q_{\lambda_z}(z) q_{\lambda_\phi}(\phi)$

$$q_{\lambda_z}(z) = \mathcal{N}(\tilde{\mu}_z, \tilde{\sigma}_z I) \qquad q_{\lambda_\phi}(\phi) = \mathcal{N}(\tilde{\mu}_\phi, \tilde{\sigma}_\phi I)$$

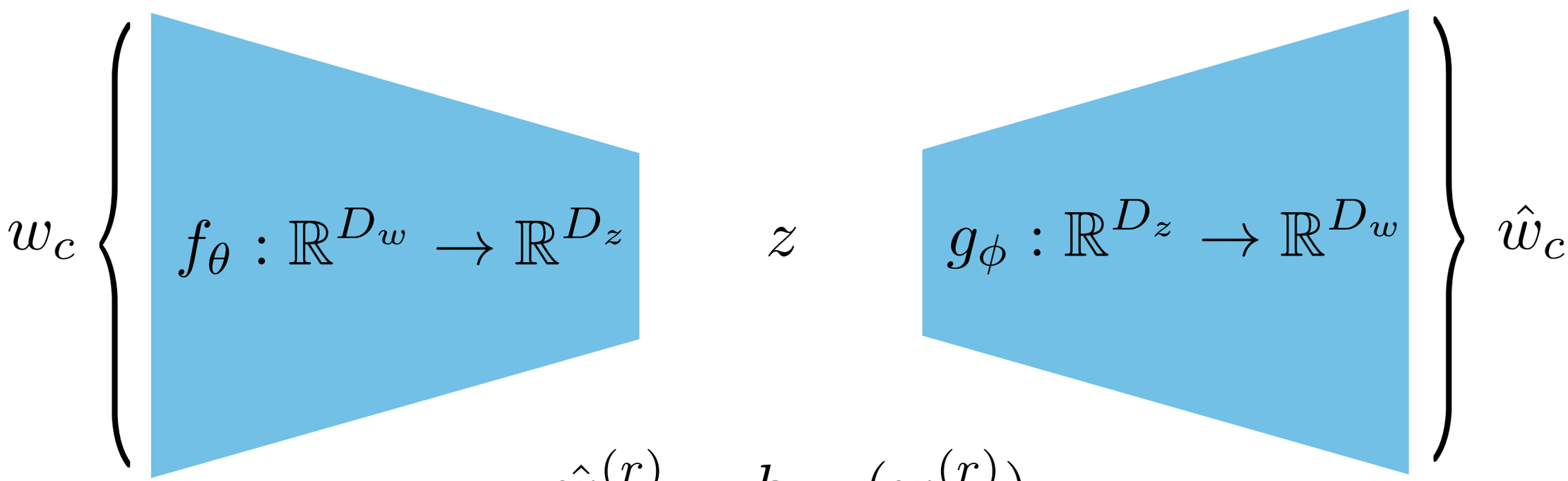
Training objective:

$$\mathcal{L}(\lambda) = \mathbb{E}_q \left[\log p(y|x, g_\phi(z))\right] - D_{KL}(q_{\lambda_z}(z)||p(z)) - D_{KL}(q_{\lambda_\phi}(\phi)||p(\phi))$$

AUTOENCODER INITIALIZATION

- FGE to collect weights $\{w_c^{(r)}\}_{r=1}^{R}$
- Learn a point estimate for the projection function

