A. Implementation and Data

Data and implementation can be found at https://github.com/HuakunShen/VCR.

B. Overview of VCR-Bench

Our method for benchmarking VCR (VCR-Bench) is outlined in Fig. 9. Step I generates a validation set that covers the full continuous range of visual changes. This is achieved by uniformly sampling from the entire domain of corruption function parameters. Step II obtains human robustness performance data needed to compute our two newly-proposed human-aware evaluation metrics: *Human-Relative Model Robustness Index* (HMRI) and *Model Robustness Superiority Index* (MRSI), which quantify the extent to which a NN can replicate or surpasses human performance, respectively. Since measuring human performance for every single image corruption function is expensive and impractical, we propose a method to reduce the cost by generalizing existing human performance data obtained for one corruption function to a class of corruption functions with similar visual effects. For example, images transformed with Gaussian Blur and Glass Blur have very similar visual effects on humans, unlike Motion Blur and Brightness. Thus, Gaussian Blur and Glass Blur, but not with Motion Blur and Brightness, thus they belong to the same class of similar corruption functions, and human performance data for one can be transferred to the other. Step III of VCR-Bench evaluates the model using the validation dataset and our human-aware metrics. Then it retrains the model to improve its robustness.

C. VCR and Its Estimation

Background: Image Quality Assessment (IQA). IQA metrics serve as quantitative measures of human objective image quality [48]. By comparing the original image and the transformed image, IQA metrics automatically estimate the perceived image quality by evaluating the perceptual "distance" between the two images [41]. This "distance" differs from simple pixel distance and varies depending on the specific IQA metric used.

One such metric is VIF (Visual Information Fidelity) [41], which evaluates the fidelity of information by analyzing the statistical properties of natural scenes within the images. VIF returns a value between 0 and 1 if the changes degrade perceived image quality, with 1 indicating the perfect quality compared to the original image; and it returns a value > 1 if the changes enhances image quality [41]. More precisely, VIF defines the visual quality of a distorted image as a ratio of the amount of information a human can extract from the distorted image versus the original reference image. The method models statistically (i) images in the wavelet domain with coefficients drawn from a Gaussian scale mixture, (ii) distortions as attenuation and additive Gaussian noise in the wavelet domain, and (iii) the human visual system (HVS) as additive white Gaussian noise in each sub-band of the wavelet decomposition. The amount of information that a human can extract from the distorted image is measured as the mutual information between the distorted image and the output of the HVS model for that image. Similarly, the amount of information that a human can extract from the reference image is measured as the mutual information between the reference image and the output of the HVS model for that image. Empirical studies have shown that VIF aligns closely with human opinions when compared to other IQA metrics [42].

We choose VIF, since it is well-established, computationally efficient, applicable to our transformations, and still performing competitively compared to newer metrics. More recent research has explored the use of feature spaces computed by deep NNs as a basis to define IQA metrics (e.g., LPIPS [56] and DISTS [4]). Even though these metrics may be applicable to a wider class of transformations than VIF, including those that affect both structure and textures, their scope may depend on the training datasets in potentially unpredictable ways. On the other hand, the scope of VIF is well-defined based on the metric's mathematical definition. In particular, VIF is suitable for evaluating corruption functions that can be locally described as a combination of signal attenuation and additive Gaussian noise in the subbands of the wavelet domain [41]. The transformations in our experiments are local corruptions that are well within this scope. Moreover, VIF performs still competitively when compared to even the newer DNN-based metrics across multiple datasets (e.g., see Table 1 in [4]). However, future work should explore VCR using other IQA metrics.

Visual Change (Δ_v). The metric (Δ_v) defined using the IQA metric VIF, as shown in Def. 3, is proposed by Hu et al. [20] to quantitatively measure the amount of visual changes in the images perceived by human observers.

Definition 3. *Let an image x, an applicable corruption function T_X with a parameter domain C and a parameter $c \in C$, s.t. $x' = T_X(x, c)$ be given. Visual change $\Delta_v(x, x')$ is a function defined as follows:*

$$\begin{cases} 0 & \textit{If VIF}(x, x') > 1 \\ 1 - \textit{VIF}(x, x') & \textit{Otherwise} \end{cases}$$

Δ_v returns a value between 0 and 1, with 0 indicating no degradation to visual quality and 1 indicating all visual information in the original image has been changed. The first case of Δ_v corresponds to changes that enhance the visual quality (when $\text{VIF}(x, x') > 1$), indicating changes do not impact human recognition of the images negatively, hence $\Delta_v = 0$. The other case deals with visible changes that degrade visual quality. Since VIF returns 1 for perfect

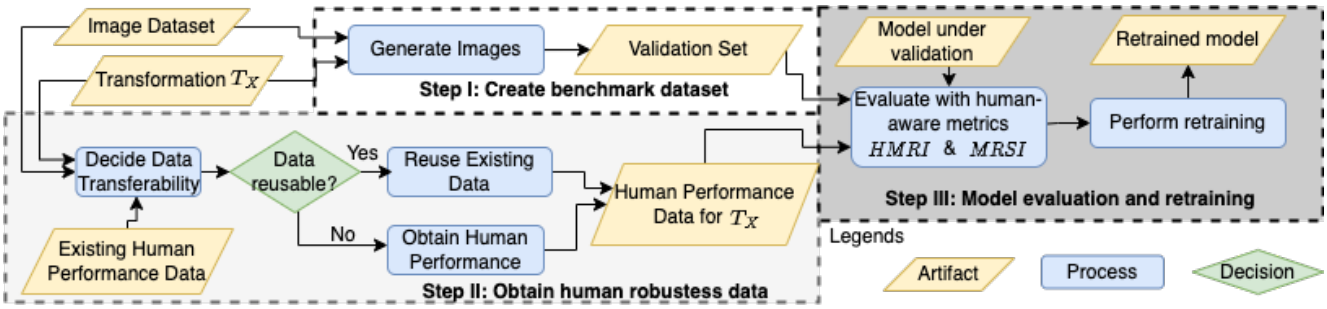


Figure 9. Our proposed method VCR-Bench for benchmarking ML robustness with humans.

quality compared to the original image, the degradation is one minus the image quality score.

Example: In Fig. 10, the visual change of the original image Fig. 10a is 0, since no changes are applied; and Fig. 10b has minimal frost added, which caused minimal change in visual quality so $\Delta_v = 0.005$; and Fig. 10c and Fig. 10d have more frost and thus higher Δ_v values 0.71 and 0.96, respectively.

VCR Estimation Algorithm. Algorithm 1 gives the pseudo-code of the VCR estimation procedure described under "Testing VCR" in the main body of the paper. The algorithm takes a model $f(x)$; a transformation T_X with its parameter domain C; an input dataset; the size N of the dataset of transformed images to be generated; the visual change resolution M, over which the model performance will be estimated; and the minimum size L of a bin to be used to estimate the performance for that bin. The input dataset consist of images $x_k \sim P_X$ for estimating VCR wrt. consistency, or images and their labels for estimating VCR wrt. accuracy. We use $M = 40$ in our experiments, which is a standard choice for calculating average precision in object detection; for example, it is used in the current version of the KITTI benchmark [7].

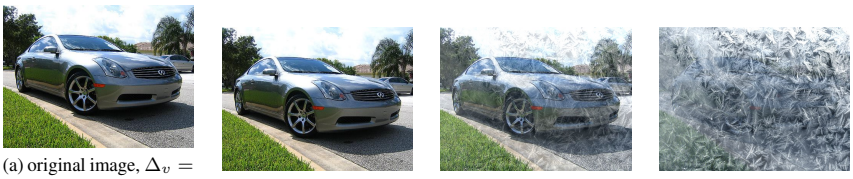
Our algorithm first initializes two histogram arrays to keep the counts of the tested data points and their consistent or accurate predictions, respectively, and an array to keep the performance data, with each of the three arrays having size M. In each iteration, the first for-loop samples an image x and transformation parameter c, and produces a transformed image x'. It then computes the visual change value v and records the result of testing $f(x')$ in the histograms. The second for-loop computes the performance data as a relative frequency of correct predictions. A monotonic smoothing spline is fit into the performance data, and the VCR is computed as the area under the spline.

