A. Impact Statements

This work strengthens the reliability of vision foundation models by systematically evaluating their robustness to common perturbations and proposing fine-tuning strategies to enhance stability without sacrificing utility. By introducing principled robustness metrics and analyzing industry-scale models, our findings highlight vulnerabilities that can impact real-world applications. Addressing these weaknesses improves the robustness of AI systems in practical settings, ensuring consistent performance across diverse conditions. Our work provides a foundation for future research on enhancing model robustness, including extending robustness analyses to language models and defending against adversarial perturbations.

B. Proof of Theorem 2

Proof. Since $||f(x)||_2 = 1$ for any x, we have the following:

$$\begin{aligned}
&||f(P(x,k_1)) - f(P(x,k_2))||_2^2 \\
=&||f(P(x,k_1))||_2^2 + ||f(P(x,k_2))||_2^2 \\
&- 2f(P(x,k_1))^T \cdot f(P(x,k_2)) \\
=&2 - 2 \cdot cos(f(P(x,k_1)), f(P(x,k_2))),
\end{aligned} \tag{11}$$

where T represents transpose and $f(P(x,k_1))^T \cdot f(P(x,k_2))$ is the inner product between two embedding vectors. Therefore, we have:

$$\begin{aligned}
&\mathcal{R}_{ed}(f,x,P,\mathbb{K}) \\
=&\frac{\max_{k_1,k_2\in\mathbb{K}} ||f(P(x,k_1)) - f(P(x,k_2))||_2}{2} \\
=&\frac{\max_{k_1,k_2\in\mathbb{K}} \sqrt{2 - 2cos(f(P(x,k_1)), f(P(x,k_2)))}}{2} \\
=&\sqrt{\frac{1 - \min_{k_1,k_2\in\mathbb{K}} cos(f(P(x,k_1)), f(P(x,k_2)))}{2}} \\
=&\sqrt{\mathcal{R}_{cs}(f,x,P,\mathbb{K})}.
\end{aligned} \tag{12}$$

Therefore, by Theorem 1, the Euclidean distance-based metric $\mathcal{R}_{ed}$ does not satisfy the worst-robustness property. □

C. DivergenceRadius Satisfies the Mathematical Properties

Proof. **Bounded domain:** The radius of any ball is non-negative, and thus $\mathcal{R}_{dr}(f,x,P,\mathbb{K}) \geq 0$. Moreover, since all embedding vectors outputted by a foundation model lie on the unit hyper-sphere, the unit ball with radius 1 can enclose all embedding vectors. Therefore, we have $\mathcal{R}_{dr}(f,x,P,\mathbb{K}) \leq 1$. Thus, we have $\mathcal{R}_{dr}(f,x,P,\mathbb{K}) \in [0,1]$.

Monotonicity: Suppose the perturbation parameter domain $\mathbb{K}_1$ expands to $\mathbb{K}_2$. If the embedding vectors of the perturbed images corresponding to the expanded perturbation parameters (i.e., parameters in $\mathbb{K}_2$-$\mathbb{K}_1$) fall outside of the minimum enclosing ball for $\mathbb{K}_1$, the minimum enclosing ball for $\mathbb{K}_2$ expands to have a larger radius, i.e., $\mathcal{R}_{dr}(f,x,P,\mathbb{K}_1) < \mathcal{R}_{dr}(f,x,P,\mathbb{K}_2)$; otherwise, the minimum enclosing ball remains the same, i.e., $\mathcal{R}_{dr}(f,x,P,\mathbb{K}_1) = \mathcal{R}_{dr}(f,x,P,\mathbb{K}_2)$. Therefore, we have $\mathcal{R}_{dr}(f,x,P,\mathbb{K}_1) \leq \mathcal{R}_{dr}(f,x,P,\mathbb{K}_2)$ if $\mathbb{K}_1 \subseteq \mathbb{K}_2$, which satisfies the monotonicity property.

Best robustness: If all the perturbed versions of the image have the same embedding vector, the minimum enclosing ball has a radius 0. Thus, we have $\mathcal{R}_{dr}(f,x,P,\mathbb{K}) = 0$ in such case, achieving the best-robustness property.

