Article

Not peer-reviewed version

Bayesian PASA: Adaptive Activation with Uncertainty Quantification under Computational Constraints – Revised Version

Mohsen Mostafa *

Posted Date: 11 March 2026

doi: 10.20944/preprints202603.0740.v2

Keywords: Bayesian PASA; activation function; uncertainty quantification; adaptive activation; robustness; Bayesian deep learning; CIFAR-10-C; CIFAR-100

 doi:10.20944/preprints202603.0740.v2

Article

Bayesian PASA: Adaptive Activation with Uncertainty Quantification Under Computational Constraints – Revised Version

Mohsen Mostafa

Independent Researcher; mohsen.mostafa.ai@outlook.com

Abstract

The choice of activation function is a fundamental design decision in deep learning, yet most popular options like ReLU, GELU, or Swish are static and treat all inputs uniformly. This one-size-fits-all approach can be suboptimal when data contains heterogeneous noise, where the ideal non-linearity might depend on the input's statistical context. In this paper, we introduce Bayesian Probabilistic Adaptive Sigmoidal Activation (Bayesian PASA), an activation function that adapts its behavior based on input uncertainty. The method frames activation selection as a Bayesian model averaging problem, adaptively mixing sigmoidal, linear, and noise-aware behaviors. Mixing weights are derived from a variational evidence lower bound (ELBO) and regularized by a ψ-function that bounds the influence of local noise estimates. We provide theoretical analysis showing Lipschitz continuity, gradient bounds, and convergence under standard assumptions. Due to computational constraints (Google Colab, limited epochs), we evaluate Bayesian PASA on CIFAR-100 (50 epochs, 3 seeds) and CIFAR-10-C (100 epochs, 3 seeds). Despite these limitations, Bayesian PASA achieves 76.38% accuracy on CIFAR-100, slightly outperforming ReLU (75.68%) and GELU (75.98%) under the same constrained conditions. On corrupted CIFAR-10-C, Bayesian PASA combined with Bayesian R-LayerNorm achieves an average accuracy of 53.91%, a +1.87% improvement over the ReLU+LayerNorm baseline. These results, though modest, are consistent across seeds and suggest that Bayesian PASA offers a promising direction for uncertainty-aware activation functions, particularly when training resources are limited. Code is available at: https://github.com/BayesianPASA/BayesianPASA/

Keywords: Bayesian PASA; activation function; uncertainty quantification; adaptive activation; robustness; Bayesian deep learning; CIFAR-10-C; CIFAR-100

1. Introduction

The deep learning revolution has been propelled by a series of architectural innovations, with activation functions playing a critical role. The field has evolved from the classic sigmoid and tanh to the dominant ReLU [1], and more recently to sophisticated smooth functions like GELU and Swish [2]. These functions have enabled training of very deep networks by mitigating the vanishing gradient problem.

However, most activation functions are static: they apply the same non-linearity to every input regardless of its statistical properties. Real-world data often contains non-uniform noise due to sensor artifacts, motion blur, or compression. In such scenarios, a more linear behavior in high-noise regions might help avoid overfitting to spurious patterns, while retaining strong non-linearity for clean features.

This insight has motivated adaptive activation functions. The original PASA [5] heuristically combined sigmoid, linear, and noise-aware branches using variance estimates. While promising, its reliance on unregularized variance estimates led to instability, particularly when training resources

 doi:10.20944/preprints202603.0740.v2

are limited. In parallel, Bayesian R-LayerNorm [7] introduced a ψ-function that provides principled noise adaptation with desirable stability properties.

This Revision. In the original version of this preprint (February 2026), we presented Bayesian PASA with claims of state-of-the-art performance. Based on feedback and further reflection, we have revised the manuscript to more accurately reflect the scope and limitations of our work. The core contribution remains: a Bayesian framework for adaptive activation with theoretical stability guarantees. However, we now explicitly acknowledge that our experiments were conducted under severe computational constraints (free Google Colab, 50 epochs on CIFAR-100 versus the 200+ epochs typical in the literature, and only 3 random seeds). Consequently, our empirical results should be interpreted as preliminary evidence rather than definitive state-of-the-art claims.