Note that this algorithm samples c uniformly, which will lead to a varying number of performance samples per point in the performance data array P. As already discussed, the number of performance samples impacts the performance estimate uncertainty at this point, and in an extreme case some of the Δ_v bins in P_i may be even empty (i.e., have value -1). These missing points are mitigated by fitting the spline over the entire Δ_v range, while anchoring it with known values for the first and last bins. In particular, the accuracy spline s_a always starts at the left with the accuracy for clean images, and the consistency spline s_p starts with 1 for models (assuming deterministic NNs).

A possible approach to obtain a sample set with a more uniform coverage of Δ_v would be to (1) fit a strictly monotonic spline into (c, Δ_v) values obtained from $c \sim \mathit{Uniform}(C)$ as in Alg. 1, (2) take a set of samples $\Delta_v \sim \mathit{Uniform}(0, 1)$, (3) map the latter to a new sample from C using the inverted spline, and repeat these steps now using the new sample from C. These steps would need to be run iteratively until a sufficient coverage is obtained. Such an algorithm would be computationally expensive, however.

D. Comparison of Δ_v Distribution

In Fig. 11 below we compare the Δ_v distribution of validation images from IMAGENET-C and those generated by our benchmark. We include all 9 corruption functions shared between IMAGENET-C and our benchmark. Note that all of our images are generated by sampling uniformly in the parameter domain, while IMAGENET-C images are generated with 5 pre-selected parameter values. We can observe two major differences in the distributions. First we can see that because of difference in the parameter values used, the Δ_v distributions between IMAGENET-C and our benchmark peak at different values. For example, for Brightness in Fig. 11a and Fig. 11b, most IMAGENET-C images have Δ_v values between 0.4 to 0.8, while most VCR-Bench images are between 0.6 and 0.9; a similar observation holds for Defocus Blur and Gaussian Blur. Second, we notice that IMAGENET-C images cannot cover all Δ_v values. Specifically, Fig. 11c for Defocus Blur shows that IMAGENET-C validation set does not contain images with Δ_v greater than 0.8 and less than 0.2. The same can be observed for all corruption functions shown in Fig. 11. These two differences indicate that, when considering the full range of visual changes that a corruption function can incur, using IMAGENET-C can lead to biased results.



(a) original image, $\Delta_v = 0$ (b) $\Delta_v = 0.005$ (c) $\Delta_v = 0.71$ (d) $\Delta_v = 0.96$

Figure 10. Examples of images from Imagenet [39] with different levels of added frost.

Algorithm 1: VCR Estimation

Input:
- model $f(x)$
- transformation T_X, with parameter domain C
- input dataset $\{x_k\}$ for consistency [or $\{(x_k, y_k)\}$ for accuracy]
- generated test set size N
- visual change resolution M
- minimum number of points per bin L

Output: estimated VCR $\hat{\mathcal{R}}_p$ [or $\hat{\mathcal{R}}_a$]

Initialize histograms $count_j$ and $correct_j$ with empty counts, for all $j \in [0..M-1]$
Initialize performance data array P_j with -1, denoting missing data points for j, for all $j \in [0..M-1]$
for $i \leftarrow 0$ **to** $N-1$ **do**
  draw random x from $\{x_k\}$ [or (x, y) from $\{(x_k, y_k)\}$]
  $c \sim Uniform(C)$
  $x' \leftarrow T_X(x, c)$
  $v \leftarrow \Delta_v(x, x')$
  $count_j \leftarrow count_j + 1$
  if $f(x') = f(x)$ [or $f(x') = y$] **then**
    $correct_j \leftarrow correct_j + 1$
for $j \leftarrow 0$ **to** $M-1$ **do**
  if $count_j \geq L$ **then**
    $P_j \leftarrow \frac{correct_j}{count_j}$
$s \leftarrow FitMonotonicSpline(P)$
$\hat{\mathcal{R}} \leftarrow \int_0^1 s(v)dv$
return $\hat{\mathcal{R}}$

E. Extra Evaluation Results

E.1. Prediction Similarity of Visually Similar Corruption Functions

In the paper, to check that human robustness data is transferable between two similar corruption functions, we checked whether the 83% confidence interval of the spine curves s_a^h and s_p^h for similar corruption functions overlap. The results for s_a^h in Fig. 8. We also include results for s_p^h in Fig. 12. We can observe that, similar to s_a^h, s_p^h for similar corruption functions are similar, thus human data is transferable.

E.2. CO2 Emission

CO2 Emission is calculated as `CO2 emissions (kg) = (Power consumption in kilowatts) x (Daily usage time in hours) x (Emissions factor in kgCO2/kWh)`

Our carbon intensity is around 25 g/kWh. During benchmark dataset generation, there is no GPU usage, and the CPU usage is 200 W. Each corruption function takes around 1.5 hour to generate a dataset with 50,000 images. During evaluation, the CPU power usage is around 160 W; and GPU power usage ranges between 50-170 W depending on the model. Each evaluation takes 30-60 minutes, depending on the corruption function type. Let's assume the power usage of other components is 50 W in total. If we assume

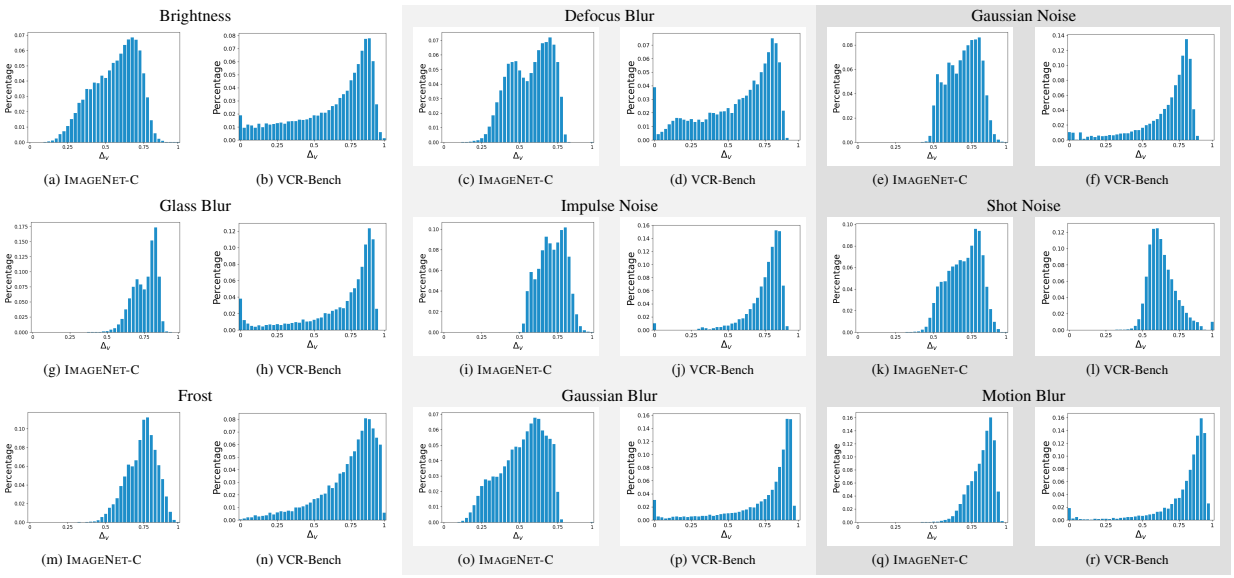
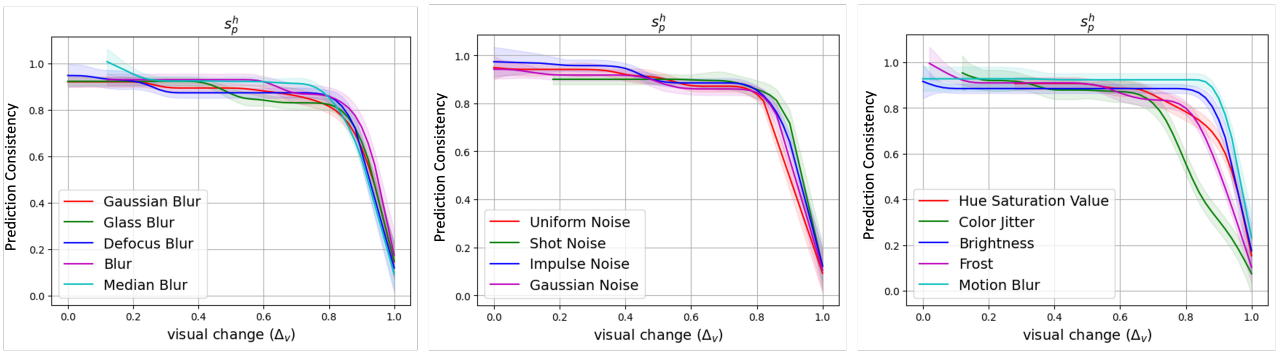


Figure 11. Comparison of Δ_v distribution between IMAGENET-C and VCR-Bench. The figures are histograms, where x-axis is Δ_v, y-axis is percentage of images.