Worst robustness: In the worst-robustness case, there exists a subdomain $\mathbb{K}' \subseteq \mathbb{K}$, where $\mathbb{K}'$ consists of discrete values and the corresponding embedding vectors are equally distributed in the embedding space, i.e., $\sum_{k\in\mathbb{K}'} f(P(x,k)) = 0$. Without loss of generality, we assume the subdomain $\mathbb{K}'$ contains n discrete values $k_1, \cdots, k_n$. Then, we have the following:

$$\sum_{i=1}^{n} f(P(x,k_i)) = \mathbf{0}. \tag{13}$$

Based on Equation 10, we have the following equation group:

$$\begin{cases}
||f(P(x,k_1))||_2^2 - 2f^T(P(x,k_1)) \cdot c + ||c||_2^2 \leq r^2 \\
||f(P(x,k_2))||_2^2 - 2f^T(P(x,k_2)) \cdot c + ||c||_2^2 \leq r^2 \\
\cdots \\
||f(P(x,k_n))||_2^2 - 2f^T(P(x,k_n)) \cdot c + ||c||_2^2 \leq r^2
\end{cases}, \tag{14}$$

where T indicates transpose of a vector. Since the embedding vectors lie on the unit hyper-sphere, we have $||f(P(x,k_i))||_2^2 = 1$ for $i = 1, \cdots, n$. After summing up the n inequalities in the equation group 14, we have the following:

$$\begin{aligned}
&n \cdot 1 - 2(\sum_{i=1}^{n} f^T(P(x,k_i))) \cdot c + n||c||_2^2 \leq nr^2, \\
\Leftrightarrow &r^2 \geq 1 + ||c||_2^2 \geq 1,
\end{aligned} \tag{15}$$

where $r = 1$ when $c = \mathbf{0}$. Since $\mathcal{R}_{dr}(f,x,P,\mathbb{K})$ is the smallest r, we have $\mathcal{R}_{dr}(f,x,P,\mathbb{K}) = 1$, achieving the worst-robustness property.

Rotational invariance: Suppose the rotation matrix is M. We observe that $||M \cdot f(P(x,k)) - M \cdot c||_2 = ||M \cdot (f(P(x,k)) - c)||_2 = ||f(P(x,k)) - c||_2 \leq \mathcal{R}_{dr}(f,x,P,\mathbb{K})$ for any $k \in \mathbb{K}$. Therefore, we have $\mathcal{R}_{dr}(M \cdot f,x,P,\mathbb{K}) \leq \mathcal{R}_{dr}(f,x,P,\mathbb{K})$. We denote the inverse matrix of the rotation matrix as M^{-1}. We have $\mathcal{R}_{dr}(f,x,P,\mathbb{K}) = \mathcal{R}_{dr}(M^{-1} \cdot M \cdot f,x,P,\mathbb{K}) \leq \mathcal{R}_{dr}(M \cdot f,x,P,\mathbb{K})$. Thus, we have $\mathcal{R}_{dr}(M \cdot f,x,P,\mathbb{K}) = \mathcal{R}_{dr}(f,x,P,\mathbb{K})$, achieving

Table 4. Benchmark datasets.

Dataset	ImageNet	Food101	NYU-Depth V2
#Training images	1,281,167	75,750	50,688
#Testing images	50,000	25,250	654
#Classes	1,000	101	-

the rotation-invariance property. Essentially, the embedding vectors are enclosed in a minimum ball with center c and radius $\mathcal{R}_{dr}(f, x, P, \mathbb{K})$ before rotation, and the embedding vectors are enclosed in a minimum ball with center Mc and the same radius after rotation. □

D. Details of Zero-shot Classification, Linear-probe Classification, and Depth Estimation

- **Zero-shot classification:** Jointly pre-trained image and text models like CLIP can perform zero-shot classification without labeled training data. Given an image x and a set of text labels $\mathcal{Y}$, CLIP represents x as an image embedding and the labels in $\mathcal{Y}$ as text embeddings, selecting the label most similar to x's embedding. For instance, CLIP's ViT-L/14 achieves 68.38% accuracy on zero-shot ImageNet classification.
- **Linear-probe classification:** Vision foundation models can be adapted for linear-probe classification using labeled data to fine-tune a simple classification head. With foundation model parameters frozen, a feed-forward classification head is added and trained on labeled data. This approach achieves high accuracy with minimal additional parameters; for example, DINO-v2 reaches 86.6% accuracy on ImageNet with a single-layer classification head.
- **Depth estimation:** Vision foundation models can also support depth estimation, predicting per-pixel depth in an image. A convolutional decoder head (with batch normalization, a 1×1 convolution, ReLU activation, and a sigmoid function) is appended to the frozen foundation model and fine-tuned on NYU-Depth V2 training data. For example, DINO-v2 achieves a root mean squared error of 0.35 on NYU-Depth V2.

