LiBRe: A Practical Bayesian Approach to Adversarial Detection

Zhijie Deng[1], Xiao Yang[1], Shizhen Xu[2], Hang Su[1], Jun Zhu[1]

[1]Dept. of Comp. Sci. and Tech., BNRist Center, Institute for AI

[1]Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University [2]RealAI

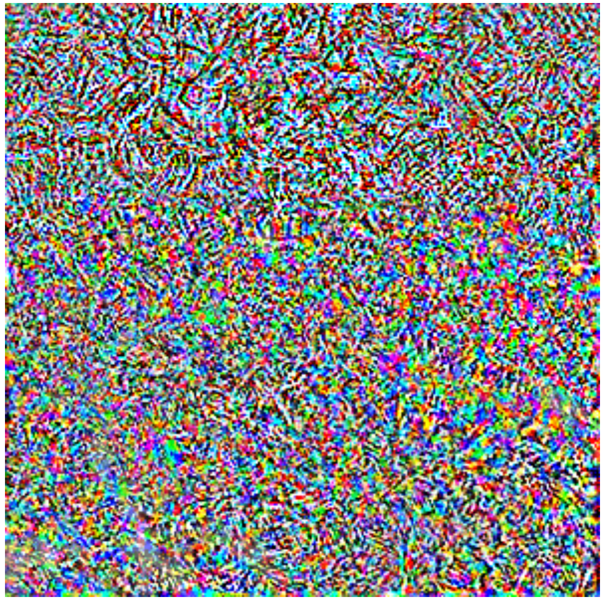
Contact: dzj17@mails.tsinghua.edu.cn

VIRTUAL JUNE 19-25

Threat from Adversarial Examples

- DNNs are vulnerable against **adversarial examples**, which are generated by adding human-imperceptible perturbations upon clean examples to deliberately cause misclassification.

Alps: 94.39%

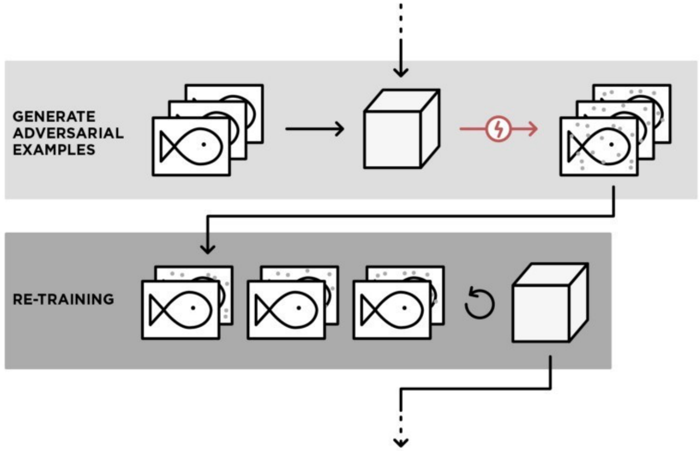


Dong et al., 2018

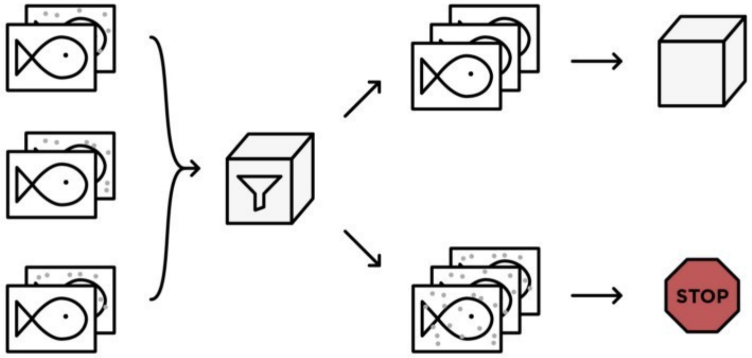
Dog: 99.99%

Current Defenses to Adversarial Examples

- **Adversarial training** methods are effective, yet cause added training overheads and undermine the predictive performance on clean data.

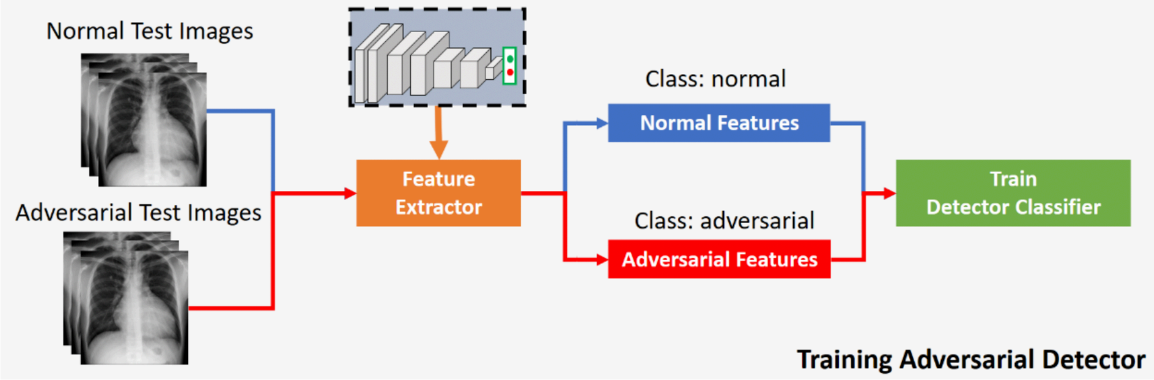


- **Adversarial detection** methods detect the adversarial examples ahead of decision making, yet are usually developed for specific tasks or attacks, thus lack the flexibility to effectively *generalize* to other tasks or attacks.