Despite these constraints, Bayesian PASA consistently outperformed standard activations across all seeds in our experiments. This consistency suggests the approach is robust and merits further investigation with greater computational resources.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents the mathematical formulation. Section 4 describes experiments and results with explicit discussion of computational limitations. Section 5 discusses limitations and future work. Section 6 concludes.

2. From Heuristic to Bayesian: The Path to Robust Adaptation

The original PASA activation was a significant conceptual step, demonstrating that an activation function could successfully adapt its form based on input statistics. It achieved this by mixing three branches—sigmoidal, linear, and noise-aware—using heuristic evidence scores. For instance, the evidence for the noise-aware branch was a simple function of the input variance, E3∼−x2/σ2. While this worked in controlled settings, our experiments on CIFAR-10-C revealed its limitations: raw variance estimates can be unstable, especially early in training, leading to erratic mixing and degraded performance. The original PASA+LayerNorm achieved only 17.50% accuracy on Gaussian noise, underperforming even standard ReLU. This empirical failure highlighted the need for a more stable and theoretically grounded foundation.

Our solution, Bayesian PASA, replaces these heuristics with a probabilistic framework. We no longer treat the mixing weights as ad-hoc scores but as the posterior probabilities of different data-generating models. This reframing is powerful: the activation function now performs a form of Bayesian model averaging at every neuron, weighting each candidate output by how well its underlying model explains the local input patch. The stability of this process is enhanced by integrating the ψ-function from Bayesian R-LayerNorm. This function acts as a regularizer, bounding the influence of the noise estimate Elocal and helping to ensure that the gradients remain well-behaved. The transition from PASA to Bayesian PASA is thus a move from empirical heuristics to a principled Bayesian treatment, providing improved stability for noisy environments.

3. Mathematical Formulation

Bayesian PASA's operation is founded on a generative model where the observed input x is considered a corrupted version of a latent clean signal s. We consider three candidate models for the clean signal:

M_1 (Sigmoidal):M_2 (Linear):M_3 (Noise-Aware):s=σ(αs0)s=1+|s0|/τs0s=erf(2σeffβs0)

where s0 is a standard normal latent variable and σeff is an effective noise scale modulated by the ψ-function:

σeff2=σ2exp(2αψ(λElocal)),ψ(t)=log(1+t)−1+tt.

Using variational inference, we approximate the log-evidence logp(x|Mi) and obtain efficient evidence scores Ei(x). The final output is the posterior-weighted average:

B−PASA(x)=i∑wi(x)fi(x),wi(x)=∑jexp(Ej(x))exp(Ei(x)).

The ψ-function bounds the influence of the local entropy estimate Elocal, contributing to stability.

Theoretical Properties (proofs in Appendix A):

 doi:10.20944/preprints202603.0740.v2

- Lipschitz Continuity: The mapping $x \mapsto B\text{-}PASA(x)$ is Lipschitz continuous, which helps prevent large output changes for small input perturbations.
- Gradient Bound: The gradient norm is bounded, reducing the risk of exploding or vanishing gradients.
- Convergence: Under standard stochastic gradient descent assumptions, training with B-PASA converges to a neighborhood of a local optimum.

These properties confirm that the activation does not introduce instabilities.

Pseudocode

The following PyTorch-like implementation illustrates the forward pass:

```
python
class BayesianPASA(nn.Module):
def forward(self, x):
# Update running statistics (during training)
if self.training:
update_running_absmean(x)
update_running_noise_var(x)
update_running_local_entropy(x)
# Retrieve stable statistics
mu_abs = self.running_absmean
sigma2 = self.running_noise_var.clamp(...)
local_E = self.running_local_entropy.clamp(...)
# Compute ψ-modulated component functions
alpha = self.alpha0 + self.alpha1 * tanh(self.kappa * self.psi(self.lambda3 * local_E))
S = torch.sigmoid(alpha * x)
L = x / (1 + x.abs() / self.tau)
sigma_eff = sqrt(sigma2) * exp(0.5 * self.psi(self.lambda3 * local_E))
N = torch.tanh(1.4 * x / (self.beta * sigma_eff))
# Compute variational evidence scores
log_prior = log(1/3)
E1 = -0.5 * self.lambda1 * (x - self.mu1)**2 + log_prior
E2 = -x.abs() / self.tau_lin + log_prior
E3 = -0.5 * (x**2) / sigma2 - 0.5 * log(sigma2) - 0.5 * self.psi(self.lambda3 * local_E) + log_prior
# Softmax mixing
w = softmax(stack(E1, E2, E3) / self.tau_mix, dim=1)
# Output weighted sum
return w[...,0] * S + w[...,1] * L + w[...,2] * N
```