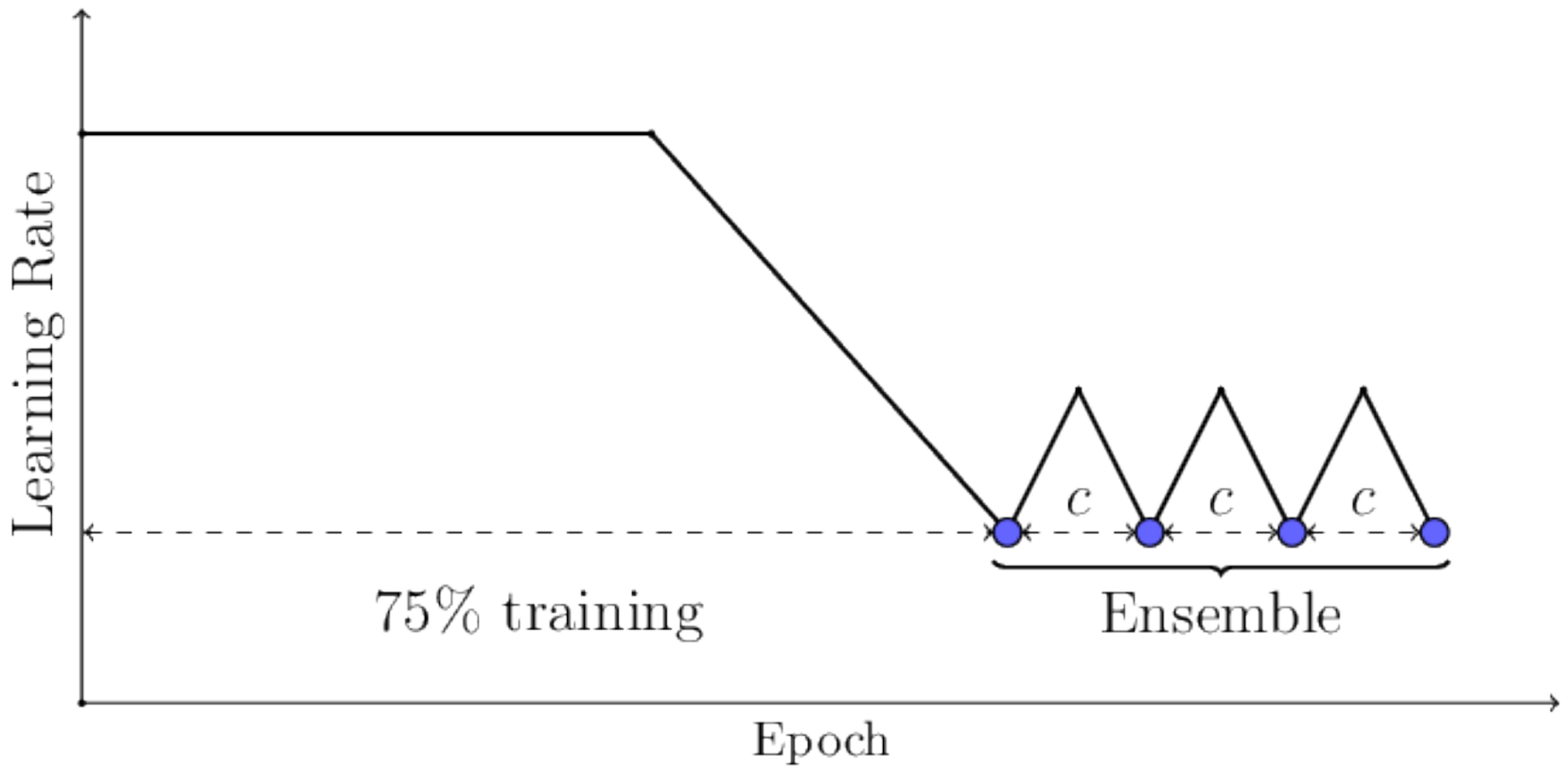
$$h_{\theta,\phi} = g_\phi \circ f_\theta$$



