NEURAL INFORMATION PROCESSING SYSTEMS

Enhancing Diversity in Bayesian Deep Learning via Hyperspherical Energy Minimization of CKA

David Smerkous[1], Qinxun Bai[2], Fuxin Li[1]

[1]Oregon State University
[2]Horizon Robotics

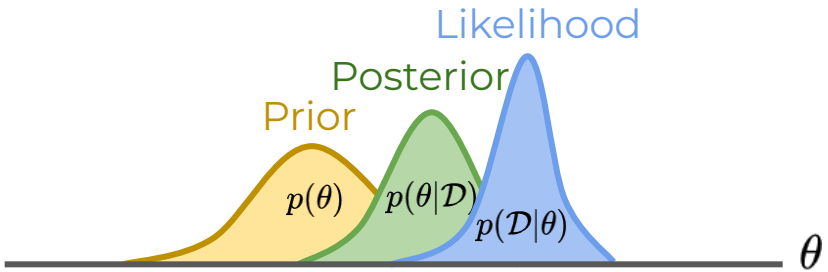
Bayesian Deep Learning

Bayesian Approach: Instead of point estimates, BDL estimates the distribution of the model parameters with prior information

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$$

Uncertainty Estimation: Improves understanding of prediction uncertainty

Applications: Transfer learning, fairness, active learning, reinforcement learning, robotics



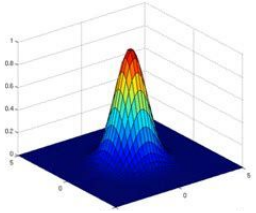
Particle Variational Inference (ParVI)

Basic: MC Dropout, Stochastic Gradient Langevin Dynamics (SGLD), and Ensemble methods.

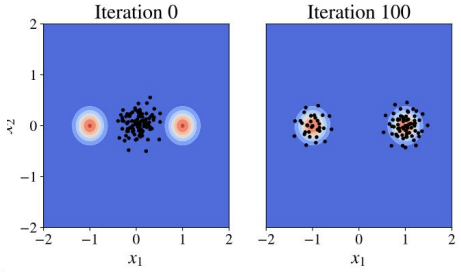
Stein Variational Gradient Descent (SVGD): iterative gradient-based variational inference algorithm on a finite set of particles **M.**

$$x_i^{\ell+1} \leftarrow x_i^{\ell} + \epsilon_\ell \hat{\phi}^*(x_i^{\ell}) \quad \text{where} \quad \hat{\phi}^*(x) = \frac{1}{n}\sum_{j=1}^{n} \underbrace{\left[k(x_j^{\ell}, x)\nabla_{x_j^{\ell}} \log p(x_j^{\ell})\right]}_{\text{Driving}} + \underbrace{\nabla_{x_j^{\ell}} k(x_j^{\ell}, x)}_{\text{Repulsive}}$$

Kernel Selection: Overwhelmingly popular choice is Radial Basis Function (RBF) with median heuristic.



$$K(\vec{x}, \vec{l}') = e^{-\frac{\|\vec{x}-\vec{l}'\|^2}{2\sigma^2}}$$



https://www.cs.toronto.edu/~duvenaud/cookbook/index.html

Issues with RBF and Ensemble Diversity

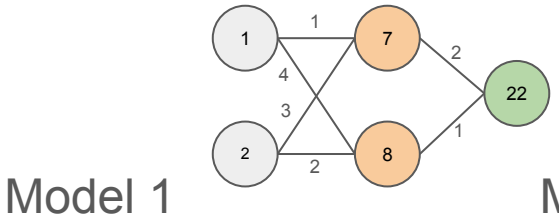
Posterior Representation: Particles that are more diverse better capture the posterior. Supported by methods like SVGD.

Challenge: what should we consider diverse and how should we better measure it?

