Quantile Activation: departing from single point estimation for better generalization across distortions

Anonymous Author(s)
Affiliation
Address
email

Abstract

A classifier is, in its essence, a function which takes an input and returns the class of the input and implicitly assumes an underlying distribution. We argue in this article that one has to move away from this basic tenet to obtain generalization across distributions. Specifically, the class of the sample should depend on the points from its *"context distribution"* for better generalization across distributions.

How does one achieve this? – The key idea is to "adapt" the outputs of each neuron of the network to its context distribution. We propose quantile activation,QACT, which, in simple terms, outputs the *relative quantile* of the sample in its context distribution, instead of the actual values in traditional networks.

The scope of this article is to validate the proposed activation across several experimental settings, and compare it with conventional techniques. For this, we use the datasets developed to test robustness against distortions – CIFAR10C, CIFAR100C, MNISTC, TinyImagenetC, and show that we achieve a significantly higher generalization across distortions than the conventional classifiers, across different architectures. Although this paper is only a proof of concept, we surprisingly find that this approach outperforms DINOv2(small) at large distortions, even though DINOv2 is trained with a far bigger network on a considerably larger dataset.

1 Introduction

Deep learning approaches have significantly influenced image classification tasks on the machine learning pipelines over the past decade. They can easily beat human performance on such tasks by non-trivial margins by using innovative ideas such as Batch Normalization [19] and other normalization techniques [3, 31], novel rectifiers such as ReLU/PReLU [26, 31, 3] and by using large datasets and large models.

However, these classification systems do not generalize across distributions [2, 34], which leads to instability when used in practice. [22] shows that deep networks with ReLU activation degrades in performance under distortions. [4] observes that there exists a feature collapse which inhibits the networks to be reliable.

Fundamental Problem of Classification: We trace the source of the problem to the fact that – *Existing classification pipelines necessitates single point prediction*, i.e., they should allow classification of a single sample given in isolation. We argue that, for good generalization, one should move away from this basic tenet and instead allow the class to be dependent on the *"context distribution"* of the sample. That is, when performing classification, one needs to know both the sample and the context of the sample for classification.

While this is novel in the context of classification systems, it is widely prevalent in the specific domain of Natural Language Processing (NLP) – The meaning of a word is dependent on the context of the

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

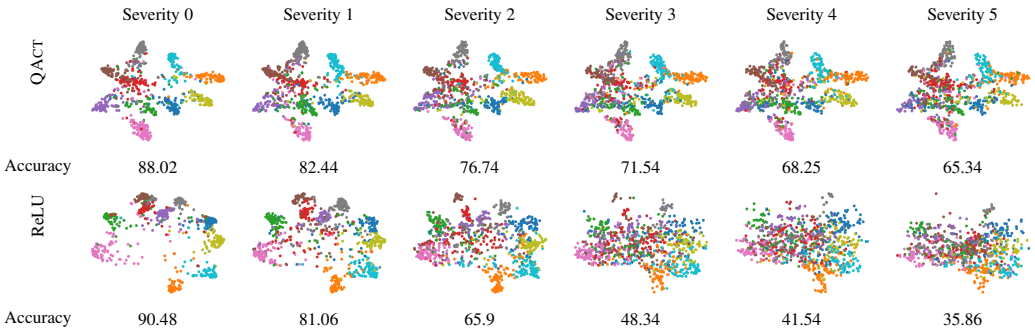


Figure 1: Comparing TSNE plots of QACT and ReLU activation on CIFAR10C with Gaussian distortions. Observe that QACT maintains the class structure extremely well across distortions, while the usual ReLU activations loses the class structure as severity increases.

word. However, to our knowledge, this has not been considered for general classification systems. Even when using dominant NLP architectures such as Transformers for vision [8], the technique has been to split the image into patches and then obtain the embedding for an individual image.

Obtaining a classification framework for incorporating context distribution: We suspect that the main reason why the context distribution is not incorporated into the classification system is – The naive approach of considering a lot of samples in the pipeline to classify a single sample is computationally expensive. We solve this problem by considering the context distribution of each neuron specifically. We introduce *Quantile Activation* (QACT), which outputs a probability depending upon the context of *all* outputs. This, however, gives rise to new challenges in training, which we address in section 2.

Figure 1 illustrates the differences of the proposed framework with the existing framework. As severity increases (w.r.t Gaussian Noise), we observe that ReLU activation loses the class structure. This behaviour can be attributed to the fact that, as the input distribution changes, the activations either increase/decrease, and due to the multiplicative effect of numerous layers, this leads to very different features. On the other hand, the proposed QACT framework does not suffer from this, since if all the pre-activations[1] change in a systematic way, the quantiles *adjust* automatically to ensure that the inputs for the next layer does not change by much. This is reflected in the fact that class structure is preserved with QACT.

Remark: Quantile activation is different from existing quantile neural network based approaches, such as regression [30], binary quantile classification [36], Anomaly Detection [24, 33]. Our approach is achieving best in-class performance by incorporating context distribution in the classification paradigm. Our approach is also markedly different from Machine unlearning which is based on selective forgetting of certain data points or retraining from scratch [32].

Contributions: A decent amount of literature on neuronal activation is available. However, to the best of our knowledge, none matches the central idea proposed in this work.

In [5], the authors propose an approach to calibrate a pre-trained classifier $f_\theta(\boldsymbol{x})$ by extending it to learn a *quantile function*, $Q(\boldsymbol{x}, \theta, \tau)$ (τ denotes the quantile), and then estimate the probabilities using $\int_\tau I[Q(\boldsymbol{x}, \theta, \tau) \geq 0.5] d\tau$[2]. They show that this results in probabilities which are robust to distortions.

1. In this article, we extend this approach to the level of a neuron, by suitably deriving the forward and backward propagation equations required for learning (section 2).
2. We then show that a suitable incorporation of our extension produces context dependent outputs at the level of each neuron of the neural network.

[1]We use the following convention – "Pre-activations" denote the inputs to the activation functions and "Activations" denote the outputs of the activation function.

[2]$I[]$ denotes the indicator function

3. Our approach contributes to achieving better generalization across distributions and is more robust to distortions, across architectures. We evaluate our method using different architectures and datasets, and compare with the current state-of-the-art – DINOv2. We show that QACT proposed here is more robust to distortions than DINOv2, even if we have considerably less number of parameters (22M for DINOv2 vs 11M for Resnet18). Additionally, DINOv2 is trained on 20 odd datasets, before being applied on CIFAR10C; in contrast, our framework is trained on CIFAR10, and produces more robust outcome (see figures 3,5).
4. The proposed QACT is consistent with all the existing techniques used in DINOv2, and hence can be easily incorporated into any ML framework.
5. We also adapt QACT to design a classifier which returns better calibrated probabilities.

Related Works on Domain Generalization (DG): The problem of domain generalizations tries to answer the question – Can we use a classifier trained on one domain across several other related domains? The earliest known approach for this is *Transfer Learning* [28, 37], where a classifier from a single domain is applied to a different domain with/without fine-tuning. Several approaches have been proposed to achieve DG, such as extracting domain-invariant features over single/multiple source domains [11, 1, 9, 29, 16], Meta Learning [17, 9], Invariant Risk Minimization [2]. Self supervised learning is another proposed approach which tries to extract features on large scale datasets in an unsupervised manner, the most recent among them being DINOv2 [27]. Very large foundation models, such as GPT-4V, are also known to perform better with respect to distribution shifts [12]. Nevertheless, to the best of our knowledge, none of these models incorporates context distributions for classification.

2 Quantile Activation

Rethinking Outputs from a Neuron: To recall – if $\mathbf{x}$ denotes the input, a typical neuron does the following – (i) Applies a linear transformation with parameters w, b, giving $w^t\mathbf{x} + b$ as the output, and (ii) applies a rectifier g, returning $g(w^t\mathbf{x} + b)$. Typically, g is taken to be the ReLU activation - $g_{relu}(x) = \max(0, x)$. Intuitively, we expect that each neuron captures an "abstract" feature, usually not understood by a human observer.