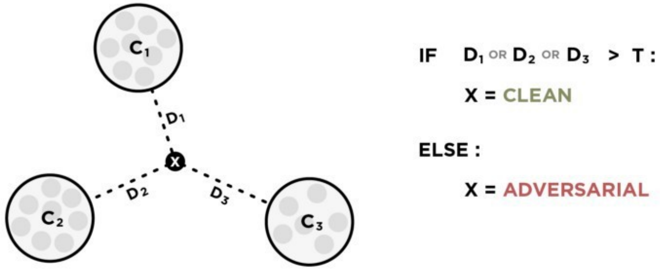
Detailed Adversarial Detection Methods

- By virtue of
  - auxiliary classifiers

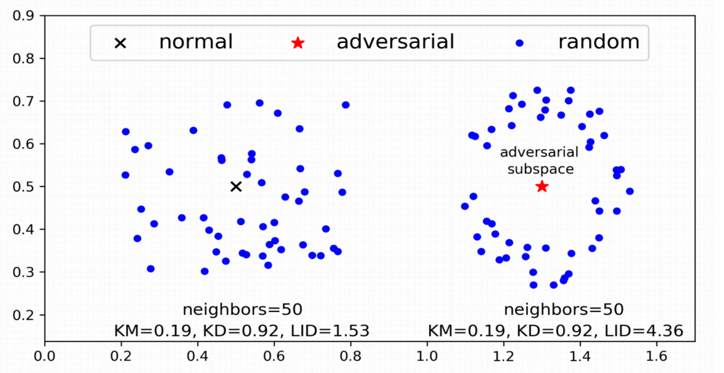


Ma et al., 2019

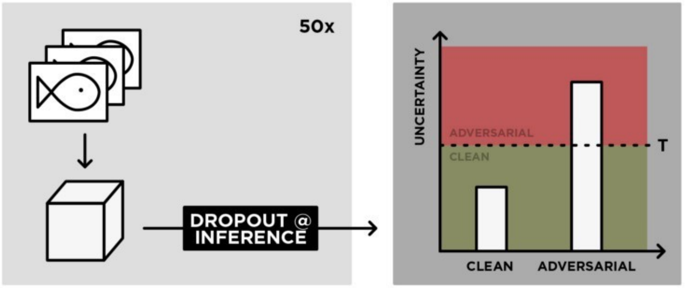
  - designed statistics



KDE based detection, Feinman et al., 2017



LID based detection, Ma et al., 2018



Dropout uncertainty based detection, Feinman et al., 2017

Detect Adversarial Examples by Bayesian Uncertainty

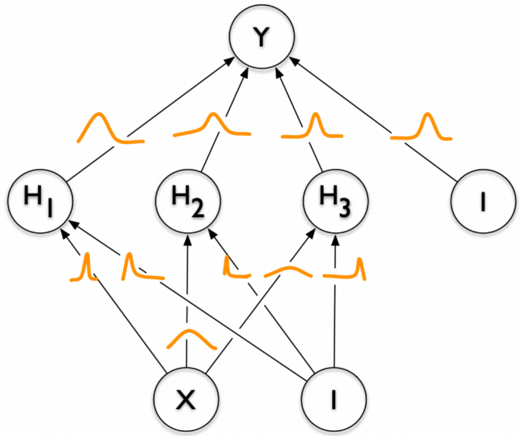
- The key motivation: think of adversarial examples as a special kind of out-of-distribution (OOD) data, and proceed in a **Bayesian** way
  - **Bayesian neural networks** (BNNs) are as **flexible** as DNNs for data fitting in various tasks, and the uncertainty yielded by them suffices to detect **heterogeneous** OOD/adversarial data in principle.

$$p(\mathbf{w}|\mathcal{D}) \propto p(\mathbf{w}) \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{w})$$