E. Evaluation Metrics $RMSE$ and $RMSE_p$

Given a downstream depth estimation model $g \circ f$, where f is a foundation model and g is a depth estimation head built on top of f. The *root mean squared error* (denoted as $RMSE$) for a testing dataset is the average root mean squared error between model's predicted depth map and the ground-truth depth map. We evaluate common perturbations to images, and thus we also consider *root mean squared error under perturbation* (denoted as $RMSE_p$) for each testing image. Specifically, given a testing image x with ground-truth depth map y, we evaluate the root mean squared error of a downstream depth estimation model $g \circ f$ for x under a perturbation function P. Without loss of generality, we assume the perturbation parameter domain $\mathbb{K}$ contains m discrete values, where m discrete values can be sampled via random sampling or equally-spaced sampling if $\mathbb{K}$ contains more than m discrete values or is continuous. Formally, the $RMSE$ of the downstream classifier $g \circ f$ for x under a perturbation function P with a perturbation parameter domain $\mathbb{K}$ is defined as below:

$$\begin{aligned} &RMSE_p(x, g, f, P, \mathbb{K}) \\ =&\frac{1}{m}\sum_{k\in\mathbb{K}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - g\circ f(P(x,k))_i)^2}, \end{aligned} \tag{16}$$

where $g \circ f(P(x, k))$ is the predicted depth map for image $P(x, k)$ and i is the index of each pixel in a depth map. Intuitively, the smaller $RMSE$ indicates the more accurate depth estimation.

F. Parameter Settings and Experimental Results of Depth Estimation

Parameter settings: We consider the downstream depth estimation model that achieves the lowest root mean squared error. Specifically, we use DINO v2 ViT-g/14 foundation model and the one-layer depth estimation head publicly released together with DINO v2 for NYU-Depth V2.

Root mean squared error $RMSE$ vs. root mean squared error under perturbation $RMSE_p$: Table 6 shows the $RMSE$ and average $RMSE_p$ under different perturbation functions of NYU-Depth V2 testing images for depth estimation model. First, we find that common perturbations degrade RMSE of downstream depth estimation models, i.e., average $RMSE_p$ is larger than $RMSE$. For instance, Frost blurring increases the $RMSE$ of depth estimation by 0.12.

Second, the increased $RMSE$ caused by a perturbation function is aligned with the average DivergenceRadius under the perturbation function. For example, in Figure 15, DINO v2 ViT-g/14 have the largest average DivergenceRadius under Frost blurring compared to other perturbation functions. Correspondingly, in Table 6, average $RMSE_p$ under Frost blurring increases the most (0.12). This occurs because a higher average DivergenceRadius indicates more diversity in the embedding vectors of perturbed images, making it more likely that the downstream depth estimation models predict inaccurate depth maps for them.

Root mean squared error under perturbation $RMSE_p$ vs. robustness value: Figure 8 shows the relationship between an image's $RMSE_p$ under a perturbation function and the image's corresponding robustness value (cosine similarity or DivergenceRadius) for the NYU-Depth V2 testing

Table 5. Foundation models.

Foundation Model	Model Family	Pre-training Algorithm	Architecture	# Parameters (M)
CLIP ViT-B/16	CLIP	Multi-modal self-supervised learning	Vision Transformer	86
CLIP ViT-L/14	CLIP	Multi-modal self-supervised learning	Vision Transformer	304
CLIP RN50	CLIP	Multi-modal self-supervised learning	ResNet	38
CLIP RN50×64	CLIP	Multi-modal self-supervised learning	ResNet	420
DINO v2 ViT-L/14	DINO v2	Self-supervised learning	Vision Transformer	304
DINO v2 ViT-g/14	DINO v2	Self-supervised learning	Vision Transformer	1136

Table 6. $RMSE$ and average $RMSE_p$ of NYU-Depth V2's testing images for depth estimation when using DINO v2 ViT-g/14 foundation model.