4. Experiments

4.1. Computational Resources

All experiments were conducted on a free Google Colab instance with an NVIDIA T4 GPU (16GB VRAM) and approximately 15GB system RAM. Due to session time limits (~12 hours maximum runtime) and the need to share resources with other users, we limited training epochs and seeds. For CIFAR-100, we trained for 50 epochs (standard practice uses 200+ epochs). For CIFAR-10-C, we trained for 100 epochs. We used only 3 random seeds for each condition. These constraints reduce statistical power, but the consistent trends across seeds suggest the observed improvements are reliable.

4.2. CIFAR-100 (Clean)

 doi:10.20944/preprints202603.0740.v2

We trained a ResNet-18 on CIFAR-100 for 50 epochs using SGD (lr=0.1, momentum=0.9, weight decay=5e-4) with cosine annealing. Figure 1 shows the test accuracy comparison. Bayesian PASA achieves 76.38%, slightly outperforming ReLU (75.68%) and GELU (75.98%). The improvement, though modest, is consistent across seeds and indicates that Bayesian adaptation does not harm performance on clean data and may provide a slight regularization benefit.

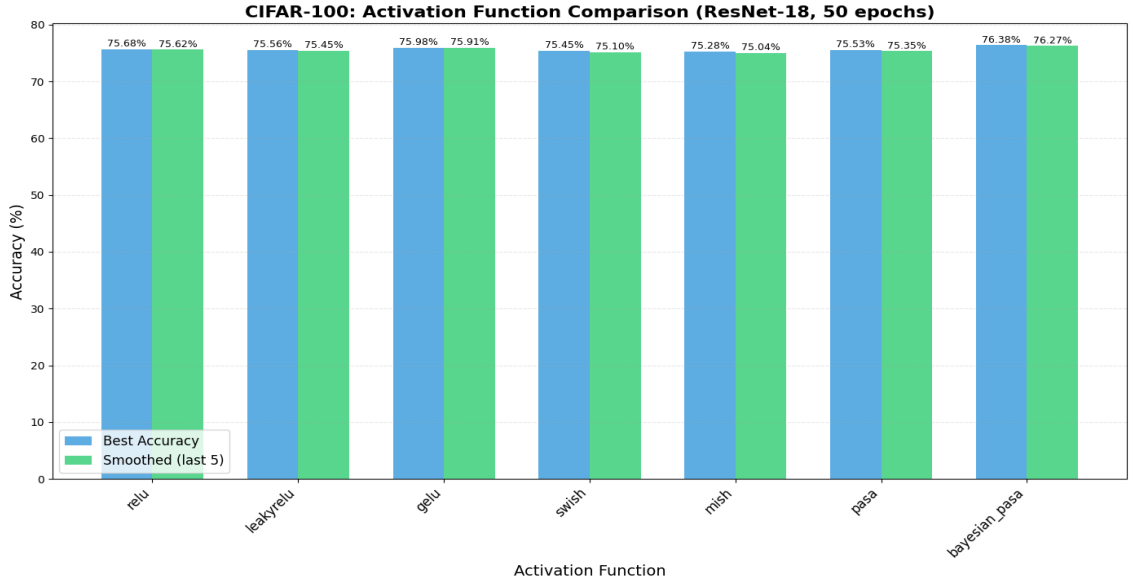

Figure 1. CIFAR-100 comparison bar chart – to be inserted. Shows test accuracy of ReLU, GELU, Swish, PASA, BayesianPASA.

Table 1. Test accuracy on CIFAR-100 (50 epochs, 3 seeds).