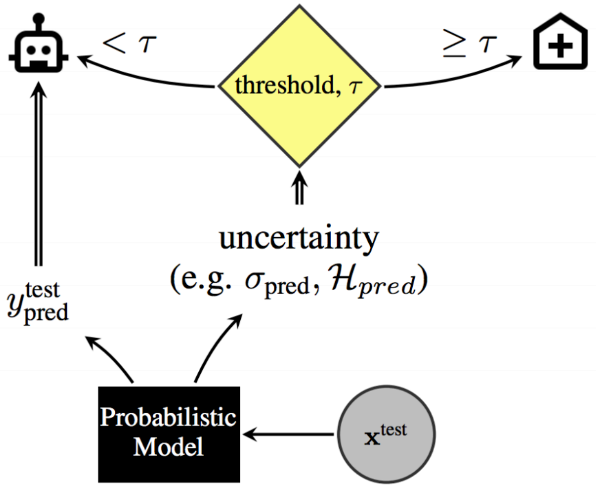
posterior inference

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

Marginalization



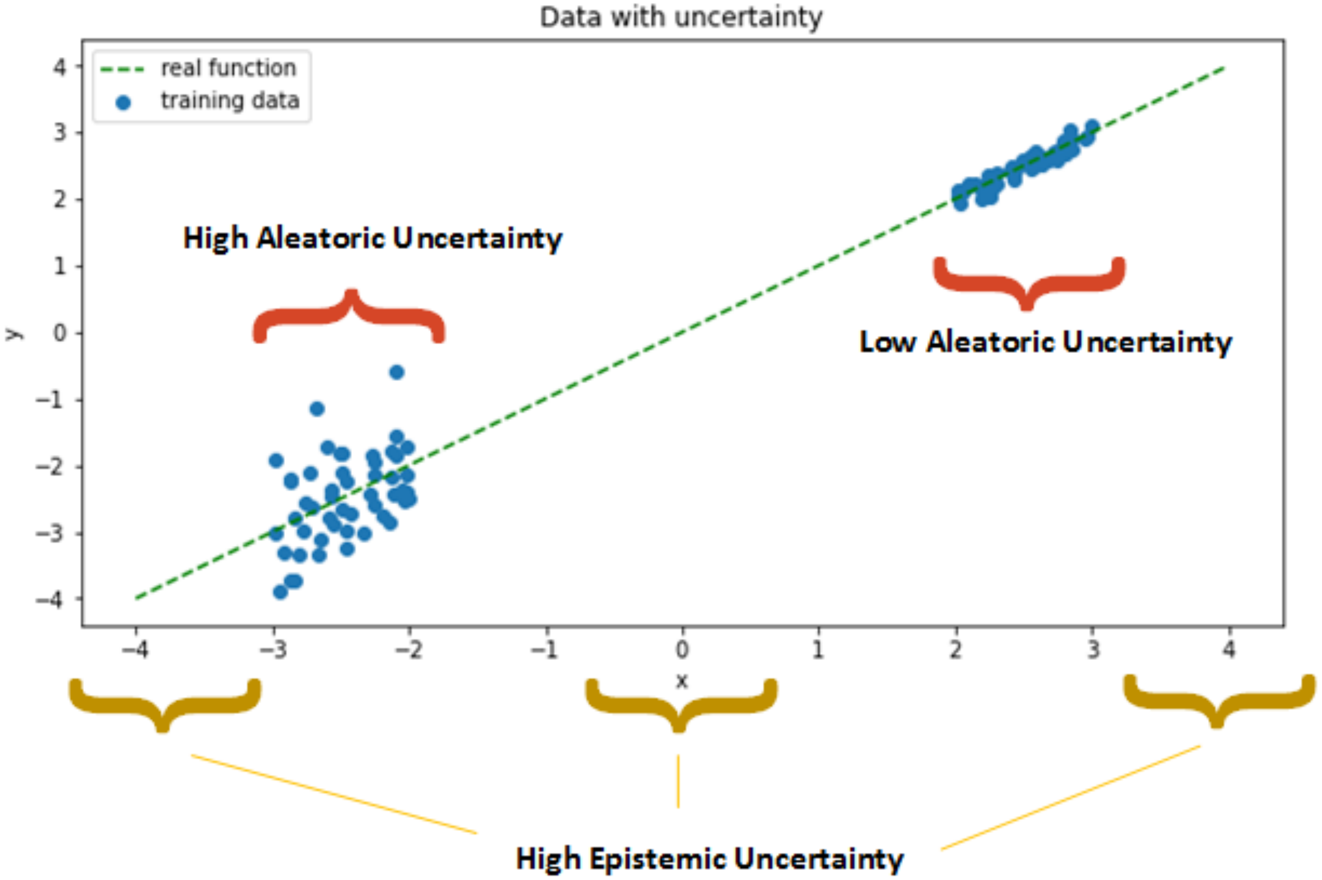
Blundell et al., 2015



Angelos Filos et al.

Two Types of Bayesian Uncertainty

- Epistemic uncertainty: uncertainty over the model (for detecting OOD)
- Aleatoric uncertainty: uncertainty over the data for a fixed model (for measuring data noise)



Approximate Inference for BNNs

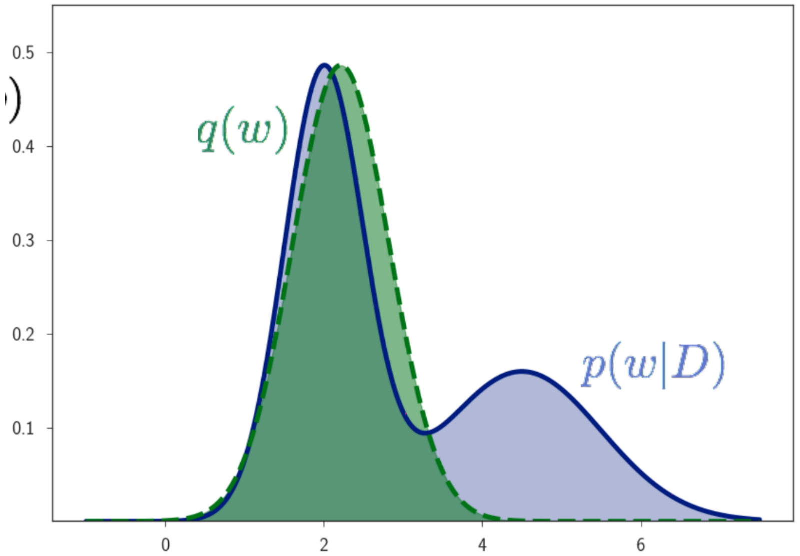
- Variational Inference [Graves, 11; Blundell et al., 15; Louizos et al., 16,17; shi et al, 18; etc.]
  - Maximize evidence lower bound (ELBO) ($q(w|\theta)$ is an introduced variational):

$$\max_{\theta} \mathcal{L}(\theta) = \mathrm{E}_{q(w|\theta)}[\log p(\mathcal{D}|w)] - \mathrm{KL}(q(w|\theta)\|p(w)) \leq \log p(D)$$

  - Reparameterizition trick:

$$q(w|\theta) = \mathcal{N}(w; \mu, \mathrm{diag}(\sigma^2)) \text{ -> } t(\theta, \epsilon) = \mu + \epsilon\sigma, \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

  - Stochastic variational inference resembles ordinary backprop



Efficient yet inducing approximation error; without the guarantee of asymptotic consistency

Approximate Inference for BNNs

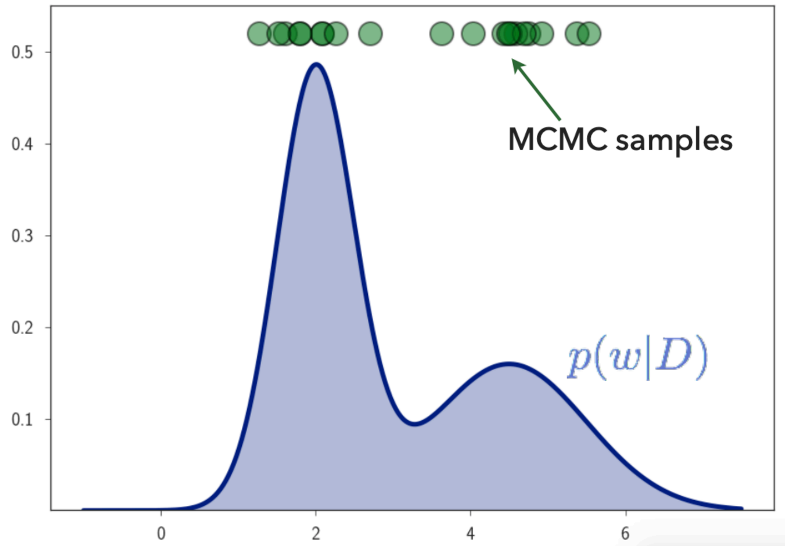
- Markov Chain Monte Carlo [Neal, 93; Welling & Teh, 11; etc.]
  - Metropolis–Hastings
  - Slice sampling
  - Hamiltonian (or Hybrid) Monte Carlo
  - Stochastic gradient Langevin dynamics, SGLD

$$w_{t+1} = w_t - \alpha_t \nabla \widetilde{U}(w_t) + \sqrt{2\alpha_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$