An alternate way to model a neuron is to consider it as predicting a latent variable $\mathbf{y}$, where $\mathbf{y} = 1$ if the feature is present and $\mathbf{y} = 0$ if the feature is absent. Mathematically, we have the following model:

$$\mathbf{z} = w^t\mathbf{x} + b + \epsilon \quad and \quad \mathbf{y} = I[\mathbf{z} \geq 0] \tag{1}$$

This is very similar to the standard latent variable model for logistic regression, with the main exception being, the *outputs* $\mathbf{y}$ *are not known* for each neuron beforehand. If $\mathbf{y}$ is known, it is rather easy to obtain the probabilities – $P(\mathbf{z} \geq 0)$. Can we still predict the probabilities, even when $\mathbf{y}$ itself is a latent variable?

The authors in [5] propose the following algorithm to estimate the probabilities:

1. Let $\{\boldsymbol{x}_i\}$ denote the set of input samples from the input distribution $\mathbf{x}$ and $\{z_i\}$ denote their corresponding latent outputs, which would be from the distribution $\mathbf{z}$
2. Assign $\mathbf{y} = 1$ whenever $\mathbf{z} > (1 - \tau)^{th}$ quantile of $\mathbf{z}$, and 0 otherwise. For a specific sample, we have $y_i = 1$ if $z_i > (1 - \tau)^{th}$ quantile of $\{z_i\}$
3. Fit the model $Q(x, \tau; \theta)$ to the dataset $\{((\boldsymbol{x}_i, \tau), y_i)\}$, and estimate the probability as,

$$P(y_i = 1) = \int_{\tau=0}^{1} I[Q(x, \tau; \theta) \geq 0.5]d\tau \tag{2}$$

The key idea: Observe that in step 2., the labelling is done without resorting to actual ground-truth labels. This allows us to obtain the probabilities on the fly for any set of parameters, only by considering the quantiles of $\mathbf{z}$.

Defining the Quantile Activation QACT Let $\mathbf{z}$ denote the pre-activation of the neuron, and let $\{\boldsymbol{z}_i\}$ denote the samples from this distribution. Let $F_{\mathbf{z}}$ denote the cumulative distribution function (CDF), and let $f_{\mathbf{z}}$ denote the density of the distribution. Accordingly, we have that $F_{\mathbf{z}}^{-1}(\tau)$ denotes

Algorithm 1 Forward Propagation for a single neuron

Input: $[z_i]$ a vector of pre-activations, $0 < \tau_1 < \tau_2 < \cdots < \tau_{n_\tau} < 1$ - a list of quantile indices at which we compute the quantiles.

Append two large values, c and $-c$, to the vector $[z_i]$.
Count n_+ = number of positive values, n_- = number of negative values, and assign the weight $w_+ = 1/n_+$ to the positive values, and $w_- = 1/n_-$ to the negative values.
Compute *weighted* quantiles $\{q_i\}$ at each of $\{\tau_i\}$ over the set $\{z_i\} \cup \{c, -c\}$
Compute $\text{QACT}(z_i)$ using the function,

$$\text{QACT}(x) = \frac{1}{n_\tau} \sum_i I[x \geq q_i] \tag{5}$$

Remember $[z_i]$, w_+, w_-, $[\text{QACT}(z_i)]$ for backward propagation.
return $[\text{QACT}(z_i)]$

Algorithm 2 Backward Propagation for a single neuron

Input: `grad_output`, $0 < \tau_1 < \tau_2 < \cdots < \tau_{n_\tau} < 1$ - a list of quantile indices at which we compute the quantiles.
Context from Forward Propagation: $[z_i]$, w_+, w_-, $[\text{QACT}(z_i)]$

Obtain a weighted sample from $[z_i]$ with weights w_+, w_- – (say) S.
Obtain a kernel density estimate, using points from S, at each of the points in z_i – (say) $\hat{f}_{\mathbf{z}}(z_i)$
Set,

$$\texttt{grad\_input} = \texttt{grad\_output} \odot [\hat{f}_{\mathbf{z}}(z_i)] \tag{6}$$

return `grad_input`

the τ^{th} quantile of $\mathbf{z}$. Using step (2) of the algorithm above, we define,

$$\text{QACT}(\boldsymbol{z}) = \int_{\tau=0}^{1} I[\boldsymbol{z} > F_{\mathbf{z}}^{-1}(1-\tau)]d\tau \underset{\tau \to (1-\tau)}{\overset{\text{Substitute}}{=}} \int_{\tau=0}^{1} I[\boldsymbol{z} > F_{\mathbf{z}}^{-1}(\tau)]d\tau \tag{3}$$

Computing the gradient of QACT: However, to use QACT in a neural network, we need to compute the gradient which is required for back-propagation. Let $\tau_{\boldsymbol{z}}$ denote the quantile at which $F_{\mathbf{z}}^{-1}(\tau_{\boldsymbol{z}}) = \boldsymbol{z}$. Then we have that $\text{QACT}(\boldsymbol{z}) = \tau_z$ since $F_{\mathbf{z}}^{-1}(\tau)$ is an increasing function. So, we have that $\text{QACT}(F_{\mathbf{z}}^{-1}(\tau)) = \tau$. In other words, we have that $\text{QACT}(\boldsymbol{z})$ is $F_{\mathbf{z}}(\boldsymbol{z})$, which is nothing but the CDF of $\mathbf{z}$. Hence, we have,

$$\frac{\partial \text{QACT}(\boldsymbol{z})}{\partial z} = f_{\mathbf{z}}(\boldsymbol{z}) \tag{4}$$

where $f_{\mathbf{z}}(\boldsymbol{z})$ denotes the density of the distribution.

Grounding the Neurons: With the above formulation, observe that since QACT is identical to CDF, it follows that, $\text{QACT}(\mathbf{z})$ is always a uniform distribution between 0 and 1, irrespective of the distribution $\mathbf{z}$. When training numerous neurons in a layer, this could cause all the neurons to learn the same behaviour. Specifically, if, half the time, a particular abstract feature is more prevalent than others, QACT (as presented above) would not be able to learn this feature. To correct this, we *enforce that positive values and negative values have equal weight*. Given the input distribution $\mathbf{z}$, We perform the following transformation before applying QACT. Let

$$\mathbf{z}^+ = \begin{cases} \mathbf{z} & \text{if } \mathbf{z} \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \mathbf{z}^- = \begin{cases} \mathbf{z} & \text{if } \mathbf{z} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

denote the truncated distributions. Then,

$$\mathbf{z}^\ddagger = \begin{cases} \mathbf{z}^+ & \text{with probability } 0.5 \\ \mathbf{z}^- & \text{with probability } 0.5 \end{cases} \tag{8}$$

From definition of $\mathbf{z}^\ddagger$, we get that the median of $\mathbf{z}^\ddagger$ is 0. This grounds the input distribution to have the same positive and negative weight.

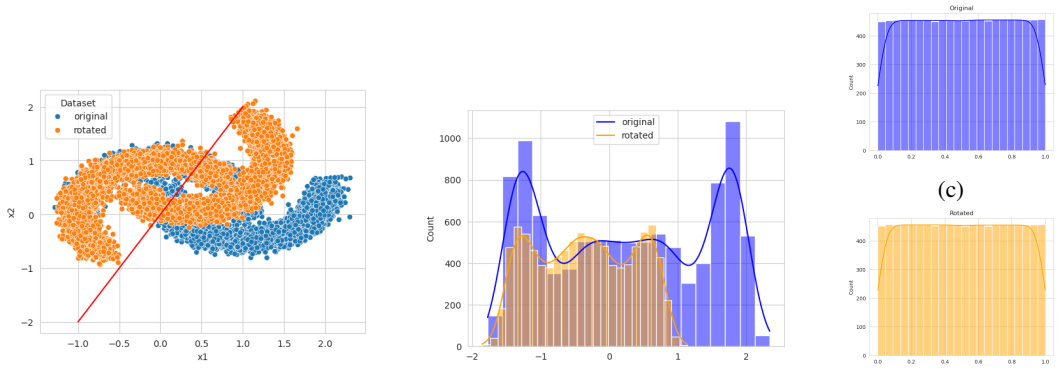


Figure 2: Intuition behind quantile activation. (a) shows a simple toy distribution of points (blue), it's distortion (orange) and a simple line (red) on which the samples are projected to obtain activations. (b) shows the distribution of the pre-activations. (c) shows the distributions of the activations with QACT of the original distribution (blue). (d) shows the distributions of the activations with QACT under the distorted distribution (orange). Observe that the distributions match perfectly under small distortions. Note that even if the distribution matches perfectly, the quantile activation is actually a deterministic function.

