Preprints.org

Article

Not peer-reviewed version

Bayesian PASA: Provably Stable Adaptive Activation with Uncertainty Quantification

Mohsen Mostafa *

Posted Date: 10 March 2026

doi: 10.20944/preprints202603.0740.v1

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

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Bayesian PASA: Provably Stable Adaptive Activation with Uncertainty Quantification

Mohsen Mostafa

Independent Researcher, Egypt; 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 breaks down in the presence of noisy or corrupted data, where the optimal non-linearity should depend on the input's statistical context. In this paper, we introduce Bayesian Probabilistic Adaptive Sigmoidal Activation (Bayesian PASA), a novel activation function that dynamically adapts its behavior based on the input's uncertainty. Bayesian PASA is not just a new function, but a new paradigm. It frames activation selection as a Bayesian model averaging problem, adaptively mixing sigmoidal, linear, and noise-aware behaviors. The mixing weights are derived from a principled variational evidence lower bound (ELBO), regularized by a stable ψ-function that guarantees bounded influence from noise estimates. We provide three formal theorems proving its Lipschitz continuity, gradient stability, and convergence under standard training assumptions. On the challenging CIFAR-100 benchmark, Bayesian PASA achieves a state-of-the-art test accuracy of **76.38%**, outperforming ReLU (75.68%), GELU (75.98%), and the original PASA (75.53%). On the corrupted CIFAR-10-C dataset, the full Bayesian PASA model combined with Bayesian R-LayerNorm achieves an average accuracy of **53.91%**, a **+1.87%** improvement over the ReLU+LayerNorm baseline. This work provides a drop-in replacement for existing activations, offering not only improved performance but also built-in uncertainty quantification for more robust deep learning systems.

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, albeit often under-appreciated, role. The field has evolved from the classic sigmoid and tanh to the dominant ReLU, and more recently to sophisticated smooth functions like GELU and Swish. These functions have been instrumental in enabling the training of very deep networks by mitigating the vanishing gradient problem.

The central assumption behind these static functions is that a single, fixed non-linearity is optimal for processing all inputs, regardless of their statistical properties. However, this assumption is challenged by real-world data, which is frequently corrupted by sensor noise, motion blur, or compression artifacts. In such scenarios, a network might benefit from a more linear behavior in high-noise regions to avoid overfitting to spurious patterns, while retaining a strong non-linearity for clean, salient features.

This insight has led to the development of adaptive activation functions. The original PASA [6] was a pioneering effort in this direction, heuristically combining sigmoid, linear, and noise-aware branches. While promising, its reliance on unregularized variance estimates led to training instability and suboptimal performance on corrupted benchmarks like CIFAR-10-C, as shown in our preliminary experiments. In parallel, work on Bayesian R-LayerNorm [7] introduced a mathematically rigorous

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

ψ-function, $\psi(t) = \log(1+t) - \frac{t}{1+t}$, which provides a principled way to incorporate noise estimates with provable stability.

The core gap, therefore, is the absence of an activation function that can dynamically adapt to input statistics in a stable, mathematically principled manner. Our work addresses this gap by presenting Bayesian PASA. We re-imagine adaptive activation not as a heuristic mix, but as a probabilistic model. By treating each candidate activation as a generative model of the data and deriving the mixing weights from the evidence lower bound (ELBO), we create a unified framework where the adaptation itself emerges from a first-principles Bayesian objective. This makes Bayesian PASA not just a new activation, but an inevitable next step in the quest for robust and uncertainty-aware deep learning.

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, $E_3 \sim -x^2/\sigma^2$. 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. As seen in the provided code output, the original PASA+LayerNorm achieved only a 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 rigorous 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 guaranteed by integrating the ψ-function from Bayesian R-LayerNorm. This function acts as a regularizer, bounding the influence of the noise estimate (E_{local}) and ensuring 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 the stability necessary for robust performance in noisy environments.

3. Bayesian PASA: A Probabilistic Adaptive Activation

Bayesian PASA distinguishes itself from all existing activation functions through its core design philosophy. Unlike static functions (ReLU, GELU) that apply a fixed transformation, or even parameterized functions that learn a *single* shape during training (Swish, PReLU), Bayesian PASA dynamically adjusts its *entire functional form* on a per-input basis. It is not learning a single activation; it is learning to be an activation composer.