  - Stochastic gradient Hamiltonian Monte Carlo, SGHMC

$$w_{t+1} = w_t + v_{t+1},\ v_{t+1} = (1-\eta)v_t - \alpha_t \nabla \widetilde{U}(w_t) + \sqrt{2(\eta - \hat{\gamma})\alpha_t}\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I)$$

  - Cyclical stochastic gradient MCMC



Non-parametric and asymptotically exact yet typically with low convergence rate

Approximate Inference for BNNs

- Particle-optimization-based Variational Inference (POVI) [Liu et al., 16; Wang et al., 19; etc.]
  - Conjoins the flexibility of being non-parametric as MCMC and the efficiency due to doing deterministic optimization as variational inference
  - Stein Variational Gradient Descent (SVGD) is one of the most popular examples:

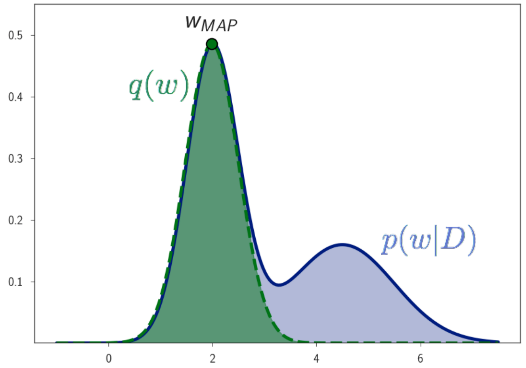
$$w_{t+1}^{(k)} = w_t^{(k)} + \epsilon\phi\left(w_t^{(k)}\right), \quad \forall k = 1, \dots, K \text{ , and } \phi(\cdot) := \mathbb{E}_{q(w)}[K(w,\cdot)\nabla_{\mathrm{w}} \log p(w|\mathcal{D}) + \nabla_{\mathrm{w}} K(w,\cdot)]$$

    - $\hat{q}(w) = \frac{1}{n}\sum_{i=1}^{n} \delta_{w^{(k)}}(w)$ replaces $q(w)$ for the above update equation
    - $\nabla_{\mathrm{w}} K(w,\cdot)$ is understood as a repulsive force to reduce the correlation between particles

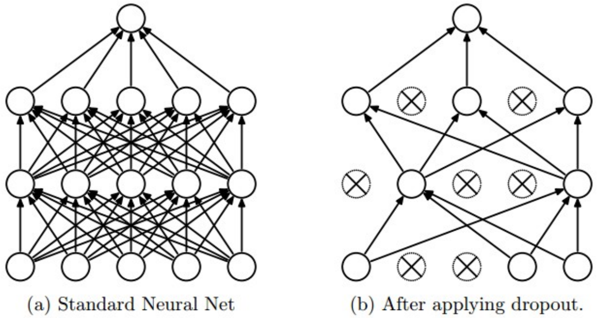
  - Yet, POVI methods may converge to degenerate posteriors due to over-parameterization, and suffer from curse of dimensionality [Wang et al., 19; Zhuo et al.,19].

Approximate Inference for BNNs

- Some practical workarounds:
  - Laplace approximation [Mackay, 92; Ritter et al, 18]
    - Compute a Gaussian posterior around the MAP with hessian
    - Less flexible



  - Monte Carlo dropout [Gal & Ghahramani, 16]
    - Take dropout as uncertainty over weights
    - Less effective



  - Deep ensemble [Lakshminarayanan et al., 17]
    - Train multiple DNNs and assemble their predictions
    - Less scalable

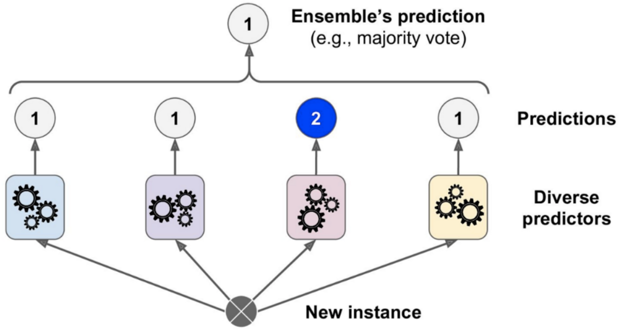
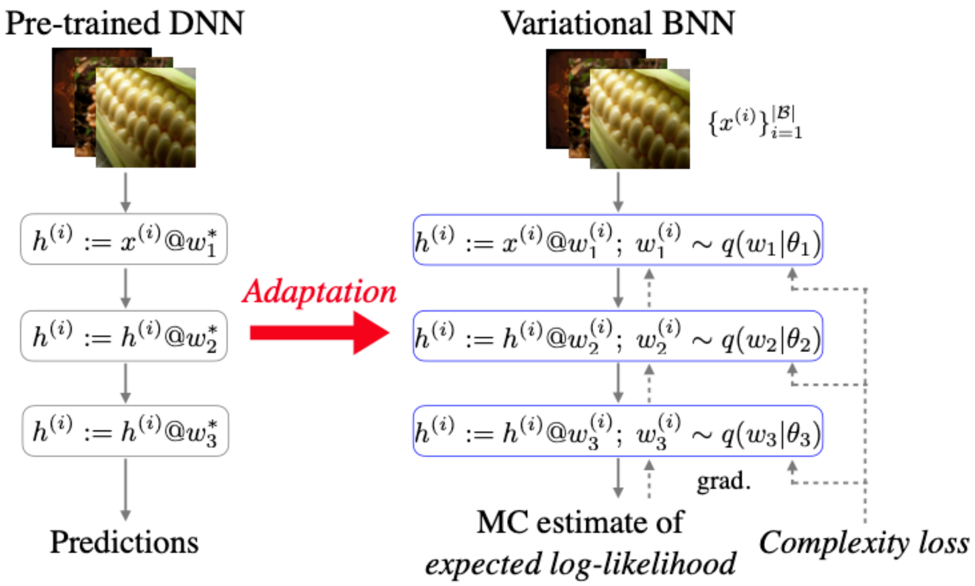