Dealing with corner cases: It is possible that during training, some neurons either only get positive values or only get negative values. However, for smooth outputs, one should still only give the weight of 0.5 for positive values. To handle this, we include two values c (large positive) and $-c$ (large negative) for each neuron. Since, the quantiles are conventionally computed using linear interpolation, this allows the outputs to vary smoothly. We take $c = 100$ in this article.

Estimating the Density for Back-Propagation: Note that the gradient for the back propagation is given by the density of $\mathbf{z}^{\ddagger}$ (weighted distribution). We use the *Kernel Density Estimation* (KDE), to estimate the density. We, (i) First sample N points with weights w_+, w_-, and (ii) then estimate the density at all the input points $[\boldsymbol{z}_i]$. This is point-wise multiplied with the backward gradient to get the gradient for the input. In this article we use $N = 1000$, which we observe gets reasonable estimates.

Computational Complexity: Computational Complexity (for a single neuron) is majorly decided by 2 functions – (i) Computing the quantiles has the complexity for a vector $[\boldsymbol{z}_i]$ of size n can be performed in $\mathcal{O}(n \log(n))$. Since this is log-linear in n, it does not increase the complexity drastically compared to other operations in a deep neural network. (ii) Computational complexity of the KDE estimates is $\mathcal{O}(Sn_\tau)$ where S is the size of sample (weighted sample from $[\boldsymbol{z}_i]$) and n_τ is the number of quantiles, giving a total of $\mathcal{O}(n + Sn_\tau)$. In practice, we consider $S = 1000$ and $n_\tau = 100$ which works well, and hence does not increase with the batch size. This too scales linearly with batch size n, and hence does not drastically increase the complexity.

Remark: Algorithms 1, and 2 provide the pseudocode for the quantile activation. For stable training, in practice, we prepend and append the quantile activation with BatchNorm layers.

Why QACT is robust to distortions? To understand the idea behind quantile activation, consider a simple toy example in figure 2. For ease of visualization, assume that the input features (blue) are in 2 dimensions, and also assume that the line of the linear projection is given by the red line in figure 2a. Now, assume that the blue input features are rotated, leading to a different distribution (indicated here by orange). Since activations are essentially (unnormalized) signed distances from the line, we plot the histograms corresponding to the two distributions in figure 2b. As expected, these distributions are different. However, after performing the quantile activation in equation 3, we have that both are uniform distribution. This is illustrated in figures 2c and 2d. This behaviour has a normalizing effect across different distributions, and hence has better distribution generalization than other activations.

3 Training with QACT

In the previous section, we described the procedure to adapt a single neuron to its context distribution. In this section we discuss how this extends to the Dense/Convolution layers, the loss functions to train the network and the inference aspect.

Extending to standard layers: The extension of equation 3 to dense outputs is straightforward. A typical output of the dense layer would be of the shape (B, N_c) - B denotes the batch size, N_c denotes the width of the network. The principle is - *The context distribution of a neuron is all the values which are obtained using the same parameters*. In this case, each of the values across the 'B' dimension are considered to be samples from the context distribution.

For a convolution layer, the typical outputs are of the form - (B, N_c, H, W) - B denotes the size of the batch, N_c denotes the number of channels, H, W denotes the sizes of the images. In this case we should consider all values across the 1st,3rd and 4th dimension to be from the context distribution, since all these values are obtained using the same parameters. So, the number of samples would be $B \times H \times W$.

Loss Functions: One can use any differentiable loss function to train with quantile activation. We specifically experiment with the standard Cross-Entropy Loss, Triplet Loss, and the recently proposed Watershed Loss [6] (see section 4). However, if one requires that the boundaries between classes adapt to the distribution, then learning similarities instead of boundaries can be beneficial. Both Triplet Loss and Watershed Loss fall into this category. We see that learning similarities does have slight benefits when considering the embedding quality.

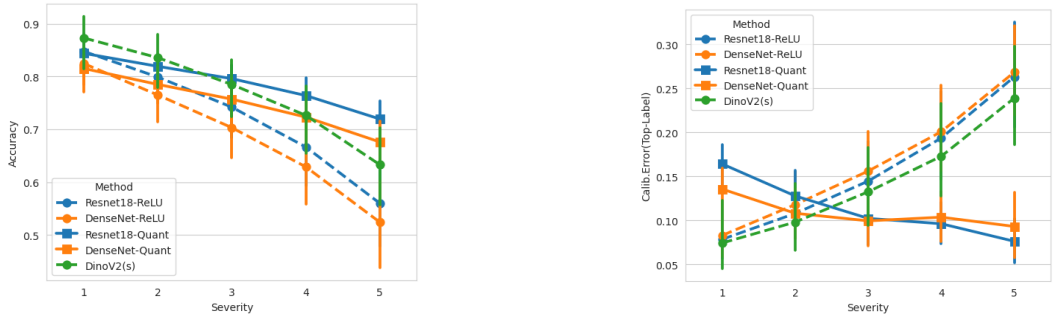
Inference with QACT: As stated before, we want to assign a label for classification based on the context of the sample. There exist two approaches for this – (1) One way is to keep track of the quantiles and the estimated densities for all neurons and use it for inference. This allows inference for a single sample in the traditional sense. However, this also implies that one would not be able to assign classes based on the context at evaluation. (2) Another way is to make sure that, even for inference on a single sample, we include several samples from the context distribution, but only use the output for a specific sample. This allows one to assign classes based on the context. In this article, we follow the latter approach.

Quantile Classifier: Observe that the proposed QACT (without normalization) returns the values in $[0, 1]$ which can be interpreted as probabilities. Hence, one can also use this for the classification layer. Nonetheless, two changes are required – (i) Traditional softmax used in conjunction with negative-log-likelihoood loss already considers "relative" activations of the classification in normalization. However, QACT does not. Hence, one should use Binary-Cross-Entropy loss with QACT, which amounts to one-vs-rest classification. (ii) Also, unlike a neuron in the middle layers, the bias of the neuron in the classification layer depends on the class imbalance. For instance, with 10 classes, one would have only $1/10$ of the samples labelled 1 and $9/10$ of the samples labelled 0. To address this, we require that the median of the outputs be at 0.9, and hence weight the positive class with 0.9 and the negative class with 0.1 respectively. In this article, whenever QACT is used, we use this approach for inference.

We observe that (figures 13 and 14) using quantile classifier on the learned features in general improves the consistency of the calibration error and also leads to the reducing the calibration error. In this article, for all networks trained with quantile activation, we use quantile classifier to compute the accracies/calibration errors.

4 Evaluation

To summarize, we make the following changes to the existing classification pipeline – (i) Replace the usual ReLU activation with QACT and (ii) Use triplet or watershed loss instead of standard cross entropy loss. We expect this framework to learn context dependent features, and hence be robust to distortions. (iii) Also, use quantile classifier to train the classifier on the embedding for better calibrated probabilities.



(a) Accuracy

(b) Top-Label Calibration Error

Figure 3: Comparing QAct with ReLU activation and Dinov2 (small) on CIFAR10C. We observe that, while at low severity of distortions QAct has a similar accuracy as existing pipelines, at higher levels the drop in accuracy is substantially smaller than existing approaches. With respect to calibration, we observe that the calibration error remains constant (up to standard deviations) across distortions.