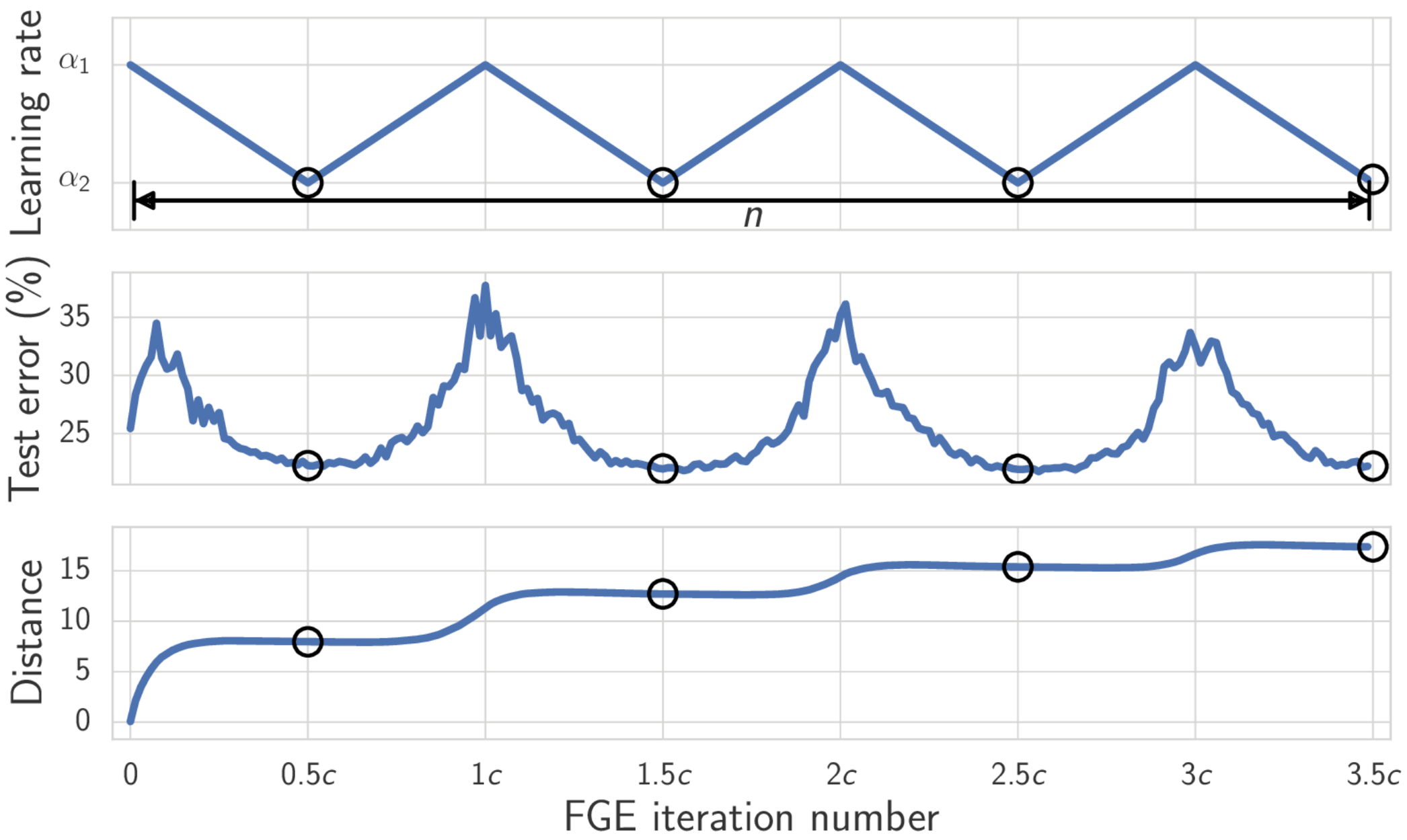
$$\hat{w}^{(r)} = h_{\theta,\phi}(w_c^{(r)})$$

FAST GEOMETRIC ENSEMBLING

- FGE to collect weights $\{w_c^{(r)}\}_{r=1}^{R}$



FAST GEOMETRIC ENSEMBLING



PREDICTION-CONSTRAINED AUTOENCODER

- FGE to collect weights $\{w_c^{(r)}\}_{r=1}^{R}$
- Learn a point estimate for the projection function

$$\mathcal{L}(\theta, \phi) = \frac{1}{R}\sum_{r=1}^{R}\underbrace{\left(w_c^{(r)} - h_{\theta,\phi}(w_c^{(r)}) + \gamma^{(r)}\right)^2}_{\text{Reconstruction Loss}} + \beta\mathbb{E}_{(x,y)\sim\mathcal{D}}\underbrace{\left[\frac{1}{R}\sum_{r=1}^{R}\log p(y|x, h_{\theta,\phi}(w_c^{(r)}))\right]}_{\text{Discriminative Loss of Reconstructed Model}}$$