Figure 7-2. Hard voting classifier predictions

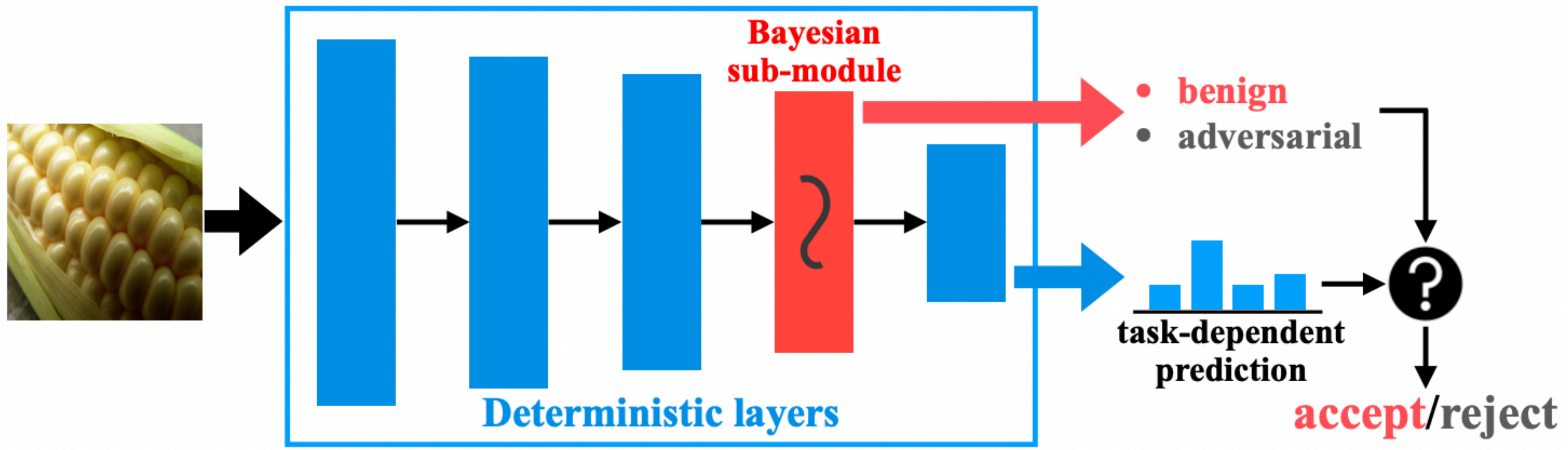
Approximate Inference for BNNs

- BayesAdapter [Deng et al., 20]
  - Obtain BNNs by fine-tuning pre-trained DNNs
  - Conjoins the complementary benefits from deterministic training and Bayesian reasoning, e.g., good performance, resistance to over-fitting, reliable uncertainty estimates, etc.
  - Exemplar reparameterization (ER)：
    - Draw a separate parameter sample for every exemplar in the mini-batch
    - Disentangle the correlation between the loss of difference instances



Lightweight Bayesian Refinement (LiBRe)

- Given a **pre-trained task-dependent** DNN
  1. LiBRe converts its last **few layers** (e.g. the last ResBlock) to be *Bayesian.*
  2. LiBRe **inherits** the **pre-trained** parameters.
  3. LiBRe launches **several**-round adversarial detection-oriented **Bayesian fine-tuning**.



Lightweight Bayesian Refinement (LiBRe)

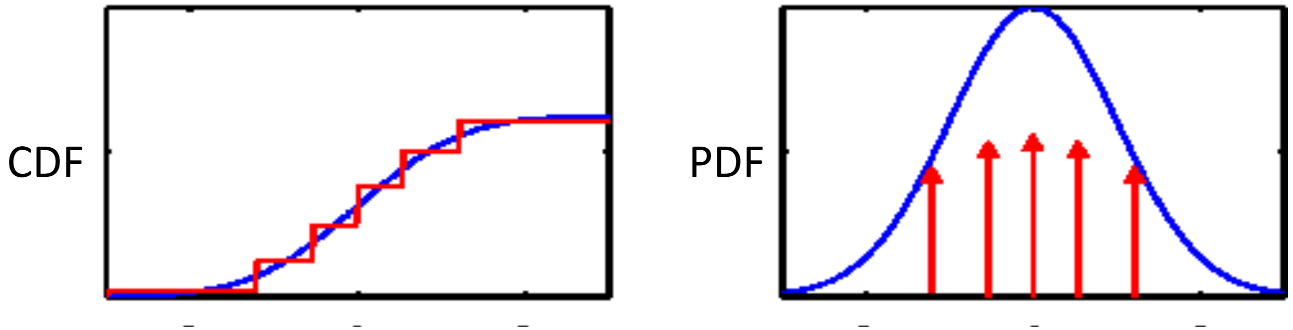
CVPR VIRTUAL JUNE 19-25

- **LiBRe** follows the ***variational inference*** pipeline for learning BNNs:

  Maximize the ELBO: $\max_{\theta} \mathrm{E}_{q(w|\theta)} \sum_i \log p(D_i|w) - KL(q(w|\theta)||p(w))$

- **Partial** Bayesian treatment: ***Few lAyer Deep Ensemble*** (FADE) variational

$$q(w|\theta) = \frac{1}{C}\sum_{c=1}^{C} \delta\left(w_b - w_b^{(c)}\right) \delta(w_{-b} - w_{-b}^{(0)})$$



- w_b: parameters of **tiny Bayesian sub-module**; w_{-b}: the other deterministic ones
- Conjoins the **expressiveness** of *deep ensemble* [Lakshminarayanan et al., 2017] and the **efficiency** of *last-layer Bayesian learning* [Kristiadi et al., 2020]
- A mixture of deltas is a singular approximating distribution, so we indeed relax $q(w|\theta)$ as ***a mixture of Gaussians with small variance*** to estimate $KL(q(w|\theta)||p(w))$

Lightweight Bayesian Refinement (LiBRe)

