AdaShift: Learning Discriminative Self-Gated Neural Feature Activation With an Adaptive Shift Factor

Sudong Cai
Graduate School of Informatics, Kyoto University
cai.sudong.t94@kyoto-u.jp

Abstract

*Nonlinearities are decisive in neural representation learning. Traditional **Act**ivation (**Act**) functions impose fixed inductive biases on neural networks with oriented biological intuitions. Recent methods leverage self-gated curves to compensate for the rigid traditional Act paradigms in fitting flexibility. However, substantial improvements are still impeded by the norm-induced mismatched feature re-calibrations (see Section 1),* i.e.*, the actual importance of a feature can be inconsistent with its explicit intensity such that violates the basic intention of a direct self-gated feature re-weighting. To address this problem, we propose to learn discriminative neural feature Act with a novel prototype, namely, **AdaShift**, which enhances typical self-gated Act by incorporating an adaptive shift factor into the re-weighting function of Act. AdaShift casts dynamic translations on the inputs of a re-weighting function by exploiting comprehensive feature-filter context cues of different ranges in a simple yet effective manner. We obtain the new intuitions of AdaShift by rethinking the feature-filter relationships from a common Softmax-based classification and by generalizing the new observations to a common learning layer that encodes features with updatable filters. Our practical AdaShifts, built upon the new Act prototype, demonstrate significant improvements to the popular/SOTA Act functions on different vision benchmarks. By simply replacing ReLU with AdaShifts, ResNets can match advanced Transformer counterparts (*e.g.*, *ResNet-50 vs. Swin-T) with lower cost and fewer parameters.*

1. Introduction

Nonlinear Act functions are indispensable for the learning of discriminative neural features [2, 7, 11, 17, 33, 39, 39, 42]. Neuronal behaviors [24, 40] originate traditional Act models, *e.g.*, Softplus [15] and ReLU [34], which are fixed and monotonic in calculations. To realize finer rectifications, recent works investigated self-gated-style Act functions based on the general prototype

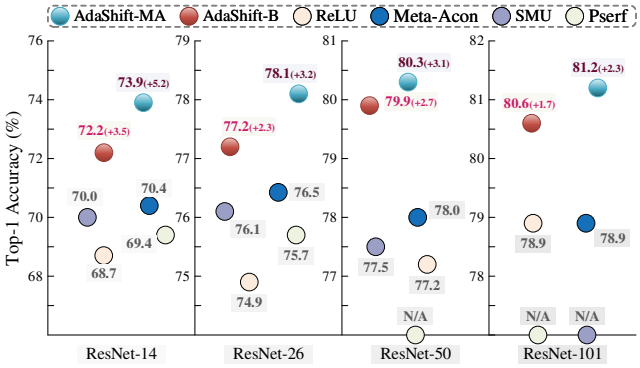

Figure 1. Comparison of our AdaShift-B and AdaShift-MA to the ReLU [34] baseline and popular/SOTA Act models [4, 5, 30] on ImageNet [13] with ResNet backbones, where the areas of the circular patterns represent the relative amount of parameters compared to the corresponding ReLU baselines. Our AdaShift-B and AdaShift-MA improve different activation functions consistently and remarkably on different backbones varying by size with negligible parameters added to the ReLU baselines.

$$\phi(x) = \varsigma(x)\, x\,, \tag{1}$$

where $x \in \mathbb{R}$ is a given feature unit (*i.e.*, scalar), $\phi : \mathbb{R} \to \mathbb{R}$ denotes the applied Act function of x, and $\varsigma : \mathbb{R} \to \mathbb{R}$ defines the re-weighting function of ϕ. As a special case, ReLU can be included in this prototype by specifying $\varsigma(x)$ as a binary masking of 0 and 1 for $x \leqslant 0$ and $x > 0$, respectively. Despite the broad applicability, ReLU leaves two practical constraints on neural Act from (1) its rigid masking on positive features, *i.e.*, unified weight assignments that possibly neutralize the discriminativeness, and (2) hard-zero-truncation on negative features that possibly leads to the "dead tensors" problem.

Recent methods addressed these by introducing smooth re-weighting functions with two assumed properties:
1. $\varsigma(x)$ is bounded (typically, $\varsigma(x) \in (0, 1)$);
2. $\varsigma(x)$ is monotonically non-decreasing about x.
These properties theoretically ensure the stability and con-

vergence of neural Act in training [45] and identify typical self-gated Act functions (*e.g.*, [16, 20, 32]) that favor feature rectifications by leaving more flexibility. However, typical self-gated functions can still fall short in adaptability to highly variational training conditions due to fixed re-weighting processes. SOTA methods [4, 5, 30] studied leveraging attention and updatable scaling/bias to enhance self-gated re-weighting by infusing more flexible inductive biases. Although effective, the substantial improvements are still hindered by the critical challenge of norm-induced *mismatched feature scoring* [6] invisible to pure biological intuitions. Cai *et al.* [6] identified the above problem based on a tailored interpretation of neural Act obtained from Multi-Criteria Decision-Making (**MCDM**, a classical problem in operational research) [9, 14, 23, 35, 37, 46, 47], where Act models were regarded as selective re-calibrators that emphasize and suppress features based on their importance scores measured by the feature-filter similarities. With this new perspective, they found that differentiated feature and filter norms possibly bias the similarities modeled with feature-filter inner products significantly, hence taking away from how important the features actually are. This inspired a rectified self-gated prototype of Act, *i.e.*,

$$\phi\left(x\right) = \varsigma\left(\varrho\left(x\right)\right) x\,, \tag{2}$$

where $\varrho\left(x\right)$ is assumed as an unknown unbiased (*i.e.*, ideal) similarity measure of x and ς preserves the property of monotonically non-decreasing of $\varsigma\left(x\right) = \varsigma\left(\varrho\left(x\right)\right)$ to $\varrho\left(x\right)$, instead of x the biased similarity. In particular, by designating $\varrho\left(x\right) = x$, prototype 2 regresses to the base form 1. Their SOTA method, IIEU [6], addressed the *mismatched feature scoring* problem by approximating $\varrho\left(x\right)$ with an adaptive **norm-decoupled importance measure** adjusted with non-local cues, thus performing amended feature re-calibrations with the rectified importance scores. Although effective, the brutal norm-decoupling in IIEU inevitably leads to constrained runtime, especially for training, due to the relatively complex gradient led by the norm-decoupled approximated similarity $\varrho\left(x\right)$.

In this paper, we present a novel Act prototype, namely, AdaShift (defined by Eq. (8)), to address the critical *mismatched feature scoring* problem in a simple yet effective manner with new intuitions in line with the MCDM interpretation. Specifically, (1) we suppose prototype 1 with properties 1 and 2 imply a critical condition that for an Act process, we have "the larger x, the more important x is," as a re-weighting function ς monotonically re-calibrates x according to its intensity. (2) By in agreement with Cai *et al.*' [6] discussion, we suppose that the importance measure of x is possibly inconsistent with the intensity of x, as the feature/filter norms influenced by the learning states of past layers and initializations can bias the current feature-filter similarity. Yet, unlike IIEU [6], we argue that feature and filter norms provide informative cues for discriminative activations and brutally decoupling norms can constrain neural features in representational capability. We identify this by rethinking the relationships of feature and filter norms from a common Softmax-based classification in a network, where we find *feature and filter norms present local and non-local cues for classifying output features, respectively*, and by generalize this understanding to general leaning blocks.