$\gamma^{(r)} \sim \mathcal{N}(0, 1)$ noise for stability

INFERENCE

$$\mathcal{L}(\lambda) = \mathbb{E}_q \left[\log p(y|x, g_\phi(z))\right] - D_{KL}(q_{\lambda_z}(z)||p(z)) - D_{KL}(q_{\lambda_\phi}(\phi)||p(\phi))$$

$$q_{\lambda_z}(z) = \mathcal{N}(\tilde{\mu}_z, \tilde{\sigma}_z I) \qquad q_{\lambda_\phi}(\phi) = \mathcal{N}(\tilde{\mu}_\phi, \tilde{\sigma}_\phi I)$$

- Initialize $\tilde{\mu}_\phi$ with ϕ^* learned by prediction-constrained autoencoder pretraining, $\tilde{\sigma}_\phi$ with a small value
- Initialize $\tilde{\mu}_z,\ \tilde{\sigma}_z$ randomly

Algorithm 1 Inference for Proj-BNN

Input: observations $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$

1. Gather multiple sets of weights $\{\mathbf{w_c}^{(r)}\}_{r=1}^R$ via Fast Geometric Ensembling (Garipov et al., 2018).

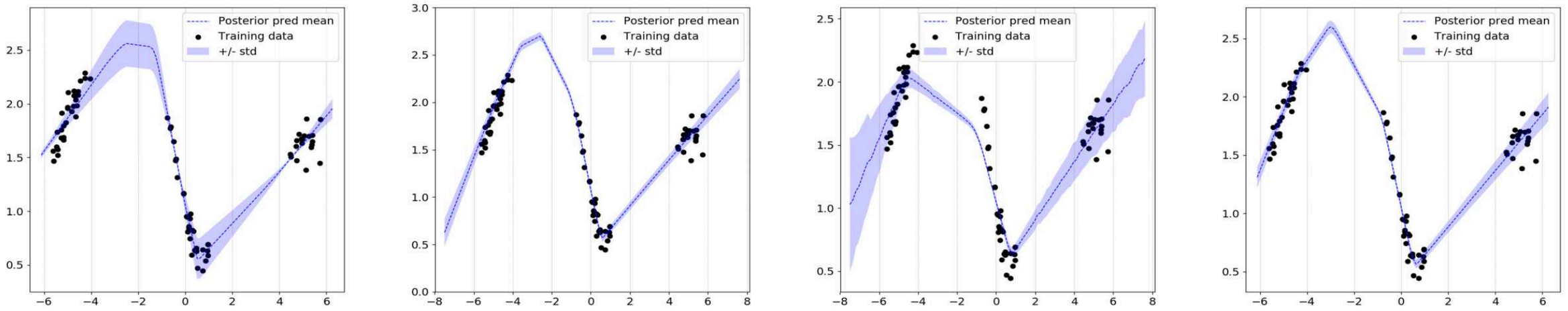
2. Train a prediction-constrained autoencoder $h_{\boldsymbol{\theta},\boldsymbol{\phi}}$ using $\{\mathbf{w_c}^{(r)}\}_{r=1}^R$ as input data to minimize the loss function described in Eq. (6).

3. Perform BBVI to learn an approximate posterior distribution over latent representations $\boldsymbol{z}$ and projection parameters $\boldsymbol{\phi}$.

    - Initialize mean variational parameters $\tilde{\boldsymbol{\mu}}_\phi$ for $\boldsymbol{\phi}$ with the autoencoder solution.
    - Optimize ELBO $\mathcal{L}(\boldsymbol{\lambda})$ in Eq. (5) to obtain $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$ closest to $p(\boldsymbol{z}, \boldsymbol{\phi}|\mathbf{y}, \mathbf{x})$ in terms of KL-divergence.

