Suppress and Diversify: Refining Robust Pathways for Corruption Robustness

Jiangang Yang [*1] **Wenhui Shi** [*1 2] **Xiaoran Xu** [2] **Wenyue Chong** [2] **Luqing Luo** [1] **Jing Xing** [1] **Jian Liu** [1]

Abstract

Model robustness against natural image corruptions is essential for safety-critical applications. While existing methods primarily focus on implicit representation learning, we provide the first systematic exploration of computational pathways to explicitly characterize internal robustness. We identify a progressive decay of robust features across network layers and establish a functional dependency between the prevalence of these features and model performance. To exploit these insights, we propose Suppress and Diversify (S&D), a non-intrusive refinement approach that enhances robustness by dynamically selecting robust pathways and diversifying them through symmetry-preserving transformations. S&D is architecture-agnostic, parameter-free, and incurs zero test-time overhead. Extensive evaluations across eight benchmarks demonstrate that S&D consistently improves performance across multiple vision tasks, diverse backbones, and complex real-world scenarios, highlighting its broad efficacy and scalability.

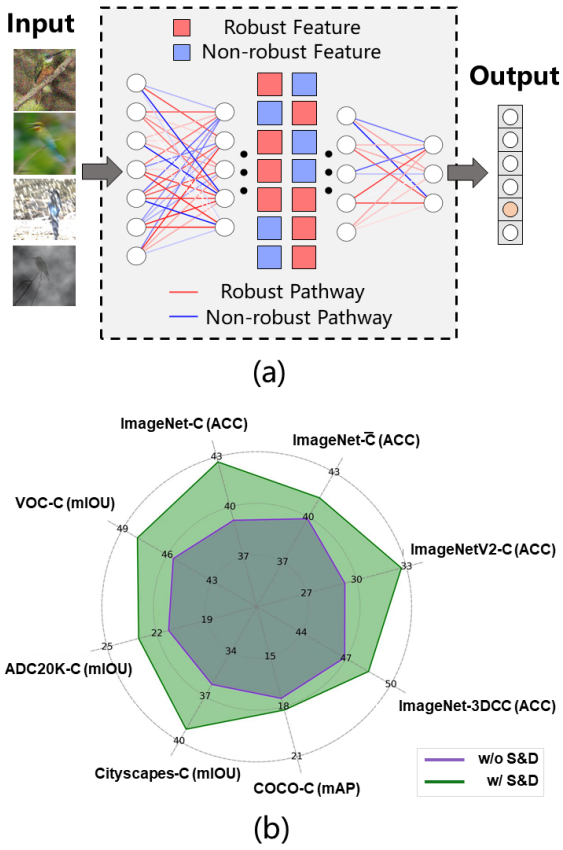


Figure 1. (a) The phenomenon of robust (non-robust) features for image corruptions . (b) The performance gains of the proposed approach across various computer vision tasks.

1. Introduction

Recent years have witnessed the success of neural networks in various computer vision tasks. However, recent works have revealed that neural networks are vulnerable to image corruptions (Hendrycks & Dietterich, 2019): image changes (e.g., noise, blur, or low lighting) that degrade visual quality can lead neural networks to make errors such as misclassifying a bicycle as a motorbike. Such vulnerability inevitably transfers to downstream 2D and 3D perception tasks (Michaelis et al., 2019; Kamann & Rother, 2021; Dong et al., 2023; Zeng et al., 2024), raising serious security concerns in safety-critical applications.

To solve this problem, prior approaches primarily focus on implicitly learning robust representations through data augmentation (Hendrycks et al., 2019; Modas et al., 2022; Qin et al., 2022), model regularization (Zhang et al., 2018; Guo et al., 2023), ensemble learning (Saikia et al., 2021; Diffenderfer et al., 2021), and biological principles (Teti et al., 2022; Dapello et al., 2020). However, explicit modeling of robust features under image corruptions remains largely under-explored. While adversarial robustness research has extensively studied feature disentanglement (Ilyas et al., 2019; Tao et al., 2022; Kim et al., 2023; Wang et al., 2024), these findings may not directly apply to natural corruptions, which exhibit global structural patterns and semantic shifts