Evaluation Protocol: To evaluate our approach, we consider the two datasets developed for this purpose – CIFAR10C, CIFAR100C, TinyImagenetC [15], MNISTC[25]. These datasets have a set of 15 distortions at 5 severity levels. To ensure diversity we evaluate our method on 4 architectures – (overparametrized) LeNet, ResNet18[14] (11M parameters), VGG[35](15M parameters) and DenseNet [18](1M parameters). The code to reproduce the results can be found at `https://anonymous.4open.science/r/QuantAct-2B41`.

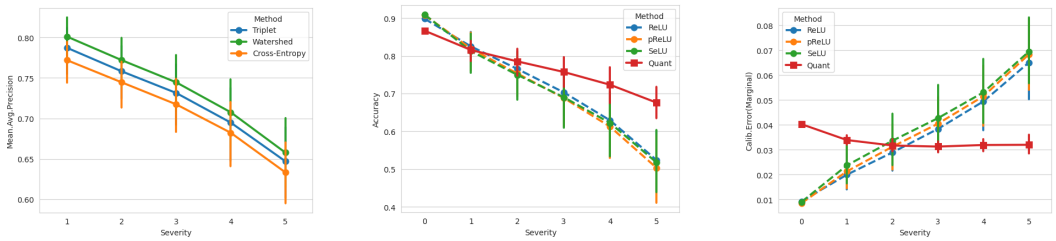
Baselines for Comparison: To our knowledge, there exists no other framework which proposed classification based on context distribution. So, for comparison, we consider standard ReLU activation [10], pReLU [13], and SELU [20] for all the architectures stated above. Also, we compare our results with DINOv2 (small) [27] (22M parameters) which is current state-of-the-art for domain generalization. Note that for DINOv2, architecture and datasets used for training are substantially different (and substantially larger) from what we consider in this article. Nevertheless, we include the results for understanding where our proposed approach lies on the spectrum. We consider the small version of DINOv2 to match the number of parameters with the compared models.

Metrics: We consider four metrics – Accuracy (ACC), calibration error (ECE) [23] (both marginal and Top-Label) and mean average precision at K (MAP@K) to evaluate the embedding. For the case of ReLU/pReLU/SELU activation with Cross-Entropy, we use the logistic regression trained on the train set embeddings, and for QAct we use the calibrated linear classifier, as proposed above. We do not perform any additional calibration and use the probabilities. We discuss a selected set of results in the main article. Please see appendix C for more comprehensive results.

Calibration error measures the reliability of predicted probabilities. In simple words, if one predicts 100 samples with (say) probability 0.7, then we expect 70 of the samples to belong to class 1 and the rest to class 0. This is measured using either the marginal or top-label calibration error. We refer the reader to [23] for details, which also provides an implementation to estimate the calibration error.

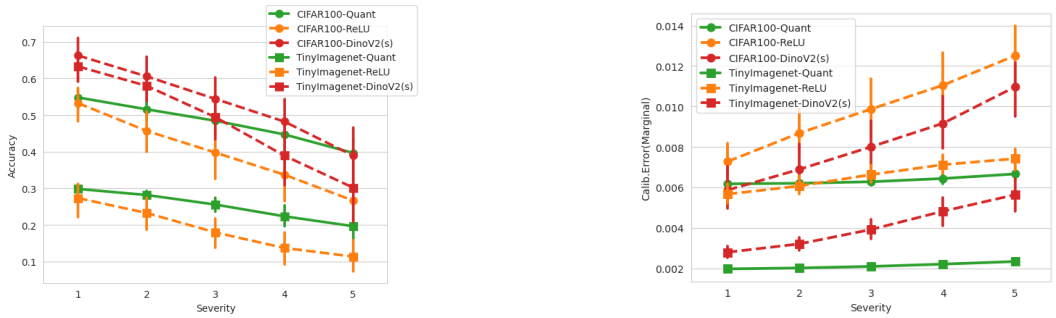
Remark: For all the baselines we use the standard Cross-Entropy loss for training. For inference on corrupted datasets, we retrain the last layer with logistic regression on the train embedding and evaluate it on test/corrupted embedding. For QAct, we as a convention use watershed loss unless otherwise stated, for training. For inference, we train the Quantile Classifier on the train embedding and evaluate it on test/corrupted embedding.

The proposed QAct approach is robust to distortions: In fig. 3 we compare the proposed QAct approach with predominant existing pipeline – ReLU+Cross-Entropy and DINOv2(small) on CIFAR10C. In figure 3a we see that as the severity of the distortion increases, the accuracy of ReLU and DINOv2 drops significantly. On the other hand, while at small distortions the results are comparable, as severity increases QAct performs substantially better than conventional approaches.



(a) (b) (c)

Figure 4: (a) Dependence on Loss functions. Here we compare watershed with other popular loss functions – Triplet and Cross-Entropy when used with QACT. We see that watershed performs slightly better with respect to MAP. (b) Comparing QACT with other popular activations – ReLU/pReLU/SELU with respect to accuracy. (c) Comparing QACT with other popular activations – ReLU/pReLU/SELU with respect to Calibration Error (Marginal). From both (b) and (c) we can conclude that QACT is notably more robust across distortions than several of the existing activation. All the plots use ResNet18 with CIFAR10C dataset.



(a) (b)

Figure 5: Results on CIFAR100C/TinyImagenetC. We compare QACT+watershed to ReLU and DinoV2 small on CIFAR100C/TinyImagenetC dataset with ResNet18. Note that the observations are consistent with CIFAR10C. (a) shows how accuracy changes across distortions. Observe that QACT is similar to DINOv2(s) with respect to embedding quality across all distortions, even if DINOv2 has 22M parameters as compared to Resnet18 11M parameters and is trained on larger datasets. (b) shows how calibration error (marginal) changes across severities. While other approaches lead to an increase in calibration error, QACT has similar calibration error across distortions.

At severity 5, QACT outperforms DINOv2. On the other hand, we observe that in figure 3b, the calibration error stays consistent across distortions.

How much does QACT depend on the loss function? Figure 4a compares the watershed classifier with other popular losses – Triplet and Cross-Entropy. We see that all the loss functions perform comparably when used in conjunction with QACT. We observe that watershed has a slight improvement when considering MAP and hence, we consider that as the default setting. However, we point out that QACT is compatible with several loss functions as well.

QACT vs ReLU/pReLU/SELU activations: To verify that most existing activations do not share the robustness property of QACT, we compare QACT with other activations in figures 4b and 4c. We observe that QACT is greatly more robust with respect to distortions in both accuracy and calibration error than other activation functions.

Results on Larger Datasets: To verify that our observations hold for larger datasets, we use CIFAR100C/TinyImagenetC to compare the proposed QACT+watershed with existing approaches. We observe on figure 5 that QACT performs comparably well as DINOv2, although DINOv2(s) has 22M parameters and is trained on significantly larger datasets. Moreover, we also observe that

QACT has approximately constant calibration error across distortions, as opposed to a significantly increasing calibration error for ReLU or DINOv2.

5 Conclusion And Future Work

To summarize, traditional classification systems do not consider the "context distributions" when assigning labels. In this article, we propose a framework to achieve this by – (i) Making the activation adaptive by using quantiles and (ii) Learning a kernel instead of the boundary for the last layer. We show that our method is more robust to distortions by considering MNISTC, CIFAR10C, CIFAR100C, TinyImagenetC datasets across varying architectures.

The scope of this article is to provide a proof of concept and a framework for performing inference in a context-dependent manner. We outline several potential directions for future research:

I. The key idea in our proposed approach is that the quantiles capture the distribution of each neuron from the batch of samples, providing outputs accordingly. This poses a challenge for inference, and we have discussed two potential solutions: (i) remember the quantiles and density estimates for single sample evaluation, or (ii) ensure that a batch of samples from the same distribution is processed together. We adopt the latter method in this article. An alternative approach would be to *learn the distribution of each neuron* using auxiliary loss functions, adjusting these distributions to fit the domain at test time. This gives us more control over the network at test time compared to current workflows.