$RMSE$		0.35
$RMSE_p$	JPEG compression	0.37 (↑ 0.02)
	Brightness adjustment	0.36 (↑ 0.01)
	Contrast adjustment	0.35 (↑ 0.00)
	Defocus blurring	0.39 (↑ 0.04)
	Elastic blurring	0.38 (↑ 0.03)
	Fog blurring	0.38 (↑ 0.03)
	Frost blurring	0.47 (↑ 0.12)
	Gaussian noise	0.37 (↑ 0.02)
	Glass blurring	0.38 (↑ 0.03)

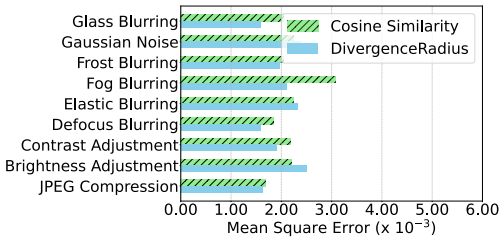
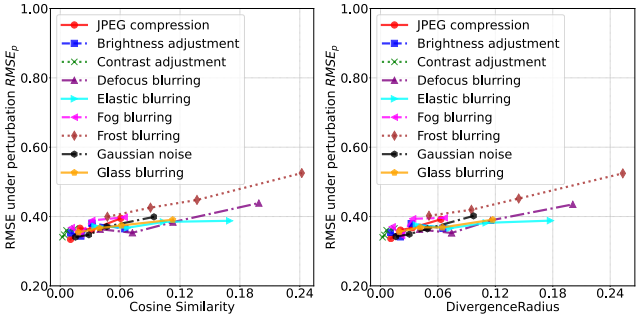


Figure 9. Mean squared error of predicting an image's $RMSE_p$ using its cosine similarity or DivergenceRadius for NYU-Depth V2 under different perturbation functions.

images and 9 perturbation functions. Specifically, given a perturbation function, for each testing image, we compute its robustness value and its $RMSE_p$ of a downstream depth estimation model, i.e., we obtain a pair (robustness value, $RMSE_p$). Then, we rank the pairs of all testing images in an increasing order according to the robustness values and divide the ranked pairs into 4 groups equally. For each group of pairs, we calculate the mean robustness value and mean $RMSE_p$, which are shown in Figure 8. Across perturbation functions, we find that $RMSE_p$ roughly increases *linearly* when the robustness metrics increases. $RMSE_p$ increases as the robustness value increases because a larger robustness value indicates more diverse embedding vectors for the perturbed images, leading to less accurate depth estimation.



(a) Zero-shot classification (b) Linear-probe classification

Figure 8. Root mean squared error under perturbation $RMSE_p$ vs. (a) cosine similarity, (b) DivergenceRadius of NYUd testing images for depth estimation when different perturbation functions are used. We use the DINO v2 ViT-g/14 foundation model.

Predicting $RMSE_p$ of an image using its robustness value: The linear relationship between $RMSE_p$ and the robustness value indicates that we can predict $RMSE_p$ of an image using its robustness value by a linear regression model. We evaluate the performance of such prediction. Towards this goal, we divide the testing images of a dataset into two halves. We use the pairs (robustness value, $RMSE_p$) of the first half of testing images to train a linear regression model, which takes a robustness value as input and outputs $RMSE_p$. Then, we evaluate this linear regression model on the pairs (DivergenceRadius, $RMSE_p$) of the second half of the testing images. Figure 9 shows the mean squared errors of the linear regression models under the 9 perturbation functions for the depth estimation model. The mean squared errors are very small, which indicates that an image's DivergenceRadius under a perturbation function can be used to accurately predict a downstream depth estimation model's $RMSE$ for the image when it is perturbed by the perturbation function.

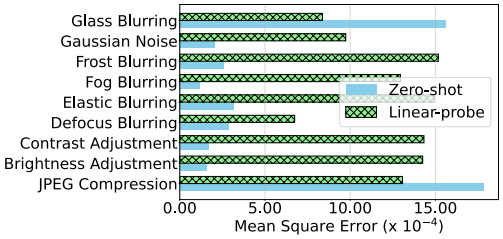
Table 7. Pearson correlations between average ACC-average ACC_p and average cosine similarity or DivergenceRadius across the 9 perturbation functions for the two datasets and downstream classifiers.