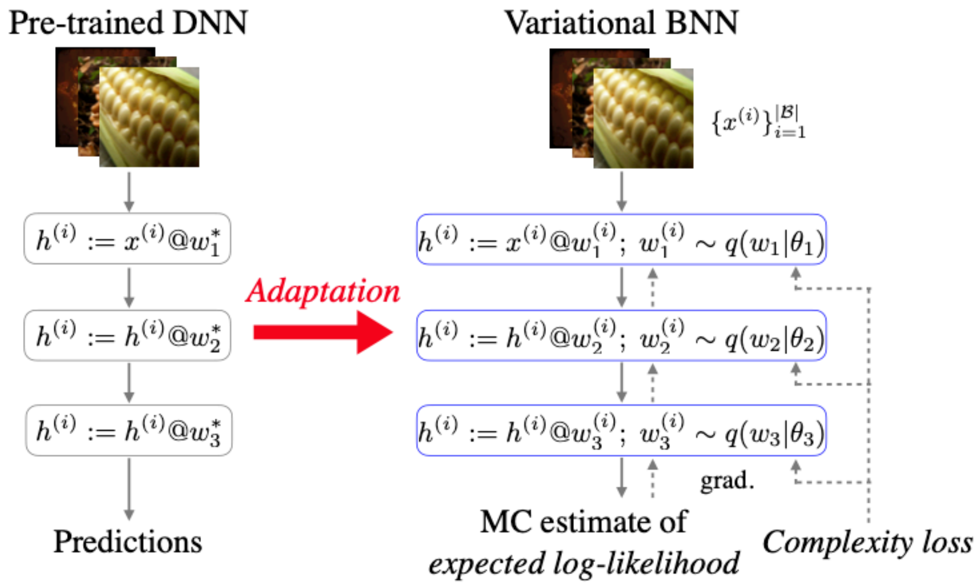
- Monte Carlo estimation of ELBO by ***reparameterization***:

$$\max_{\theta} \mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}_i} \log p\left(\mathcal{B}_i \middle| w_b^{(c)}, w_{-b}^{(0)}\right), c \sim \{1, 2, \ldots, C\}, \mathcal{B} \subset D$$

  - **Variance reduction** by ***Exemplar reparameterization*** [Deng et al., 2020]

$$\max_{\theta} \mathcal{L}^* = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}_i} \log p\left(\mathcal{B}_i \middle| w_b^{(c_i)}, w_{-b}^{(0)}\right), c_i \sim \{1, 2, \ldots, C\}\ \forall i = 1, \ldots, |\mathcal{B}|$$

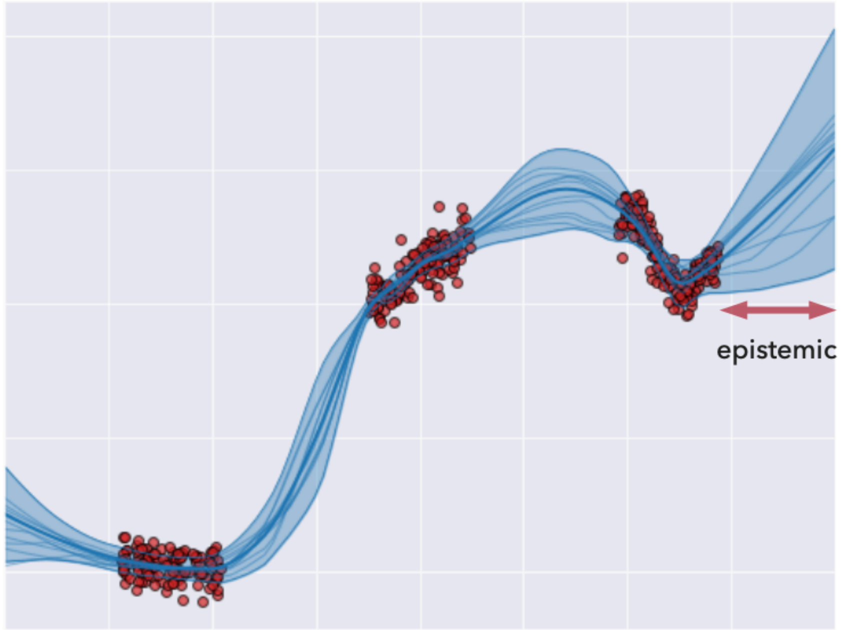
- **Stochastic variational inference as Bayesian fine-tuning**



Lightweight Bayesian Refinement (LiBRe)

- Detect adversarial examples with ***epistemic uncertainty:***
  - A typical metric: softmax variance [Feinman et al., 2017, Smith and Gal, 2018], but not universal (e.g. in regression)
  - A **more generic** metric: ***feature variance***

$$Unc = \frac{1}{T-1}\left(\sum_{t=1}^{T}\left\|z^{(t)}\right\|_2^2 - T\left\|\frac{1}{T}\sum_{t=1}^{T} z^{(t)}\right\|_2^2\right)$$ ($z^{(t)}$ is the feature under $w^{(t)}, t = 1, \ldots, T$)



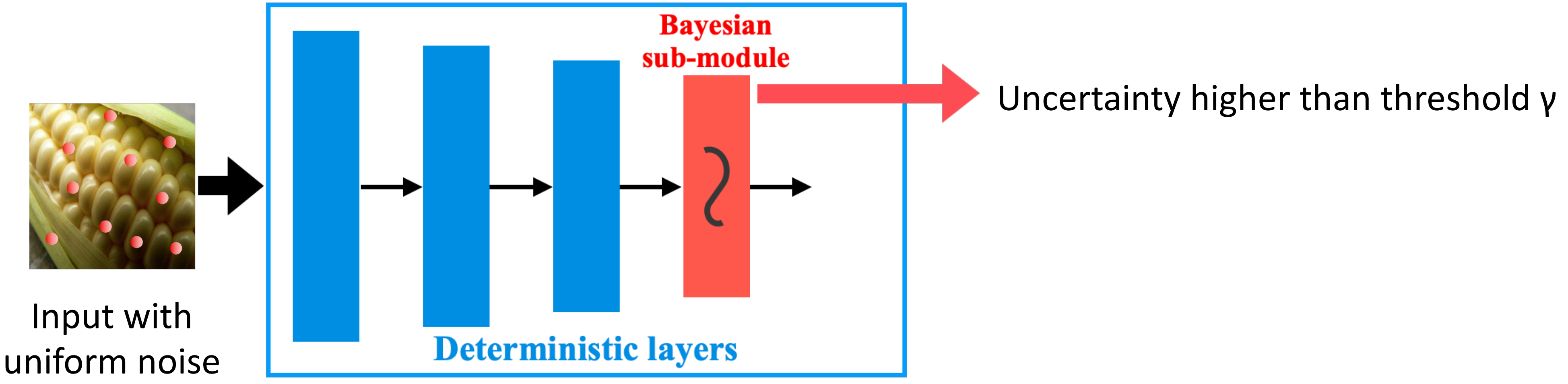
Lightweight Bayesian Refinement (LiBRe)

