Richer Bayesian Last Layers with Subsampled NTK Features

ICML 2026

Presenter: Dongyoon Kim

26.07.16

Preliminaries — Notation

Two roles for $\mathbf{x}$ (paper's convention).

$$\underbrace{\mathbf{x}_i \in \mathbb{R}^d}_{\text{a single input point}} \quad , \qquad \underbrace{\mathbf{x} := (\mathbf{x}_1, \ldots, \mathbf{x}_N)}_{\text{training set (used as subscript)}} \quad , \qquad \underbrace{\mathbf{x}' := (\mathbf{x}'_1, \ldots, \mathbf{x}'_{N'})}_{\text{test set (used as subscript)}} .$$

Function arguments take single points; *matrix subscripts* take collections.

$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$	training data
$f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}, \ \boldsymbol{\theta} \in \mathbb{R}^P$	neural network with parameters $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}$	MAP estimate
$\boldsymbol{\phi}^p(\mathbf{x}_i) := \nabla_{\boldsymbol{\theta}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) \in \mathbb{R}^p$	eNTK feature of *one* point
$\boldsymbol{\Phi}^p_{\mathbf{x}} \in \mathbb{R}^{N \times p}$	feature matrix (rows are $\boldsymbol{\phi}^p(\mathbf{x}_i)^\top$)
$k^p_{\mathbf{x},\mathbf{x}'} := \boldsymbol{\Phi}^p_{\mathbf{x}} \boldsymbol{\Phi}^{p\top}_{\mathbf{x}'} \in \mathbb{R}^{N \times N'}$	eNTK Gram matrix (train $\times$ test)
$\sigma^2, \ S^{\text{ntk}}_{\mathbf{x}',\mathbf{x}'} \in \mathbb{R}^{N' \times N'}$	noise variance; posterior-predictive cov.
$\boldsymbol{\phi}^r(\mathbf{x}_i) \in \mathbb{R}^r, \ k^r_{\mathbf{x},\mathbf{x}'}, \ S^{\text{bll}}_{\mathbf{x}',\mathbf{x}'}$	last-layer analogues (BLL), r = last-layer width

Three sizes to keep separate: $d =$ input dim, $N/N' =$ # train/test points, $p = m + r =$ # params.

What is Neural Tangent Kernel?

The NTK *emerges naturally* from gradient-descent training dynamics. (Shown here for one point $(\mathbf{x}_i, y_i)$; sum over i for the full dataset.)

Step 1 — Continuous-time gradient flow. $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_k)$ becomes, as $\eta \to 0$,

$$\dot{\boldsymbol{\theta}}(t) = -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}(t)).$$

Step 2 — MSE loss. With $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}(f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2$,

$$\dot{\boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_i) \big(f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i\big).$$

Step 3 — Chain rule: output evolution at a test point $\mathbf{x}'_j$.

$$\dot{f}_{\boldsymbol{\theta}}(\mathbf{x}'_j) = \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}'_j)^\top \dot{\boldsymbol{\theta}} = -\underbrace{\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}'_j)^\top \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_i)}_{=: H_{\boldsymbol{\theta}}(\mathbf{x}'_j, \mathbf{x}_i)} \big(f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i\big).$$

Neural Tangent Kernel and eNTK

Neural Tangent Kernel (NTK).

In the *infinite-width* limit, training a neural network by gradient descent is equivalent to **kernel regression** with kernel

$$k_{\mathrm{NTK}}(\mathbf{a}, \mathbf{b}) = \mathbb{E}_{\boldsymbol{\theta}_0}\big[\langle \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{a}),\, \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{b})\rangle\big], \qquad \mathbf{a}, \mathbf{b} \in \mathbb{R}^d.$$

Empirical NTK (eNTK).