II. Since the aim of the article was to establish a proof-of-concept, we did not focus on scaling, and use only a single GPU for all the experiments. To extend it to multi-GPU training, one needs to synchronize the quantiles across GPU, in a similar manner as that for Batch-Normalization. We expect this to improve the statistics, and to allow considerably larger batches of training.

III. On the theoretical side, there is an interesting analogy between our quantile activation and how a biological neuron behaves. It is known that when the inputs to a biological neuron change, the neuron adapts to these changes [7]. Quantile activation does something very similar, which leads to an open question – can we establish a formal link between the adaptability of a biological neuron and the accuracy of classification systems?

IV. Another theoretical direction to explore involves considering distributions not just at the neuron level, but at the layer level, introducing a high-dimensional aspect to the problem. The main challenge here is defining and utilizing *high dimensional quantiles*, which remains an open question [21].

Broad Impact: In this article, we propose an approach to maintain calibration and generalization across small distortions. While, we do not foresee any direct societal consequences of our work, we expect the potential future consequences of the technique to reduce the bias in the following ways – (i) Since we do not assume normal distribution, our approach is likely to handle long tails better than existing methods. This would help in reducing the dataset bias where marginal groups are less represented. (ii) Note that the output of each QACT layer is a uniform distribution. This can allow us to understand the working of each layer in isolation and possibly reduce the black-box nature of the current classification systems. (iii) Moreover, by directly modifying the context distribution of each neuron, one can easily make the networks more reliable without resorting to expensive re-training the entire network.

References

[1] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *European Conf. Mach. Learning*, 2019.

[2] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 2019.

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

[4] Jens Behrmann, Paul Vicol, Kuan-Chieh Wang, Roger B. Grosse, and Jörn-Henrik Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *Artificial Intelligence and Statistics*, 2021.

[5] Aditya Challa, Snehanshu Saha, and Soma Dhavala. Quantprob: Generalizing probabilities along with predictions for a pre-trained classifier. *arXiv:2304.12766*, 2023.

[6] Aditya Challa, Sravan Danda, and Laurent Najman. A novel approach to regularising 1nn classifier for improved generalization. *arXiv:2402.08405*, 2024.

[7] Colin WG Clifford, Michael A Webster, Garrett B Stanley, Alan A Stocker, Adam Kohn, Tatyana O Sharpee, and Odelia Schwartz. Visual adaptation: Neural, psychological and computational aspects. *Vision research*, 2007.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. on Learning Representations*, 2021.

[9] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Neural Inform. Process. Syst.*, 2019.

[10] Kunihiko Fukushima. Correction to "visual feature extraction by a multilayered network of analog threshold elements". *IEEE Trans. Syst. Sci. Cybern.*, 1970.

[11] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proc. Int. Conf. Comput. Vision*, 2015.

[12] Zhongyi Han, Guanglin Zhou, Rundong He, Jindong Wang, Tailin Wu, Yilong Yin, Salman H. Khan, Lina Yao, Tongliang Liu, and Kun Zhang. How well does gpt-4v(ision) adapt to distribution shifts? A preliminary investigation. *arXiv:2312.07424*, 2023.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. Int. Conf. Comput. Vision*, 2015.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[15] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Int. Conf. on Learning Representations*, 2019.

[16] Shoubo Hu, Kun Zhang, Zhitang Chen, and Laiwan Chan. Domain generalization via multidomain discriminant analysis. In *Uncertainity in Artificial Intelligence*, 2019.

[17] Bincheng Huang, Si Chen, Fan Zhou, Cheng Zhang, and Feng Zhang. Episodic training for domain generalization using latent domains. In *Int. Conf. on Cogni. Systems and Signal Process.*, 2020.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243. URL `https://doi.org/10.1109/CVPR.2017.243`.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. Mach. Learning*, 2015.

[20] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Neural Inform. Process. Syst.*, 2017.

[21] Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005. doi: 10.1017/CBO9780511754098.

[22] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Int. Conf. Mach. Learning*, 2020.

[23] Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Neural Inform. Process. Syst.*, 2019.

[24] Zhong Li and Matthijs van Leeuwen. Explainable contextual anomaly detection using quantile regression forests. *Data Min. Knowl. Discov.*, 2023.

[25] Norman Mu and Justin Gilmer. MNIST-C: A robustness benchmark for computer vision. *arXiv:1906.02337*, 2019.

[26] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Int. Conf. Mach. Learning*, 2010.

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2023.

[28] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 2010.

[29] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *Int. Conf. Mach. Learning*, 2020.

[30] Tejas Prashanth, Snehanshu Saha, Sumedh Basarkod, Suraj Aralihalli, Soma S. Dhavala, Sriparna Saha, and Raviprasad Aduri. Lipgene: Lipschitz continuity guided adaptive learning rates for fast convergence on microarray expression data sets. *IEEE ACM Trans. Comput. Biol. Bioinform.*, 2022.

[31] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Neural Inform. Process. Syst.*, 2016.

[32] Aditi Seetha, Satyendra Singh Chouhan, Emmanuel S Pilli, Vaskar Raychoudhury, and Snehanshu Saha. Dievd-sf: Disruptive event detection using continual machine learning with selective forgetting. *IEEE Transactions on Computational Social Systems*, 2024.

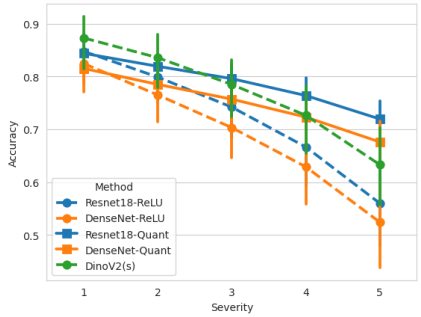
[33] Hogeon Seo, Seunghyoung Ryu, Jiyeon Yim, Junghoon Seo, and Yonggyun Yu. Quantile autoencoder for anomaly detection. In *AAAI,Workshop on AI for Design and Manufacturing (ADAM)*, 2022.

[34] Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. Stable learning via sample reweighting. In *AAAI Conf. on Artificial Intelligence*, 2020.

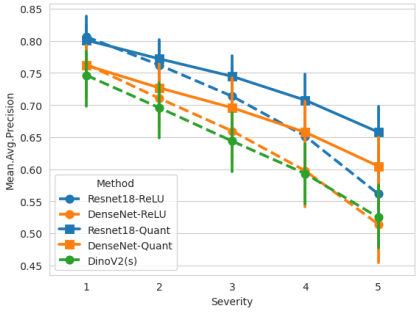
[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. on Learning Representations*, 2015.

[36] Anuj Tambwekar, Anirudh Maiya, Soma S. Dhavala, and Snehanshu Saha. Estimation and applications of quantiles in deep binary classification. *IEEE Trans. Artif. Intell.*, 2022.

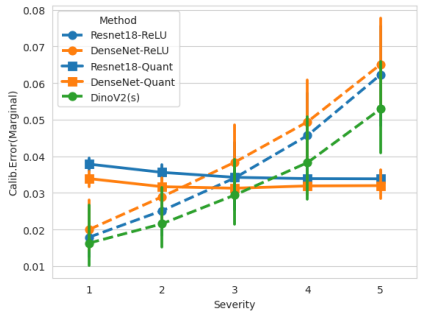
[37] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proc. IEEE*, 2021.



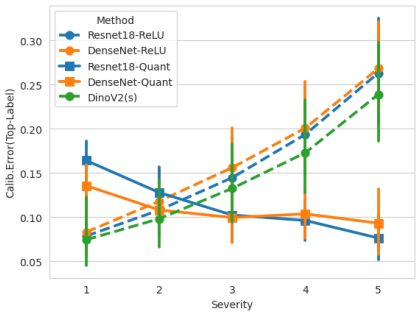
(a) Accuracy



(b) Mean Average Precision (MAP@100)



(c) Marginal Calibration Error



(d) Top-Label Calibration Error

Figure 6: Comparing QACT with ReLU activation and Dinov2 (small).