Result: Approximate posterior $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$

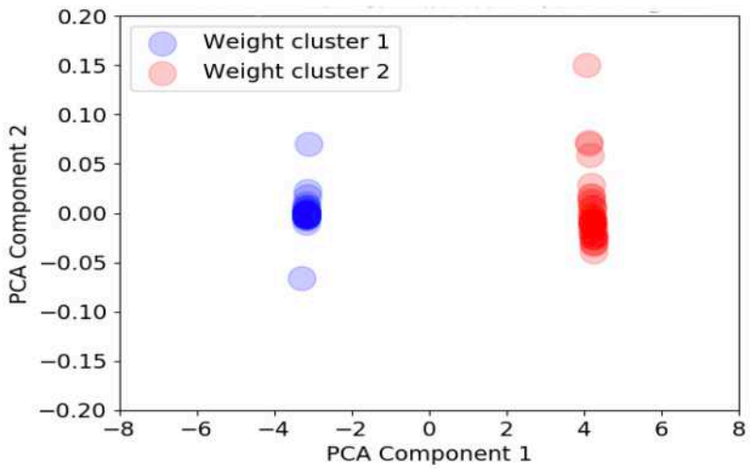
RESULTS: SYNTHETIC REGRESSION



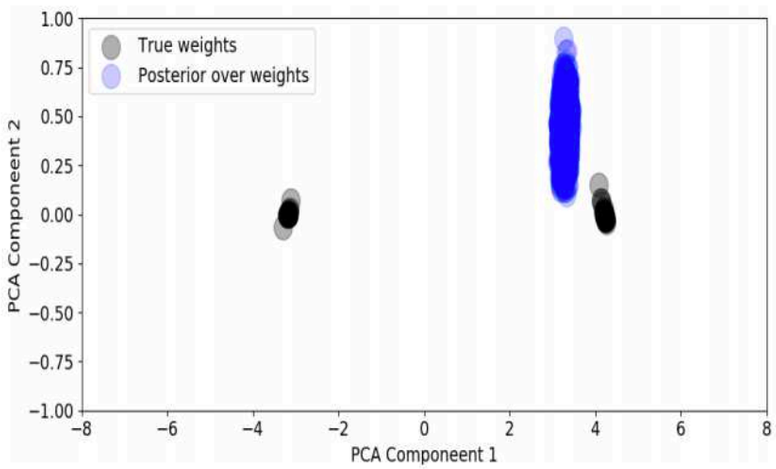
(a) Proj-BNN ($D_z = 2$) (b) BbB (c) MNF (d) MVG

Figure 1: Inferred predictive posterior distribution for a toy data set drawn from a NN with 1-hidden layer, 20 hidden nodes and RBF activation functions. **Proj-BNN is able to learn a plausible predictive mean and better capture predictive uncertainties.**