Based on the assumptions (1) and (2), in AdaShift, we introduce an adaptive shift factor Δ, leveraged on the complementary tensor-level non-local context, which learns to approximate $\varrho\left(x\right)$ by $\hat{\varrho}\left(x\right) = x + \Delta$, thus imposing dynamic inductive biases to a monotonic curve ς to rectify its intensity-based re-weighting on x by exploiting different ranges of local/non-local context of the current learning states in an interactive manner. We identify that Δ can be effectively learned to introduce remarkable improvements to networks by even surprisingly simple approaches that aggregate tensor-level channel/spatial interactions, *e.g.*, only by a vanilla LayerNorm [3] operator casted on a vector of channel statistics (*e.g.*, channel mean responses), with negligible parameters and computational cost. This allows us to propose a brand-new class of Act models, *i.e.*, practical **AdaShift**(s) by embodying the shift factor Δ with different derivatives. In particular, we mainly present two practical AdaShifts as examples, where we refer to the one that solely casts an embedded LayerNorm operator on the channel statistic vector as AdaShift-**B** (*i.e.*, -**B**asic) and we introduce AdaShift-**MA** that enhances AdaShift-B by exploiting finer-grained tensor-level context cues with a **M**inimalist-style self-**A**ttention operation, which applies LayerNorm operators to calculate Q-K-V attention and removes all the heavy linear projections to preserve the high efficiency of activation. More extensions can be created by varying Δ with finer aggregational operators for tensor-level cues (see Sec. 4.3). From a different perspective, we regard the essence of AdaShift as an adaptive fine-grained adjustment of the re-weighting curve ς *w.r.t.* x, hence creating improved ς dynamically, with the incorporated awareness of different ranges of mutual-complementary local and non-local information This avoids the explicit manual modifications to ς, which can be excessively challenging due to the ultra-complexity of underlying mappings.

The contributions of our work are three-fold: (1) We introduce a novel activation prototype with new intuitions, *i.e.*, AdaShift, to learn discriminative self-gated neural Act. (2) Based on (1), we present efficient practical AdaShifts that improve current SOTA Act models significantly. (3) We extensively validate our methods with various vision benchmarks and our new intuitions with targeted ablation studies. Code is disseminated at https://github.com/SudongCAI/AdaShift.

2. Related Work

As a maxout approximation to Softplus, ReLU rectified positive and negative inputs by binary masking of 0 and 1, respectively. This paradigm encouraged various derivatives. LeakyReLU [31] suggested a slight leakage factor to the negative interval to make use of negative inputs. PReLU [18] involved negative inputs in parameter updating by an updatable slope. ELU [12] imposes exponential rectifications on negative features. Recent efforts have been taken to develop self-gated-style functions by varying the re-weighting curves ς. As representative methods, SiLU [16] re-weighted features by a Sigmoid function and GELU [20] instead leveraged a Gauss-Error-Function-based (ERF) function to realize finer feature rectifications. Inspired by SiLU, Mish [32] suggested a composite function of Tanh and Softplus. Although demonstrating clear accuracy gains to basic Act functions, typical self-gated functions still found limitations in adaptability.

To compensate for fitting flexibility, SOTA methods introduced auxiliary trainable scaling/bias terms and embedded contextual cues to self-gated Act. Swish [36] extended SiLU by assigning a learnable scaling factor to the input, *i.e.*, $\phi(x) = \varsigma(\kappa x)\, x$, where $\kappa \in \mathbb{R}$. ACON-C [30] further extended Swish by introducing a learnable bound. Meta-ACON [30] enhanced ACON-C by generalizing SE-Net-based [21] channel attention to predict a content-aware input scaling factor. Several SOTA works also investigated new approaches to the ERF-based Act. Biswas *et al.* [4] proposed two trainable derivatives of GELU, namely, ErfAct and Pserf, where the former and the later employed exponential and Softplus functions with updatable coefficients to scale the activation inputs, respectively. Encouraged by ACONs, Smooth Maximum Units (*i.e.*, SMU-1 and SMU) [5] suggested an ERF-based Act with flexible upper and lower bounds. These new ideas significantly extended the design space of self-gated Act while still leaving the norm-induced *mismatched feature scoring* [6] problem unsettled, which put a critical constraint on further discriminativeness.

Cai [6] clarified the *mismatched feature scoring* problem and presented IIEU as the initial solution. IIEU was learned with a tailored paradigm to eliminate the norm-induced feature-filter similarity biases by explicit norm-decoupling. This idea demonstrated SOTA improvements on different networks, especially for small-size models. However, the brutal norm-decoupling on Act inputs likely neutralizes the discriminativeness. This lies in the new observation that feature/filter norms contain informative local details and dataset-level non-local cues for optimizing network parameters. In contrast, our prototype, *AdaShift*, learns discriminative feature Acts by comprehensively exploiting local and contextual cues of three different ranges in a particularly simple but effective manner. As a core spirit, unlike IIEU, AdaShift addresses norm-induced *mismatched feature scoring* by temperate dynamic adjustments that evolve a vanilla self-gated ς by adapting to the current learning states. This saves the meaningful norm-related cues and enables AdaShift to improve popular/SOTA Acts.

3. Intuition and Method

In this section, we first discuss our new intuitions that inspire AdaShift prototype and then present two novel practical AdaShift derivatives that achieve SOTA improvements over neural Act models with low computational cost, which we refer to as AdaShift-B and AdaShift-MA, respectively.

3.1. Preliminaries

Our discussion adopts the preliminary settings suggested by Cai in IIEU [6], which first considers a set of simple settings with image inputs: (1) A network includes T sequential learning layers indexed by $\tau = 1, 2, \ldots, T$. (2) Let $\mathbf{X}^\tau \in \mathbb{R}^{C^\tau \times H^\tau \times L^\tau}$ denotes the input feature map of the layer-τ, where C^τ and $H^\tau \times L^\tau$ show the number of channels and the spatial resolution, respectively. (3) The learning of the layer-τ at a spatial location $(h, l) \in \Omega_{H^\tau \times L^\tau}$ is denoted by $x_c^{\tau+1}(h,l) := \phi(\tilde{x}_c^\tau(h,l))$, where $\boldsymbol{w}^\tau(c) \in \mathbb{R}^{C^\tau}$ and $\boldsymbol{x}^\tau(h,l) \in \mathbb{R}^{C^\tau}$ denote the vectorial filter-c and feature $\boldsymbol{x}^\tau(h,l) \in \mathbb{R}^{C^\tau}$, respectively; $\Omega_{H^\tau \times L^\tau}$ represents the spatial lattice of $\mathbf{X}^\tau$ and $\tilde{x}_c^\tau(h,l) = \langle \boldsymbol{w}^\tau(c), \boldsymbol{x}^\tau(h,l) \rangle$ denotes the feature-filter inner product. Note that (a) the layer-τ includes $C_{\tau+1}$ filters; (b) ϕ denotes a given Act function and we rewrite form 2 as $\phi(\tilde{x}_c^\tau(h,l)) = \varsigma(\tilde{x}_c^\tau(h,l))\, \tilde{x}_c^\tau(h,l)$ for clarity (also applicable to prototype 2), where ς is the re-weighting function for feature re-calibration.

