Responses to Reviewer iGq4

Comment 1: The intuition why the proposed method is better is not provided. There are many ways to generalize the existing activation functions, but why would TMAF work?

Response 1: We generalize ReLU to TMAF using matrix-vector multiplication, which is mathematically natural and simple. This generalization is different from classical generalization strategy in the literature. One advantage is that TMAF includes discontinuous activation functions and continuous as special cases. Please have a look at the Remark on page 3.

Comment 2: Lack of thorough compassion to support the effectiveness of TMAF. There have been a lot of widely used activation functions, as mentioned in the introduction section: PLU, Leaky ReLU, PLU, Softplus, ELU, SELU, GELU etc. However, the authors did not compare TMAF with most of them, making the effectiveness of TMAF questionable.

Response 2: In the revised version, we compared TMAF also with Parametric ReLU in the image classification problems. Parametric ReLU includes Leaky ReLU as a special case. TMAF outperforms Parametric ReLU on all examples that we have tested.

Comment 3: The task of sin/cos function approximation is quite simple compared to fitting the highly non-convex neural networks. So section 3.1 and 3.2 might not be very informative.

Response 3: We agree that section 3.1 is relatively simple. However, approximating a highly oscillating function, as the one in section 3.2, is a difficult task in general. As it is easily seen from the experiments in the manuscript, the standard network structures with gradient-descent-like training process do not work at all as approximants to such functions. In contrast, TMAF is able to produce good approximations.

Comment 4: The experiments on MNIST/CIFAR is based on very lousy baselines — The 2-hidden-layer network or ResNet18 is by no means the state-of-the-art methods, not even close. So using them in the experiments is not convincing. It would be better to have TMAF plugged in the latest networks, such as EfficientNet and Vision Transformers to test its efficacy.

Response 4: Our aim is to prove a concept, namely, that TMAF outperforms the NN approximations using standard approaches. Although Resnet18 is not the state-of-the-art method, our numerical experiments provide evidence that TMAF is robust. As the referee rightfully mentions, employing TMAF in more efficient (the latest NN, using deeper layer structure) is desirable. It is, however, also a topic of our current and future research.

Comment 5: The CIFAR-100 result is missing in Table 3, making the paper look incomplete and probably written in a hurry.

Response 5: We added a Cifar-100 result in the revised version.

Responses to Reviewer MSbu

Comment 1: My main concern with this paper is that there is already very similar work on this area of research, namely [1]. In [1], we find a learnable piecewise function which leads to the activation function PWLU. If we were to consider a different PWLU for every neuron in the network, we would have basically the same approach as the one introduced here.

Response 1: We were not aware of the recent work on PWLU activation functions when writing the paper. In the revised version, we included remarks on the difference between PWLU and TMAF in the Remark on pages 2 and 3 and explanations under Equation (6). In particular, TMAF includes discontinuous activation functions as special cases while PWLU only allows continuous piecewise polynomials.

Comment 2: On the experimental section, there is no comparison to other learnable activation functions such as PAU, F-RELU or APL, PWLU, making it difficult for the reader to stack TMAF against the state of the art. Moreover, the number of architectures and datasets evaluated is relatively small in comparison to other papers in the same area of research. Finally, it seems like the paper only does one run for each of the experiments which might be problematic to estimate the actual performance of the activation function.

Response 2: We tested the approximation properties of the TMAF against a representative set of other learnable activation functions (parametric ReLU) and as our results convinsingly show that TMAF provides a competitive and robust approximation when compared to these NN approximations. A comprehensive comparisons with all of the NN constructions available would require substantial time and while it is one of the future research goals of our team we consider it falling beyond the scope of this paper.

Comment 3: There are other minor improvements that could be done to the paper on the presentation front. It would be helpful for the reader to see the plots of the synthetic experiments overlapped with the ones produced by the networks. Although that space might be better used for a more comprehensive empirical evaluation.

Response 3: The NNs (TMAF or other activation functions) fall in the category of nonlinear/adaptive approximation methods. The mathematical theory shows that the nonlinear approximation results in errors which are always better than the ones provided by synthetic (linear) approximations. As this is well known fact, and because of the restricted space, we did not consider such comparison.

Responses to Reviewer mR17

Comment 1: application of activation σ returns a vector, while D returns a matrix. How can this be?

Response 1: D returns a matrix. But in TMAF D is multipplied with a column vector to obtain another column vector. Then, obviously, this operation results in an activation returning a vector. The multiplication by D and a vector is not a dense vector matrix multiplication. If $D = \text{diag}(z)$, then $Dy = z \odot y$ where $\odot$ is the well known Schur product.

Comment 2: η_ℓ is defined after it is used in the beginning of section 2.

Response 2: In the revised version, we define $\eta_L = \eta_{L,\mathcal{W},\mathcal{B}}$ in Equation (2) to avoid such confusions.