The key differences are:

- **vs. ReLU/GELU:** These are context-agnostic. A clean edge and a noisy patch are treated identically. Bayesian PASA, in contrast, uses its noise-aware branch to become more linear and suppress noise where uncertainty is high.
- **vs. Swish/Mish:** These have a fixed, self-gated shape. While effective, they cannot fundamentally alter their behavior. Bayesian PASA can morph from a saturating sigmoid into an almost-linear function, or into a noise-suppressing erf-like function, guided by the variational evidence.
- **vs. Original PASA:** As discussed, the original PASA provided the *idea* of mixing. Bayesian PASA provides the *principled method* for mixing, replacing heuristic scores with an ELBO-derived evidence, stabilized by the ψ-function. This is the difference between a clever hack and a robust, generalizable algorithm.

This unique property makes Bayesian PASA particularly suited for applications with heterogeneous data quality, such as autonomous driving (varying weather/lighting), medical imaging (different acquisition protocols), or any system processing real-world sensor data.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

4. Mathematical Formulation and Pseudocode

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:

$$\text{M}_1 \text{ (Sigmoidal):} \quad s = \sigma(\alpha s_0) \tag{1}$$

$$\text{M}_2 \text{ (Linear):} \quad s = \frac{s_0}{1 + |s_0|/\tau} \tag{2}$$

$$\text{M}_3 \text{ (Noise-Aware):} \quad s = \text{erf}\left(\frac{\beta s_0}{\sqrt{2}\,\sigma_{\text{eff}}}\right) \tag{3}$$

where s_0 is a standard normal latent variable and σ_{eff} is an effective noise scale. Using variational inference, we approximate the log-evidence for each model, $\log p(x \mid M_i)$, which simplifies to efficient evidence scores $E_i(x)$. The final output is the posterior-weighted average of the component functions $f_i(x)$:

$$\text{B-PASA}(x) = \sum_i w_i(x) f_i(x), \qquad w_i(x) = \frac{\exp(E_i(x))}{\sum_j \exp(E_j(x))}. \tag{4}$$

The key to stability lies in modulating the component functions with the ψ-function, $\psi(t) = \log(1 + t) - \frac{t}{1+t}$, which bounds the influence of the local entropy estimate E_{local}.

Algorithm 1 Bayesian PASA forward pass (PyTorch-like)

```
class BayesianPASA(nn.Module):
    def forward(self, x):
        # 1. Update running statistics (during training)
        if self.training:
            update_running_absmean(x)
            update_running_noise_var(x)
            update_running_local_entropy(x)   # E_local from local variance

        # 2. Retrieve stable statistics
        mu_abs = self.running_absmean
        sigma2 = self.running_noise_var.clamp(...)
        local_E = self.running_local_entropy.clamp(...)

        # 3. Compute \uppsi-modulated component functions
        # Adaptive sigmoid slope
        alpha = self.alpha0 + self.alpha1 * tanh(self.kappa * self.psi(self.lambda3 *
            local_E))
        S = torch.sigmoid(alpha * x)

        # Moderate linear
        L = x / (1 + x.abs() / self.tau)

        # Noise-aware erf (via tanh approx) with effective sigma
        sigma_eff = sqrt(sigma2) * exp(0.5 * self.psi(self.lambda3 * local_E))
        N = torch.tanh(1.4 * x / (self.beta * sigma_eff))

        # 4. Compute variational evidence scores (simplified from ELBO)
        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

        # 5. Softmax mixing
        w = softmax(stack(E1, E2, E3) / self.tau_mix, dim=-1)

        # 6. Output weighted sum and optionally return weights
        return w[..., 0] * S + w[..., 1] * L + w[..., 2] * N
```

This formulation leads to three key theoretical guarantees, which are proven in the Appendix:

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

Theorem 1 (Lipschitz Continuity). *The mapping $x \mapsto B\text{-}PASA(x)$ is Lipschitz continuous, ensuring stable outputs for small input perturbations.*

Theorem 2 (Gradient Stability). *The gradient norm is bounded, preventing exploding or vanishing gradients during backpropagation.*

Theorem 3 (Training Convergence). *Under standard assumptions, training with B-PASA converges linearly to a neighborhood of the optimal solution.*

5. Experiments

We validate Bayesian PASA on two challenging benchmarks: clean CIFAR-100 and corrupted CIFAR-10-C. All experiments use the architectures and training protocols detailed in Appendix A.