Note that (1) in discussions of intuitions, we temporarily omit normalization layers (*e.g.*, BatchNorm [22] and LayerNorm [3]) and biases for simplicity (if not specified) and consider them in the formulations of practical methods; (2) for a convolution operation with $K \times K$ field, the supposed settings can be simply met by vectorizing the neighborhood of features/filters to the shape $C^\tau \cdot K^2$ from $C^\tau \times K \times K$. (3) We omit the layer index (*i.e.*, τ) and pixel coordinate (*i.e.*, (h, l)) in the subsequent text for simplified notations. For example, $\boldsymbol{w}^\tau(c)$, $\boldsymbol{x}^\tau(h,l)$, and $\tilde{x}_c^\tau(h,l)$ are denoted by $\boldsymbol{w}$, $\boldsymbol{x}$, and $\tilde{x}$, respectively. By following the MCDM interpretation [6], we regard (1) a filter $\boldsymbol{w}$ as a learnable ideal candidate which in MCDM [23, 35, 37, 46] denotes the acquirable or virtual optimal decision/choice applied to measure the performance of an alternative candidate by the similarity: (2) a feature vector $\boldsymbol{x}$ as an alternative candidate and its importance score about the corresponding criteria is measured by its similarity to the filter $\boldsymbol{w}$.

3.2. AdaShift: Intuition and Prototype

We begin by clarifying our new intuitions that inspire AdaShift. First, based on the above understanding with the

preliminary settings, (1) we identify that typical self-gated Act functions (*e.g.*, [16, 20, 32]) based on the prototype 1 with properties 1 and 2 imply a critical condition that the importance score of a feature vector $\boldsymbol{x}$ about the criteria of a filter $\boldsymbol{w}$ is (strictly) positively correlated to the intensity of the input of Act, *i.e.*, $\tilde{x}$ the feature-filter inner-product. This lies in the fact that their re-weighting functions, *i.e.* ς, are assumed monotonically non-decreasing about $\tilde{x}$. (2) However, as feature/filter norms can bias the intensity of an inner product as a similarity measure, the implied condition in (1) is likely violated. Therefore, we suppose that the unbiased (*i.e.*, ideal) similarity measure (denoted by $\varrho(\tilde{x})$) of $\boldsymbol{x}$ to $\boldsymbol{w}$ is not strictly consistent with $\tilde{x}$ over the whole domain (*i.e.*, we agree that the *mismatched feature scoring* [6] problem is applicable to self-gated Act). (3) The analysis in (2) indicates that a basic solution to address *mismatched feature scoring* for self-gated Act is to introduce appropriate ς completely in line with the unbiased similarity measure. However, due to the extreme complexity of underlying mappings of neural learning, the accurate definition of $\varrho(\tilde{x})$ can be excessively difficult. Cai *et al.* [6] proposes to approximate $\varrho(\tilde{x})$ by a tailored learnable prototype, namely, IIEU, leveraging explicit norm-decoupling, *i.e.*,

$$\phi(\tilde{x}) = \varsigma\left(\frac{\tilde{x}}{\|\boldsymbol{x}\|\,\|\boldsymbol{w}\|} + \nu\right)\tilde{x}\,, \tag{3}$$

where $\|\boldsymbol{x}\|\,\|\boldsymbol{w}\| > 0$ is assumed and ν is a trainable bias term to enhance fitting flexibility. Whereas, this paradigm inevitably brings relatively complex gradients, as the (partial) derivative of $s(\boldsymbol{w}) = \frac{\tilde{x}}{\|\boldsymbol{x}\|\|\boldsymbol{w}\|}$ *w.r.t.* $\boldsymbol{w}$ is computed by

$$\nabla_{\boldsymbol{w}} s(\boldsymbol{w}) = \frac{\|\boldsymbol{w}\|^2\boldsymbol{x} - \boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{x}\|\,\|\boldsymbol{w}\|^3}\,, \tag{4}$$

where T is the transpose operation of matrix/vector.

Further, we argue that brutally decoupling feature and filter norms (*i.e.*, $\|\boldsymbol{x}\|$ and $\|w\|$) from $\tilde{x}$ likely neutralize the discriminativeness of activated features, as we identify

Intuition 1. Feature and filter norms present local and dataset-level non-local cues, respectively.

We obtain this intuition by rethinking a common classification process with a Softmax-based classifier. Below, we formalize our discussion of Intuition 1.

Discussion 1. We consider a common Softmax-based classification process that takes the vectorial outputs of the classification head (*i.e.*, the last linear layer) as inputs. Let

(1) $\boldsymbol{w}(i) \in \mathbb{R}^C$ denotes a learned filter from the classification head which includes N filters in total, *i.e.* N is the number of classes to categorize and $\boldsymbol{w}(i)$ is learned to represent the class-i;

(2) $\boldsymbol{x} \in \mathbb{R}^C$ denotes a vectorized (*i.e.*, average-pooled) feature inputted to the classification head, served as the learned representation of a raw exemplar (*e.g.*, image);

(3) $\tilde{x}_i = \langle \boldsymbol{w}(i), \boldsymbol{x} \rangle \in \mathbb{R}$ is the corresponding feature-filter inner-products of $\boldsymbol{x}$ and $\boldsymbol{w}(i)$;

(4) $b_i \in \mathbb{R}$ denotes the learned bias term added to the linear projections induced by the filter $\boldsymbol{w}(i)$.