Finite-width, practical analogue: evaluate the tangent kernel at the trained (MAP) parameters $\hat{\boldsymbol{\theta}}$,

$$k^p(\mathbf{a}, \mathbf{b}) := \langle \nabla_{\boldsymbol{\theta}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{a}),\, \nabla_{\boldsymbol{\theta}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{b})\rangle = \boldsymbol{\phi}^p(\mathbf{a})^\top \boldsymbol{\phi}^p(\mathbf{b}).$$

Reading the notation. The kernel *function* $k^p(\mathbf{a}, \mathbf{b})$ takes two single points; stacking over training / test sets gives the Gram *matrix* $k^p_{\mathbf{x},\mathbf{x}'} \in \mathbb{R}^{N \times N'}$.

From Neural Network to Bayesian Linear Model

Step 1 — Linearize $f_{\boldsymbol{\theta}}$ around $\hat{\boldsymbol{\theta}}$. For any input $\mathbf{a} \in \mathbb{R}^d$,

$$f_{\boldsymbol{\theta}}(\mathbf{a}) \approx \underbrace{f_{\hat{\boldsymbol{\theta}}}(\mathbf{a})}_{\text{fixed scalar}} + \underbrace{\boldsymbol{\phi}^p(\mathbf{a})^\top}_{\text{fixed vector in } \mathbb{R}^p =: \Delta\boldsymbol{\theta},\ \text{unknown in } \mathbb{R}^p} \underbrace{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})}.$$

Everything is computed once at $\hat{\boldsymbol{\theta}}$; the only unknown is $\Delta\boldsymbol{\theta}$.

Step 2 — Bayesian linear regression.
Gaussian prior on the deviation and Gaussian likelihood:

$$\Delta\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p), \qquad y_i \mid \mathbf{x}_i, \boldsymbol{\theta} \sim \mathcal{N}\big(f_{\boldsymbol{\theta}}(\mathbf{x}_i),\, \sigma^2\big)\,.$$

Weight-space $\leftrightarrow$ Function-space (GP view)

Compute mean and covariance directly. For any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, writing $f^{lin}(\mathbf{a}) = f_{\hat{\boldsymbol{\theta}}}(\mathbf{a}) + \boldsymbol{\phi}^p(\mathbf{a})^\top \Delta\boldsymbol{\theta}$ with $\Delta\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_p)$:

$$\mathbb{E}[f^{lin}(\mathbf{a})] = f_{\hat{\boldsymbol{\theta}}}(\mathbf{a}) + \boldsymbol{\phi}^p(\mathbf{a})^\top \underbrace{\mathbb{E}[\Delta\boldsymbol{\theta}]}_{=\mathbf{0}} = f_{\hat{\boldsymbol{\theta}}}(\mathbf{a}),$$

$$\mathrm{Cov}\big(f(\mathbf{a}), f(\mathbf{b})\big) = \boldsymbol{\phi}^p(\mathbf{a})^\top \underbrace{\mathbb{E}[\Delta\boldsymbol{\theta}\,\Delta\boldsymbol{\theta}^\top]}_{=\mathbf{I}_p} \boldsymbol{\phi}^p(\mathbf{b}) = k^p(\mathbf{a}, \mathbf{b}).$$

Joint Gaussianity. For any finite set of points, the corresponding output vector is a linear function of the Gaussian $\Delta\boldsymbol{\theta}$, hence jointly Gaussian. This is the definition of a Gaussian process:

$$\boxed{f \;\sim\; \mathcal{GP}\big(f_{\hat{\boldsymbol{\theta}}},\; k^p\big).}$$

Preliminaries — Posterior Predictive

Step 3 — Condition on $\mathcal{D}$ (Gaussian conjugacy). Recall each kernel block is a feature outer product: $k^p_{\mathbf{a},\mathbf{b}} = \mathbf{\Phi}^p_{\mathbf{a}} \mathbf{\Phi}^{p\top}_{\mathbf{b}}$. Under the GP prior, the joint distribution is Gaussian:

$$\begin{pmatrix} f^{\text{lin}}(\mathbf{x}') \\ {\scriptstyle N'\times 1} \\ \mathbf{y} \\ {\scriptstyle N\times 1} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}') \\ f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} k^p_{\mathbf{x}',\mathbf{x}'} & k^p_{\mathbf{x}',\mathbf{x}} \\ {\scriptstyle N'\times N'} & {\scriptstyle N'\times N} \\ k^p_{\mathbf{x},\mathbf{x}'} & k^p_{\mathbf{x},\mathbf{x}} + \sigma^2 \mathbf{I}_N \\ {\scriptstyle N\times N'} & {\scriptstyle N\times N} \end{pmatrix} \right).$$