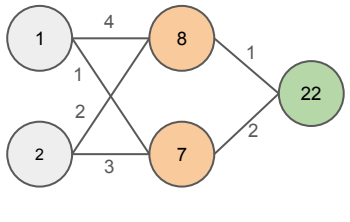
RBF: Works with small networks, but fails with larger ones (D'Angelo et al.)

Current limitations:

- Euclidean based kernels - using L_1,L_2 - lack scale and permutation invariance
- Curse of dimensionality in high-dimensional parameter spaces



Model 1



Model 2

$L_1(M_1, M_2)=10$

Centered Kernel Alignment

Goal: construct a differentiable kernel that is invariant to isotropic scaling and permutations

Solution: Centered Kernel Alignment

CKA between two models

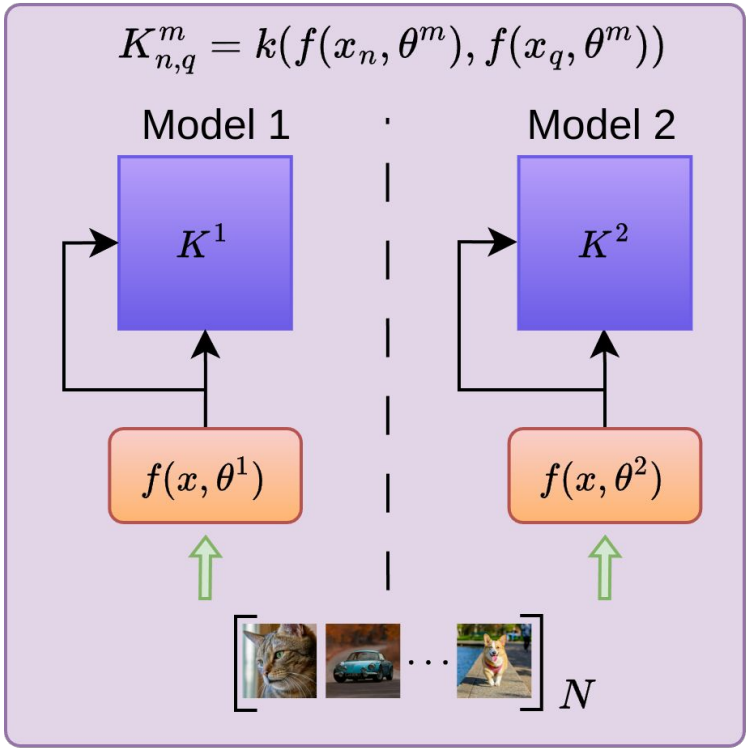
$$\mathrm{HSIC}(K^1, K^2) = \frac{1}{(N-1)^2}\mathrm{tr}(K^1 H K^2 H) \qquad H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^\intercal$$

$$\mathrm{CKA}(K^1, K^2) = \frac{\mathrm{HSIC}(K^1, K^2)}{\sqrt{\mathrm{HSIC}(K^1, K^1)\mathrm{HSIC}(K^2, K^2)}}$$

Ensemble metric: Pairwise CKA

$$\mathrm{CKA}_{\mathrm{pw}}(\mathcal{K}) = \frac{1}{LM(M-1)} \sum_{l=1}^{L} \sum_{\substack{m,m'=1 \\ m \neq m'}}^{M,M} \mathrm{CKA}(K_l^m, K_l^{m'})$$