Note that we consider $\forall \boldsymbol{w}(i), \boldsymbol{x}$, $\boldsymbol{w}(i) \neq \boldsymbol{0}$ and $\boldsymbol{x} \neq \boldsymbol{0}$ (*i.e.*, $\|w(i)\| \neq 0$ and $\|x\| \neq 0$) to ensure a meaningful classification. Without loss of generality, let us discuss an assumed case that $\boldsymbol{x}$ is categorized as the class-i. That is, for an arbitrarily given filter $\boldsymbol{w}(j)$ different from $\boldsymbol{w}(i)$, we have the following inequality holds for any $i \neq j$:

$$\begin{aligned}
&\frac{e^{\tilde{x}_i + b_i}}{\sum_{c=1}^{C} e^{\tilde{x}_c + b_c}} > \frac{e^{\tilde{x}_j + b_j}}{\sum_{c=1}^{C} e^{\tilde{x}_c + b_c}} \Longleftrightarrow e^{\tilde{x}_i + b_i} > e^{\tilde{x}_j + b_j} \\
&\Longleftrightarrow e^{\langle \boldsymbol{w}(i), \boldsymbol{x} \rangle + b_i} > e^{\langle \boldsymbol{w}(j), \boldsymbol{x} \rangle + b_j} \\
&\Longleftrightarrow e^{\|\boldsymbol{w}(i)\|\|\boldsymbol{x}\|\cos\theta_{\boldsymbol{w}(i),\boldsymbol{x}} + b_i} > e^{\|\boldsymbol{w}(j)\|\|\boldsymbol{x}\|\cos\theta_{\boldsymbol{w}(j),\boldsymbol{x}} + b_j}\,.
\end{aligned} \tag{5}$$

Then, as exponential function is monotonically increasing on $\mathbb{R}$, we have inequality 5 equivalent to

$$\|\boldsymbol{w}(i)\|\,\|\boldsymbol{x}\|\cos\theta_{\boldsymbol{w}(i),\boldsymbol{x}} + b_i > \|\boldsymbol{w}(j)\|\,\|\boldsymbol{x}\|\cos\theta_{\boldsymbol{w}(j),\boldsymbol{x}} + b_j\,. \tag{6}$$

As biases are fixed after learning, let $\alpha = b_j - b_i$, we can rewrite the inequality 6 as

$$\|\boldsymbol{w}(i)\|\cos\theta_{\boldsymbol{w}(i),\boldsymbol{x}} - \|\boldsymbol{w}(j)\|\cos\theta_{\boldsymbol{w}(j),\boldsymbol{x}} > \frac{\alpha}{\|\boldsymbol{x}\|}\,. \tag{7}$$

In particular, our major observations from inequality 7 are:

1 For the cases where $\|\boldsymbol{x}\| \gg |\alpha|$, *i.e.* $\frac{\alpha}{\|\boldsymbol{x}\|}$ close to 0, the classification of $\boldsymbol{x}$ is (almost) determined by *the filter norms* (e.g., $\|\boldsymbol{w}(i)\|, \forall i \in \{1, 2, \ldots, C\}$) *and norm-decoupled feature-filter similarities,* i.e. *cosine similarities in the discussed case* (e.g., $\cos\theta_{\boldsymbol{w}(i),\boldsymbol{x}}$).

2 For where the feature norms $\|\boldsymbol{x}\|$ and the (absolute intensities of) learned biases $|\alpha|$ are comparable, or $\|\boldsymbol{x}\| \ll |\alpha|$ (hardly exist, as biases are typically small values to avoid neutralizing feature details and over-fittings), the norm-decoupled feature-filter similarities, filter norms, and feature norms are all non-trivial.

3 The norm-decoupled feature-filter similarities and the filter norms are decisive factors to classify $\boldsymbol{x}$, ***regardless of the relative relationship of*** $\|\boldsymbol{x}\|$ and α.

In general, these findings indicate that

1. Filter norms prevalently possess dataset-level non-local cues, and filters leverage these non-trivial context cues to cast significant influences on feature recognitions.
2. The feature norm cues are particularly meaningful when the feature norms are relatively small or close to the learned biases. This attribute induces conditional influences on feature recognition. Intuitively, features with small norms reflect relatively lower confidences/higher uncertainties of identification, therefore feature norms become informative to present private details.

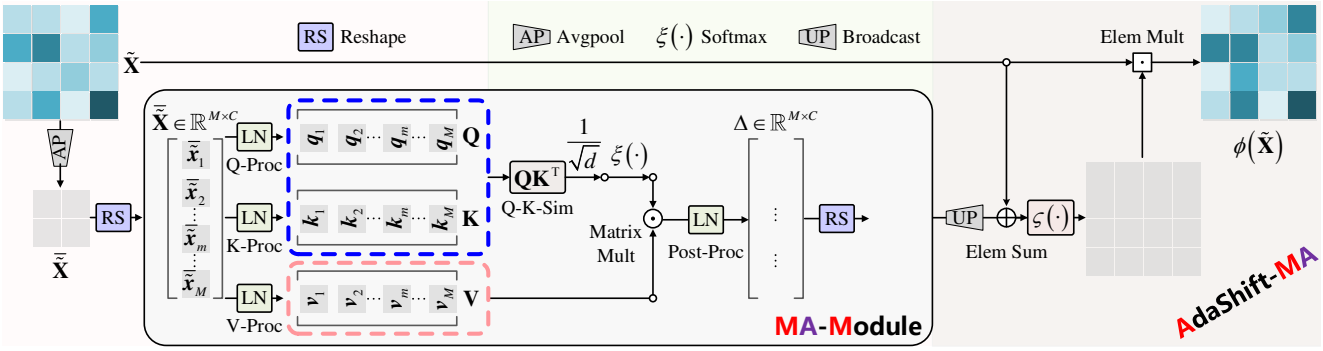

Figure 2. Illustration of AdaShift-MA. $M = \lceil H/K_H \rceil \cdot \lceil L/K_L \rceil$. "Elem" denotes "Element-wise" and "Mult" denotes "Multiplication."

Further discussions are included in Supp.

We generalize Intuition 1 to common learning layers where the filters are employed to select feature tokens by the feature-filter inner products and identify a promising solution to alleviate norm-induced biases is to cast gentle adaptive adjustments on feature/filter norms, or $\tilde{x}$ them-self since norms are components of $\tilde{x}$. We suppose a key to realizing effective adaptive adjustments is to incorporate ***complementary learning cues*** to compensate for self-gated recalibration and propose Adashift prototype, *i.e.*,

$$\phi\left(\tilde{x}\right) = \varsigma\left(\tilde{x} + \Delta\right)\tilde{x}\,, \tag{8}$$

where Δ defines a learnable shift factor to perform an efficient fine-grained translation on $\tilde{x}$ by exploiting tensor-level context cues; ς denotes a typical self-gated re-weighting function where ***we apply a Sigmoid function by default*** (*i.e.*, the same as SiLU's [16] ς), yet demonstrate wild applicability to various options of ς of different self-gated Act functions (shown in Supp). Ensured by the simple prototype, AdaShift is efficient in both inference and gradient calculation, where the (partial) derivative of $\boldsymbol{w}$ is

$$\begin{aligned}\nabla_{\boldsymbol{w}}\phi\left(\boldsymbol{w}\right) &= \frac{\partial\left(\varsigma\left(\tilde{x}+\Delta\right)\tilde{x}\right)}{\partial \boldsymbol{w}} \\ &= \frac{\partial\varsigma\left(\tilde{x}+\Delta\right)}{\partial\left(\tilde{x}+\Delta\right)}\frac{\partial\left(\tilde{x}+\Delta\right)}{\partial \boldsymbol{w}}\tilde{x} + \frac{\partial\tilde{x}}{\partial \boldsymbol{w}}\varsigma\left(\tilde{x}+\Delta\right) \\ &= \varsigma'\left(\tilde{x}+\Delta\right)\tilde{x}\left(\boldsymbol{x} + \frac{\partial\Delta}{\partial \boldsymbol{w}}\right) + \varsigma\left(\tilde{x}+\Delta\right)\boldsymbol{x}\,,\end{aligned} \tag{9}$$

where the shift factor Δ is assumed as a function of $\boldsymbol{w}$. Eq. (9) indicates that AdaShift can work at a low training cost by employing a relatively simple Δ.