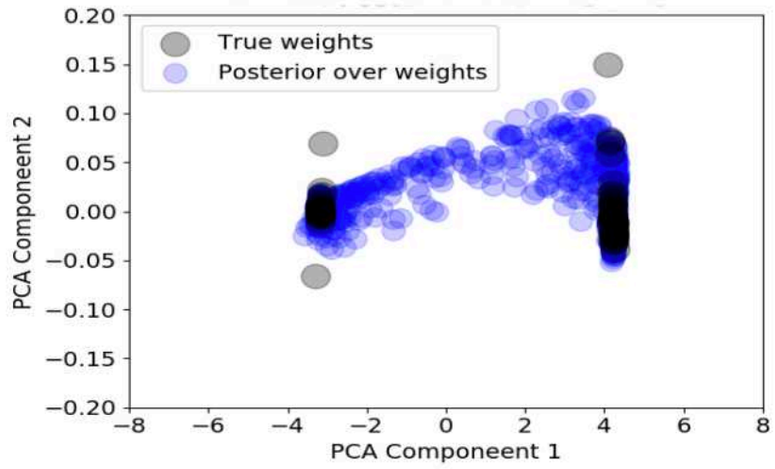
RESULTS: CAPTURING BIMODAL POSTERIOR



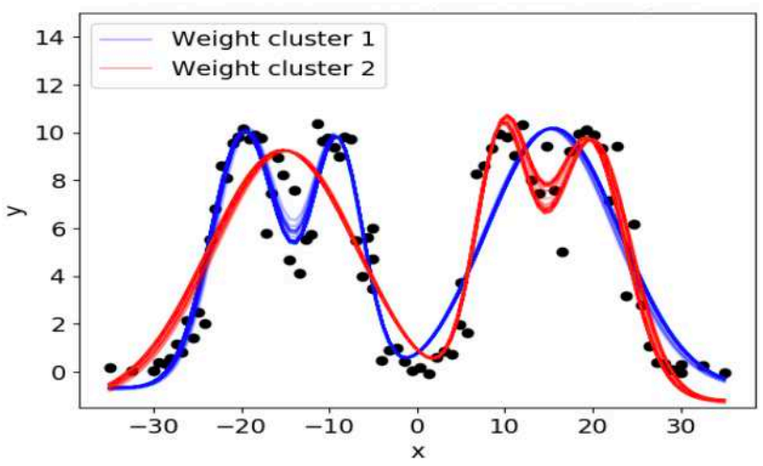
(a) Projection of true weights



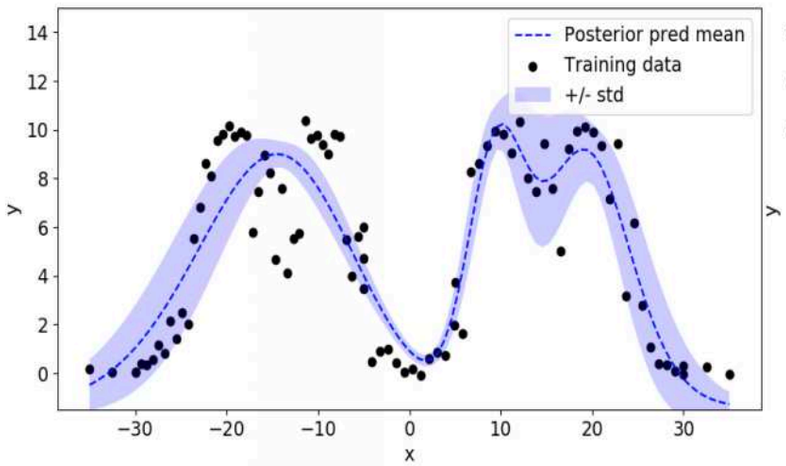
(b) BbB posterior over weights



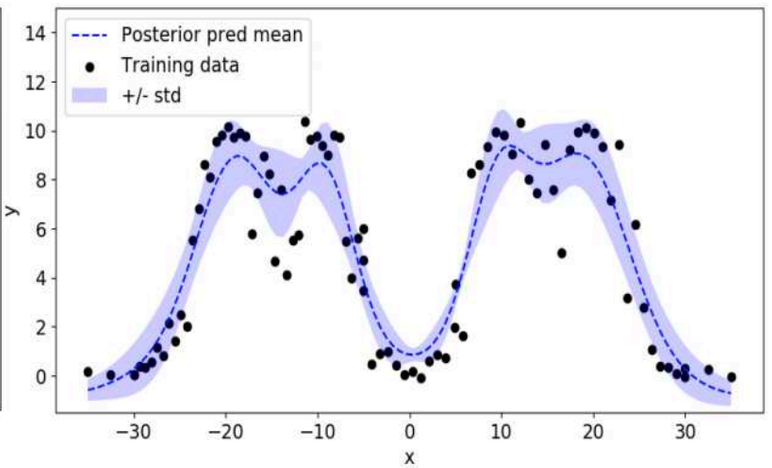
(c) Proj-BNN posterior over weights



(d) Functions from true weights

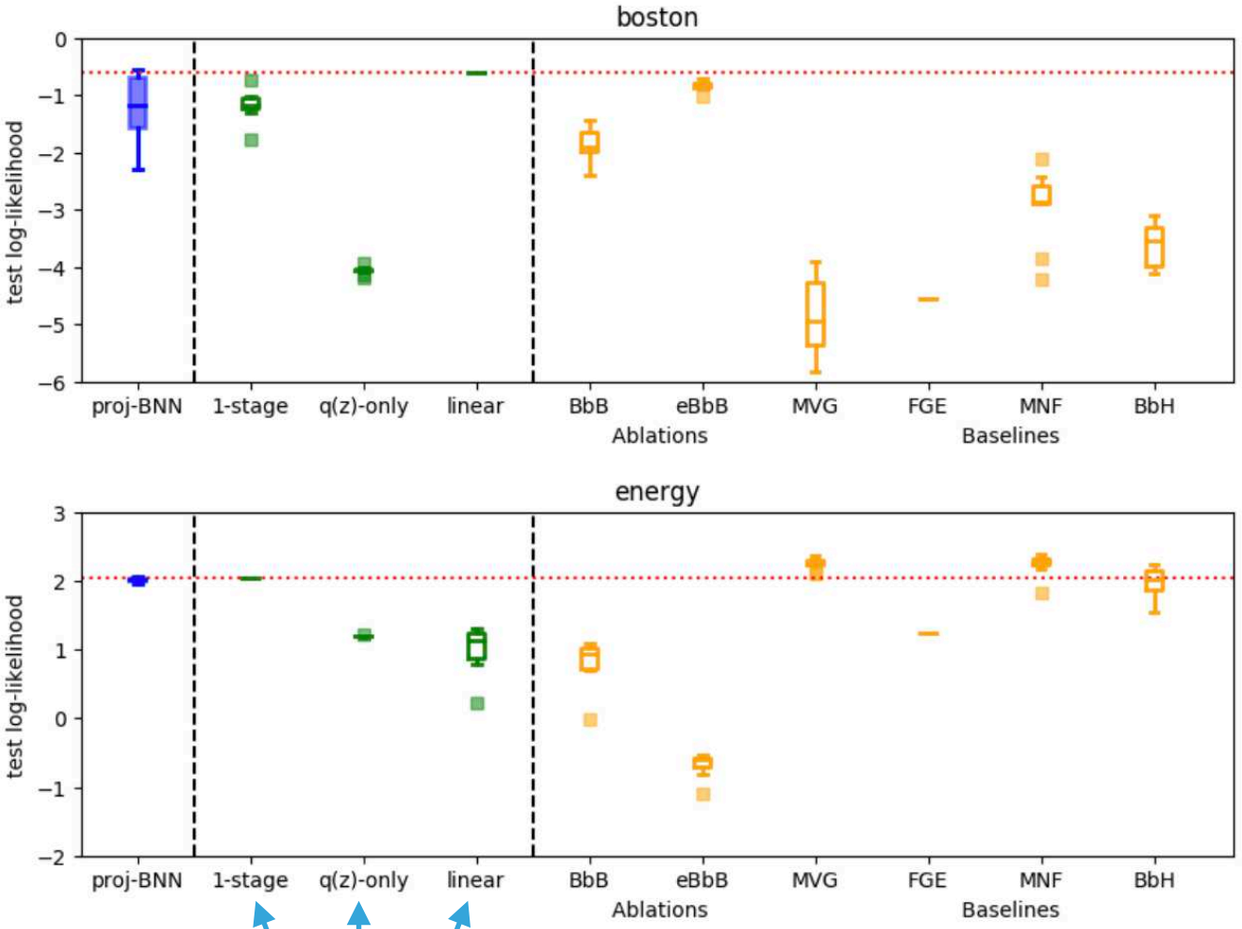