(a) Cosine similarity

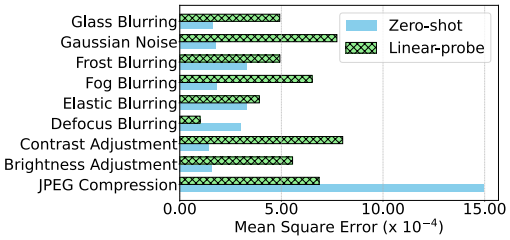
	Zero-shot Classification	Linear-probe Classification
ImageNet	0.91	0.89
Food101	0.94	0.92

(b) DivergenceRadius

	Zero-shot Classification	Linear-probe Classification
ImageNet	0.92	0.89
Food101	0.94	0.93

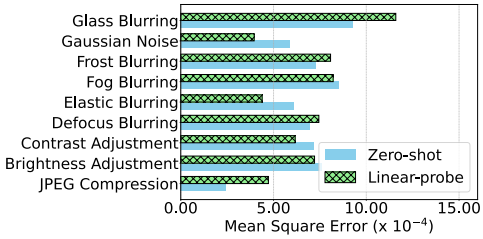


(a) Cosine similarity

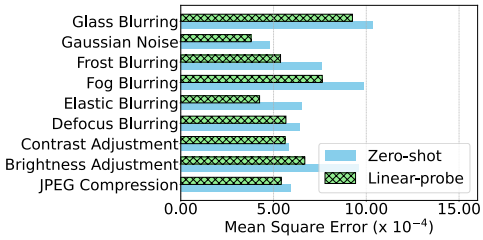


(b) DivergenceRadius

Figure 10. Mean squared error of predicting an image's ACC_p using its (a) cosine similarity or (b) DivergenceRadius for ImageNet under different perturbation functions in two downstream classifiers.

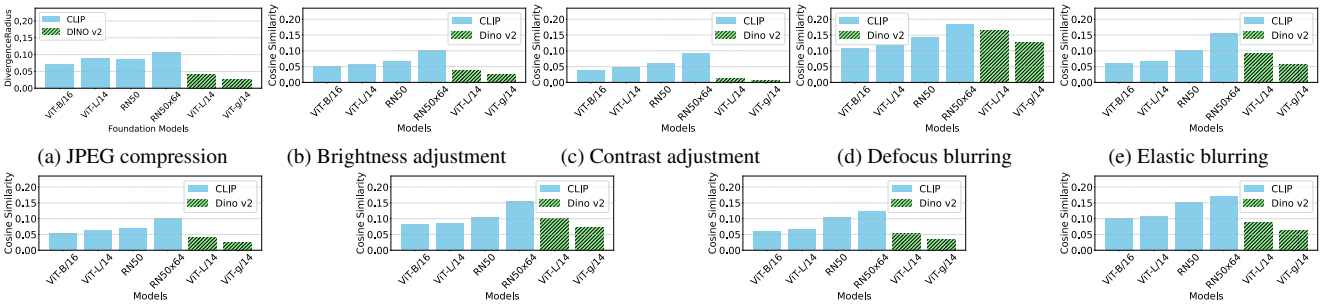


(a) Cosine similarity



(b) DivergenceRadius

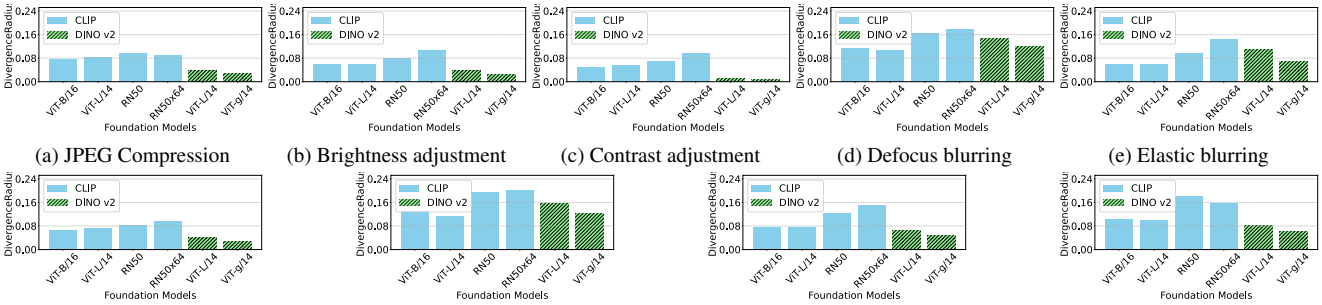
Figure 11. Mean squared error of predicting an image's ACC_p using its (a) cosine similarity or (b) DivergenceRadius for Food101 under different perturbation functions in two downstream classifiers.