We further clarify the working mechanism of AdaShift by comprehensively comparing it to other prospective prototypes (Sec. 4.3) and SOTA self-gated Act functions built on the modified prototypes of 1 (*e.g.*, [5, 30]) (in Supp) with a tailored analysis, where we identify a set of critical properties, *i.e.*, whether a self-gated prototype is capable of (1) casting flexible variations to inputs, *e.g.*, varying an input from a positive value to negative based on the current learning states; (2) realizing fine-grained adjustments to the inputs; (3) constraining the changes of (1) and (2) within the re-weighting processes without influential leakages.

3.3. Practical Method

We present **AdaShift-B** (-**B**asic) and **AdaShift-MA** (-**M**inimalist **A**ttention) (Fig. 2) as two examples of practical AdaShifts by embodying the adaptive shift factor Δ with two different efficient designs. AdaShift-B adaptively translates inputs only by leveraging a LayerNorm (LN) to learn tensor-level non-local cues on a global vector of channel statistics. AdaShift-MA further improves AdaShift-B by incorporating finer-grained tensor-level non-local cues, dynamically, with a minimalist self-attention-based module embedded in the re-weighting function ς. Our practical AdaShifts demonstrate SOTA improvements to Act functions with negligible parameters and computational cost.

AdaShift-B. For AdaShift-B, we let Δ be

$$\Delta = \left[\mathrm{LN}\left(\mathrm{avgpool}_{H\times L}\left(\tilde{\mathbf{X}}\right)\right)\right]_c\,, \tag{10}$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^{C\times H\times L}$ denotes the input tensor and c denotes the channel index of $\tilde{x}$ (for alignment); $\mathrm{avgpool}_{H\times L}$ denotes the average-pooling on the global spatial extent $\Omega_{H\times L}$ to generate a vector of channel global statistics $\bar{\tilde{\boldsymbol{x}}} \in \mathbb{R}^C$. LN denotes the LayerNorm to gather tensor-level non-local cues from the channel global statistics.

AdaShift-MA. We propose a minimalist self-attention-based Δ for AdaShift-MA, *i.e.*,

$$\Delta = \left[\mathrm{MA}\left(\mathrm{avgpool}_{K_H\times K_L}\left(\tilde{\mathbf{X}}\right)\right)\right]_c\left(h_K, l_K\right)\,, \tag{11}$$

where $\mathrm{avgpool}_{K_H\times K_L}$ denotes a non-overlapped local average-pooling with a kernel-size of $K_H \times K_L$ ($K_H, K_L \in \mathbb{Z}^+$), which produces a patch of channel local statistics $\bar{\tilde{\mathbf{X}}} \in \mathbb{R}^{C\times\left\lceil\frac{H}{K_H}\right\rceil\times\left\lceil\frac{L}{K_L}\right\rceil}$; $(h_K, l_K) = \left(\left\lceil\frac{h}{K_H}\right\rceil, \left\lceil\frac{l}{K_L}\right\rceil\right)$ means the spatial index corresponding to $\tilde{x}$. In particular, MA avoids heavy FLOPs and parameters by replacing all the linear projections with vanilla LayerNorm operations. We

suppose this change is feasible for self-gated neural Act, as the core is to mine effective non-local cues to induce dynamic yet gentle adjustments on inputs within the ς.

Noting that the above formulations of Δ are tailored to normalized inputs $\tilde{x}$ (*e.g.*, feature-filter inner-products processed by BN or LN), otherwise triggering an imbalanced summation of $\tilde{\mathbf{X}}$ and Δ, since Z-Scoring in a normalization layer (*i.e.*, $\frac{\tilde{\mathbf{X}}-\mu_x}{\sigma_x}$, where μ_x and σ_x are the concerned mean and standard deviation, respectively) actually casts pre-scalings on inputs. We suppose this will impede the effective parameter update hence resulting in accuracy drops (discussed in Sec. 4.3). The version of practical AdaShift with tailored modifications for MetaFormer blocks with unnormalized Act inputs (in FFNs) is described in Supp.

4. Experiment

We evaluate the effectiveness and versatility of our practical AdaShifts on various vision benchmark datasets, *i.e.*, ImageNet [13] and CIFAR-100 [25] image classification; COCO [27] object detection (*in Supp*); KITTI-Materials [8] road scene material segmentation (*in Supp*). Our AdaShift-B and -MA are validated by comprehensive experimental comparisons with popular/SOTA Act functions, *i.e.*, (1) Softplus [15], ReLU [34], and ReLU derivatives [12, 18, 31]; (2) popular static self-gated families including [16, 20, 32]; (3) SOTA dynamic self-gated families including [4, 5, 30, 36]; (4) others: [1, 6, 10, 29, 33]. We further validate our AdaShift prototype through extensive ablation studies and analysis of the key observations corresponding to our intuitions and methodological clarifications in Sec. 3.

4.1. ImageNet Classification

Implementation details. We evaluate our practical AdaShifts with three popular kinds of networks of various model sizes, *i.e.*, ResNet [19] and two lightweight networks, MobileNetV2 [38] (in Supp) and ShuffleNetv2 [28] (in Supp), where the baseline networks adopt ReLU as the Act function. For fair comparisons, we adopt the basic CNN training-evaluation protocols [30, 48] for all the implemented ResNets and MobileNetV2(s)/ShuffleNetv2(s), respectively (As detailed in Supp). Our experiments are conducted with $4\times$ A6000 GPUs.

Experimental results. We report the comparative results of our AdaShift-B/-MA and popular/SOTA Act functions with various networks on ImageNet in Tabs. 1 and 2, where our major observations are 3-fold: (1) AdaShift-B enjoys significant improvements over the popular/SOTA Act methods on different networks of various model sizes and AdaShift-MA boosts AdaShift-B further. Our AdaShifts achieve these large accuracy gains with negligible computational costs and additional parameters to the ReLU baselines (a detailed efficiency analysis is added in Supp). (2) Compared to the current SOTA, IIEU [6], AdaShift-B achieves superior accuracies on deep ResNets with far higher practical efficiency in both training and testing phases (*i.e.*, measured by throughput) by simpler inference and gradient computations (as detailed in Supp). This validates the significant applicability and practicality of our methods. (3) Enhanced by AdaShifts, networks of relatively small sizes and higher efficiencies can outperform/match the counterparts with far larger scales and deeper layers, *e.g.*, ResNet-50s with AdaShift-B and -MA show remarkable improvements to the large-size ResNet-101 with nearly half the model size and computational cost. These validate our AdaShift for discriminative neural feature Act.