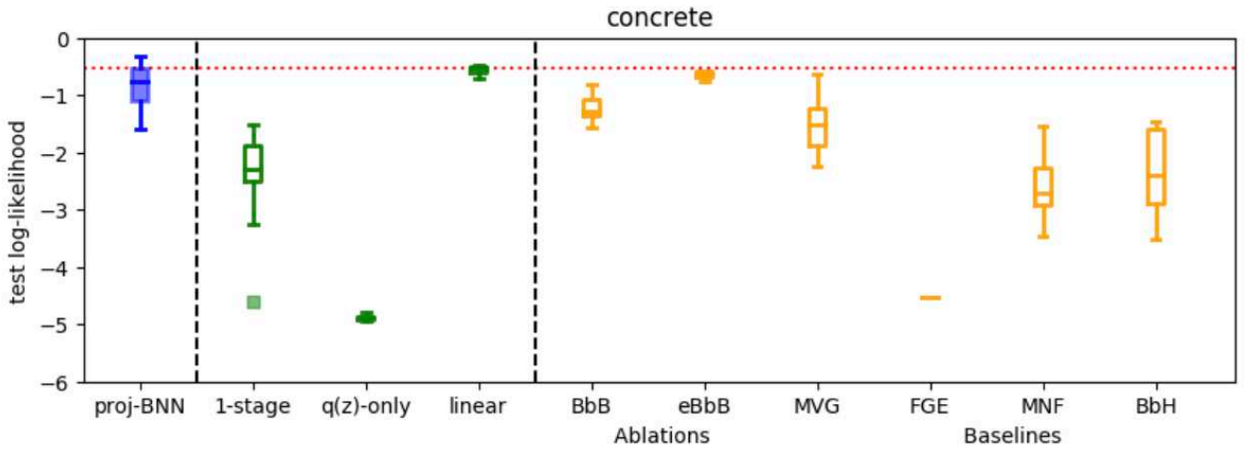
(e) BbB posterior predictive

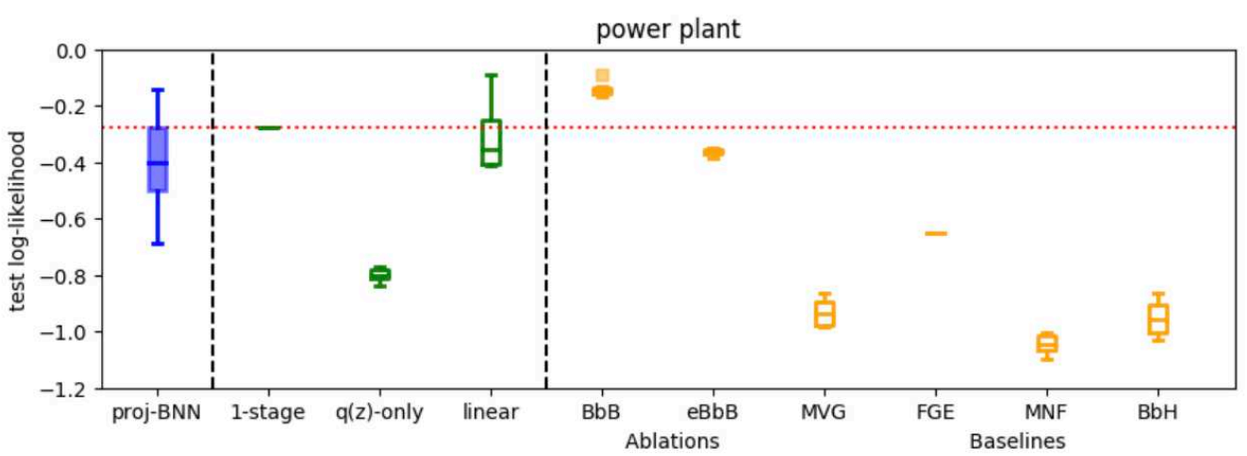


(f) Proj-BNN posterior predictive

RESULTS: UCI REGRESSION







ablation study of ProjBNN

PROJECTED BNNS TAKEAWAYS

- Can capture multiple modes of posterior
- Using VI in latent space (rather than weight space), we can capture correlations between parameters
- Seems to be harder to scale to large-scale image problems

SUMMARY OF BOTH PAPERS

- We can capture geometric properties of the posterior in subspaces (linear or non-linear) of the parameter space
- Even with very low-dimensional subspaces we can get significant improvements in predictions and uncertainty
- More rich subspaces translate into better performance