Gaussian conditioning identity $\Rightarrow$ predictive covariance:

$$\text{Cov}[f^{\text{lin}}(\mathbf{x}') \,|\, \mathcal{D}] \;=\; \underbrace{k^p_{\mathbf{x}',\mathbf{x}'} \;-\; k^p_{\mathbf{x}',\mathbf{x}} \Big(k^p_{\mathbf{x},\mathbf{x}} + \sigma^2 \mathbf{I}_N \Big)^{-1} k^p_{\mathbf{x},\mathbf{x}'}}_{= S^{\text{ntk}}_{\mathbf{x}',\mathbf{x}'} \,\in\, \mathbb{R}^{N'\times N'}}.$$

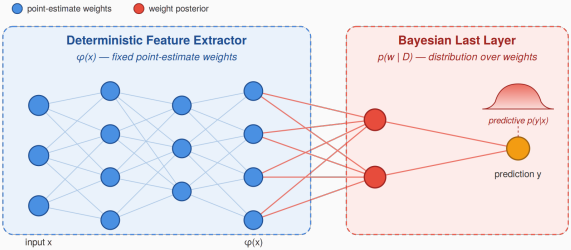
Bottleneck. $N \times N$ inverse above. **Woodbury identity** rewrites it as

$$\left(\sigma^2 \mathbf{I}_N + \mathbf{\Phi}^p \mathbf{\Phi}^{p\top}\right)^{-1} = \sigma^{-2} \mathbf{I}_N - \sigma^{-2} \mathbf{\Phi}^p \underbrace{\left(\sigma^2 \mathbf{I}_p + \mathbf{\Phi}^{p\top} \mathbf{\Phi}^p\right)}_{p\times p}{}^{-1} \mathbf{\Phi}^{p\top},$$

swapping $N \times N$ for a $p \times p$ inverse. For full eNTK $p \gg N$ (millions of params) — Woodbury doesn't help.

Motivation

Bayesian Last Layers (BLLs) provide a convenient and computationally efficient way to estimate uncertainty in neural networks.



Key problem: BLL underestimates epistemic uncertainty because it applies a Bayesian treatment only to the final layer, ignoring uncertainty induced by earlier layers.

Methodology — Motivation

Two extremes so far.

	feature dim.	cost	uncertainty
Standard BLL	r	cheap	*under*estimates
Full eNTK / LLA	$p = m{+}r$	infeasible	richer

Goal. eNTK-quality uncertainty at BLL cost.

Key observation. The eNTK spectrum concentrates on few directions $\Rightarrow$ many of the p feature dimensions carry little information.
(*Spectrum* = eigenvalues of $\mathbf{\Phi}_{\mathbf{x}}^{p}\mathbf{\Phi}_{\mathbf{x}}^{p\top}$, i.e. squared singular values of $\mathbf{\Phi}_{\mathbf{x}}^{p}$; rapid decay $\Rightarrow$ only a few singular directions matter.)

Idea. Approximate non-last-layer features by last-layer features:

$$\boldsymbol{\phi}^{m}(\mathbf{x}_i) \;\approx\; \mathbf{A}\,\boldsymbol{\phi}^{r}(\mathbf{x}_i), \qquad \mathbf{A} \in \mathbb{R}^{m\times r}.$$

Inference lives in the r-dim last-layer space, richer than plain BLL.