It is worth noting that these results on ImageNet are all achieved by conducting the basic CNN training recipes [48] of 120 epochs with the raw data augmentations. By applying an improved 300-epoch CNN training recipe [44] inspired by the standard Transformer training recipe [41], we demonstrate that ResNets can match advanced Transformers with fewer parameters and higher efficiency just by simply replacing ReLU with our AdaShifts (in Supp).

4.2. CIFAR-100 Classification

Implementation details. We conduct experimental comparisons of our AdaShift-B and -MA with popular/SOTA Act functions on CIFAR-100 with a public CIFAR version [43] of ResNets which have fewer parameters and computations than the ImageNet network counterparts. For fair comparisons, we train each network from scratch using the standard training recipes [26] (as detailed in Supp).

Experimental results. Tab. 3 reports the experimental results, where our **AdaShift-B** and **-MA** improve the popular/SOTA Act functions remarkably, which are consistent with the evaluations on ImageNet. These validate the applicability of our AdaShift(s) for datasets of different scales.

4.3. Ablation Study

AdaShift prototype. We propose the AdaShift prototype based on Intuition 1. As a complementary investigation to Sec. 3.2, we verify our prototype through a targeted ablation study, where we set a series of **C**ontrol **G**roups (**CG**s) of modified AdaShift-B(s) built on various prospective prototypes of Act functions and compare our original AdaShift-B to these CGs. In Tab. 4, we specify the prototypes of CGs and report the comparative results on CIFAR100 with CIFAR-ResNet-56, where ReLU is the baseline. Note that (1) all the compared methods use Sigmoid as the ς; (2) Δ is defined by Eq. (10) (Δ_1 and Δ_2 are assigned independently) and κ is specified as channel-wise trainable parameters; (3) CG-1 and CG-2 are equivalent to SiLU [16] and Swish [36], respectively. Our major observations and the supposed explanations are 4-fold: (1) AdaShift yields the highest accuracy among all the compared prototypes. *This*

Table 1. Comparison of different Act functions with the small-size ResNet-14 and -26 [19] backbones on ImageNet. We train each network from scratch with the same training recipes, where "$(+\cdot)$" presents the improvements in Top-1 accuracy of our AdaShift-B and -MA over the ReLU baselines. "NaN" means failed training.

Backbone	ResNet-14 [19]			ResNet-26 [19]		
Activation	#Params.	FLOPs	Top-1 (%)↑	#Params.	FLOPs	Top-1 (%)↑
ReLU [34]	10.1M	1.5G	68.7	16.0M	2.4G	74.9
LeakyReLU [31]	10.1M	1.5G	68.8	16.0M	2.4G	74.9
Softplus [15]	10.1M	1.5G	69.5	16.0M	2.4G	75.7
ELU [12]	10.1M	1.5G	69.1	16.0M	2.4G	75.5
GELU [20]	10.1M	1.5G	69.6	16.0M	2.4G	75.7
SiLU [16]	10.1M	1.5G	69.6	16.0M	2.4G	75.8
Mish [32]	10.1M	1.5G	69.4	16.0M	2.4G	75.8
Swish [36]	10.1M	1.5G	69.9	16.0M	2.4G	76.1
ErfAct [4]	10.1M	1.5G	NaN	16.0M	2.4G	75.7
Pserf [4]	10.1M	1.5G	69.4	16.0M	2.4G	75.7
SMU [5]	10.1M	1.5G	70.0	16.0M	2.4G	76.1
SMU-1 [5]	10.1M	1.5G	68.5	16.0M	2.4G	75.1
ACON-C [30]	10.1M	1.5G	69.0	16.0M	2.4G	75.6
Meta-ACON [30]	10.1M	1.5G	70.4	16.1M	2.4G	76.5
AdaShift-B (Ours)	10.1M	1.5G	**72.2**(***+3.5***)	16.0M	2.4G	**77.2**(***+2.3***)
AdaShift-MA (Ours)	10.1M	1.5G	**73.9**(***+5.2***)	16.1M	2.4G	**78.1**(***+3.2***)

Table 2. Comparison of different Act functions with ResNet-50 and -101 [19] backbones on ImageNet. We report the implemented results for our AdaShift-B/-MA and the official results for all the other compared models. "N/A" denotes non-applicable/unknown.

Activation	Backbone	#Params.	FLOPs	Top-1(%)↑
ReLU [34]	ResNet-50 [19]	25.6M	4.1G	77.2
+SE-Net [21]		28.1M	4.1G	77.8
PReLU [18]		25.6M	4.1G	77.1
PWLU [49]		N/A	N/A	77.8
SMU [5]		25.6M	4.1G	77.5
SMU-1 [5]		25.6M	4.1G	76.9
FReLU [29]		25.7M	4.0G	77.6
DY-ReLU [10]		27.6M	N/A	77.2
ACON-C [30]		25.6M	3.9G	76.8
Mt-ACON [30]		25.8M	3.9G	78.0
IIEU [6]		25.6M	4.2G	79.7
AdaShift-B		25.6M	4.1G	**79.9**(***+2.7***)
AdaShift-MA		25.7M	4.2G	**80.3**(***+3.1***)
ReLU [10]	ResNet-101 [19]	44.5M	7.8G	78.9
+SE-Net [21]		49.3M	7.9G	79.3
FReLU [29]		45.0M	7.8G	77.9
ACON-C [30]		44.6M	7.6G	77.9
Mt-ACON [30]		44.9M	7.6G	78.9
IIEU [6]		44.7M	7.9G	80.3
AdaShift-B		44.6M	7.8G	**80.6**(***+1.7***)
AdaShift-MA		44.9M	8.1G	**81.2**(***+2.3***)

validates our designs for the AdaShift prototype. (2) CG6 which equals to $\phi(\tilde{x}') = \varsigma(\tilde{x}')\tilde{x}', \tilde{x}' = \tilde{x} + \Delta$ improves CG1 and CG2 but leads to accuracy drops to AdaShift-B. *This demonstrates that (a) the tensor-level non-local cues are contributive to adaptive feature translations; (b) the* mismatch feature scoring *problem of Act is hard to be eliminated by the direct adjustments on features outside ς and instead, the adaptive adjustments on the re-weighting curve about the input features can be more effective.* (3) CG7 which employs two ways of Δ(s) to shift features from both inside and outside of ς fails to improve AdaShift-B. *This validates that an Act cannot cumulate the contributions led by the same non-local cues.* (4) CG3, CG4, and CG5 perform significantly inferior to CG1, CG2, and AdaShift-B. This validates our intuitions clarified in (the last paragraph of) Sec. 3.2, *which we discuss in detail in Supp.*