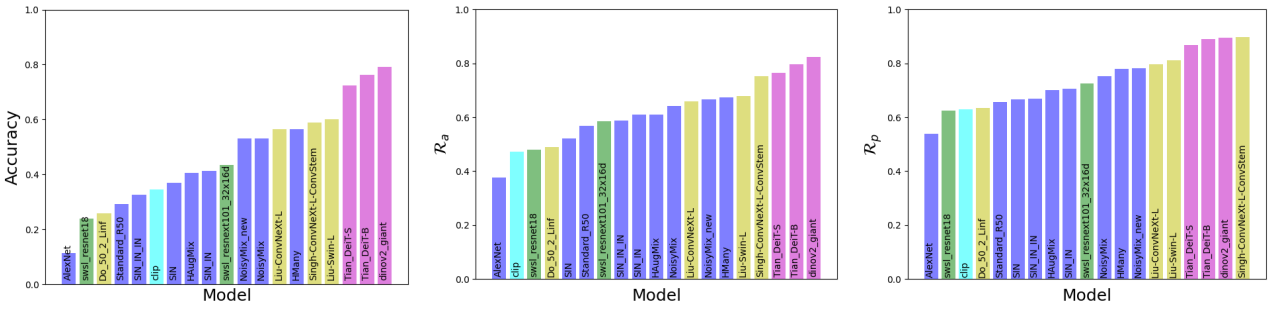
(a) Blur corruption functions (b) Noise corruption functions (c) Dissimilar corruption functions

Figure 12. Comparing human performance spline curves s_p^h for similar and dissimilar corruption functions. For each curve, the coloured region around the curve is the 83% confidence interval used for comparison of similarity [25].

the total power usage is $((200 + 50) \times 1.5 + (170 + 160 + 50))/1{,}000 = 0.755$ kWh for each experiment, the CO2 emission is $0.755 \times 25 = 18.875$ g for each experiment (corruption function type).

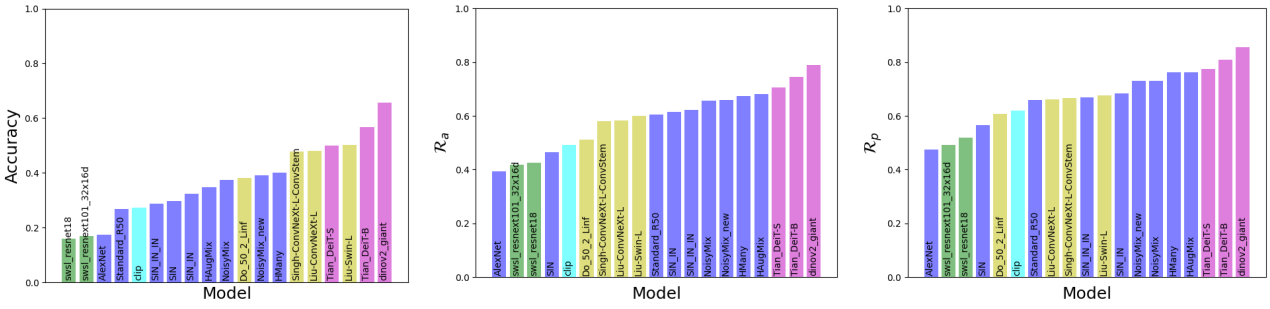
E.3. VCR Evaluation

In the main body of the paper, we have compared VCR robustness results with IMAGENET-C on Gaussian Noise, and we presented the assessing VCR in relation to human performance with our human-aware metrics *HMRI* and *MRSI* for Gaussian Noise and Shot Noise. Below, we first include the comparison between VCR and IMAGENET-C for all IMAGENET-C 9 corruption functions we studied. Then, include detailed evaluation results with our human-aware metrics for all 12 other corruption functions we studied.



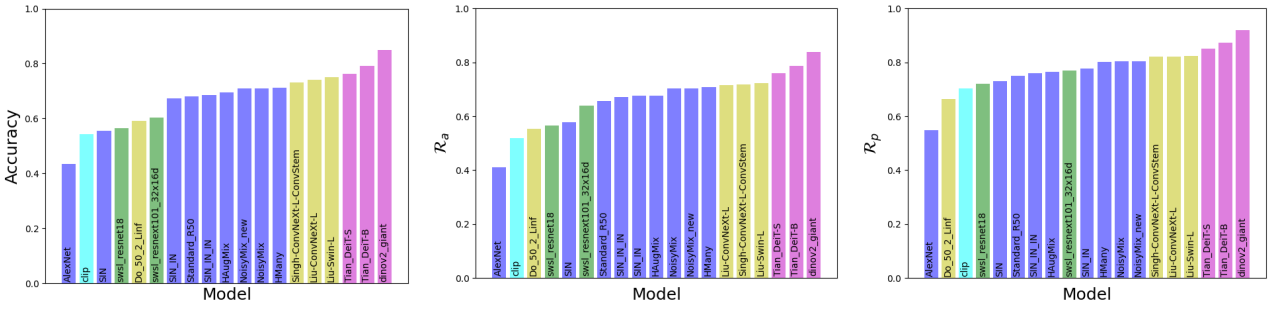
(a) IMAGENET-C Gaussian Noise Accuracy (b) Gaussian Noise $\hat{\mathcal{R}}_a$ (c) Gaussian Noise $\hat{\mathcal{R}}_p$

Figure 13. Comparison between IMAGENET-C and VCR with Gaussian Noise.



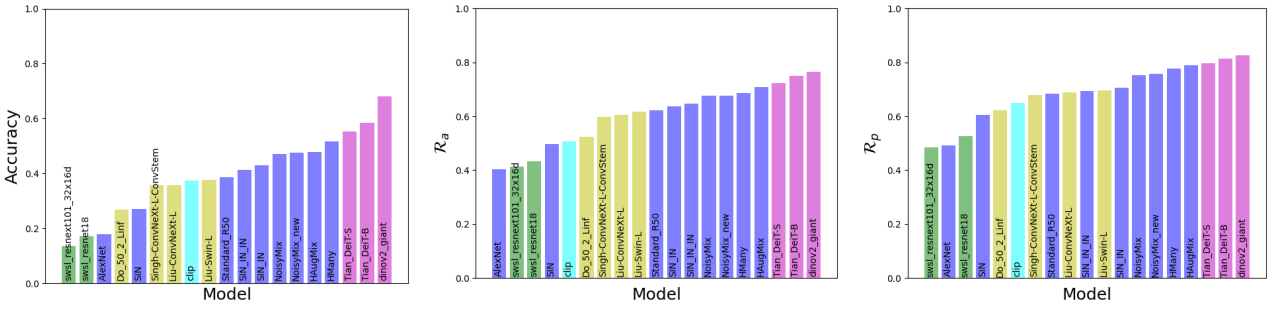
(a) IMAGENET-C Glass Blur Accuracy (b) Glass Blur $\hat{\mathcal{R}}_a$ (c) Glass Blur $\hat{\mathcal{R}}_p$

Figure 14. Comparison between IMAGENET-C and VCR with Glass Blur.