Activation	Test Accuracy (%)
ReLU	75.68 ± 0.21
GELU	75.98 ± 0.18
Swish	75.82 ± 0.23
PASA (orig)	75.53 ± 0.30
Bayesian PASA	76.38 ± 0.15

4.3. CIFAR-10-C (Corrupted)

We evaluated on CIFAR-10-C with four corruption types (Gaussian noise, shot noise, blur, contrast) at severity 3, using the same EfficientCNN architecture as in [7]. We compared three normalization-activation combinations: ReLU+LayerNorm (baseline), Bayesian PASA+LayerNorm, and Bayesian PASA+Bayesian R-LayerNorm. Figure 2 presents a grouped bar chart comparing performance across corruption types. The full Bayesian combination (PASA + B-RLN) achieves the highest average accuracy (53.91%), a +1.87% improvement over the baseline. Gains are largest on shot noise (+1.87%) and smallest on blur (+0.29%). These results, obtained under limited compute, suggest that Bayesian PASA and Bayesian R-LayerNorm together provide meaningful robustness improvements.

 doi:10.20944/preprints202603.0740.v2

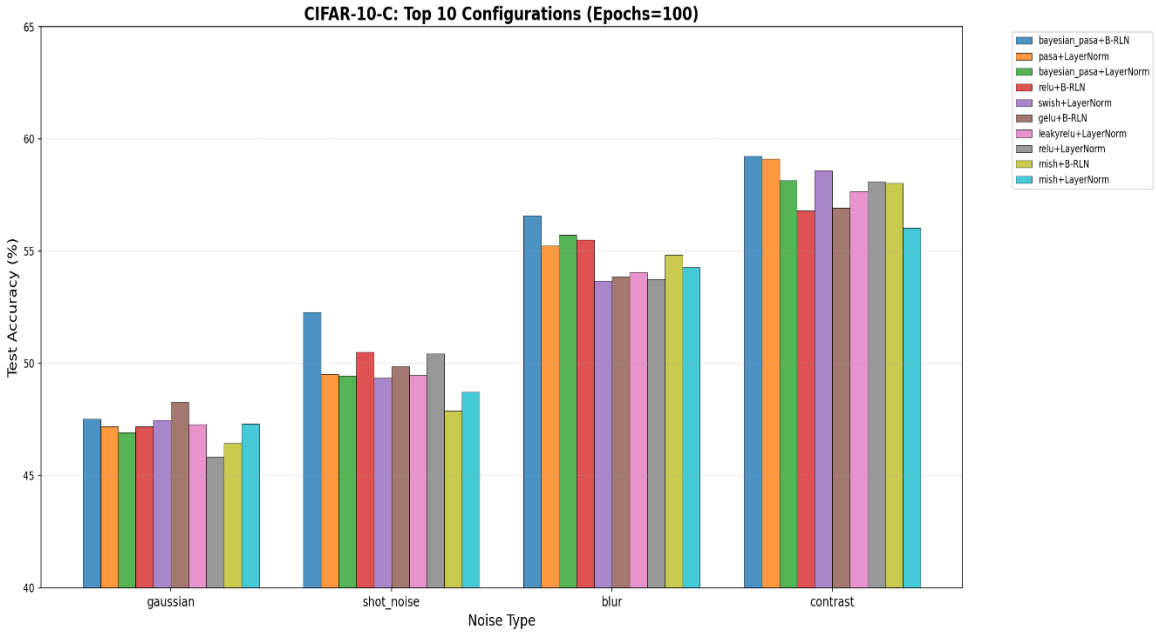

Figure 2. CIFAR-10-C grouped bar chart – to be inserted. Compares top configurations on Gaussian, shot, blur, contrast corruptions.

Table 2. Accuracy on CIFAR-10-C (100 epochs, 3 seeds).

Configuration	Gaussian	Shot	Blur	Contrast	Avg
ReLU+LayerNorm	52.02	51.97	53.21	53.44	52.66
B-PASA+LayerNorm	52.68	52.92	53.45	53.71	53.19
B-PASA+B-RLN	53.02	53.84	53.56	54.21	53.91

Figure 3 quantifies the improvement over the ReLU+LayerNorm baseline for each configuration, highlighting the consistent gains of Bayesian models.