(a) JPEG compression (b) Brightness adjustment (c) Contrast adjustment (d) Defocus blurring (e) Elastic blurring

(f) Fog blurring (g) Frost blurring (h) Gaussian noise (i) Glass blurring

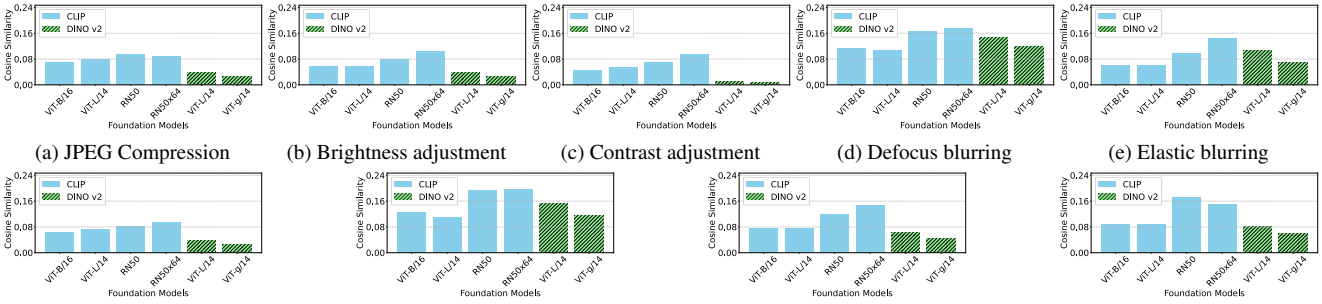
Figure 12. Average cosine similarity of ImageNet testing images for different foundation models and perturbation functions.



(a) JPEG Compression (b) Brightness adjustment (c) Contrast adjustment (d) Defocus blurring (e) Elastic blurring

(f) Fog blurring (g) Frost blurring (h) Gaussian blurring (i) Glass blurring

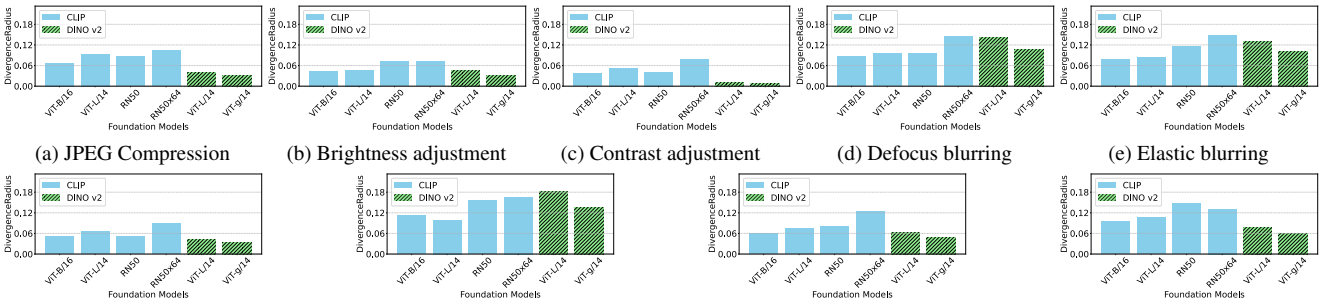
Figure 13. Average DivergenceRadius of Food101 testing images for different foundation models and perturbation functions.



(a) JPEG Compression (b) Brightness adjustment (c) Contrast adjustment (d) Defocus blurring (e) Elastic blurring

(f) Fog blurring (g) Frost blurring (h) Gaussian blurring (i) Glass blurring

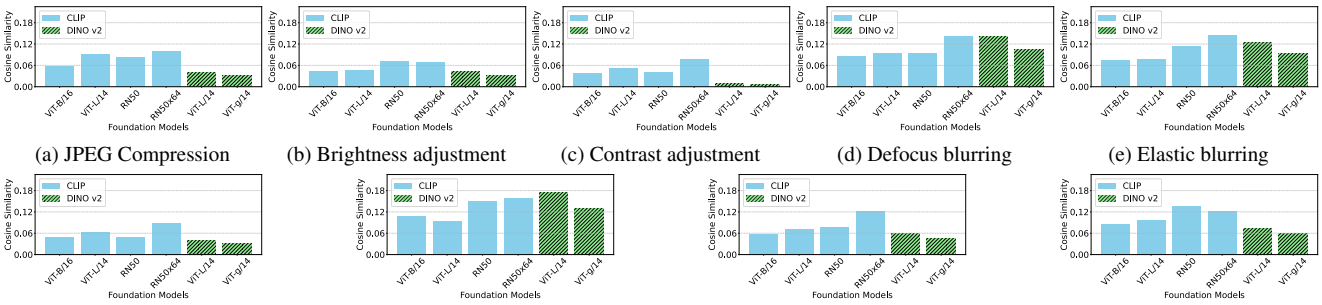
Figure 14. Average cosine similarity of Food101 testing images for different foundation models and perturbation functions.