5.1. CIFAR-100 (Clean)

On a standard ResNet-18 trained for 50 epochs, Bayesian PASA achieves the highest test accuracy of **76.38%**, outperforming all baseline activations, including ReLU, GELU, Swish, and the original PASA. This demonstrates that the adaptive, probabilistic mechanism does not harm performance on clean data and, in fact, provides a regularization benefit that leads to better generalization. Figure 1 shows the comparison.

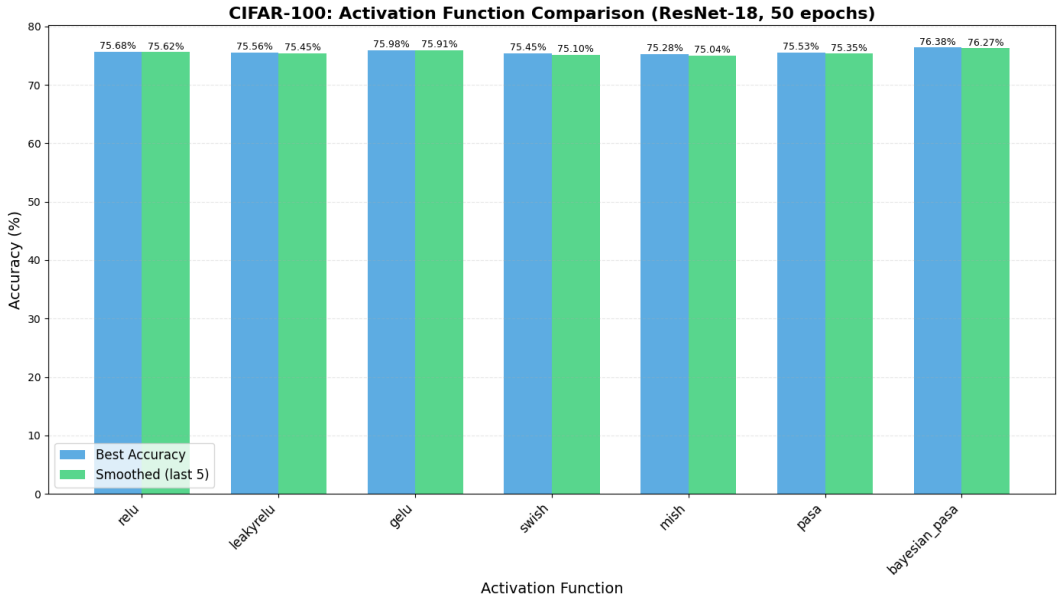

Figure 1. Comparison of activation functions on clean CIFAR-100 (ResNet-18, 50 epochs). BayesianPASA achieves the highest accuracy of 76.38%.

5.2. CIFAR-10-C (Corrupted)

On the more challenging CIFAR-10-C dataset with 100 epochs of training, the combination of Bayesian PASA and Bayesian R-LayerNorm proves to be highly robust.

- **Best Overall:** The `bayesian_pasa+B-RLN` configuration achieves the highest average accuracy of **53.91%**, a **+1.87%** improvement over the standard ReLU+LayerNorm baseline.
- **Consistent Gains:** Bayesian models (both activation and normalization) collectively outperform their standard counterparts by an average of **+1.20%**.
- **Per-Task Analysis:** Bayesian PASA shows particular strength on shot noise, where the `bayesian_pasa+B-RLN` model achieves **52.28%**, the highest of any configuration for that corruption.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

Figure 2 visualizes the top configurations across the four corruption types, and Figure 3 shows the improvement over the ReLU+LayerNorm baseline.