Hypothesis: balanced summation of $\tilde{x}$ and Δ. We suppose the balanced summation of $\tilde{x}$ and Δ is critical to ensure the effectiveness of AdaShifts (as discussed in Sec. 3.3). To investigate this hypothesis, we compare the original AdaShift-B with three modified AdaShift-B(s) which serve as the targeted control groups: (1) Ada-CG1 which degrades the Δ from $\mathrm{LN}(\tilde{\tilde{x}})$ to $\gamma\tilde{\tilde{x}} + \beta$ by removing the Z-Scoring of LN; (2) Ada-CG2 which replaces the LN in Δ by a linear layer; (3) Ada-CG3, unlike Ada-CG2, which instead applies a linear projection before LN such that the balanced summation is preserved. Tab. 5 reports the comparative results on CIFAR-100 using CF-ResNet-56 backbone, where Ada-CG1 and -CG2 that violate the balanced summation both demonstrate inferior accuracies to the original AdaShift-B. Particularly, although CG2 leverages a linear layer with considerable extra parameters to impose compensated flexibility to CG1, it still fails to improve AdaShift-B due to the imbalanced summation. In contrast, CG3 which saves the balanced summation paradigm achieves meaningful accuracy gains to AdaShift-B. This

Table 3. Comparison of different Act functions on CIFAR-100. We train each model 8 times and report the mean $\pm$ std of the Top-1.

Activation	#Params.	ReLU[34]	PReLU[18]	ELU[12]	SiLU[16]	GELU[20]	Mish[32]	Swish[36]	AN-C[30]	Mt-AN[30]	Pserf[4]	SMU-1[5]	SMU[5]	**AdaS-B**	**AdaS-MA**
CF-RN-29	0.3M	70.5±0.3	70.1±0.5	72.6±0.2	72.0±0.4	71.4±0.3	72.1±0.3	71.5±0.3	70.9±0.2	72.2±0.3	71.6±0.2	70.7±0.3	71.1±0.4	**73.7**±0.4	**74.3**±0.3
CF-RN-56	0.6M	74.4±0.3	73.2±0.4	74.7±0.3	75.3±0.4	75.3±0.3	75.2±0.3	74.8±0.2	74.1±0.3	75.7±0.2	75.3±0.2	74.7±0.2	74.9±0.3	**76.5**±0.3	**77.0**±0.4

Table 4. Ablation study on different prospective prototypes that apply learnable adjustments and leverage tensor non-local cues.

Activation	Prototype	#Params.	Top-1(%)↑
ReLU	—	0.6M	74.4±0.3
Proto-CG1	$\phi(\tilde{x}) = \varsigma(\tilde{x})\,\tilde{x}$	0.6M	75.3±0.4
Proto-CG2	$\phi(\tilde{x}) = \varsigma(\kappa\tilde{x})\,\tilde{x}$	0.6M	74.8±0.2
Proto-CG3	$\phi(\tilde{x}) = \varsigma(\Delta\tilde{x})\,\tilde{x}$	0.6M	***73.4±0.3***
Proto-CG4	$\phi(\tilde{x}) = \varsigma(\kappa\tilde{x} + \Delta)\,\tilde{x}$	0.6M	***73.7±0.3***
Proto-CG5	$\phi(\tilde{x}) = \varsigma(\Delta_1\tilde{x} + \Delta_2)\,\tilde{x}$	0.6M	***73.6±0.2***
Proto-CG6	$\phi(\tilde{x}) = \varsigma(\tilde{x} + \Delta)\,(\tilde{x} + \Delta)$	0.6M	75.9±0.3
Proto-CG7	$\phi(\tilde{x}) = \varsigma(\tilde{x} + \Delta_1)\,(\tilde{x} + \Delta_2)$	0.6M	76.2±0.4
AdaShift-B	$\phi(\tilde{x}) = \varsigma(\tilde{x} + \Delta)\,\tilde{x}$	0.6M	**76.5±0.3**

Table 5. Ablation study on the hypothesis of imbalanced summation of $\tilde{x}$ and Δ, where we report the mean $\pm$ std of the Top-1.

Activation	Backbone	#Params.	FLOPs	Top-1(%)↑
ReLU [34]	CF-ResNet-56 [19]	0.6M	90.7M	74.4±0.3
Ada-CG1		0.6M	91.8M	76.0±0.4
Ada-CG2		1.2M	92.4M	76.3±0.2
Ada-CG3	CF-ResNet-56 [19]	1.2M	92.4M	**77.1±0.3**
AdaShift-B		0.6M	91.8M	**76.5±0.3**

Table 6. Ablation study on the meaning of non-local cues for Δ. We report the mean $\pm$ std of the Top-1 on CIFAR100.

Activation	Backbone	#Params.	FLOPs	Top-1(%)↑
ReLU [34]	CF-ResNet-56 [19]	0.6M	90.7M	74.4±0.3
Δ-CG1		0.6M	90.7M	75.3±0.4
Δ-CG2		0.6M	90.7M	75.1±0.4
AdaShift-B	CF-ResNet-56 [19]	0.6M	91.8M	**76.5±0.3**

validates our hypothesis.

Feature translation w/ or wo/ non-local cues. We suppose the tensor-level non-local cues incorporated by Δ are the critical complementary information to perform adaptive feature translations. We experimentally investigate this hypothesis by comparing AdaShift-B with two tailored control groups, *i.e.*, modified AdaShift-B(s) (1) removing Δ from the re-weighting process, hence regressing to SiLU [16] (CG1); (1) leveraging a plain Δ that shifts input features by the trainable channel-wise biases (CG2). Tab. 6 report the results, where we have two major observations: (1) AdaShift-B enjoys significant improvements to both CG1 and CG2; (2) CG1 and CG2 demonstrate close accuracies. These validate our hypothesis.

Extending the practical AdaShifts with simple modifications. We further validate the extensibility of the proposed AdaShift prototype by introducing 3 new practical AdaShift derivatives, namely, AdaShift-MA-N1, -N2, and -N3, modified from AdaShift-MA with simple ideas (the diagrams are depicted in Supp). Compared to AdaShift-MA, (1) AdaShift-MA-N1 jointly attends to the main and the residual features through a united attention process. That is, for a layer that converges the main and the residual features, AdaShift-MA-N1 produces two patches of local channel statistics of the main and the residual features, respectively, and concatenates these two patches along the spatial axis to generate the extended keys and values. The queries are the simple aggregation of the two patches to constrain the complexity. This modification adds zero parameters to AdaShift-MA. (2) AdaShift-MA-N2 uses a pre-linear-projection before the post-LN to incorporate further fitting flexibility. To avoid excessive parameters, this change is only applied to where the inputs are un-expanded features. (3) AdaShift-MA-N3 jointly applies the modifications (1) and (2), simultaneously. Tab. 7 reports the comparative results of different AdaShift derivatives on ImageNet, where AdaShift-MA-N1, -N2, and -N3 all achieve practical improvements on AdaShift-B and -MA. In particular, AdaShift-MA-N3 which combines the modifications (1) and (2) demonstrates superior accuracy to other derivatives. These verify the extensibility of our AdaShift prototype.