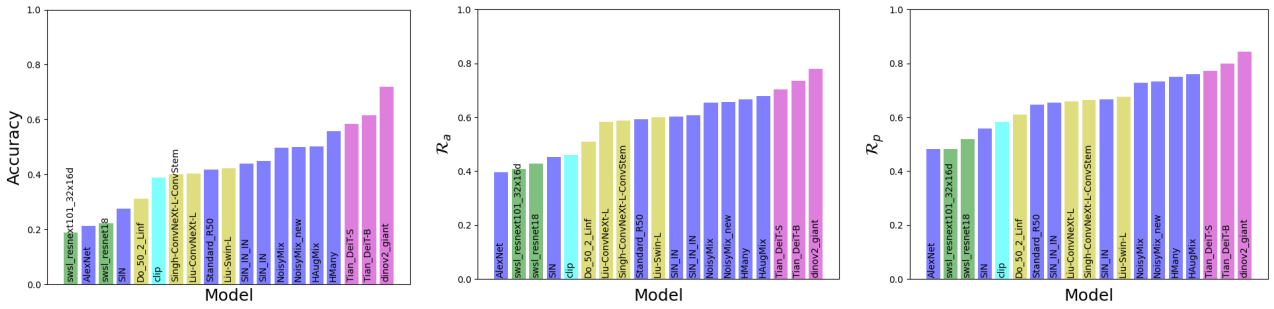
(a) IMAGENET-C Brightness Accuracy (b) Brightness $\hat{\mathcal{R}}_a$ (c) Brightness $\hat{\mathcal{R}}_p$

Figure 15. Comparison between IMAGENET-C and VCR with Brightness.



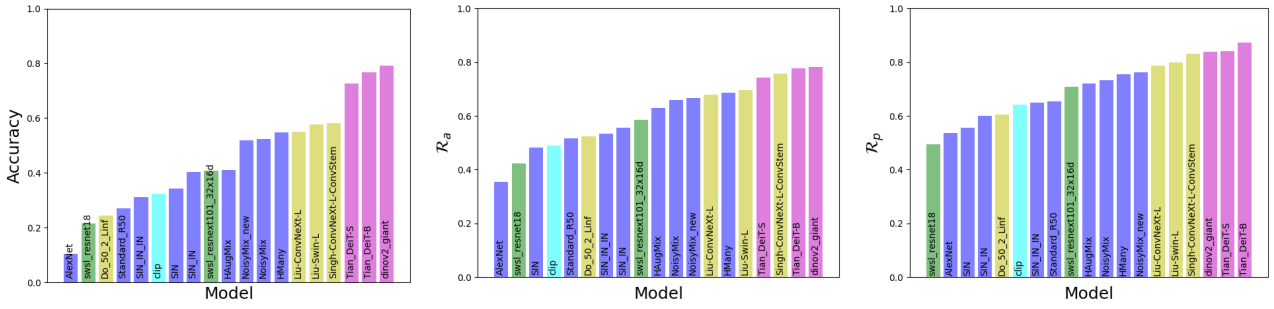
(a) IMAGENET-C Defocus Blur Accuracy (b) Defocus Blur $\hat{\mathcal{R}}_a$ (c) Defocus Blur $\hat{\mathcal{R}}_p$

Figure 16. Comparison between IMAGENET-C and VCR with Defocus Blur.



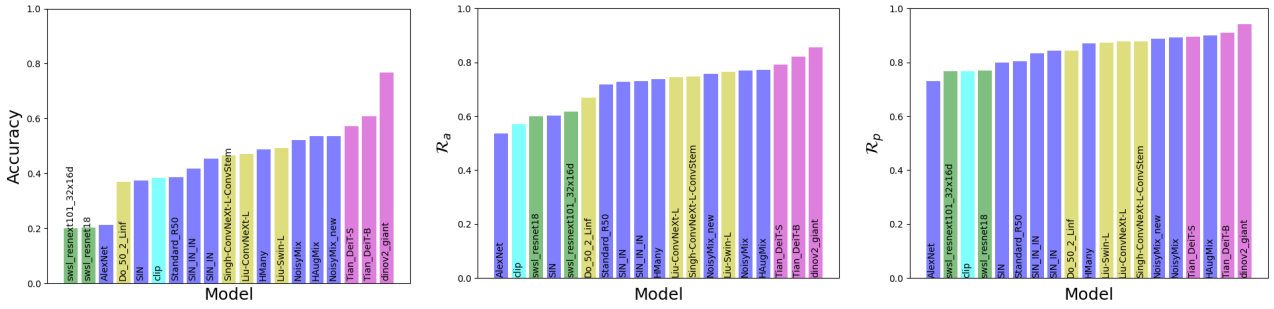
(a) IMAGENET-C Gaussian Blur Accuracy (b) Gaussian Blur $\hat{\mathcal{R}}_a$ (c) Gaussian Blur $\hat{\mathcal{R}}_p$

Figure 17. Comparison between IMAGENET-C and VCR with Gaussian Blur.



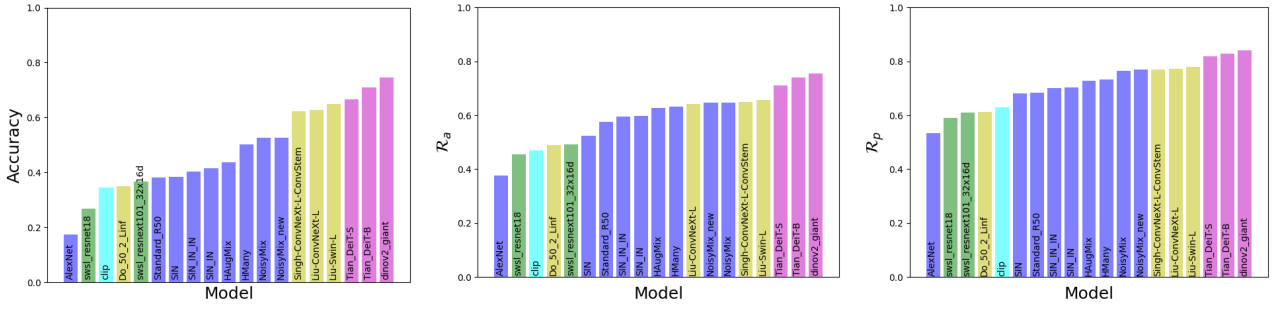
(a) IMAGENET-C Shot Noise Accuracy (b) Shot Noise $\hat{\mathcal{R}}_a$ (c) Shot Noise $\hat{\mathcal{R}}_p$

Figure 18. Comparison between IMAGENET-C and VCR with Shot Noise.



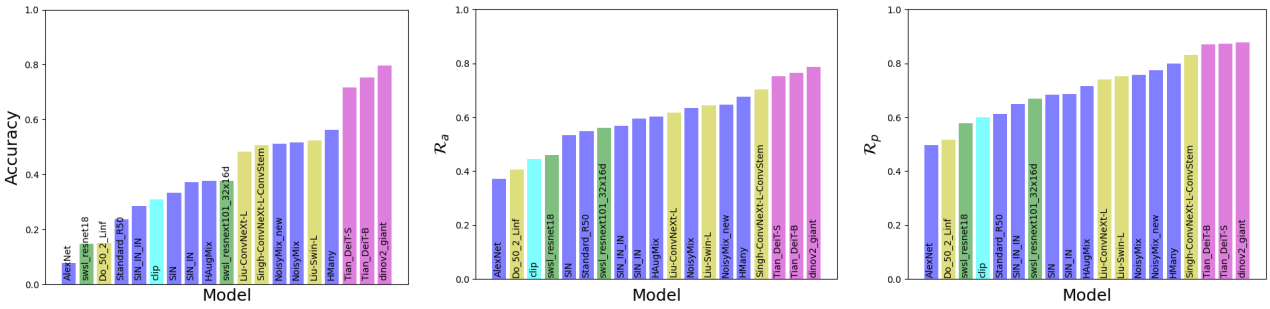
(a) IMAGENET-C Motion Blur Accuracy (b) Motion Blur $\hat{\mathcal{R}}_a$ (c) Motion Blur $\hat{\mathcal{R}}_p$

Figure 19. Comparison between IMAGENET-C and VCR with Motion Blur.