CVPR VIRTUAL JUNE 19-25

- Adversarial example **free uncertainty correction**

$$\max_{\theta} \mathcal{R} = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}_i} \min\left(\left\| \tilde{z}_i^{(c_{i,1})} - \tilde{z}_i^{(c_{i,2})} \right\|_2^2, \gamma\right).$$

  - $\tilde{z}_i^{(c_{i,j})}$ refers to the feature of i[th] training instances with **uniform** input perturbations under parameter sample $w^{(c_{i,j})} = \{w_b^{(c_{i,j})}, w_{-b}^{(0)}\}$.
  - This is **necessary** as adversarial examples can easily destroy the uncertainty based adversarial detection if there is no uncertainty correction [Grosse et al., 2018]



Experiments

- We perform Bayesian fine-tuning for **only 6** epochs on ImageNet.
- LiBRe preserves **non-degraded accuracy** while demonstrating **near-perfect capacity of detecting adversarial examples**.

Method	Prediction accuracy ↑		AUROC of adversarial detection under *model transfer* ↑			
	TOP1	TOP5	PGD	MIM	TIM	DIM
MAP	76.13%	92.86%	-	-	-	-
MC dropout [17]	74.86%	92.33%	0.660	0.723	0.695	0.605
LMFVI	76.06%	92.92%	0.125	0.200	0.510	0.018
MFVI	75.24%	92.58%	0.241	0.205	0.504	0.150
LiBRe	**76.19%**	**92.98%**	**1.000**	**1.000**	**0.982**	**1.000**

Table 1: Left: comparison on accuracy. Right: comparison on AUROC of adversarial detection under *model transfer*. (ImageNet)

Method	FGSM	BIM	C&W	PGD	MIM	TIM	DIM	FGSM-ℓ_2	BIM-ℓ_2	PGD-ℓ_2
KD [14]	0.639	1.000	0.999	1.000	1.000	0.999	0.624	0.633	1.000	1.000
LID [39]	0.846	0.999	0.999	0.999	0.997	0.999	0.762	0.846	0.999	0.999
MC dropout [17]	0.607	1.000	0.980	1.000	1.000	0.999	0.628	0.577	0.999	0.999
LMFVI	0.029	0.992	0.738	0.943	0.996	0.997	0.021	0.251	0.993	0.946
MFVI	0.102	1.000	0.780	0.992	1.000	0.999	0.298	0.358	0.952	0.935
LiBRe	**1.000**	0.984	0.985	0.994	0.996	0.994	**1.000**	**0.995**	0.983	0.993

Table 2: Comparison on AUROC of adversarial detection for *regular attacks* ↑. (ImageNet)

Experiments

- Face recognition

Method	Softmax				CosFace				ArcFace			
	MAP	*MCD*	*LMFVI*	*LiBRe*	*MAP*	*MCD*	*LMFVI*	*LiBRe*	*MAP*	*MCD*	*LMFVI*	*LiBRe*
VGGFace2	**0.9256**	0.9254	0.9198	0.9246	0.9370	0.9370	0.9360	**0.9376**	0.9356	0.9334	**0.9358**	0.9348
LFW	**0.9913**	0.9898	0.9912	0.9892	0.9930	0.9932	0.9920	**0.9935**	0.9933	0.9930	0.9933	**0.9943**
CPLFW	0.8630	**0.8638**	0.8610	0.8598	0.8915	0.8890	**0.8925**	0.8910	0.8808	0.8803	0.8833	**0.8837**
CALFW	0.9107	0.9110	0.9087	**0.9120**	0.9327	0.9345	0.9333	**0.9352**	0.9292	**0.9300**	0.9250	0.9283
AgedDB-30	**0.9177**	0.9170	0.9128	0.9167	**0.9435**	0.9422	0.9387	0.9433	0.9327	0.9317	**0.9337**	**0.9337**
CFP-FP	0.9523	**0.9543**	0.9480	0.9489	0.9564	0.9567	0.9583	**0.9597**	**0.9587**	0.9586	0.9554	0.9573
CFP-FF	0.9873	0.9870	**0.9874**	**0.9874**	**0.9927**	0.9926	0.9916	**0.9927**	0.9914	0.9910	0.9911	**0.9921**

Table 3: Accuracy comparison on face recognition ↑. *MCD* is short for *MC dropout*. **Bold** refers to the best results under specific loss function. **Blue bold** refers to the overall best results.

Attack	Softmax			CosFace			ArcFace		
	MC dropout	*LMFVI*	*LiBRe*	*MC dropout*	*LMFVI*	*LiBRe*	*MC dropout*	*LMFVI*	*LiBRe*
FGSM	0.866	0.155	**1.000**	0.889	0.001	**1.000**	0.794	0.001	**1.000**
BIM	1.000	1.000	0.999	1.000	1.000	0.999	1.000	1.000	1.000
PGD	1.000	0.992	0.999	1.000	0.998	0.998	1.000	0.990	1.000
MIM	1.000	1.000	0.999	1.000	1.000	0.999	1.000	1.000	1.000
TIM	1.000	1.000	0.999	1.000	1.000	0.998	1.000	1.000	1.000
DIM	0.910	0.025	**1.000**	0.850	0.000	**1.000**	0.746	0.000	**1.000**
FGSM-ℓ_2	0.860	0.659	**1.000**	0.825	0.014	**0.999**	0.660	0.002	**0.999**
BIM-ℓ_2	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.000
PGD-ℓ_2	1.000	0.996	0.999	1.000	0.999	1.000	1.000	0.994	1.000

Table 4: Comparison on adversarial detection AUROC ↑. We report the averaged AUROC over the verification datasets. (face recognition)