Table 7. Comparison of the ReLU baseline and different practical AdaShift derivatives on ImageNet using ResNet-50 [19] backbone.

Activation	Backbone	#Params.	FLOPs	Top-1(%)↑
ReLU [34]	ResNet-50 [19]	25.6M	4.1G	77.2
AdaS-B		25.6M	4.1G	79.9
AdaS-MA		25.7M	4.2G	80.3
AdaS-MA-N1	ResNet-50 [19]	25.8M	4.3G	**80.4**
AdaS-MA-N2		28.3M	4.4G	**80.5**
AdaS-MA-N3		28.3M	4.4G	**80.6**

5. Conclusion

We propose to learn discriminative self-gated neural feature Act with a novel AdaShift prototype inspired by the new intuitions of feature-filter context in neural learning. AdaShift adaptively translates the Act inputs by comprehensively exploiting informative local/non-local cues of different ranges, therefore performing fine-grained adjustments to the feature re-weighting in a particularly simple yet effective manner. Built on the new prototype, our practical AdaShifts significantly improve popular/SOTA Act functions on various vision benchmarks with only negligible computational cost and parameters added to ReLU baseline.

References

[1] Forest Agostinelli, Matthew Hoffman, Peter Sadowski, and Pierre Baldi. Learning activation functions to improve deep neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 6

[2] Pranjal Awasthi, Alex Tang, and Aravindan Vijayaraghavan. Efficient algorithms for learning depth-2 neural networks with general relu activations. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2, 3

[4] Koushik Biswas, Sandeep Kumar, Shilpak Banerjee, and Ashish Kumar Pandey. Erfact and pserf: Non-monotonic smooth trainable activation functions. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 1, 2, 3, 6, 7, 8

[5] Koushik Biswas, Sandeep Kumar, Shilpak Banerjee, and Ashish Kumar Pandey. Smooth maximum unit: Smooth activation function for deep networks using smoothing maximum technique. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 5, 6, 7, 8

[6] Sudong Cai. Iieu: Rethinking neural feature activation from decision-making. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 5796–5806, 2023. 2, 3, 4, 6, 7

[7] Sudong Cai, Yulan Guo, Salman Khan, Jiwei Hu, and Gongjian Wen. Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[8] Sudong Cai, Ryosuke Wakaki, Shohei Nobuhara, and Ko Nishino. Rgb road scene material segmentation. In *Proc. Asian Conference on Computer Vision (ACCV)*, 2022. 6

[9] Shyi-Ming Chen, Shou-Hsiung Cheng, and Tzu-Chun Lan. Multicriteria decision making based on the topsis method and similarity measures between intuitionistic fuzzy values. *Information Sciences*, 367:279–295, 2016. 2

[10] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic relu. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 6, 7

[11] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[12] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *Proc. International Conference on Learning Representations (ICLR)*, 2016. 3, 6, 7, 8

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1, 6

[14] Yucheng Dong, Yating Liu, Haiming Liang, Francisco Chiclana, and Enrique Herrera-Viedma. Strategic weight manipulation in multiple attribute decision making. *Omega-International Journal of Management Science*, 75:154–164, 2018. 2

[15] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2000. 1, 6, 7

[16] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107: 3–11, 2018. 2, 3, 4, 5, 6, 7, 8

[17] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proc. International Conference on Machine Learning (ICML)*, 2013. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 3, 6, 7, 8

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 7, 8

[20] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 2, 3, 4, 6, 7, 8

[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42 (8):2011–2023, 2020. 3, 7

[22] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. International Conference on Machine Learning (ICML)*, pages 448–456, 2015. 3

[23] Deepa Joshi and Sanjay Kumar. Interval-valued intuitionistic hesitant fuzzy choquet integral based topsis method for multi-criteria group decision making. *European Journal of Operational Research*, 248(1):183–191, 2016. 2, 3

[24] Minjoon Kouh. *Toward a more biologically plausible model of object recognition*. PhD thesis, MIT, 2007. 1

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009. 6

[26] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective Kernel Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–519, 2019. 6

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 6

[28] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proc. European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 6

[29] Ningning Ma, Xiangyu Zhang, and Jian Sun. Funnel activation for visual recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 351–368. Springer, 2020. 6, 7

[30] Ningning Ma, Xiangyu Zhang, Ming Liu, and Jian Sun. Activate or not: Learning customized activation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8032–8042, 2021. 1, 2, 3, 5, 6, 7, 8

[31] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML Workshop*, 2013. 3, 6, 7

[32] Diganta Misra. Mish: A self regularized non-monotonic neural activation function. In *Proc. British Machine Vision Conference (BMVC)*, 2020. 2, 3, 4, 6, 7, 8

[33] Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Padé Activation Units: End-to-end Learning of Flexible Activation Functions in Deep Networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2020. 1, 6

[34] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. International Conference on Machine Learning (ICML)*, 2010. 1, 6, 7, 8

[35] Jindong Qin, Xinwang Liu, and Witold Pedrycz. An extended todim multi-criteria group decision making method for green supplier selection in interval type-2 fuzzy environment. *European Journal of Operational Research*, 258(2): 626–638, 2017. 2, 3

[36] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. In *Proc. Workshop Track of the 6th International Conference on Learning Representations (ICLR)*, 2018. 3, 6, 7, 8

[37] Jafar Rezaei. Best-worst multi-criteria decision-making method: Some properties and a linear model. *Omega-International Journal of Management Science*, 64:126–130, 2016. 2, 3

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 6

[39] Gabriel Schwartz and Ko Nishino. Recognizing Material Properties from Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(8):1981–1995, 2020. 1

[40] Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *PNAS*, 104(15):6424–6429, 2007. 1

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training Data-Efficient Image Transformers & Distillation Through Attention. In *Proc. International Conference on Machine Learning (ICML)*, 2021. 6

[42] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[43] Weiaicunzai. pytorch-cifar100. https://github.com/weiaicunzai/pytorch-cifar100. 6

[44] Ross Wightman, Hugo Touvron, and Herve Jegou. Resnet strikes back: An improved training procedure in timm. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*, 2021. 6

[45] Lei Wu. Learning a Single Neuron for Non-monotonic Activation Functions. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2022. 2

[46] Ye Xu, Ye Li, Lijun Zheng, Liang Cui, Sha Li, Wei Li, and Yanpeng Cai. Site selection of wind farms using gis and multi-criteria decision making method in wafangdian, china. *Energy*, 207:118222, 2020. 2, 3

[47] Hong-Bin Yan, Tieju Ma, and Van-Nam Huynh. On qualitative multi-attribute group decision making and its consensus measure: A probability based perspective. *Omega-International Journal of Management Science*, 70:94–117, 2017. 2

[48] Jingkai Zhou, Varun Jampani, Zhixiong Pi, Qiong Liu, and Ming-Hsuan Yang. Decoupled Dynamic Filter Networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[49] Yucong Zhou, Zezhou Zhu, and Zhao Zhong. Learning specialized activation functions with the piecewise linear unit. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 12095–12104, 2021. 7