Methodology — eNTK Feature Decomposition

Split parameters by layer. With a linear final layer:

$$\boldsymbol{\theta} = \mathrm{vec}(\boldsymbol{\theta}^{\leq l}, \boldsymbol{\theta}^{l+1}) \in \mathbb{R}^p, \qquad \boldsymbol{\theta}^{\leq l} \in \mathbb{R}^m, \quad \boldsymbol{\theta}^{l+1} \in \mathbb{R}^r, \quad p = m + r.$$

Last-layer (NNGP) features. Gradient w.r.t. the last-layer weights and bias is exactly the last hidden layer's post-activation (augmented with a 1 for the bias):

$$\boldsymbol{\phi}^r(\mathbf{x}_i) \;:=\; \nabla_{\boldsymbol{\theta}^{l+1}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) \;=\; \mathrm{vec}\big(\mathbf{a}^l_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i),\, 1\big) \in \mathbb{R}^r.$$

Non-last-layer features. $\boldsymbol{\phi}^m(\mathbf{x}_i) := \nabla_{\boldsymbol{\theta}^{\leq l}} f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i) \in \mathbb{R}^m$.

Full eNTK features decompose as:

$$\boldsymbol{\phi}^p(\mathbf{x}_i) \;=\; \mathrm{vec}\big(\boldsymbol{\phi}^m(\mathbf{x}_i),\, \boldsymbol{\phi}^r(\mathbf{x}_i)\big) \in \mathbb{R}^{m+r}. \tag{6}$$

Standard **BLL** keeps only $\boldsymbol{\phi}^r$; standard **eNTK/LLA** uses the full $\boldsymbol{\phi}^p$. We approximate the $\boldsymbol{\phi}^m$ contribution using $\boldsymbol{\phi}^r$.

Methodology — Approximating ϕ^m by ϕ^r

Assumption. There exists $\mathbf{A} \in \mathbb{R}^{m \times r}$ such that $\boldsymbol{\phi}^m(\mathbf{x}_i) \approx \mathbf{A}\,\boldsymbol{\phi}^r(\mathbf{x}_i)$ for every training point. **(7)**

Fit $\mathbf{A}$ by least squares on the training data:

$$\min_{\mathbf{A}} \sum_{i=1}^{N} \left\| \boldsymbol{\phi}^m(\mathbf{x}_i) - \mathbf{A}\,\boldsymbol{\phi}^r(\mathbf{x}_i) \right\|_2^2. \tag{8}$$

Whenever $N \geq r$, $\boldsymbol{\Phi}_{\mathbf{x}}^{r\top}\boldsymbol{\Phi}_{\mathbf{x}}^{r}$ has full rank and

$$\mathbf{A} \;=\; \boldsymbol{\Phi}_{\mathbf{x}}^{m\top}\boldsymbol{\Phi}_{\mathbf{x}}^{r}\big(\boldsymbol{\Phi}_{\mathbf{x}}^{r\top}\boldsymbol{\Phi}_{\mathbf{x}}^{r}\big)^{-1} \in \mathbb{R}^{m \times r}. \tag{9}$$

Stack into a single lifting matrix.

$$\mathbf{B} := \begin{pmatrix} \mathbf{A} \\ \mathbf{I}_r \end{pmatrix} \in \mathbb{R}^{(m+r)\times r} \quad\Longrightarrow\quad \boldsymbol{\phi}^p(\mathbf{x}_i) \;\approx\; \mathbf{B}\,\boldsymbol{\phi}^r(\mathbf{x}_i) \;=:\; \boldsymbol{\phi}^B(\mathbf{x}_i). \tag{10, 11}$$

Equivalently $\boldsymbol{\Phi}_{\mathbf{x}}^{p} \approx \boldsymbol{\Phi}_{\mathbf{x}}^{r}\mathbf{B}^{\top}$.

Methodology — Modified Posterior Covariance

Approximate kernel matrix. $k^B_{\mathbf{x},\mathbf{x}'} := \mathbf{\Phi}^r_{\mathbf{x}}\,\mathbf{B}^\top\mathbf{B}\,\mathbf{\Phi}^{r\top}_{\mathbf{x}'} \in \mathbb{R}^{N\times N'}$. The $r \times r$ Gram $\mathbf{B}^\top\mathbf{B}$ acts as a data-driven metric on last-layer features, replacing the identity used by standard BLL.