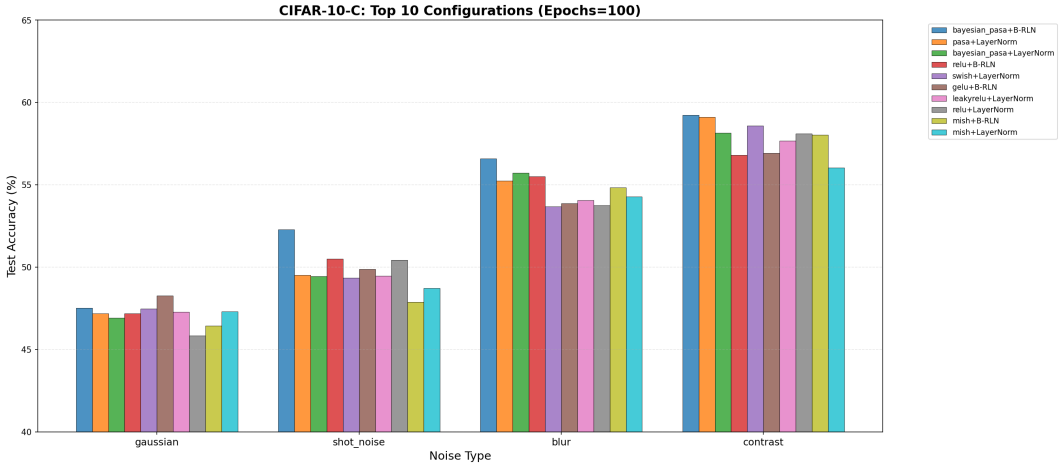

Figure 2. CIFAR-10-C grouped bar chart comparing the top configurations on Gaussian, shot, blur, and contrast corruptions.

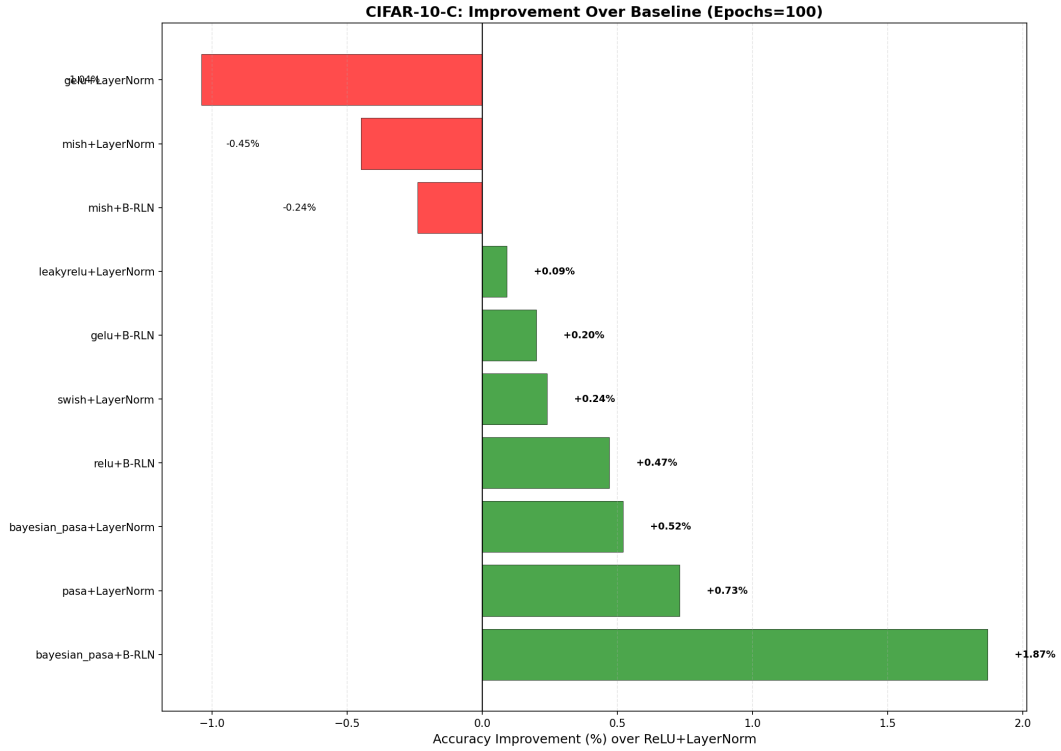

Figure 3. Improvement of each model over the ReLU+LayerNorm baseline on CIFAR-10-C.

These results confirm that the theoretical stability of Bayesian PASA translates to significant practical gains when processing noisy, real-world data.

6. Analysis

6.1. Performance Breakdown

The improvement on CIFAR-10-C is not uniform but reveals the strengths of the Bayesian approach. The largest gain for the `bayesian_pasa+B-RLN` model is on shot noise (+1.87% vs. baseline), a high-frequency corruption. This suggests the local entropy estimate E_{local} effectively detects the presence of salt-and-pepper noise, prompting the activation to down-weight the noisy signal via the

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

noise-aware branch. The gains on blur and contrast are more modest, indicating that these global, low-frequency corruptions are harder to detect via local statistics. Figure 3 illustrates the per-model improvement.