[1]Institute of Microelectronics, Chinese Academy of Sciences, Beijing, China [2]University of Chinese Academy of Science, Beijing, China. Correspondence to: Jiangang Yang <yangjiangang@ime.ac.cn>, Jian Liu <liujian@ime.ac.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

rather than local pixel-level noise. The nature of features under such structured distortions remains poorly understood, raising several critical questions: (a) *Do neural networks similarly exhibit distinct non-robust features under image corruptions?* (b) *Can we explicitly manipulate robust (non-robust) features to enhance corruption robustness?*

To address these questions, we provide a systematic analysis to verify the existence and impact of non-robust features. We first formalize the robustness of computational pathways via the local Lipschitz constant (Cisse et al., 2017; Fazlyab et al., 2019), revealing that robust and non-robust features are prevalent across the network. Our investigation identifies a progressive decay of robust features along the network depth, suggesting a non-homogeneous evolution of internal robustness. Furthermore, through linear probing and loss landscape analysis, we establish a functional dependency between robust feature prevalence and overall performance. These findings demonstrate that the aggregation of robust features dictates the robustness of downstream sub-networks and the full model under corruptions.

Building on these insights, we propose Suppress and Diversify (S&D), a non-intrusive refinement approach. The core principles are: (1) selecting pathways that generate robust features, and (2) diversifying them. Specifically, we introduce a Memory-aware Dynamic Selection Mechanism (MDSM) that maintains pathway groups in a memory bank, dynamically updated by a fitness score balancing feature invariance and discriminability. To prevent overfitting, a Structure-consistent Path Tweaking Strategy (SPTS) subsequently filters these pathways and generates augmented counterparts via symmetry-preserving transformations. This strategy enhances the diversity of robust representations while adhering to structural invariants. Finally, S&D ensures the network is guided by a diverse set of robust computational pathways, significantly improving performance under corruptions with zero test-time overhead.

S&D is a lightweight refinement requiring no inference-time overhead or structural modifications. We validate its efficacy across diverse benchmarks for classification, detection, and segmentation. S&D consistently enhances the intrinsic robustness of various architectures and provides orthogonal gains when integrated with state-of-the-art methods. Our contributions are summarized as follows:

- We provide the first systematic exploration of non-robust features under image corruptions, revealing how the prevalence of these features determines the robustness across sub-networks and the overall model.
- We introduce S&D, a novel training-time refinement that suppresses non-robust pathways and diversifies robust ones. S&D is architecture-agnostic and parameter-free, enabling plug-and-play integration.
- Extensive evaluations across diverse benchmarks and vision tasks demonstrate that S&D consistently enhances corruption robustness and OOD generalization.

2. Related Work

2.1. Understanding Corruption Robustness

There has been growing interest in understanding corruption robustness. Early work compares human and neural-network robustness under image corruptions (Geirhos et al., 2018b). A major line of research explains robustness through model bias, especially texture/shape bias: texture bias plays a key role in corruption performance (Geirhos et al., 2018a). Complementary perspectives analyze robustness in the Fourier domain by separating low- and high-frequency effects (Yin et al., 2019), and connect robustness with decision-boundary thickness (Yang et al., 2020). Large-scale evaluations show that several bias-based assumptions do not hold for network robustness (Gavrikov & Keuper, 2024). Closest to our focus, recent work implicitly models patch-induced non-robust features in Vision Transformers and shows that models exploit these patterns (Qin et al., 2022; Guo et al., 2023). However, no prior work explicitly models robust or non-robust features as a direct lens for understanding corruption robustness.

2.2. Enhancing Corruption Robustness

Enhancing corruption robustness is a major research focus, with benchmarks spanning multiple vision tasks (Hendrycks & Dietterich, 2019; Michaelis et al., 2019; Kamann & Rother, 2021). A dominant direction is data augmentation, which consistently improves robustness (Geirhos et al., 2018a; Yin et al., 2019; Guo et al., 2023). AugMix (Hendrycks et al., 2019) mixes diverse image operations, and later methods extend mixing to style (Hendrycks et al., 2021; Zhou et al., 2024), spatial (Modas et al., 2022), and frequency domains (Yucel et al., 2023; Vaish et al., 2024). Other lines combine augmentations via ensembles (Saikia et al., 2021; Diffenderfer et al., 2021) or propose training recipes coupling augmentation with regularization (Wightman et al., 2021; Touvron et al., 2021). Beyond training pipelines, bio-inspired architectures and learning paradigms are explored (Dapello et al., 2020; Teti et al., 2022; Li et al., 2019; Safarani et al., 2021). Fine-tuning strategies also target corruption-robust sub-networks (Lee et al., 2022; Guo et al., 2022); among them, EWS (Guo et al., 2022), closely related to our approach, uses knowledge distillation with a model-specific controller to refine non-robust sub-networks. In contrast, our method requires no additional model structures and can be seamlessly integrated into existing deep-learning frameworks.