Gram Matrix Construction

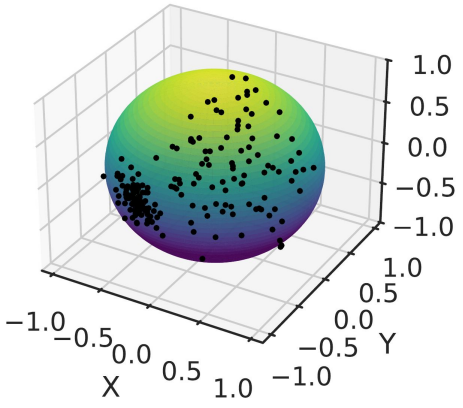


Hyperspherical Energy

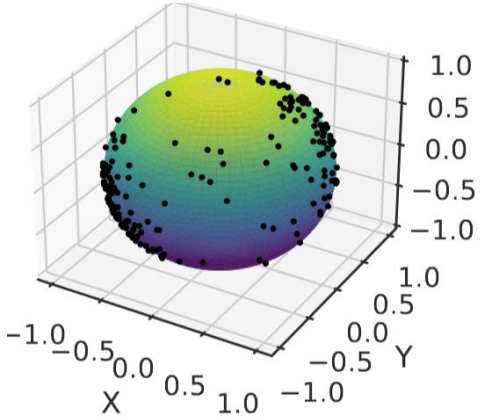
Alternative perspective of pairwise CKA: projection of feature gram matrices vectorized and projected onto unit *hypersphere* and minimization of pairwise cosine similarity.

Potential deficiency: small gradients when model CKAs are similar as gradient of cos is -sin