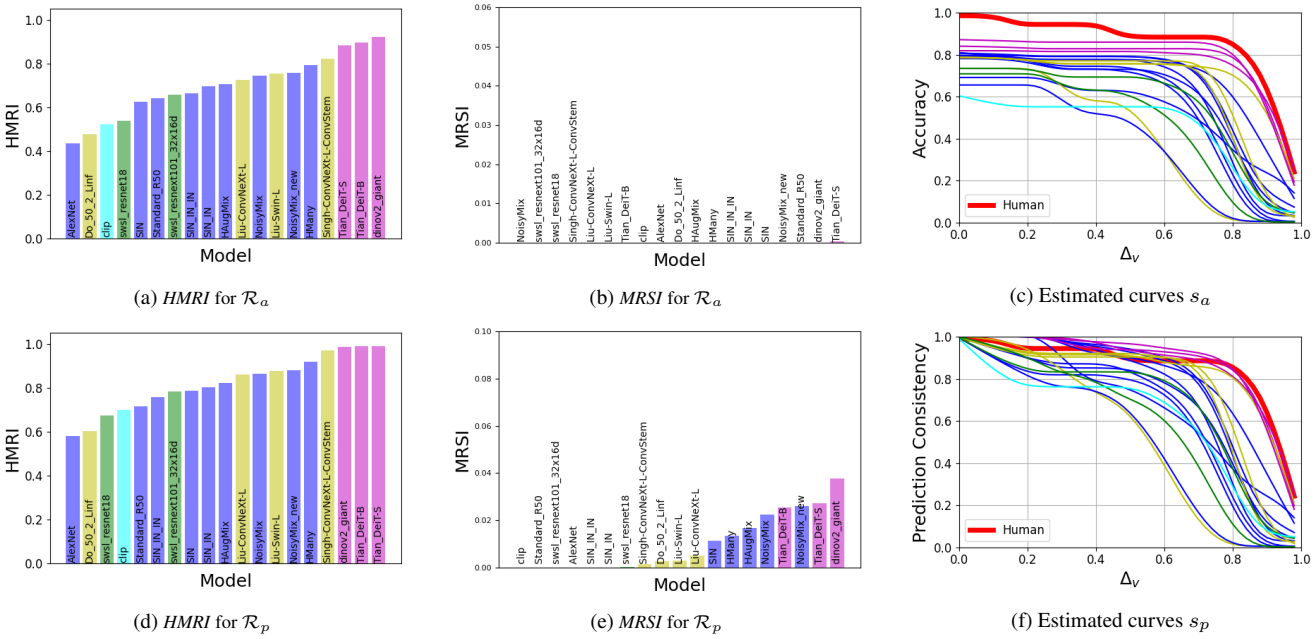
(a) IMAGENET-C Frost Accuracy (b) Frost $\hat{\mathcal{R}}_a$ (c) Frost $\hat{\mathcal{R}}_p$

Figure 20. Comparison between IMAGENET-C and VCR with Frost.



(a) IMAGENET-C Impulse Noise Accuracy (b) Impulse Noise $\hat{\mathcal{R}}_a$ (c) Impulse Noise $\hat{\mathcal{R}}_p$

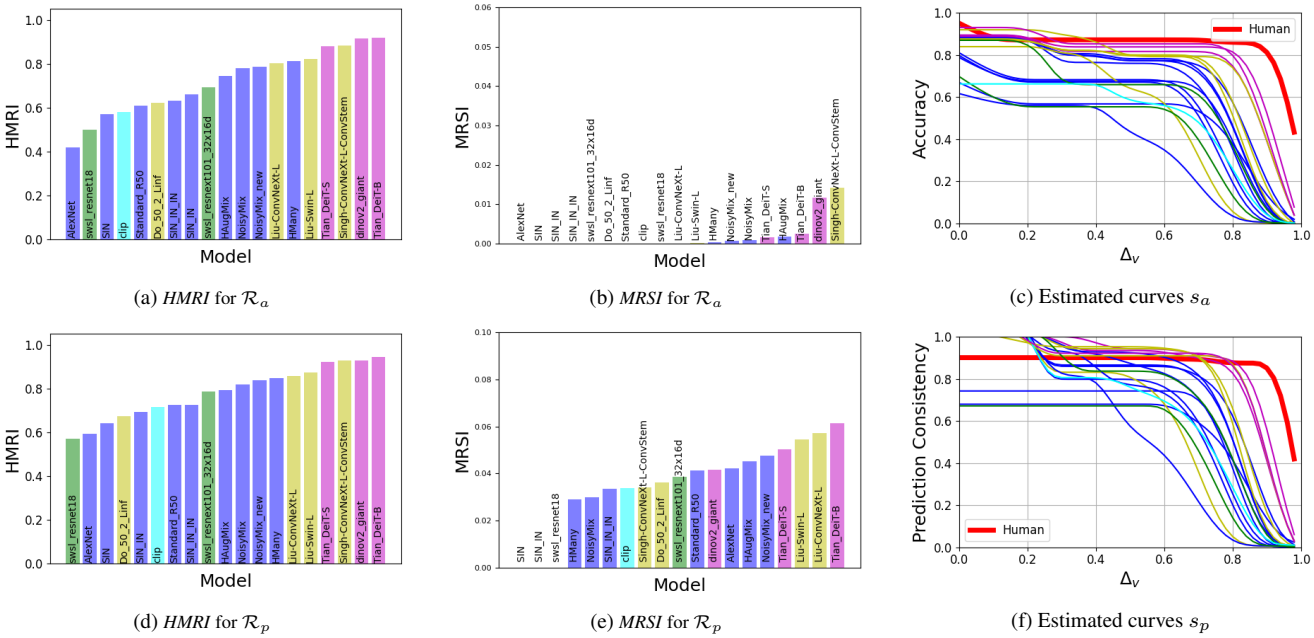
Figure 21. Comparison between IMAGENET-C and VCR with Impulse Noise.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

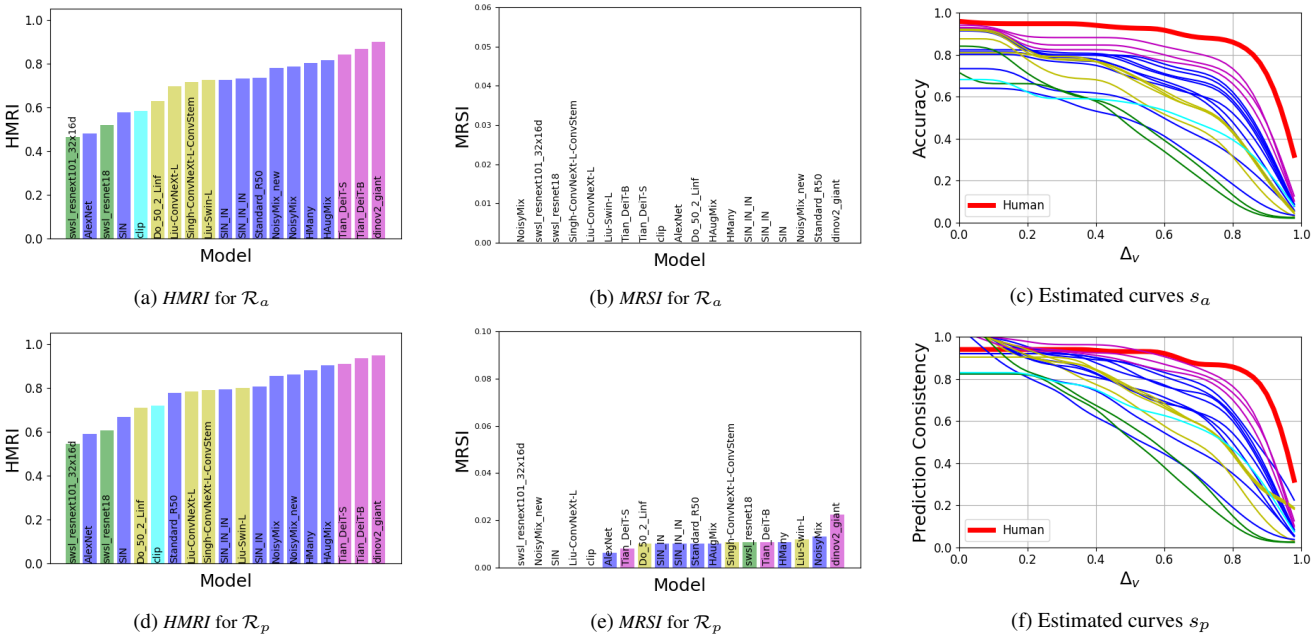
Figure 22. VCR evaluation results for Impulse Noise.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

