We thank the reviewers for their valuable comments and suggestions, and list our responses as follows.

To Reviewer #1:

1. The performance results of the stochastic pooling [32] are shown in Table A and will be included in the revised paper.

2. Eq. 10 is derived through the following variable transformation. Suppose y is a random variable whose probability density function is Gaussian, $\mathtt{q}(y) = \frac{1}{\sqrt{2\pi}\sigma_0}\exp\{-\frac{1}{2\sigma_0^2}(y-\mu_0)^2\}$. The target random variable x is obtained via softplus transformation by $x = \mathtt{softplus}(y) \Leftrightarrow y = \mathtt{softplus}^{-1}(x) = \log[\exp(x) - 1]$. Then, we apply the relationship of $\mathtt{q}(y)dy = \mathtt{p}(x)dx$ and $\frac{dy}{dx} = \frac{\exp(x)}{\exp(x)-1}$ to provide $\mathtt{p}(x) = \frac{1}{\sqrt{2\pi}\sigma_0}\frac{\exp(x)}{\exp(x)-1}\exp\{-\frac{1}{2\sigma_0^2}(\log[\exp(x)-1]-\mu_0)^2\}$ (Eq.10). Such detailed explanation about Eq.10 will be added to the revised paper.

3. We will clearly describe that this work focuses on local pooling and the method is applied to all the local pooling layers in a CNN; e.g., pool1&2 in Table 2a of 13-layer Net.

To Reviewer #2:

1. From the viewpoint of the increased number of parameters, we show the effectiveness of the proposed method in comparison with the other types of modules that adds the same number of parameters; NiN [LCY14] using 1×1 conv, ResNiN which adds an identity path to the NiN module as in ResNet [7], and squeeze-and-excitation (SE) module [HSS18]. For fair comparison, they are implemented by using the same 2-layer MLP as ours (Eq.12) of C^2 parameters with appropriate activation functions and are embedded before pool1&2 layers in the 13-layer Net (Table 2a) so as to work on the feature map fed into the *max* pooling layer; the detailed architecture is shown in the left-bottom figure. The performance results are shown in Table B, demonstrating that our method most effectively leverages the additional parameters to improve performance. This comparison result will be included in the revised paper.

2. The approximation in Eq.15 is *heuristically* determined so as to represent $\mathrm{E}[\eta]$ in a simple analytic form. That is, under the condition of $\sigma_0 \leq 1$, we *manually* tune the form and the parameters of the residual term, $0.115\sigma_0^2\frac{4\exp(0.9\mu_0)}{(1+\exp(0.9\mu_0))^2}$, toward minimizing the residual error between $\mathtt{softplus}(\mu_0)$ and $\int \log[1+\exp(\tilde{\epsilon})]\mathcal{N}(\tilde{\epsilon};\mu_0,\sigma_0)d\tilde{\epsilon}$ which is empirically computed by means of sampling. Then, Eq.16 is presented as the *most roughly* approximated form for Eq.15 by *ignoring* the above-mentioned residual term which exhibits at most 0.115 residual error. The rough approximation is introduced since it is practically useful for fast computation at inference without degrading performance (lines146-150).

3. In the preliminary experiment, we confirmed that the log-Gaussian makes it almost impossible to train CNNs; due to introducing the log-Gaussian module, the training loss is not favorably reduced during the end-to-end learning.

To Reviewer #3:

1. As mentioned in lines 174-178, the computation overhead of the proposed method is caused by the GAP+MLP to estimate the two parameters of $\{\mu_0, \sigma_0\}$ at training and only one μ_0 at inference; $O(HWC)$ in GAP and $O(C^2)$ in MLP. For example, in ResNet-50 which requires 3.86GFLOPs, our method increases the computation by only 0.017GFLOPs.

2. Table C shows the performance of ResNet-50 on the adversarial attack via FGSM [GSS15] which adds perturbation by $\epsilon\mathtt{sign}(\nabla_I\mathcal{L}(I,t))$ to an input (test) image I according to its label t and the loss function $\mathcal{L}$. Compared to the other pooling methods, our method exhibits favorable robustness against the attack while the Mixed pooling endowed with stochastic training also works well. This result motivates our future work to further analyze the proposed pooling method, especially in terms of stochastic training in the pooling, from this viewpoint of robustness to input perturbations.

Table A: Performance on ImageNet.

Method	ResNet-50 Top-1	ResNet-50 Top-5	ResNeXt-50 Top-1	ResNeXt-50 Top-5
Stochastic	25.47	7.87	25.02	7.73
iSP-Gauss	21.37	5.68	20.66	5.60

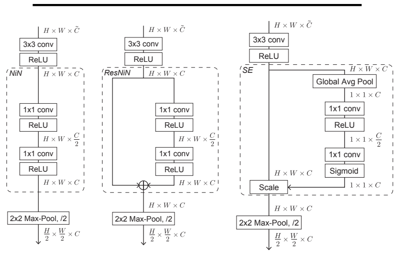

Table B: Performance comparison on Cifar100 dataset by 13-layer Net.

Method	Error (%)
NiN	24.49±0.13
ResNiN	24.33±0.16
SE	23.99±0.07
iSP-Gaussian	23.52±0.37

← Module architecture of the comparison methods. They utilize the same 2-layer MLP as in our method.

Table C: Performance results of ResNet-50 on ImageNet dataset through adversarial attack by FGSM. $\epsilon = 0$ means *no* adversarial attack, producing to the original results in Table 3c.

Method	$\epsilon=0$ Top-1	$\epsilon=0$ Top-5	$\epsilon=0.1$ Top-1	$\epsilon=0.1$ Top-5	$\epsilon=0.2$ Top-1	$\epsilon=0.2$ Top-5	$\epsilon=0.3$ Top-1	$\epsilon=0.3$ Top-5
skip	23.53	7.00	42.89	14.64	56.58	22.20	66.25	28.73
avg	22.61	6.52	40.35	13.22	53.99	20.13	63.97	26.62
max	22.99	6.71	45.39	15.41	60.93	23.59	71.03	30.64
Mixed	23.32	6.77	37.55	12.11	49.83	17.90	58.99	23.27
DPP	22.52	6.63	42.70	14.02	58.12	21.88	68.77	28.79
Gated	22.27	6.33	41.23	13.29	55.84	20.66	66.41	27.58
GFGP	21.79	5.95	38.11	11.85	50.44	17.70	60.06	23.26
iSP-Gaussian	21.37	5.68	37.42	11.27	50.26	17.52	60.02	23.24

References

[LCY14] M. Lin, Q. Chen, and S. Yan. Network in Network. In ICLR, 2014.
[HSS18] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In CVPR, pp. 7132-7141, 2018.
[GSS15] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. In ICLR, 2015.