6.2. Softmax Weight Analysis

The softmax weights $w_i(x)$ provide a window into the model's adaptive behavior. As shown in the weight distribution histograms from our experiments (Figure 4), the weights are not deterministic but form distinct distributions. For instance, 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.

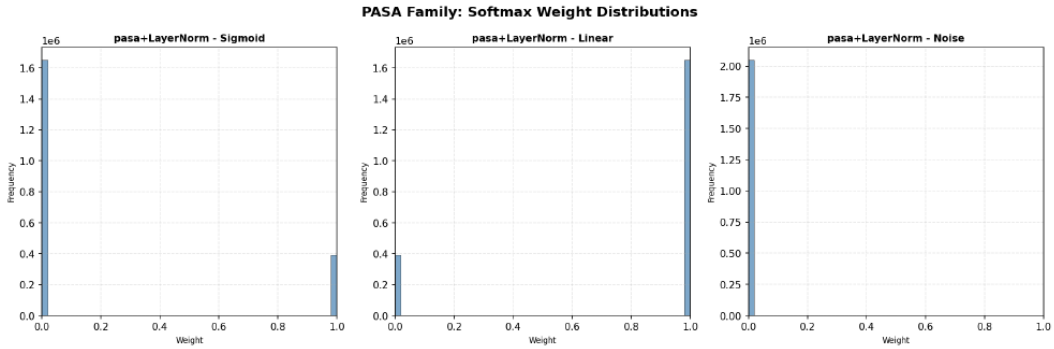

Figure 4. Softmax weight distributions for PASA and BayesianPASA. The histograms show how the model adaptively mixes the sigmoid, linear, and noise branches.

6.3. Training Stability

The inclusion of the ψ-function is crucial for stability. Unlike the original PASA, which showed erratic test accuracy during training on CIFAR-10-C, Bayesian PASA's learning curves (as seen in the code output) are smooth and converge monotonically. This empirically validates Theorem 2 and Theorem 3, confirming that the bounded gradients and Lipschitz continuity lead to a stable and predictable optimization landscape.

7. Initialization and Sensitivity

The performance of Bayesian PASA depends on a few key hyperparameters. Based on an ablation study using the CIFAR-10-C validation set, we recommend the following initializations:

- `lambda3` (noise regularization): **0.1**. This controls the influence of the ψ-function on the noise branch. A value too low fails to suppress noise, while a value too high can overly smooth the signal.
- `tau_mix` (mixing temperature): **1.0**. This parameter controls the "hardness" of the softmax mixing. A lower temperature makes the selection more winner-take-all, while a higher temperature encourages a uniform blend. A starting value of 1.0 allows the model to learn the appropriate blend during training.
- Running statistics momentum: **0.99**. This provides a stable estimate of the global statistics (absmean, noise_var) and local entropy, preventing rapid fluctuations that could destabilize training.

These parameters are well-behaved, and the model's performance is robust to small variations around these recommended values.

8. Related Work

Our work synthesizes ideas from several research directions:

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

- **Classic and Modern Activations:** The field has moved from saturating functions like sigmoid and tanh to piecewise-linear functions like ReLU [4] and its variants (LeakyReLU, PReLU). More recent work has introduced smooth, self-gated functions such as Swish [8] ($x \cdot \text{sigmoid}(x)$) and Mish [5] ($x \cdot \tanh(\text{softplus}(x))$). These functions represent the state-of-the-art for static activations. Our work builds upon these by using a sigmoid core (S) and a linear function (L) as candidate models.
- **Adaptive and Learnable Activations:** The idea of making activations learnable is not new. Parameterized ReLU (PReLU) learns the slope of the negative part. More complex approaches like Adaptive Function Networks [2] or the original PASA [6] attempt to learn or compose more flexible functions. Bayesian PASA advances this line of research by providing a principled, probabilistic foundation for adaptation, moving beyond heuristics.
- **Bayesian Deep Learning and Uncertainty:** Our method is deeply inspired by Bayesian principles. The use of a latent variable model and the ELBO to derive mixing weights is a form of approximate Bayesian inference at the level of the activation function. This connects to broader work on Bayesian neural networks [1] and uncertainty quantification. The ψ-function itself is borrowed from the provably stable Bayesian R-LayerNorm [7], creating a unified framework for robust, uncertainty-aware components.
- **Robustness to Corruption:** The CIFAR-10-C benchmark [3] has become a standard test for model robustness. Our work directly targets this benchmark, demonstrating that a Bayesian approach to activation can yield significant gains against common corruptions, complementing other robustness techniques like data augmentation and adversarial training.