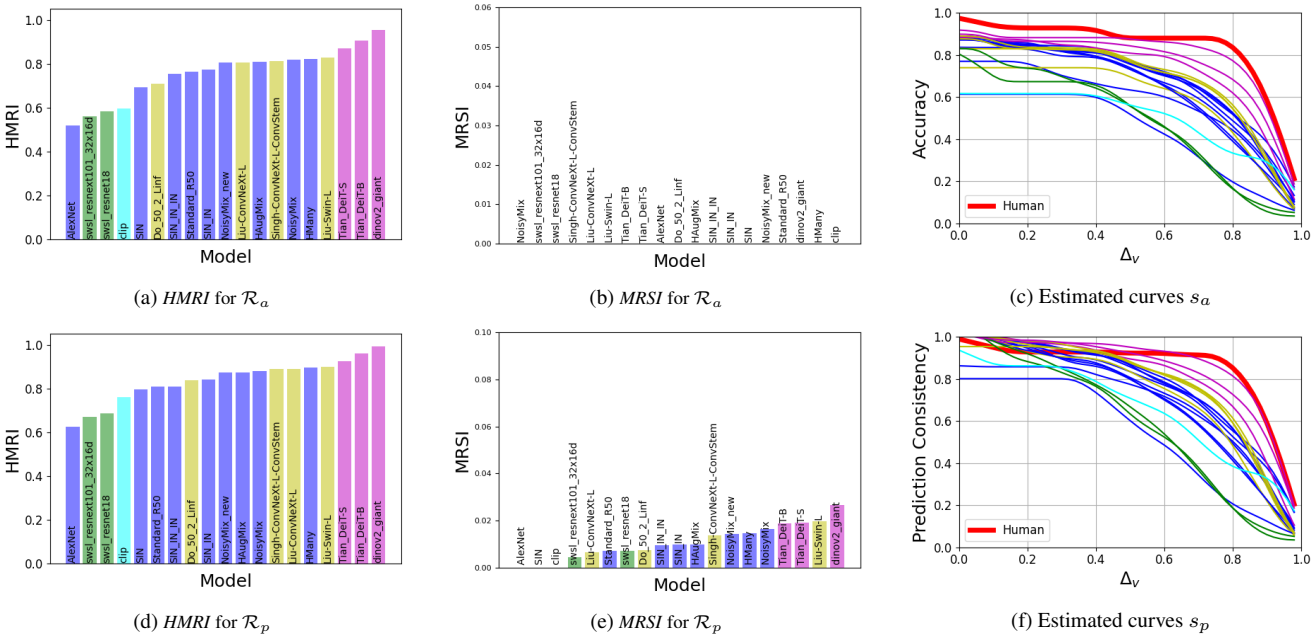
Figure 23. VCR evaluation results for Shot Noise.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

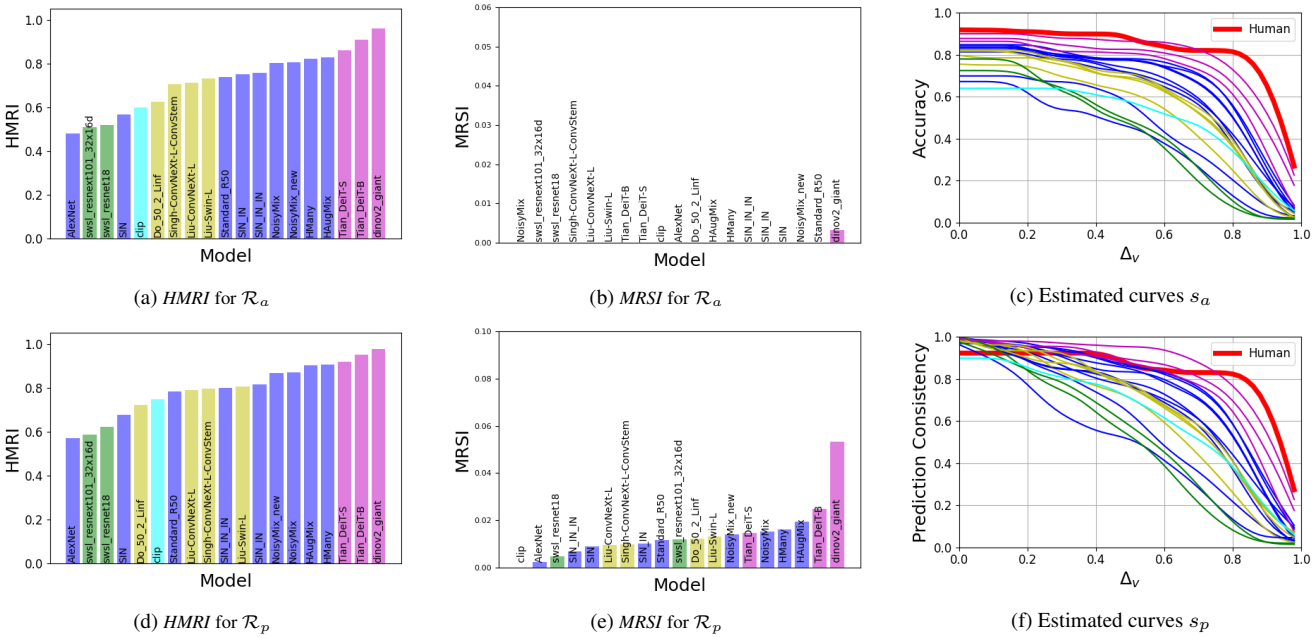
Figure 24. VCR evaluation results for Blur.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

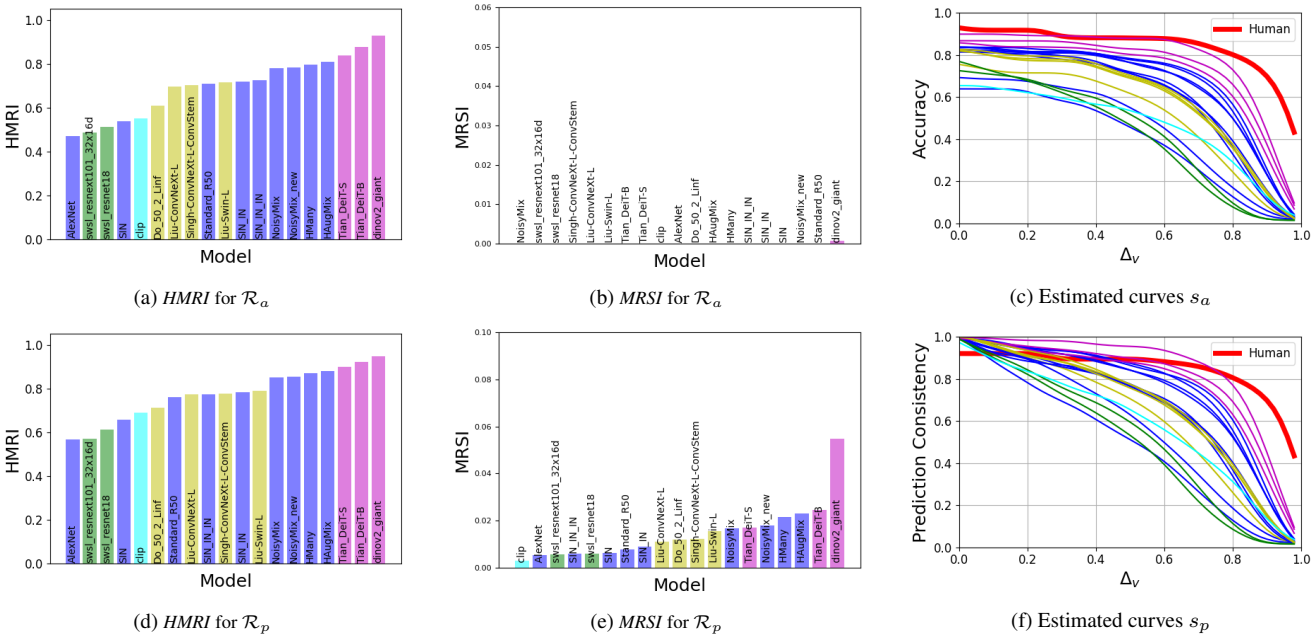
Figure 25. VCR evaluation results for Median Blur.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