(a) JPEG Compression (b) Brightness adjustment (c) Contrast adjustment (d) Defocus blurring (e) Elastic blurring

(f) Fog blurring (g) Frost blurring (h) Gaussian blurring (i) Glass blurring

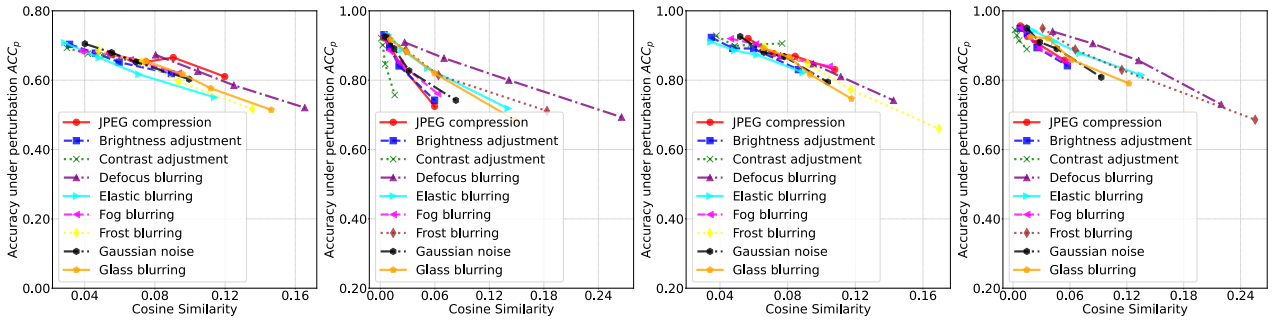
Figure 15. Average DivergenceRadius of NYU-Depth V2 testing images for different foundation models and perturbation functions.



(a) JPEG Compression (b) Brightness adjustment (c) Contrast adjustment (d) Defocus blurring (e) Elastic blurring

(f) Fog blurring (g) Frost blurring (h) Gaussian blurring (i) Glass blurring

Figure 16. Average cosine similarity of images in NYU-Depth V2 for different foundation models and perturbation functions.



(a) ImageNet: Zero-shot classification (b) ImageNet: Linear-probe classification (c) Food101: Zero-shot classification (d) Food101: Linear-probe classification

Figure 17. Accuracy under perturbation ACC_p vs. cosine similarity of ImageNet and Food101 testing images for zero-shot classification and linear-probe classification when different perturbation functions are used. Zero-shot classification is based on the CLIP ViT-L/14 foundation model and linear-probe classification is based on the DINO v2 ViT-g/14 foundation model.

Table 8. Details of perturbation functions.

Perturbation Function	Key Perturbation Parameter $\mathbb{K}$	Domain $\mathbb{K}$	Maximally Distorted Example
JPEG Compression	Quality factor	$[30, 70]$	
Brightness Adjustment	Value in Hue-Saturation -Value space	$[0.1, 0.5]$	
Contrast Adjustment	Amplifying factor of deviations from the mean	$[0.3, 0.7]$	
Defocus Blurring	Bluring disk kernel	$[1, 5]$	
Elastic Blurring	Scaling factor	$[0.01, 0.05]$	
Fog Blurring	Density of fog	$[0.5, 2.5]$	
Frost Blurring	Weight of the frost	$[0.2, 0.6]$	
Gaussian Noise	Standard deviation	$[0.02, 0.10]$	
Glass Blurring	Standard deviation	$[0.2, 1.0]$	

Table 9. Impact of λ on average DivergenceRadius and ACC of zero-shot classification for two datasets before/after enhancement, where the foundation model is CLIP ViT-L/14.

λ	DivergenceRadius	ACC (%)
0.0	0.01	0.16
0.5	0.05	66.61
1.0	0.06	67.99
5.0	0.07	68.31