A Experiment details for figure 1

We consider the features obtained from ResNet18 with both QACT and RelU activations for the datasets of CIFAR10C with gaussian_noise at all the severity levels. Hence, we have 6 datasets in total. To use TSNE for visualization, we consider 1000 samples from each dataset and obtain the combined TSNE visualizations. Each figure shows a scatter plot of the 2d visualization for the corresponding dataset.

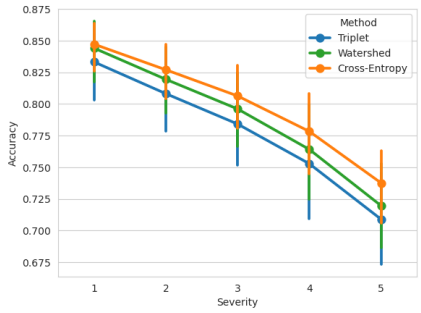
B Compute Resources and Other Experimental Details

All experiments were performed on a single NVidia GPU with 32GB memory with Intel Xeon CPU (10 cores). For training, we perform an 80:20 split of the train dataset with seed 42 for reproducibility. All networks are initialized using default pytorch initialization technique.

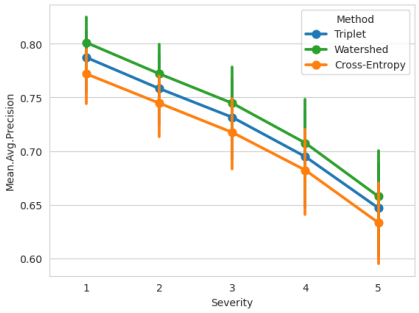
We use Adam optimizer with initial learning rate $1e-3$. We use ReduceLROnPlateau learning rate scheduler with parameters – factor=0.1, patience=50, cooldown=10, threshold=0.01, threshold_mode=abs, min_lr=1e-6. We monitor the validation accuracy for learning rate scheduling. We also use early_stopping when the validation accuracy does not increase by 0.001.

C Extended Results Section

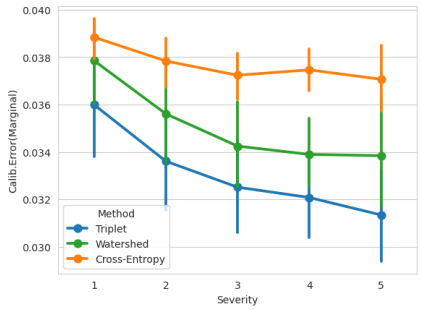
Comparing QACT + watershed and ReLU+Cross-Entropy: Figure 6 shows the corresponding results. The first experiment compares QACT + watershed with ReLU + Cross-Entropy on two standard networks – ResNet18 and DenseNet. With respect to accuracy, we observe that while at severity 0, ReLU + Cross-Entropy slightly outperforms QACT + watershed, as severity increases QACT + watershed is far more stable. We even outperform DinoV2(small) (22M parameters) at severity 5. Moreover, with respect to calibration error, we see a consistent trend across distortions. As [5] argues, this helps in building more robust systems compared to one where calibration error increases across distortions.



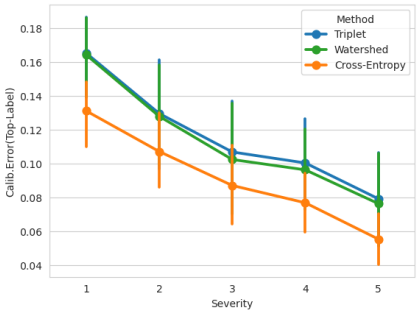
(a) Accuracy



(b) Mean Average Precision (MAP@100)



(c) Marginal Calibration Error



(d) Top-Label Calibration Error

Figure 7: Triplet vs Watershed vs Cross-Entropy

Does loss function make a lot of difference? Figure 7 compares three different loss functions Watershed, Triplet and Cross-Entropy when used in conjunction with QACT. We observe similar trends across all loss functions. However, Watershed performs better with respect to Mean Average Precision (MAP) and hence we use this as a default strategy.

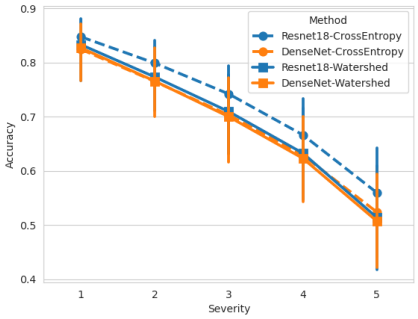
Why Mean-Average-Precision? – We argue that the key indicator of distortion invariance should be the quality of embedding. While, accuracy (as measured by a linear classifier) is a good metric, a better one would be to measure the Mean-Average-Precision. With respect to calibration error, due to the scale on the Y-axis, the figures suggest reducing calibration error. However, the standard deviations overlap, and hence, these are assumed to be constant across distortions.

How well does watershed perform when used with ReLU activation? Figure 8 shows the corresponding results. We observe that both the watershed loss and cross-entropy have large overlaps in the standard deviations at all severity levels. So, this shows that, when used in conjunction with ReLU watershed and cross-entropy loss are very similar. But in conjunction with QACT, we see that watershed has a slightly higher Mean-Average-Precision.

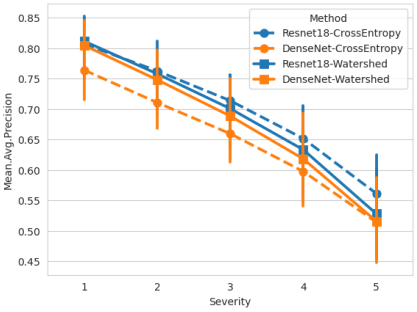
What if we consider an easy classification task? In figure 9, we perform the comparison of QACT+Watershed and ReLU and cross-entropy on MNISTC dataset. Across different architectures, we observe a lot less variation (standard deviation) of QACT+Watershed compared to RelU and cross-entropy. This again suggests robustness against distortions of QACT+Watershed.

Comparing with other popular activations: Figures 10 and 11 shows the comparison of QACT with ReLU, pReLU and SeLU. We observe the same trend across ReLU, pReLU and SeLU, while QACT is far more stable across distortions.

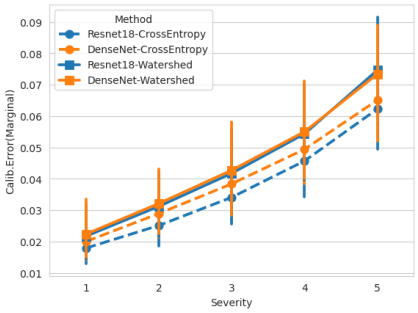
Results on CIFAR100/TinyImagenetC: Figure 12 compares QACT+Watershed and ReLU+Cross-Entropy on CIFAR100C dataset. We also include the results of QACT+Cross-Entropy vs. ReLU+Cross-Entropy on TinyImagenetC. The results are consistent with what we observe on CIFAR10C, and hence, draw the same conclusions as before.



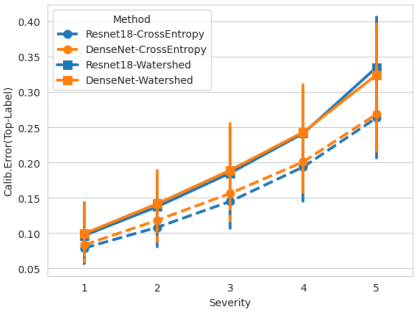
(a) Accuracy



(b) Mean Average Precision (MAP@100)