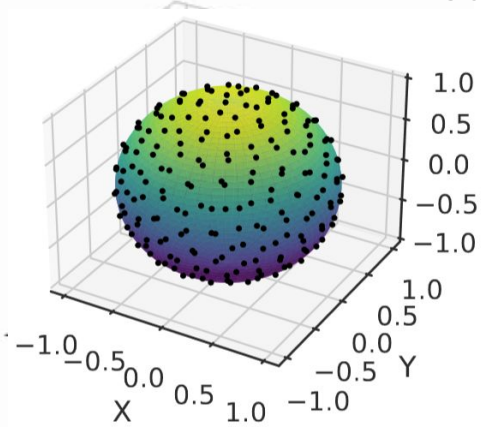
Initial Points



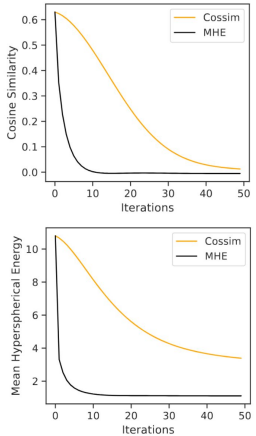
Pairwise Cossim



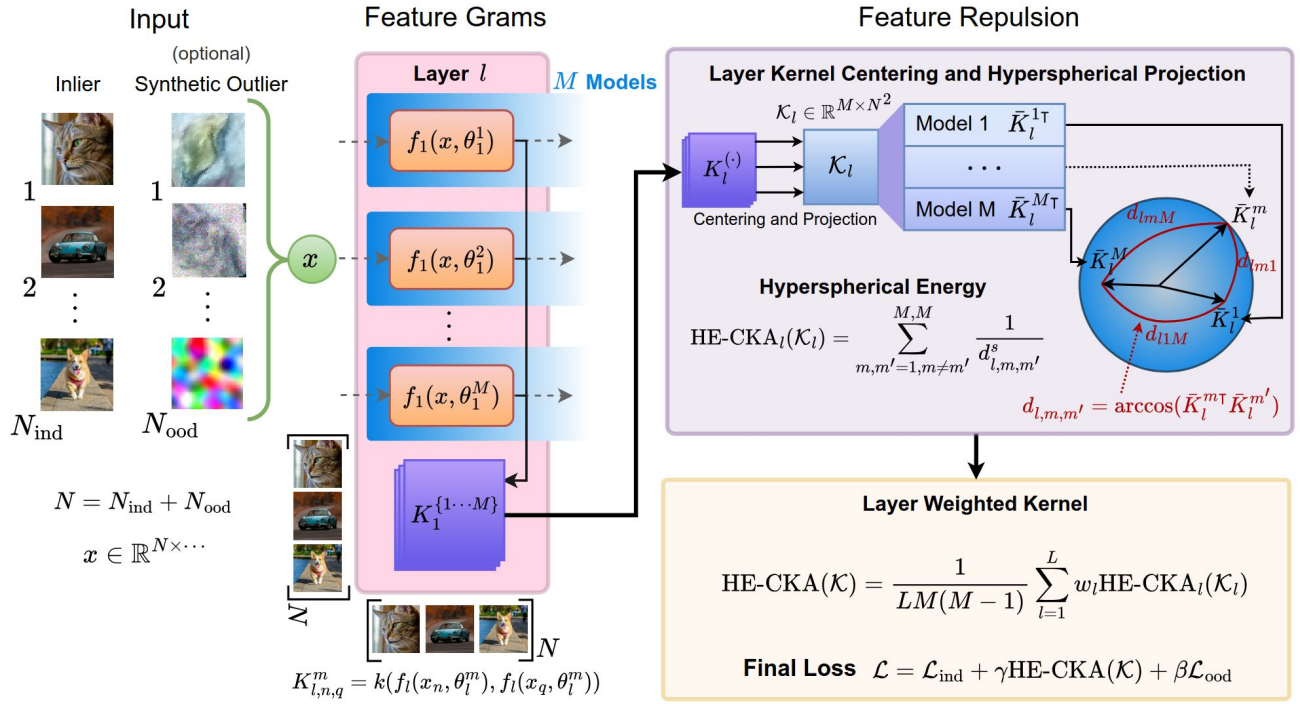
Hyperspherical Energy



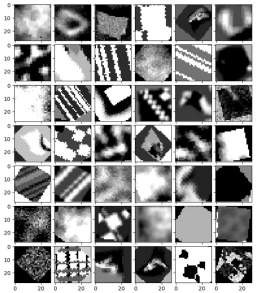
HE vs Cossim



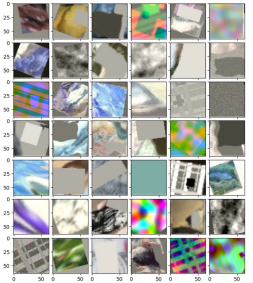
Architecture



Synthetic OOD



MNIST



TinyImgNet

Noise: Perlin, Simplex, Gaussian, random shapes

ID2OOD: augmentations such as blur, elastic transformation, cutout, etc

Synthetic four class classification

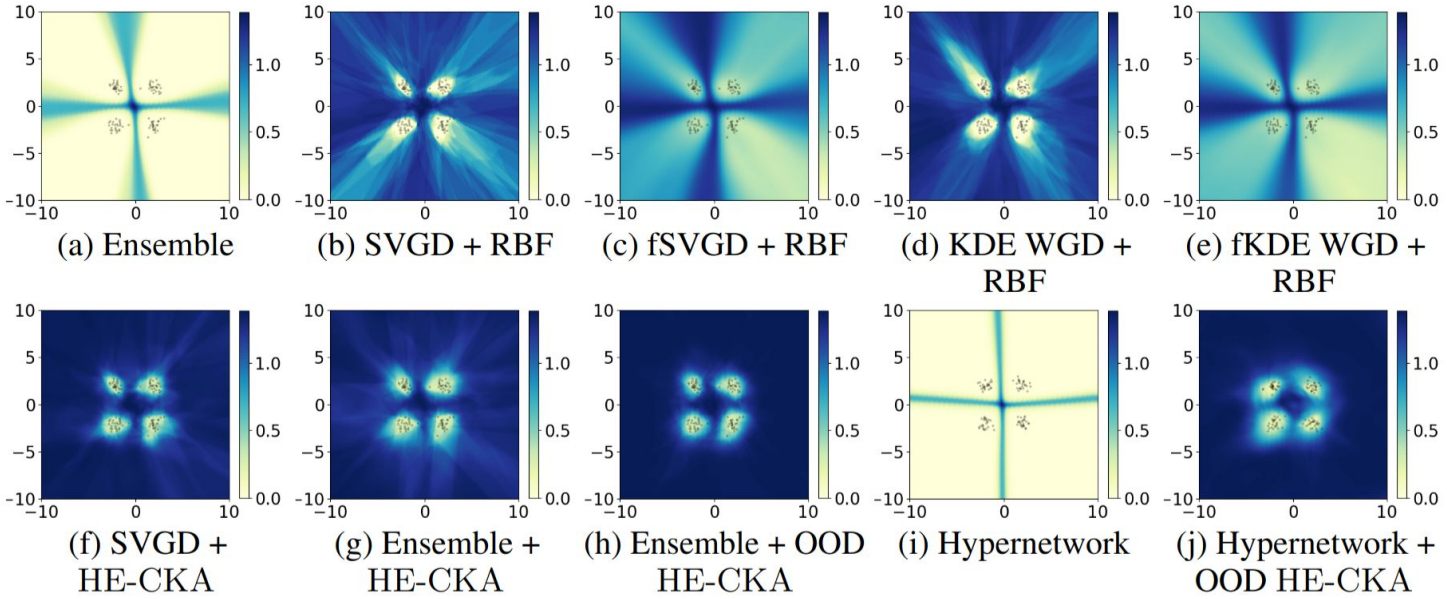
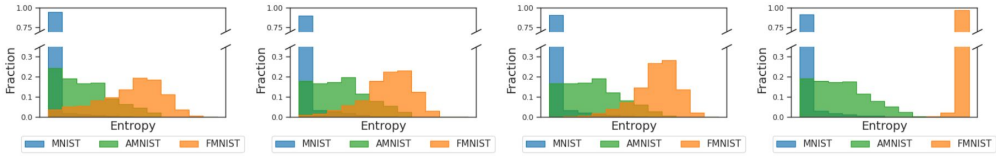


Figure 3: Predictive entropies (PE) on a four-cluster 2D classification task. Darker values indicate higher entropy, lower confidence regions, and lighter values indicate higher confidence regions. (b) and (d) use an RBF kernel on ensemble member weights, whereas (c) and (e) use an RBF kernel on ensemble member outputs. (f) and (g) use the HE-CKA, RBF feature kernel, for feature diversity on inlier