From full eNTK to Rich-BLL: substitute $k^p \to k^B$.

$$\underbrace{S^{\mathrm{ntk}}_{\mathbf{x}',\mathbf{x}'} = k^p_{\mathbf{x}',\mathbf{x}'} - k^p_{\mathbf{x}',\mathbf{x}}\left(k^p_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}_N\right)^{-1}k^p_{\mathbf{x},\mathbf{x}'}}_{\text{full eNTK}} \xrightarrow{k^p \to k^B} \underbrace{S^B_{\mathbf{x}',\mathbf{x}'} = k^B_{\mathbf{x}',\mathbf{x}'} - k^B_{\mathbf{x}',\mathbf{x}}\left(k^B_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}_N\right)^{-1}k^B_{\mathbf{x},\mathbf{x}'}}_{\text{Rich-BLL eNTK}}$$

Comparison at a glance.

	features	Gram/metric	effective dim.
Standard BLL	$\boldsymbol{\phi}^r$	$\mathbf{I}_r$	r
Full eNTK / LLA	$\boldsymbol{\phi}^p$	$\mathbf{I}_p$	$p = m{+}r$
Rich-BLL ($\boldsymbol{\phi}^B$)	$\mathbf{B}\boldsymbol{\phi}^r$	$\mathbf{B}^\top\mathbf{B}$	r

Methodology — Cholesky Reformulation

Predictive covariance in the reduced feature space.

$$S^B_{\mathbf{x}',\mathbf{x}'} = \mathbf{\Phi}^r_{\mathbf{x}'} \left(\frac{1}{\sigma^2} \mathbf{\Phi}^{r\top}_{\mathbf{x}} \mathbf{\Phi}^r_{\mathbf{x}} + (\mathbf{B}^\top \mathbf{B})^{-1} \right)^{-1} \mathbf{\Phi}^{r\top}_{\mathbf{x}'}. \tag{12}$$

Cholesky factorization. Let

$$\mathbf{B}^\top \mathbf{B} = \mathbf{L}\mathbf{L}^\top, \qquad \mathbf{L} \in \mathbb{R}^{r \times r},$$

where $\mathbf{L}$ is lower triangular, and define

$$\boldsymbol{\phi}^L(\mathbf{x}) := \mathbf{L}^\top \boldsymbol{\phi}^r(\mathbf{x}) \in \mathbb{R}^r. \tag{13}$$

Theorem 3.2. Using $\boldsymbol{\phi}^L(\mathbf{x}) \in \mathbb{R}^r$ is equivalent to using $\boldsymbol{\phi}^B(\mathbf{x}) \in \mathbb{R}^{m+r}$, and

$$S^B_{\mathbf{x}',\mathbf{x}'} = \mathbf{\Phi}^L_{\mathbf{x}'} \left(\frac{1}{\sigma^2} \mathbf{\Phi}^{L\top}_{\mathbf{x}} \mathbf{\Phi}^L_{\mathbf{x}} + \mathbf{I}_r \right)^{-1} \mathbf{\Phi}^{L\top}_{\mathbf{x}'} \tag{14}$$

$$= \mathbf{\Phi}^r_{\mathbf{x}'} \mathbf{L} \left(\frac{1}{\sigma^2} \mathbf{L}^\top \mathbf{\Phi}^{r\top}_{\mathbf{x}} \mathbf{\Phi}^r_{\mathbf{x}} \mathbf{L} + \mathbf{I}_r \right)^{-1} \mathbf{L}^\top \mathbf{\Phi}^{r\top}_{\mathbf{x}'}. \tag{15}$$

Methodology — Subsampling for Scalability

Remaining N-dependent bottleneck. Although the matrix inversion is only $r \times r$, constructing the feature covariance still requires all N training examples:

$$\mathbf{\Phi}_{\mathbf{x}}^{L\top}\mathbf{\Phi}_{\mathbf{x}}^{L} = \sum_{i=1}^{N} \boldsymbol{\phi}^{L}(\mathbf{x}_i)\boldsymbol{\phi}^{L}(\mathbf{x}_i)^{\top} =: N\hat{\mathbf{\Sigma}}_N. \tag{17}$$