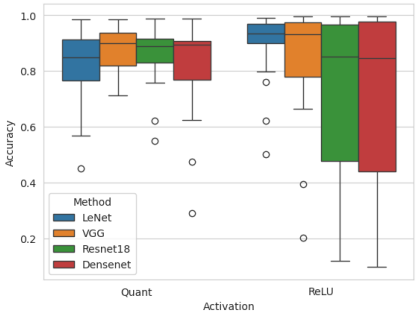


(c) Marginal Calibration Error

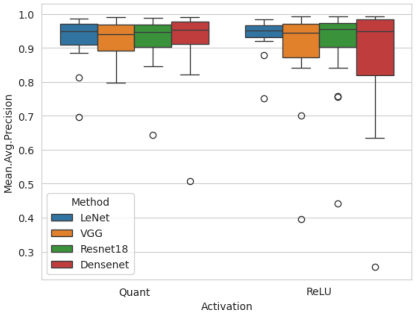


(d) Top-Label Calibration Error

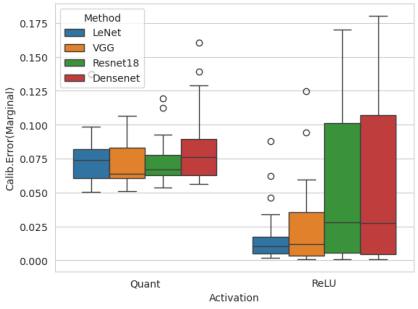
Figure 8: Watershed vs Cross-Entropy when using ReLU activation



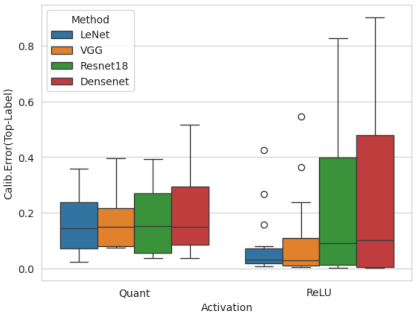
(a) Accuracy



(b) Mean Average Precision (MAP@100)

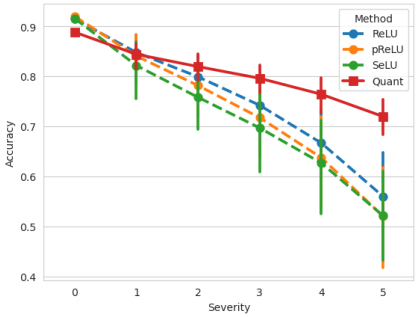


(c) Marginal Calibration Error

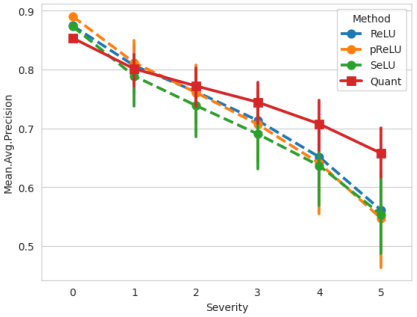


(d) Top-Label Calibration Error

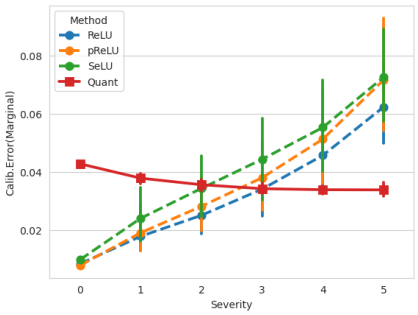
Figure 9: Results on MNIST



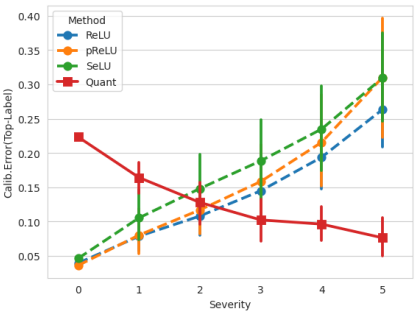
(a) Accuracy



(b) Mean Average Precision (MAP@100)

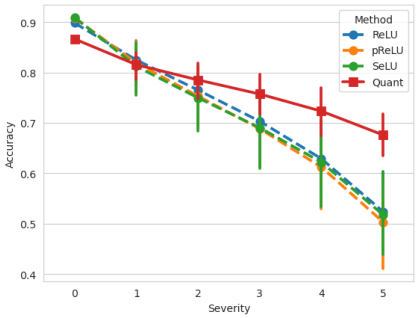


(c) Marginal Calibration Error

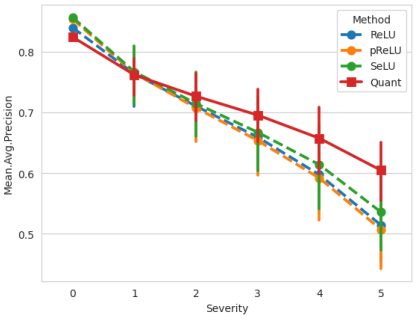


(d) Top-Label Calibration Error

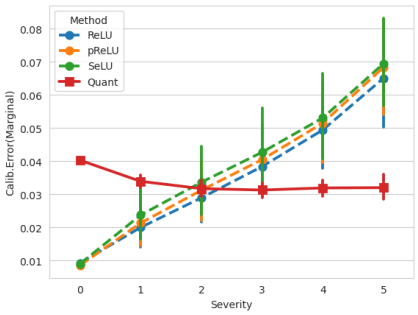
Figure 10: QACTvs ReLU vs pReLU vs Selu activations on ResNet18



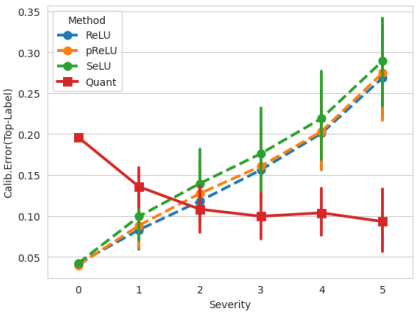
(a) Accuracy



(b) Mean Average Precision (MAP@100)

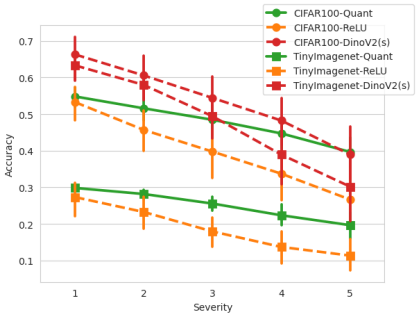


(c) Marginal Calibration Error

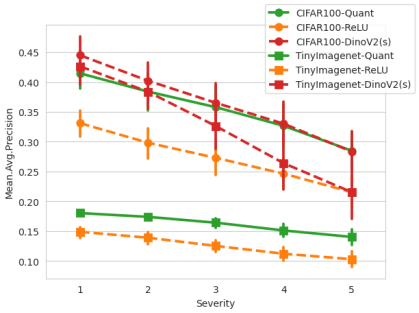


(d) Top-Label Calibration Error

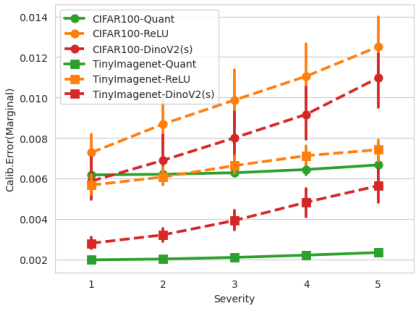
Figure 11: QACTvs ReLU vs pReLU vs Selu activations on Densenet



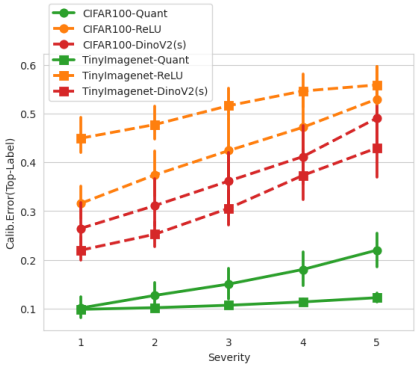
(a) Accuracy



(b) Mean Average Precision (MAP@100)