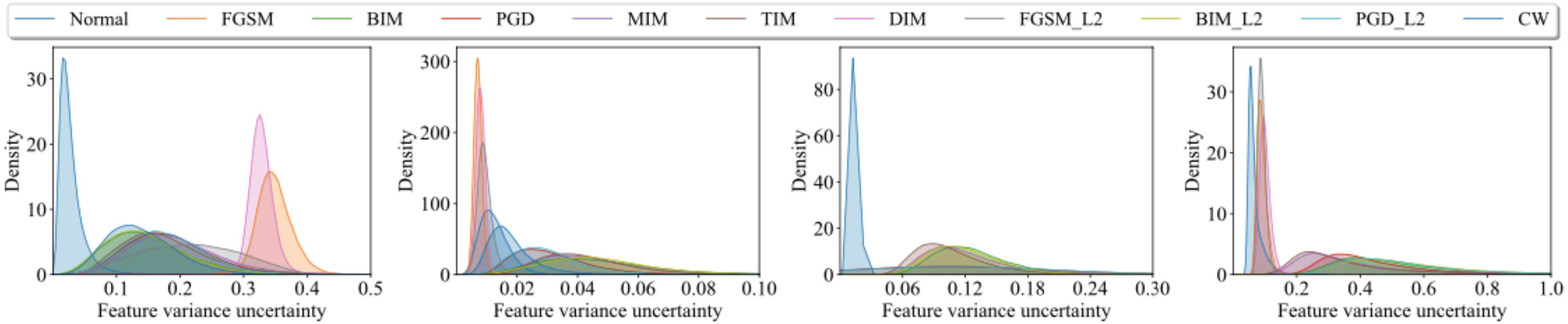
Experiments

- Object detection

Method	Object detection		Adversarial detection			
	mAP@.5	mAP@.5:.95	FGSM	BIM	PGD	MIM
MAP	0.559	0.357	-	-	-	-
LiBRe	0.545	0.344	0.957	0.936	0.972	0.966

Table 5: Results on object detection. (COCO)

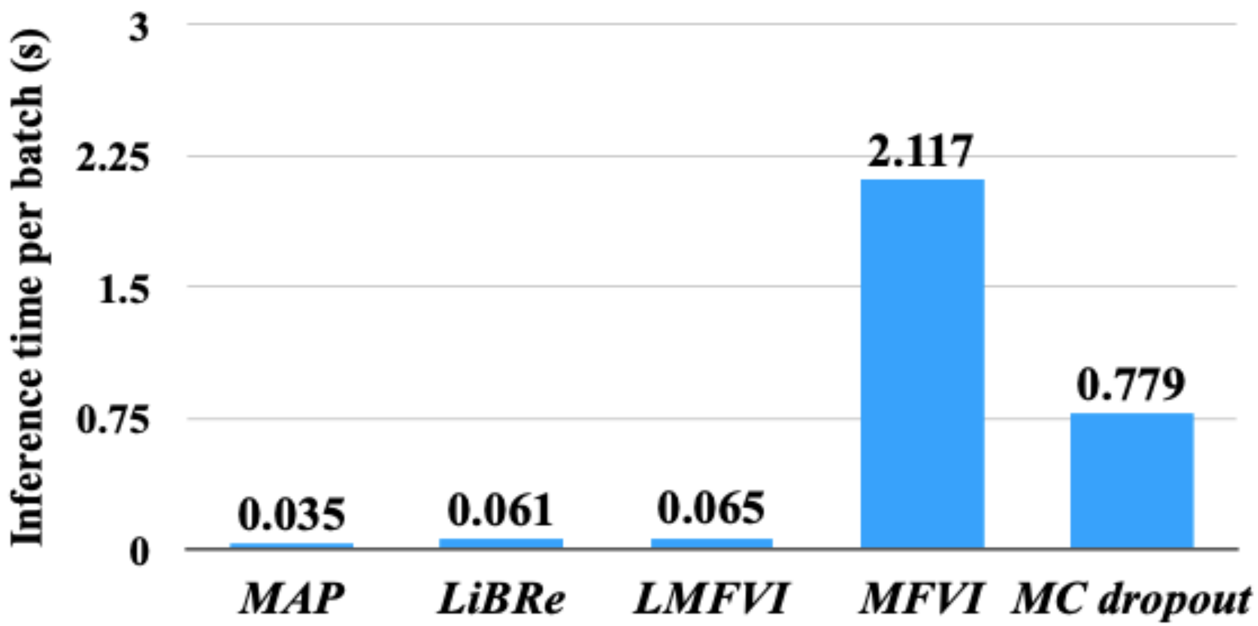
- Visualization for the population of uncertainty estimates



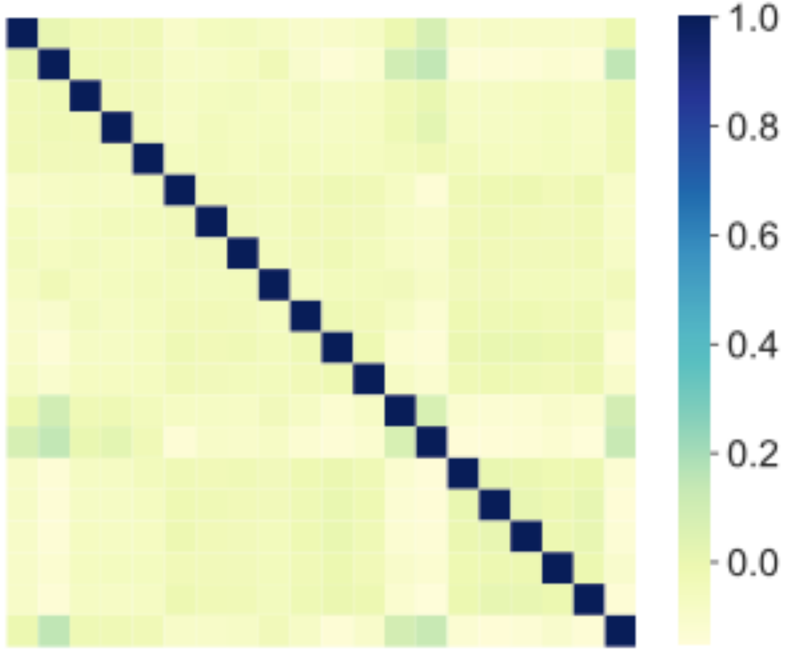
(a) *LiBRe*, ImageNet **(b) *LMFVI*, ImageNet** **(c) *LiBRe*, Face** **(d) *MC dropout*, Face**

Figure 2: The histograms for the *feature variance* uncertainty of normal and adversarial examples given by *LiBRe* or the baselines.

Experiments



(a) Inference speed comparison



(b) Candidate similarity in the posterior

Thanks