points. Both (h) and (j) use HE-CKA and OOD entropy terms. All methods were trained on an ensemble of 30 four layer MLPs for 1k iterations with the same seeds.

Dirty-MNIST and FashionMNIST

Inlier

Outlier



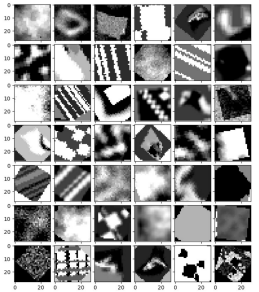
(a) Ensemble (b) SVGD + RBF (c) SVGD + HE-CKA (d) Ensemble + OOD HE-CKA

Figure 6: Predictive softmax entropy between MNIST, Dirty-MNIST (with aleatoric uncertainty), and OOD Fashion-MNIST. Utilizing an ensemble of 5 LeNets. It can be seen that HE-CKA and OOD HE-CKA better separates the inlier Dirty-MNIST from outlier Fashion-MNIST.

Table 1: OOD detection results with inlier Dirty-MNIST and outlier Fashion MNIST, over 5 runs. All models were trained on a LeNet, with HE-CKA and CKA_{pw} utilizing a cosine similarity feature kernel. One exception to predictive entropy (PE) report is DDU, which uses feature space density, indicated by a star, to calculate AUROC Mukhoti et al. (2023). More training details can be found in Appendix C.