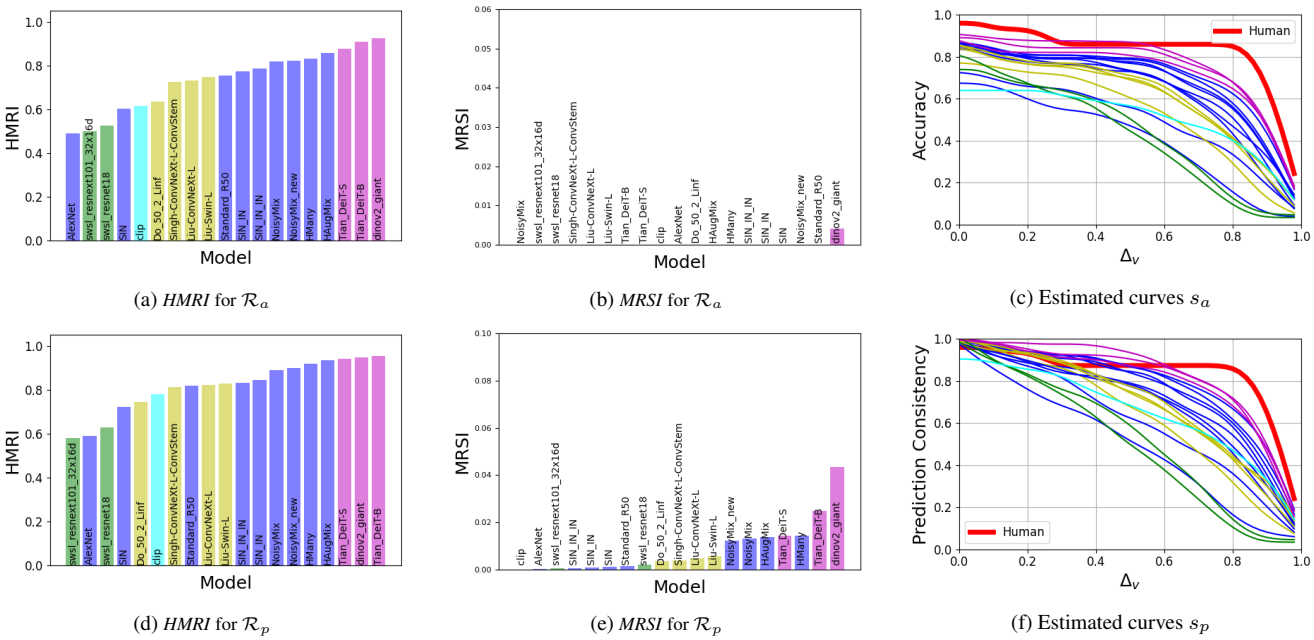
Figure 26. VCR evaluation results for Glass Blur.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

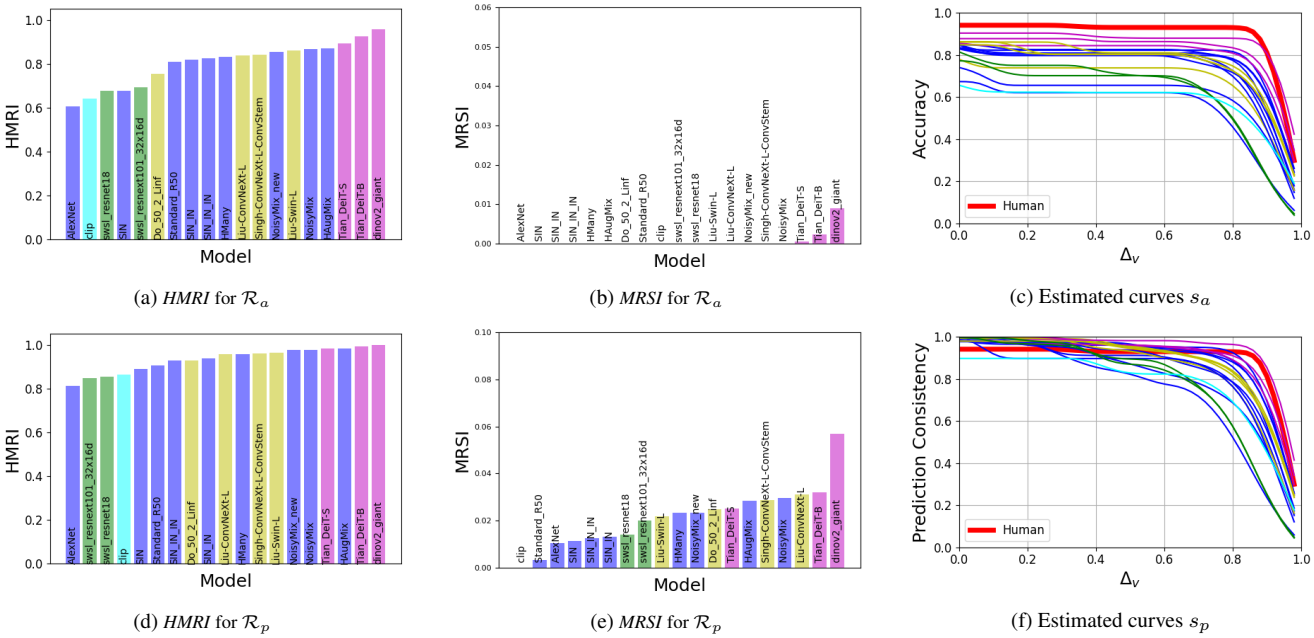
Figure 27. VCR evaluation results for Gaussian Blur.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

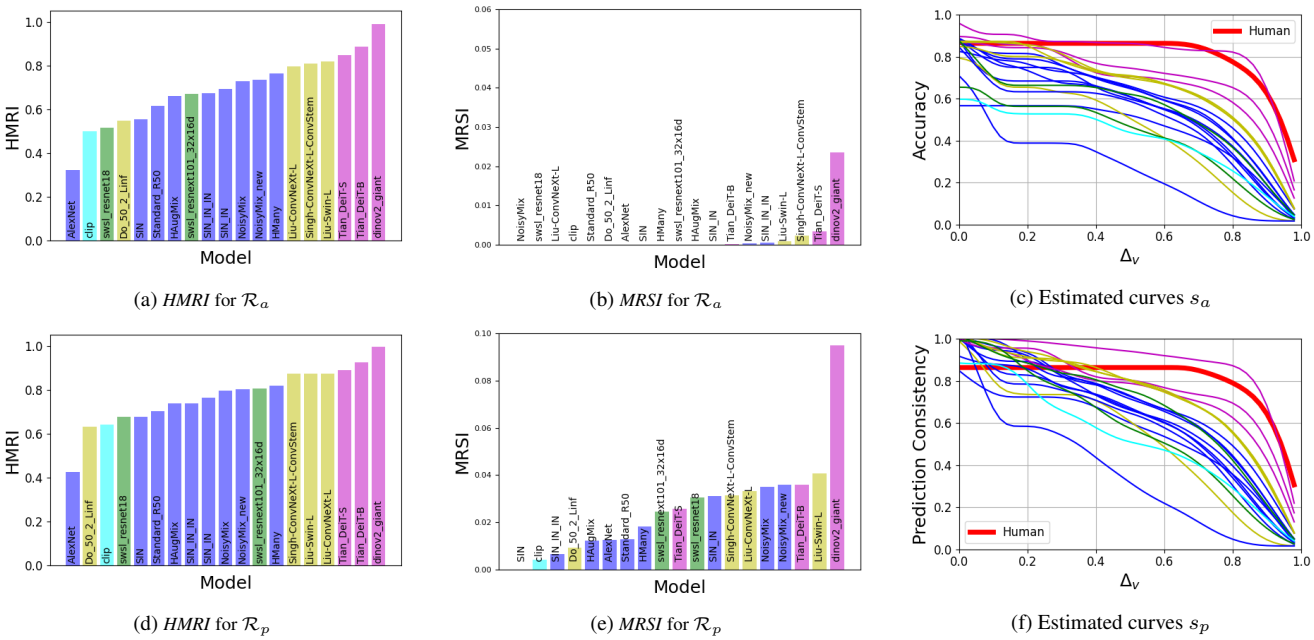
Figure 28. VCR evaluation results for Defocus Blur.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