Comment 3: is m a hyperparameter? How is it selected?

Response 3: According to the defition of α_ℓ, m is the number of intervals used in α_ℓ. For ReLU we have $m = 2$. The intervals are clearly given in each experiment and m can vary in general. To keep the computational cost low, in our numerical experiments, we fix m to be a small integer.

Comment 4: I also don't understand numerical results:...

Response 4: We have modified the notation in the revised version to clarify the results better.

Responses to Reviewer MkDq

Comment 1: the paper does not provide a review of related works and fails to mention connections to existing trainable (or untrainable) AFs. The form s(x)x is also the base of the Swish AF (where s(x) is the sigmoid) or its variants. Many AFs are built by piecewise linear or constant parts (s-shaped ReLU, adaptive piecewise linear unit, ...) (Apicella et al., 2021). The idea here is to combine Swish with a piecewise constant s(x), but the way it is described in the paper is convoluted (where the original AF is replaced with D(s(x))x, where D is a diagonal matrix).

Response 1: In the revised version, we have added more references in the introduction as well as in Remark on page 3.

Comment 2: COMPUTATIONAL COST: the paper says "it is observed that the computational time of D and T is comparable to the classical ReLU", but evaluting (6) or even worse (7) is clearly more expensive than a simple max(0, x). Concrete results on the overhead should be given to justify this sentence.

Response 2: We remove this sentence in the revised version. But for a finite number m of intervals in the definition of TMAF (for example $m = 2$ for ReLU), it is clear that the computational cost can be controlled and similar to that of ReLU or Parametric ReLU.

Comment 3: EXPERIMENTS: more realistic experiments should be provided, by comparing a larger set of trainable/fixed baselines (Swish, SELU, ...), letting the networks train to convergence (see, e.g., Fig. 7), provide standard deviation of the results, etc.

Response 3: In all experiments of the revised version, we compare TMAF with ReLU and/or the trainable Parametric ReLU. These activation functions can provide general NN approximations with typical behavior close to the one exhibitted by the AF mentioned by the reviewer.

Responses to Reviewer FB5F

Comment 1: First, the introduction does not cover the literature on adaptive activation functions and lack several references...

Response 1: In the revised version, we added more references in the introduction.

Comment 2: I do not understand why we would need a matrix-vector multiplication (especially in the case where the matrix is diagonal), but also for non-diagonal matrix.

Response 2: We generalize ReLU to TMAF using matrix-vector multiplication, which is mathematically natural and simple. This generalization is different from classical generalization strategy such as piecewise constant or continuous continuous piecewise linear units in the literature. Please have a look at the Remark on page 3.

Comment 3: The authors mention that the computational time is comparable to ReLU but do not report any timings to support this claim. Moreover, for the small network size that the authors employ in section 3, the increase of the network parameters will likely impact the training time.

Response 3: ReLU corresponds to a fixed TMAF with $m = 2$ (two) intervals for each layer in the NN. In TMAF we choose m to be a fixed integer ($m = 5$ is typical) and this asymptotically has the same computational cost as usual ReLU, namely,

$$\text{computational-cost(TMAF)} \lesssim \frac{5}{2}(\text{computational-cost(ReLU)})$$

Comment 4: The authors comment this by saying that "increasing the number of intervals in TMAF

only lead to a few more training parameters", I disagree with this since the increase would grow as number of layers x number of neurons per layer so is very significant for the network size used.

Response 4: In the revised version, we modify the notation. In fact, TMAF uses the same activation function for each neuron in the same layer. So the increase of cost would be only proportional to the number of layers and is independent of number of neurons.

Comment 5: Similarly, the experiments on CIFAR-10 and CIFAR-100 are not really conclusive as the authors employ a residual network instead of standard convolution networks. Moreover, they don't do any comparison with other (adaptive) activation functions. I also do not think that the reported accuracy is closed to the state-of-the-art. I encourage the authors to read the section 4.3 of (arXiv:1907.06732), which performs comparisons between several activation functions with different architectures on the CIFAR-10 dataset, The reported performances are all higher than 90%, which is way higher than the ones reported by the authors in Table 3.

Response 5: We do appreciate this comment. State-of-the-art activation functions appear almost daily (weekly, monthly, quarterly). It is impossible to compare with all such new constructions. We have clearly shown that TMAF outperforms a representative set of NN in approximation. In our tests as representative set we use ReLU and Parametric ReLU (which includes Leaky ReLU). Overall the NN with TMAF has efficiency comparable to a ReLU NN. The accuracy of the TMAF-NN in approximating is better and is robust with respect to the oscillations in the data. It is reasonable to conclude that such a pool of numerical tests is sufficient to prove a concept and we expect to include in our future research a comparison with all competitive activation functions that are available in the literature.