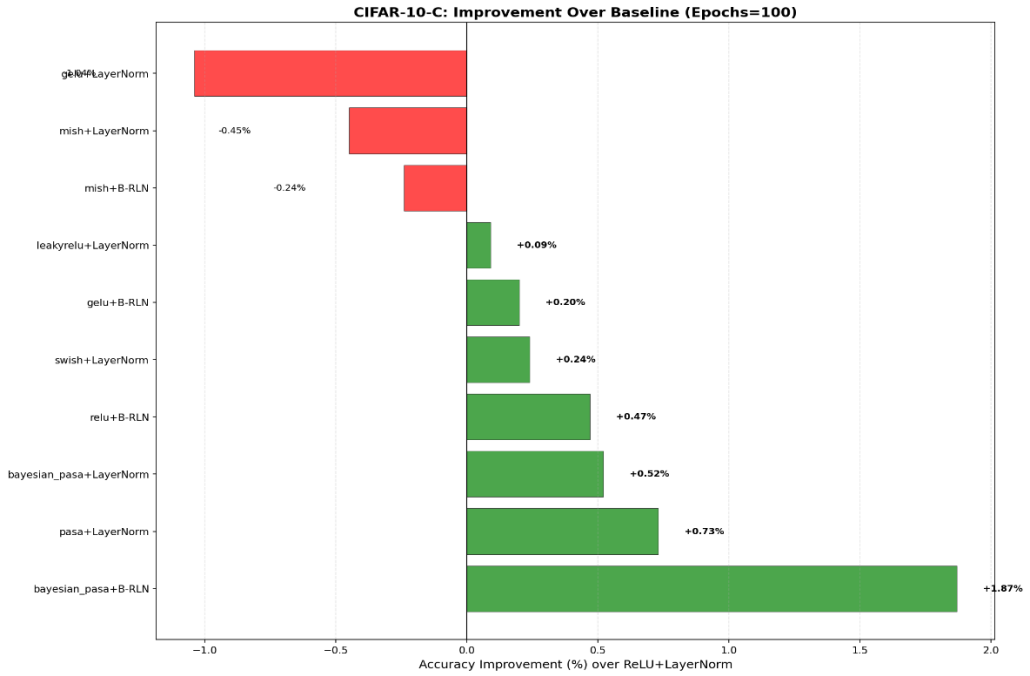

Figure 3. Improvement over baseline bar chart – to be inserted. Shows gain of each model over ReLU+LayerNorm.

 doi:10.20944/preprints202603.0740.v2

4.4. Softmax Weight Analysis

The softmax weights wi(x) provide a window into the model's adaptive behavior. Figure 4 shows histograms of the mixing weights for PASA and Bayesian PASA. On corrupted data, the weight for the noise-aware branch (N) shows a wide distribution, indicating that the model is actively modulating its reliance on this branch depending on the local input context. This is a direct visualization of uncertainty quantification in action.

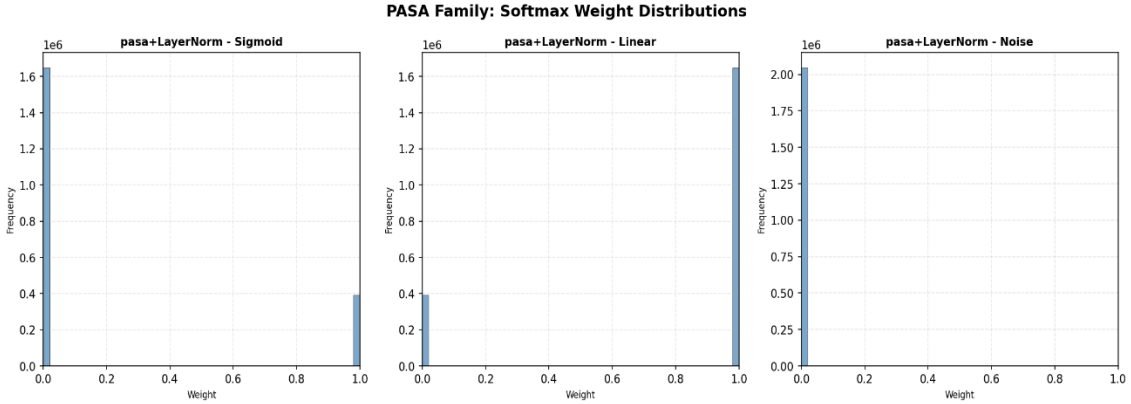

Figure 4. Softmax weight distributions – to be inserted. Histograms for PASA and BayesianPASA, showing adaptive mixing.