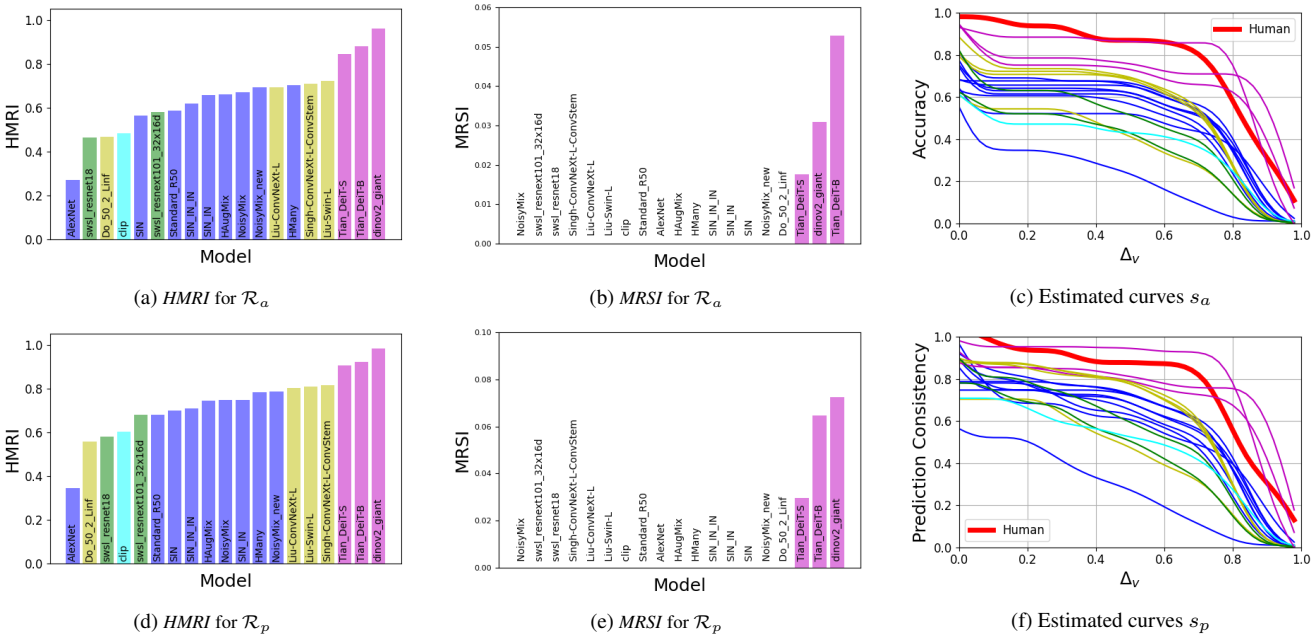
Figure 29. VCR evaluation results for Motion Blur.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

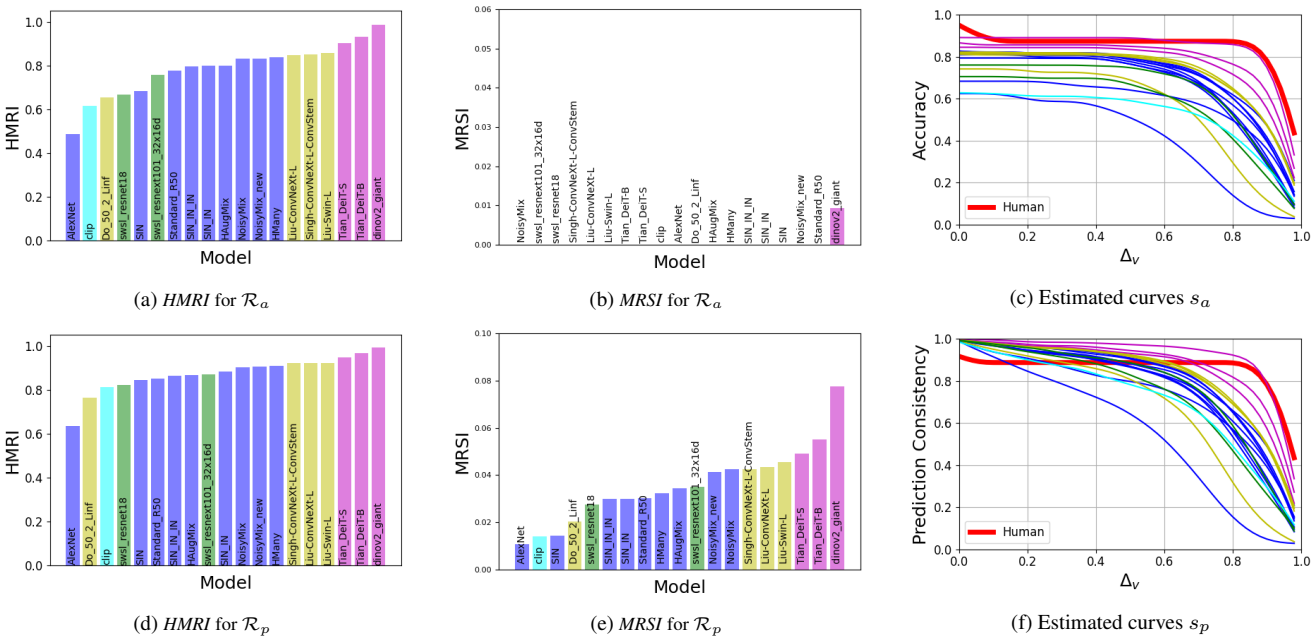
Figure 30. VCR evaluation results for Hue Saturation Value.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

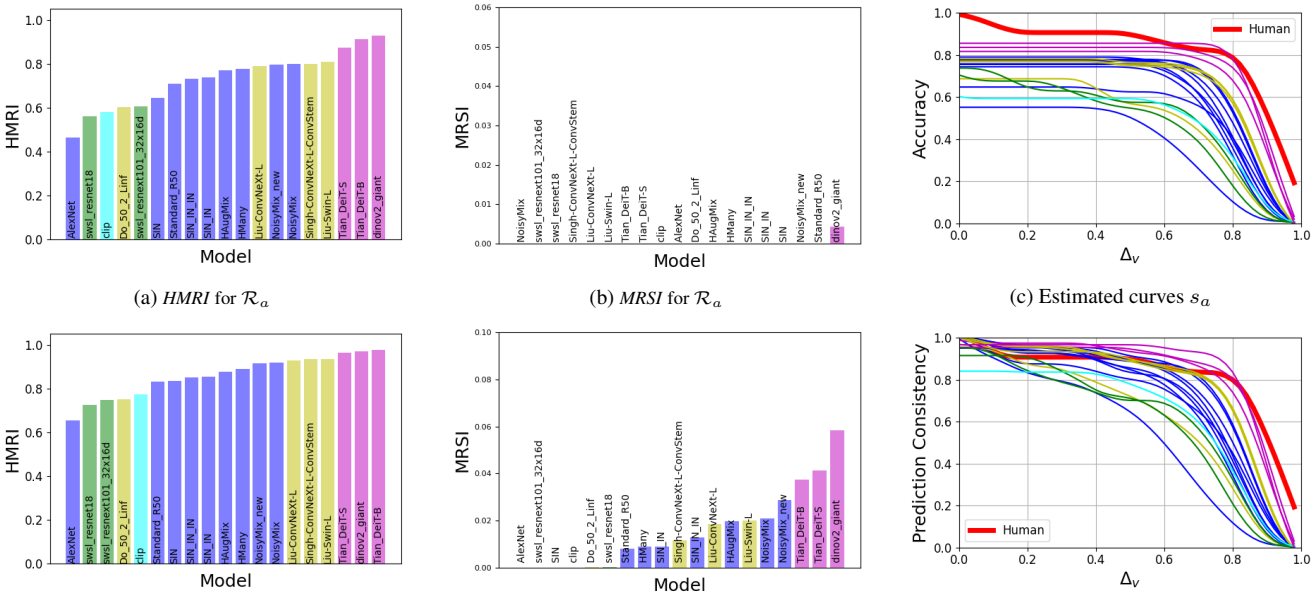
Figure 31. VCR evaluation results for Color Jitter.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

Figure 32. VCR evaluation results for Brightness.



(a) *HMRI* for $\mathcal{R}_a$ (b) *MRSI* for $\mathcal{R}_a$ (c) Estimated curves s_a

(d) *HMRI* for $\mathcal{R}_p$ (e) *MRSI* for $\mathcal{R}_p$ (f) Estimated curves s_p

Figure 33. VCR evaluation results for Frost.