Uniform subsampling. Draw indices $s_1, \ldots, s_k$ uniformly without replacement from $\{1, \ldots, N\}$ and approximate

$$N\hat{\mathbf{\Sigma}}_N \approx \frac{N}{k}\sum_{i=1}^{k} \boldsymbol{\phi}^{L}(\mathbf{x}_{s_i})\boldsymbol{\phi}^{L}(\mathbf{x}_{s_i})^{\top} =: N\hat{\mathbf{\Sigma}}_k. \tag{18}$$

Rich-BLL(S) predictive covariance. Replacing the full feature covariance with its subsampled estimate gives

$$S^{B,k}_{\mathbf{x}',\mathbf{x}'} := \mathbf{\Phi}^{L}_{\mathbf{x}'}\left(\frac{1}{\sigma^2}N\hat{\mathbf{\Sigma}}_k + \mathbf{I}_r\right)^{-1}\mathbf{\Phi}^{L\top}_{\mathbf{x}'}. \tag{19}$$

Feature-covariance construction: $\mathcal{O}(Nr^2) \longrightarrow \mathcal{O}(kr^2)$

Results — UCI Regression (NLL ↓)

Method	Boston	Concrete	Energy	Power	Wine
Rich-BLL	2.61±0.04	3.10±0.03	0.75±0.03	2.78±0.01	1.00±0.01
Rich-BLL (S)	2.62±0.04	3.10±0.04	0.78±0.03	2.78±0.01	1.01±0.01
NNGP (BLL)	2.78±0.06	3.39±0.06	0.92±0.01	2.82±0.01	1.03±0.01
VBLL	2.75±0.06	3.19±0.05	0.74±0.04	2.78±0.01	1.05±0.01
MAP	2.79±0.09	4.77±0.15	0.93±0.03	3.28±0.02	1.02±0.00
RBF GP	2.84±0.09	3.24±0.13	**0.66**±0.04	2.89±0.01	**0.97**±0.01
Dropout	2.62±0.05	3.19±0.02	1.80±0.03	2.81±0.01	1.01±0.01
Ensemble	**2.48**±0.09	3.15±0.03	0.94±0.02	**2.75**±0.02	0.98±0.01
SWAG	2.65±0.01	3.19±0.03	1.20±0.06	2.79±0.01	1.04±0.01
BBB	2.54±0.04	**2.99**±0.03	1.22±0.01	2.77±0.01	1.05±0.01

Results — Image Classification & OOD Detection

Method	NLL ↓	ECE ↓	AUROC ↑ (SVHN)	AUROC ↑ (CIFAR-100)
Rich-BLL (S)	$0.56_{\pm 0.00}$	$0.029_{\pm 0.01}$	$0.91_{\pm 0.02}$	$\mathbf{0.65}_{\pm 0.01}$
NNGP (BLL)	$0.58_{\pm 0.00}$	$0.046_{\pm 0.01}$	$0.88_{\pm 0.01}$	$0.62_{\pm 0.02}$
LL Laplace	$0.57_{\pm 0.00}$	$0.044_{\pm 0.00}$	$0.84_{\pm 0.01}$	$0.55_{\pm 0.00}$
SNGP	$0.89_{\pm 0.00}$	$0.027_{\pm 0.001}$	$0.85_{\pm 0.01}$	$0.52_{\pm 0.00}$
LL Dropout	$0.56_{\pm 0.00}$	$0.031_{\pm 0.002}$	$0.78_{\pm 0.02}$	$0.62_{\pm 0.00}$
Dropout	$\mathbf{0.31}_{\pm 0.05}$	$\mathbf{0.013}_{\pm 0.003}$	$\mathbf{0.92}_{\pm 0.001}$	$0.61_{\pm 0.02}$
Ensemble	$0.47_{\pm 0.00}$	$0.030_{\pm 0.002}$	$0.80_{\pm 0.20}$	$\mathbf{0.65}_{\pm 0.01}$
BBB	$1.39_{\pm 0.02}$	$0.220_{\pm 0.007}$	$0.89_{\pm 0.01}$	$0.54_{\pm 0.00}$