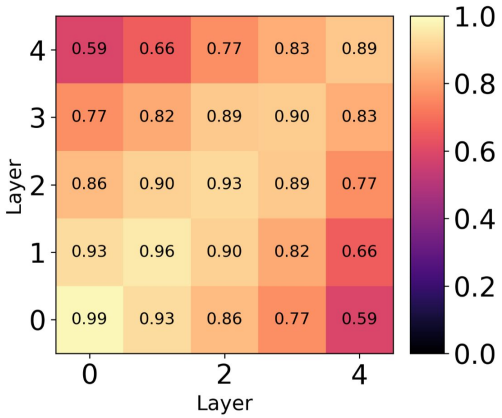
Model	NLL (↓)	Accuracy (↑)	ECE (↓)	AUROC FashionMNIST (↑)	
				PE	MI
DDU	0.278 ± 0.001	82.177 ± 0.032	3.952 ± 0.317	94.168 ± 3.425*	–
Single	$\mathbf{0.272 \pm 0.002}$	82.299 ± 0.166	2.908 ± 0.129	65.935 ± 10.669	50.000 ± 0.000
Ensemble	$\mathbf{0.271 \pm 0.001}$	83.915 ± 0.084	$\mathbf{1.306 \pm 0.098}$	86.095 ± 1.608	96.065 ± 0.798
SVGD+RBF	0.304 ± 0.001	83.560 ± 0.072	3.178 ± 0.076	91.003 ± 1.155	98.083 ± 0.516
SVGD+CKA_{pw}	0.377 ± 0.003	82.351 ± 0.150	7.359 ± 0.211	89.195 ± 4.260	$\mathbf{99.207 \pm 0.160}$
SVGD+HE-CKA	0.298 ± 0.002	83.879 ± 0.110	2.846 ± 0.153	94.380 ± 1.332	$\mathbf{99.213 \pm 0.147}$
Hypernet	0.278 ± 0.003	81.157 ± 0.174	4.253 ± 0.092	46.393 ± 3.545	64.856 ± 2.768
Hypernet+OOD HE-CKA	0.325 ± 0.014	82.398 ± 0.628	3.058 ± 0.665	98.073 ± 0.951	77.548 ± 9.526
Ensemble+HE-CKA	0.306 ± 0.001	83.684 ± 0.029	3.174 ± 0.120	94.656 ± 1.095	98.866 ± 0.148
Ensemble+OOD HE-CKA	0.277 ± 0.001	$\mathbf{84.090 \pm 0.049}$	1.712 ± 0.061	$\mathbf{99.996 \pm 0.001}$	$\mathbf{99.742 \pm 0.506}$