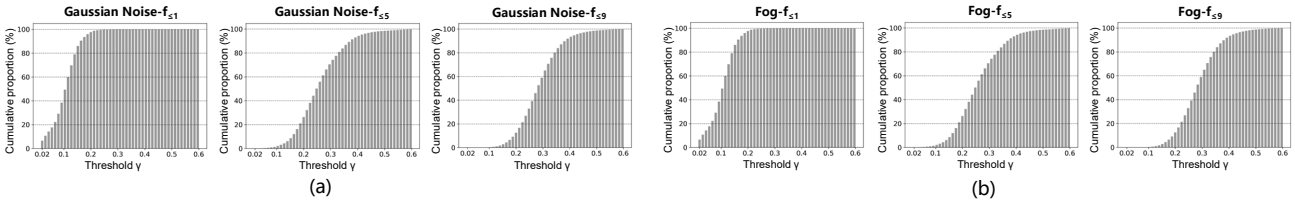


Figure 2. The cumulative distribution of γ-robust features at different sub-networks under the corruption of (a) Gaussian Noise and (b) Fog. The consistent shift toward higher thresholds in deeper layers indicates a progressive scarcity of robust features as the network depth increases, revealing that high-level representations are more sensitive to structured corruptions.

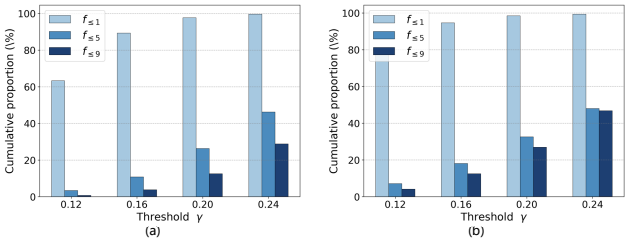


Figure 3. Robust feature proportions across thresholds γ under (a) Gaussian Noise and (b) Fog. The downward trend across stages ($f_{\leq 1} \rightarrow f_{\leq 9}$) reveals systemic robustness decay.

3. Methodology

3.1. Problem Formulation

We consider the robustness of a neural network f_θ under natural image corruptions. Formally, let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a clean dataset from $\mathbb{P}(X, Y)$. A corruption function $h \in \mathcal{H}$ transforms x_i into a corrupted version $h(x_i)$. Following the covariate shift assumption (Gavrikov & Keuper, 2024), where $\mathbb{P}(X) \neq \mathbb{P}(h(X))$ but the label remains invariant, our objective is to minimize the expected risk across the corruption space $\mathcal{H}$:

$$\min_\theta \mathbb{E}_{h\sim H}[\mathbb{E}_{X,Y\sim\mathbb{P}(X,Y)}[\mathcal{L}(f_\theta(h(X)), Y)]], \quad (1)$$

where $\mathcal{L}$ denotes a standard supervised loss (e.g., cross-entropy). Unlike adversarial robustness works that emphasize input-level sensitivity (Ilyas et al., 2019), we posit that for natural corruptions, robustness is fundamentally tied to the stability of internal features. To formalize this, we decompose f_θ into L layers: $f_\theta(\cdot) = f_L \circ f_{L-1} \circ \cdots \circ f_1$, where $f_{\leq l}$ maps the input into a feature map Z_l. Each neuron in layer l constitutes a **computational pathway** $P_{l,k} : X \rightarrow Z_{l,k}$ (abbreviated as P). The robustness of f_θ thus depends on whether these pathways can produce invariant features $Z_{l,k}$ under corruption h.

3.2. Robust and Non-robust Features

3.2.1. Characterizing Robust Features and Pathways

The concept of robust features was introduced by (Ilyas et al., 2019) via a pixel-level disentanglement framework to interpret adversarial examples. Subsequent studies extended this analysis to the layer (Yan et al., 2021) and neuron levels (Zhang et al., 2020; Madaan et al., 2020) to enhance adversarial robustness. However, while adversarial perturbations are typically high-dimensional, non-semantic noise affecting the local pixel manifold, natural corruptions (e.g., fog, blur) exhibit global structural patterns that manifest as semantic shifts within the latent space. Despite this, the nature of robust and non-robust features under such structured distortions remains poorly understood, particularly regarding their explicit disentanglement. To fill this gap, we formalize robust features under image corruptions through the lens of computational pathways.