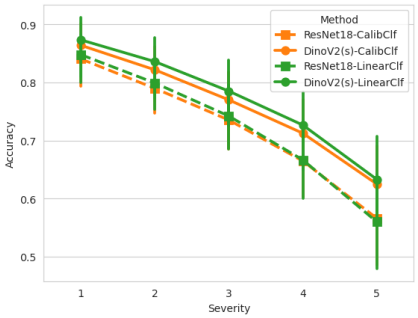


(c) Marginal Calibration Error

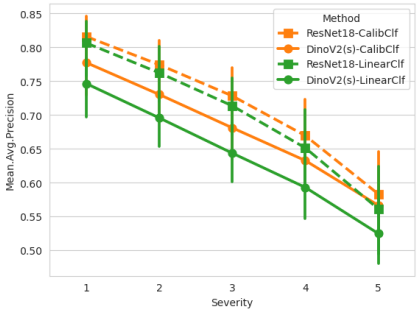


(d) Top-Label Calibration Error

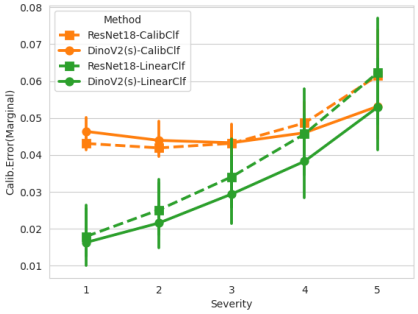
Figure 12: QACTvs ReLU on Resnet18+CIFAR100



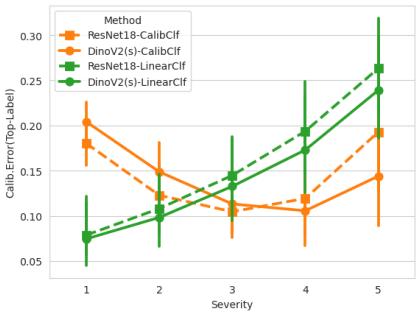
(a) Accuracy



(b) Mean Average Precision (MAP@100)

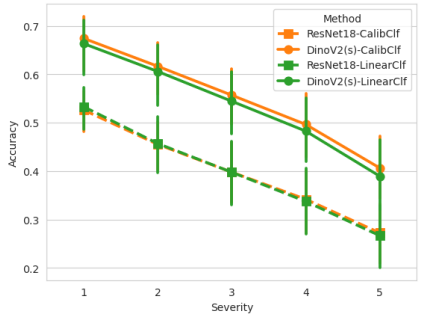


(c) Marginal Calibration Error

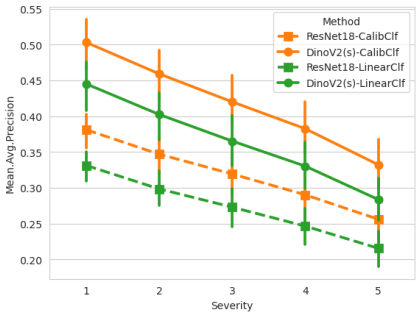


(d) Top-Label Calibration Error

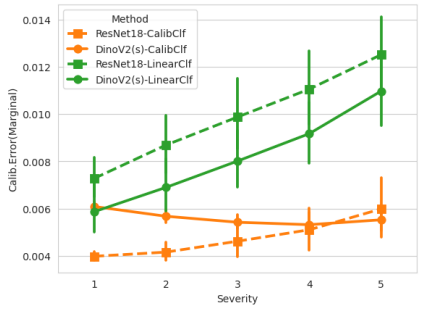
Figure 13: Effect of Quantile Classifier. We use ResNet18 and DinoV2 architectures on CIFAR10.



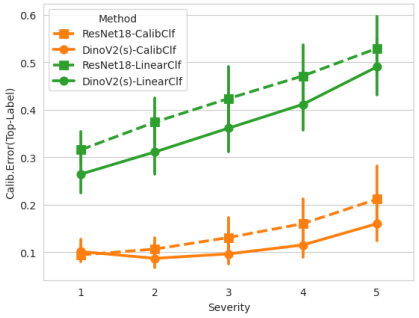
(a) Accuracy



(b) Mean Average Precision (MAP@100)



(c) Marginal Calibration Error



(d) Top-Label Calibration Error

Figure 14: Effect of Quantile Classifier. We use ResNet18 and DinoV2 architectures on CIFAR100.

Effect of Quantile Classifier: Figures 13 and 14 shows the effect of quantile classifier on standard ResNet10/DinoV2 outputs with CIFAR10C/CIFAR100C datasets. While the accuracy values are almost equivalent, we observe a "flatter" trend of the calibration errors, sometimes reducing the error as in the case of CIFAR100C.

D Watershed Loss

The authors in [6] proposed a novel classifier – *watershed classifier*, which works by learning similarities instead of the boundaries. Below we give the brief idea of the loss function, and refer the reader to the original paper for further details.

1. Let $(\boldsymbol{x}_i, y_i)$ denote the samples in each batch, and let f_θ denote the embedding network. $f_\theta(\boldsymbol{x}_i)$ denotes the corresponding embedding.
2. Starting from randomly selected seeds in the batch, propagate the labels to all the samples. Let $\hat{y_i}$ denote the estimated samples. For each $f_\theta(\boldsymbol{x}_i)$ and for each label l, obtain the nearest neighbour in the samples in the set,

$$\mathcal{S}_l = \{f_\theta(\boldsymbol{x}_i) \mid \hat{y_i} = y_i = l\} \tag{9}$$

   that is, all the samples of class l labelled correctly. Denote this nearest neighbour using $f_\theta(\boldsymbol{x}_{i,l,1nn})$.
3. Then the loss is given by,

$$\text{Watershed Loss} = \frac{-1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} \sum_{l=1}^{L} I[y_i = l] \log \left(\frac{\exp\left(-\|f_\theta(\boldsymbol{x}_i) - f_\theta(\boldsymbol{x}_{i,l,1nn})\|\right)}{\sum_{j=1}^{L} \exp\left(-\|f_\theta(\boldsymbol{x}_i) - f_\theta(\boldsymbol{x}_{i,j,1nn})\|\right)} \right) \tag{10}$$

Why Watershed Loss?: Observe that the loss in equation 10 implicitly learns representations consistent with the RBF kernel, which is known to be translation invariant. Minimizing this loss function, hence, will learn translation invariant kernels. This is important for obtaining networks robust to distortions.

If one uses (say) Cross Entropy loss, then the features learned would be such that the classes are linearly separable. Contrast this with watershed, which instead learns a similarity between two points in a translation invariant manner.

Remark: Observe that the watershed loss is very similar to metric learning losses. The authors in [6] claim that this offers better generalization, and show that this is consistent with 1NN classifier. Moreover, they show that this classifier (without considering f_θ) has a VC dimension which is equal to the number of classes. While metric learning losses are similar, there is no such guarantee with respect to classification. This motivated our choice of using watershed loss over other metric learning losses.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: Yes. The main contribution is a proof-of-concept that one should move away from single point estimation for better generalization across distortions.
   Guidelines:
   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: Since, the scope is only a proof-of-concept, we have not considered scaling to large datasets/models in this work. Scaling these ideas would require rethinking current strategies and does not fit perfectly into existing framework. Moreover, the proposed approach slightly more resource intensive than ReLU activation.
   Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We do not consider theoretical aspects in this article. While there are interesting theoretical connections, we leave it for future work.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide an anonymous link to generate all the results provided in the article. Moreover, we describe all the hyper-parameters used in the appendix as well.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully

might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**
   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?
   Answer: [Yes]
   Justification: We provide the code using an anonymous link at `https://anonymous.4open.science/r/QuantAct-2B41`. The datasets are all public datasets which can be downloaded.
   Guidelines:
   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?
   Answer: [Yes]
   Justification: We explain the experimental setting in complete detail in the article.
   Guidelines:
   - The answer NA means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The datasets used incorporate 15 kinds of distortions across 5 severity levels. We report the error bars across the 15 kinds of distortions which should provide a good picture of the reliability of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer:[Yes]

   Justification: Yes. we include the entire information about the compute resources in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have tried to maintain the double-blind policy to the maximum extent possible.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In this article we propose a novel way to allow robustness against distortions. We do not expect any negative societal impacts. There could be positive societal impact since this can potentially stop hallucinations/bias of the ML models.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not use any datasets/models that have high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original articles of all the datasets/models we use in the article. We have ensured that these can be used with credit for academic purposes.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We share the code at `https://anonymous.4open.science/r/QuantAct-2B41`.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.