Synthetic OOD

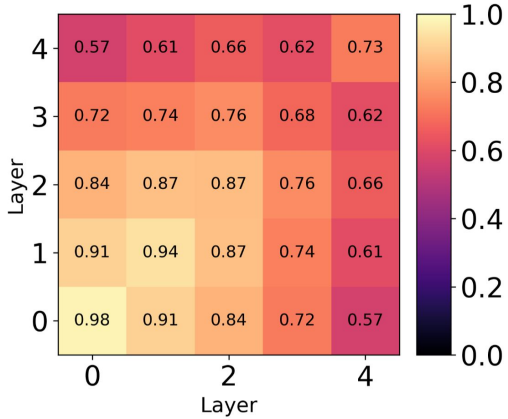


Dirty MNIST Ensemble CKA Plots

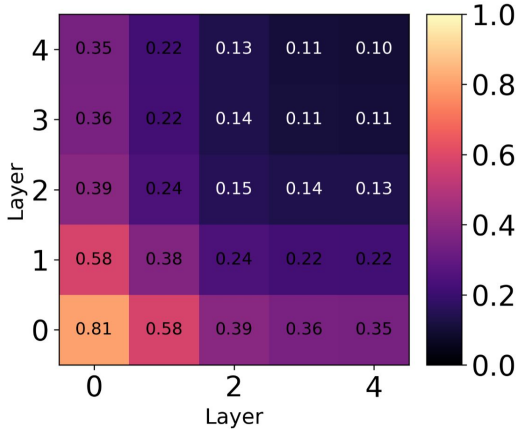
Deep Ensemble



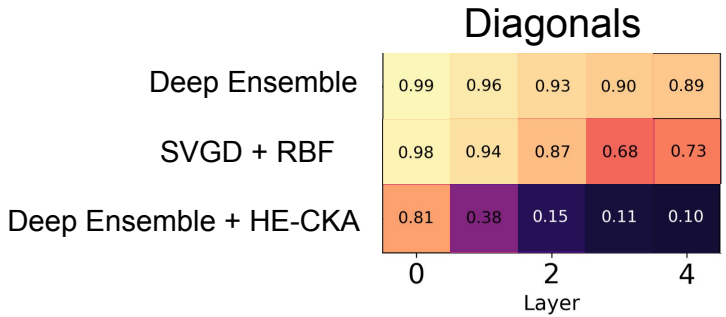
SVGD + RBF



Deep Ensemble + HE-CKA



Diagonals



Used unbiased CKA estimator for plotting. Each square represents the mean pairwise CKA in the ensemble for that layer

CIFAR10 / TinyImageNet

Table 3: OOD results on CIFAR-10 vs SVHN. Methods used a ResNet18 ensemble of size 5. $(\cdot)$ indicates ensemble size.

Model	OOD Method	NLL (↓)	Accuracy (↑)	ECE (↓)	AUROC PE SVHN (↑)
WideResNet28-10+SN Mukhoti et al. [2023] (1)	GMM	–	95.97	0.85	97.86
Ensemble	PE	0.122	**96.34**	1.08	96.18
SVGD+RBF	PE	0.143	95.71	1.19	95.37
SVGD+CKA_{pw}	PE	0.125	96.25	**0.41**	96.07
SVGD+HE-CKA	PE	0.124	96.23	0.63	96.01
Ensemble+HE-CKA	PE	**0.120**	96.23	0.59	96.71
Ensemble+OOD HE-CKA	PE	0.123	96.24	0.58	**99.86**

Table 4: Performance of a five member ResNet18 ensemble trained on TinyImageNet. All models utilized a pretrained deep ensemble with no repulsive term, then fine tuned for 30 epochs for each method (including deep ensemble). Methods utilizing CKA_{pw} and HE-CKA utilized a linear feature kernel.

Model	NLL (↓)	ID Accuracy (↑)	ECE (↓)	AUROC PE (↑)		
				SVHN	CIFAR 10/100	Textures (DTD)
Ensemble	0.775	62.95	8.90	89.81	66.85/67.33	68.96
SVGD+RBF	0.926	61.87	16.10	92.76	72.23/73.73	65.67
SVGD+CKA_{pw}	0.835	60.15	8.26	94.08	78.40/79.48	66.48
SVGD+HE-CKA	**0.732**	61.36	**3.71**	94.10	72.05/72.86	70.75
Ensemble+HE-CKA	0.784	**63.10**	9.82	92.65	72.13/71.68	70.69
Ensemble+OOD HE-CKA	0.786	61.88	8.02	**99.31**	**81.56/87.64**	**90.94**

CKA for Bayesian Deep Learning

CKA: Provides a more intuitive kernel to measure model similarity compared to RBF

HE: Realized pairwise CKA can be reformulated as a hyperspherical energy optimization problem.

Synthetic Outlier: Increase feature diversity on obvious outlier examples and provide negative signals for likelihood. ***Significant*** *improvement in outlier detection* without sacrificing inlier performance.

Limitations:

- Larger memory requirements (feature gram matrices/gradients)
- Requires tuning of multiple hyperparameters for good performance (ex: layer weighting)
- Hyperparameters are sensitive to dataset and particle number

Enhancing Diversity in Bayesian Deep Learning via Hyperspherical Energy Minimization of CKA

David Smerkous, Qinxun Bai, Fuxin Li

Code publicly available at

https://github.com/Deep-Machine-Vision/he-cka-ensembles