Definition 3.1 (Robust features)**.** Given a normalized distortion metric D, a feature $z_{l,k}$ is said to be γ-robust ($\gamma > 0$) for corruption h if

$$D(z_{l,k}, \hat{z}_{l,k}) < \gamma, \quad (2)$$

where $\hat{z}_{l,k}$ denotes the corrupted version of $z_{l,k}$ induced by applying the corruption h to the input.

Definition 3.2 (Robust pathways)**.** Given a normalized distortion metric D, a computational pathway P is said to be γ-robust ($\gamma > 0$) for corruption h if

$$\mathbb{E}_{x\sim\mathbb{P}(X)}[D(P(x), P(h(x)))] < \gamma. \quad (3)$$

Note that these definitions isolate robustness as a property of features and pathways, independent of their impact on predictions; Section 3.2.3 later investigates this connection.

3.2.2. Identifying Robust Features: Layer-wise Distribution and Evolution

To demystify the internal behavior of robust features, we quantify their layer-wise distribution across the network. We employ the local Lipschitz constant (Cisse et al., 2017;

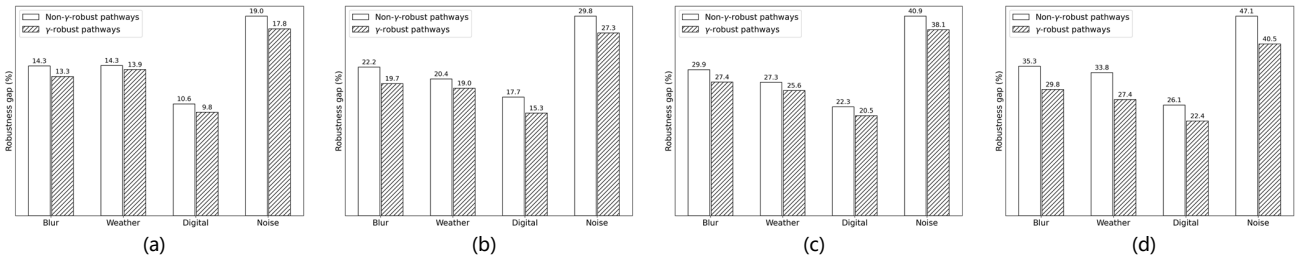


Figure 4. Impact of pathway robustness on sub-networks and the full model. We report the robustness gap (lower is better) for (a) $f_{\leq 1}$, (b) $f_{\leq 5}$, (c) $f_{\leq 9}$, and (d) f_θ when preserving either γ-robust or non-γ-robust pathways.

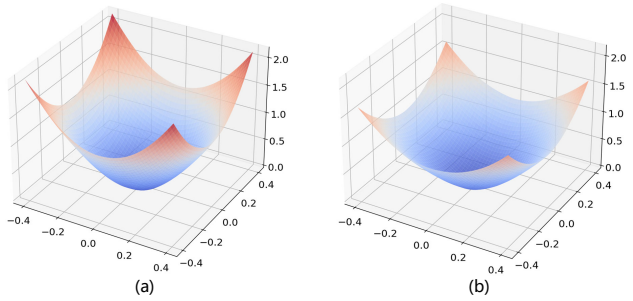


Figure 5. Loss landscapes of f_θ under input perturbations. (a) Non-γ-robust pathways exhibit a sharp landscape, whereas (b) γ-robust pathways show a flatter surface.

Szegedy et al., 2013) as the distortion metric D in Equation (2), denoted as D_{Lip}:

$$D_{\text{Lip}}(z_{l,k}, \hat{z}_{l,k}) = \frac{||z_{l,k} - \hat{z}_{l,k}||}{||x - h(x)||}, \tag{4}$$

where $||\cdot||$ represents the normalized Frobenius norm. This metric effectively captures the sensitivity of internal features relative to input-level perturbations. Using Equation (2), Figure 2 (a)-(b) visualize the cumulative distribution of γ-robust features for a pretrained ResNet18 model under Gaussian noise and Fog, respectively. To provide a more intuitive and quantitative comparison, Figure 3 summarizes the proportion of robust features at specific thresholds γ.