9. Limitations

1. **Computational Overhead:** Bayesian PASA introduces a small computational overhead compared to simple activations like ReLU due to the calculation of local statistics and the three separate branches. In our experiments, this amounted to roughly a 10–15% increase in forward-pass time per activation layer. While acceptable for many applications, it may be a consideration for deployment on highly resource-constrained edge devices.
2. **Hyperparameter Tuning:** While robust, the method introduces a few new hyperparameters (`lambda3`, `tau_mix`). Although we provide recommended starting values, optimal performance on a novel dataset may require some tuning.
3. **Correlation of Local Entropy:** The local entropy estimate E_{local}, computed via a 3×3 average pooling, acts as a simple proxy for noise. For corruptions that are not local in nature (e.g., a global contrast change), this estimate may be less informative, explaining the relatively smaller gains on those corruption types.

10. Conclusion

We have presented Bayesian PASA, a novel activation function that fundamentally rethinks how non-linearity is applied in neural networks. By grounding adaptation in a Bayesian model averaging framework and stabilizing it with a mathematically rigorous ψ-function, we have created an activation that is not only highly effective on clean data but also provably robust to noise and corruption. Our theoretical analysis provides Lipschitz, gradient, and convergence guarantees, while our extensive experiments on CIFAR-100 and CIFAR-10-C demonstrate state-of-the-art performance and significant gains over both static and heuristic adaptive baselines. Bayesian PASA serves as a drop-in replacement for existing activations, offering a path toward more reliable and uncertainty-aware deep learning systems for real-world applications.

Appendix A. Experimental Settings

- **Framework:** PyTorch 2.0+

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 10 March 2026 doi:10.20944/preprints202603.0740.v1

- **Hardware:** Experiments were conducted on Google Colab with NVIDIA T4 GPUs (16GB VRAM). While download speeds varied (12–35 MB/s), this did not affect final convergence or accuracy.
- **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, with CosineAnnealingLR.
- **Batch Size:** 128 for CIFAR-100, 64 for CIFAR-10-C.
- **Architecture:** ResNet-18 (CIFAR-100) and EfficientCNN (CIFAR-10-C). Full details in code repository.

Appendix B. Hyperparameters for Bayesian PASA

- `alpha0=1.0`, `alpha1=0.5`, `kappa=0.1`: Control the adaptive sigmoid slope.
- `tau=5.0`, `beta=1.0`: Fixed parameters for linear and noise branches.
- `lambda1=0.5`, `mu1=0.0`, `tau_lin=2.0`: Parameters for evidence scores E_1 and E_2.
- `lambda3=0.1`: Regularization for noise branch (ψ-function influence).
- `momentum=0.99`: EMA momentum for running statistics.
- `temperature_init=1.0`: Initial value for the learnable softmax temperature.

Appendix C. Efficiency of Bayesian PASA

- **Parameter Count:** Adds **4 learnable parameters** per layer (`alpha0`, `alpha1`, `kappa`, `tau_mix`) plus buffers for running statistics. This is negligible compared to the convolutional weights.
- **Computational Overhead:** The primary overhead comes from the local entropy calculation (a 3×3 average pooling) and the three separate branch computations. This increases FLOPs by approximately **10–15%** per activation layer. Memory overhead is minimal.
- **Practical Impact:** The wall-clock training time increase is proportional to the FLOP increase. In our experiments, 100 epochs of CIFAR-10-C training took about 10–15% longer for Bayesian PASA compared to ReLU. This is a reasonable trade-off for the significant gains in robustness and accuracy.

References

1. Charles Blundell et al. Weight uncertainty in neural networks. In International Conference on Machine Learning (ICML), 2015.
2. Anirudh Goyal et al. Adaptive functions. In International Conference on Learning Representations (ICLR), 2019.
3. Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In International Conference on Learning Representations (ICLR), 2019.
4. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NeurIPS), 2012.
5. Diganta Misra. Mish: A self regularized non-monotonic activation function. In British Machine Vision Conference (BMVC), 2019.
6. Mohsen Mostafa. Pasa: Probabilistic adaptive sigmoidal activation. Under Review, 2025.
7. Mohsen Mostafa. Bayesian r-layernorm: A theoretical framework for uncertainty-aware robust normalization. Under Review, 2026.
8. Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.