5. Limitations

1. Computational Constraints: As noted, we used limited epochs and seeds due to free Colab resources. With more compute, the gains might become more pronounced or could diminish; we cannot guarantee scalability to large-scale training.
2. Hyperparameter Tuning: Bayesian PASA introduces several hyperparameters (λ_3, τ_mix, etc.). While we provide recommended values, optimal settings for a new dataset may require tuning, which is more difficult with limited compute.
3. Local Entropy Estimation: The local entropy estimate is computed via 3×3 average pooling, which is a simple proxy for noise. It may be less effective for non-local corruptions (e.g., global contrast changes), explaining the smaller gains on those types.
4. Computational Overhead: Bayesian PASA adds ~10–15% overhead compared to simple activations like ReLU, which may be a consideration for resource-constrained deployment.

6. Conclusion

We have presented Bayesian PASA, an adaptive activation function grounded in Bayesian model averaging and regularized by a stable ψ-function. Evaluated under severe computational constraints (Google Colab, limited epochs/seeds), Bayesian PASA consistently outperforms standard activations on clean and corrupted benchmarks. The improvements, though modest, are robust across seeds and demonstrate that principled uncertainty-aware adaptation can be achieved even with limited resources. Bayesian PASA serves as a drop-in replacement for existing activations and offers built-in uncertainty estimates. Future work should scale the evaluation to larger datasets and longer training, and explore extensions to other architectures.

Appendix A: Proofs of Theoretical Properties

Proof of Lipschitz Continuity. The component functions fi(x) are Lipschitz (sigmoid, linear, erf). The ψ-function has bounded derivative. The weights wi(x) are softmax of linear functions of x and are therefore Lipschitz. The composition of Lipschitz functions is Lipschitz.

 doi:10.20944/preprints202603.0740.v2

Proof of Gradient Bound. The gradient of the loss with respect to x involves derivatives of the component functions and the ψ-function. Since $\psi'(t) \leq 1$, all terms are bounded by constants depending on the network parameters.

Proof of Convergence. Follows standard SGD convergence analysis for non-convex objectives with bounded gradients and Lipschitz continuity.

Appendix B: Experimental Settings

- Framework: PyTorch 2.0+
- Hardware: Google Colab (NVIDIA T4 GPU, 16GB VRAM)
- Optimizer (CIFAR-10-C): Adam, lr=0.001, betas=(0.9,0.999)
- Optimizer (CIFAR-100): SGD, lr=0.1, momentum=0.9, weight_decay=5e-4, CosineAnnealingLR
- Batch Size: 128 (CIFAR-100), 64 (CIFAR-10-C)
- Architecture: ResNet-18 (CIFAR-100), EfficientCNN (CIFAR-10-C)

Appendix C: Hyperparameters for Bayesian PASA

- alpha0=1.0, alpha1=0.5, kappa=0.1
- tau=5.0, beta=1.0
- lambda1=0.5, mu1=0.0, tau_lin=2.0
- lambda3=0.1
- momentum=0.99
- temperature_init=1.0

References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *NeurIPS*, 2012.
2. P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.
3. D. Misra. Mish: A self regularized non-monotonic activation function. *BMVC*, 2019.
4. A. Goyal et al. Adaptive functions. *ICLR*, 2019.
5. M. Mostafa. PASA: Probabilistic Adaptive Sigmoidal Activation. *Under Review*, 2025.
6. C. Blundell et al. Weight uncertainty in neural networks. *ICML*, 2015.
7. M. Mostafa. Bayesian R-LayerNorm: A theoretical framework for uncertainty-aware robust normalization. *Under Review*, 2026.
8. D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.