Our empirical analysis reveals a non-homogeneous evolution of robustness along the network depth. Specifically, while shallow sub-networks (e.g., $f_{\leq 1}$) retain a high proportion of γ-robust features, this proportion exhibits a progressive decay as information propagates into deeper sub-networks (e.g., $f_{\leq 5}$ and $f_{\leq 9}$). As evidenced by the sharp contrast in Figure 3, the available pool of robust features shrinks drastically in deeper layers, particularly at stringent thresholds. This trend suggests that as receptive fields expand to capture global structural patterns, features become increasingly susceptible to semantic shifts induced by structured corruptions. Consistent results for additional corruption types are further detailed in Appendix D.1, underscoring that internal robustness degradation is a pervasive challenge across diverse distortion categories.

Remark 1. Under image corruptions, robust features exhibit a progressive decay across neural layers, indicating that their prevalence diminishes significantly as features propagate into deeper stages of the network.

3.2.3. A CLOSER LOOK AT ROBUST FEATURES: CAUSAL IMPACT AND LOSS LANDSCAPES

Based on the above observation, we take a closer look at the collective behavior of robust (non-robust) features under image corruptions. To explore this, we investigate the causal role of these features by isolating robust and non-robust pathways and quantifying their impact on the robustness of both downstream sub-networks and the full model. For a given corruption h, we score each computational pathway P in the stem stage (l_1) of a pretrained ResNet-18 using the empirical mean of D_{Lip}:

$$\bar{D}_{\text{Lip}}(P, h) = \frac{1}{N} \sum_{i=1}^{N} \frac{||P(x_i) - P(h(x_i))||}{||x_i - h(x_i)||}. \tag{5}$$

To evaluate their causal role, we partition pathways into γ-robust and non-γ-robust groups based on the top and bottom 50% of scores. We selectively activate one group at layer l_1 while zeroing the other. For a fair comparison, we perform linear probing by attaching and fine-tuning a linear head on the final feature maps of each sub-network $f_{\leq l_2}$ and the full model f_θ using clean data. Comprehensive analysis across 15 corruption types in Appendix D.2 further confirms the consistent advantage of γ-robust pathways.

We evaluate the influence of these pathway groups using the robustness gap (Taori et al., 2020; Fang et al., 2022; Tu et al., 2025), defined as the accuracy degradation from clean to corrupted data. Figure 4 reveals a consistent trend across all sub-networks and the full model: γ-robust pathways yield a significantly narrower gap compared to their non-robust counterparts. This disparity suggests that model robustness is sustained by this specific subset of invariant features. To provide a geometric explanation, we visualize the loss landscapes of f_θ in Figure 5. Notably, γ-robust pathways exhibit a significantly flatter loss surface, which

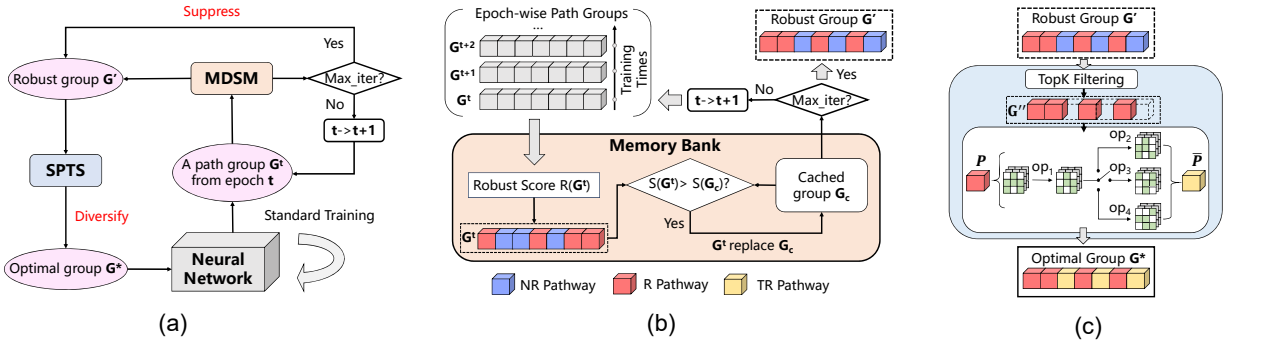


Figure 6. The pipeline of (a) S&D, (b) MDSM, and (c) SPTS. NR, R and TR denote non-robust, robust and tweaked robust pathways, respectively.

ensures stable predictions in the vicinity of clean inputs and effectively minimizes the robustness gap.

Remark 2. The robustness of the full model is functionally dependent on the prevalence of robust features, which exert a direct impact on preserving performance across downstream layers and sub-networks under corruptions.

3.3. The S&D Refinement

3.3.1. Overview

Motivated by the hierarchical decay of robust features, we propose the S&D refinement to enhance corruption robustness through pathway manipulation (Figure 6 (a)). S&D consists of two synergistic components: (1) MDSM, which identifies a stable pathway group G' generating robust representations, and (2) SPTS, which constructs an optimal group G^* by diversifying G' while preserving its structural invariants. Notably, S&D is non-intrusive, requiring no architectural modifications or interference with standard training, ensuring seamless integration into mainstream frameworks.

3.3.2. MDSM: Selecting Robust Pathways

The Memory-aware Dynamic Selection Mechanism (MDSM) identifies robust pathways by maintaining a candidate group G_c in a memory bank (Figure 6 (b)). At epoch t, a candidate collection $G^t = \{P_{l,1}, \ldots, P_{l,K}\}$ is deterministically constructed from the current model state, with fixed indexing inherited from layer l. This collection is evaluated against G_c using a fitness score $\mathcal{S}$ that balances feature stability with discriminability:

$$\mathcal{S}(G^t) = \frac{1}{K}\sum_{k=1}^{K}[R(P_{l,k}(x), P_{l,k}(\hat{x}))] + \lambda \cdot \sigma(n/N), \quad (6)$$

where X and $\hat{X}$ represent a batch of clean images and their corresponding corrupted versions. R is a similarity metric (e.g., L_2 or CKA), and n and N denote the current and total scheduled MDSM steps at the epoch level. λ is a hyperparameter balancing two terms and is set to 1. The second term, governed by a sigmoid-like schedule $\sigma(\cdot)$, prioritizes discriminability as training matures, inspired by findings that deeper layers develop more discriminative representations over time (Zeiler & Fergus, 2014; Yosinski et al., 2014; Zhang et al., 2021). Using Equation (6), the memory bank iteratively updates G_c over N steps to output the robust pathway group G'. Ablations in Appendix B.4 and B.3 confirm that S&D yields consistent gains regardless of the specific choice of R or the transformations for $\hat{x}$. Notably, MDSM operates during epoch gaps, ensuring zero interference with standard training gradients.

3.3.3. SPTS: Diversifying Robust Pathways

To prevent overfitting to specific robust patterns, we propose the Structure-consistent Pathway Tweaking Strategy (SPTS) to diversify G'. As illustrated in Figure 6 (c), SPTS first filters the top-K most robust pathways:

$$G^{''} = \mathrm{TopK}(G^{'}, R), \quad (7)$$

where $\mathrm{TopK}(\cdot)$denotes the operation that retains the K pathways in $G^{'}$ according to their R values, yielding the new group $G^{''}$. K is typically set to $|G'|/2$; the impact of varying K is detailed in Appendix B.5. To restore the group capacity, we generate augmented pathways $\bar{P}$ by applying symmetry-preserving transformations to the final weight matrix $A \in \mathbb{R}^{C\times W\times H}$ of pathways in G'':

$$\begin{aligned} \bar{A} &= \mathtt{op}_i(\mathtt{op}_1(A)) \\ \text{subject to} \quad & \mathtt{op}_i \sim U\{\mathtt{op}_2, \mathtt{op}_3, \mathtt{op}_4\}, \end{aligned} \quad (8)$$

where U denotes uniform sampling from a predefined set of operations. The transformation suite includes: `Channel shuffling` ($\mathtt{op}_1$), `Horizontal flipping` ($\mathtt{op}_2$), `Vertical flipping` ($\mathtt{op}_3$), and `Matrix transpose` ($\mathtt{op}_4$), with implementation details in Appendix A.1. This specific combination was identified as a highly effective configuration through comparative experiments in Appendix B.2. The augmented pathways $\bar{P}$ (parameterized by $\bar{A}$) introduce stochastic variations into

Table 1. We report Top-1 Accuracy (↑) for ImageNet and mean Corruption Error (mCE) for its corrupted variants (C, $\bar{\text{C}}$, 3DCC, and V2-C). Avg.mCE (↓) averages these four error rates. $[\cdot]$ indicates ensemble-based results; $(+)$ and $(-)$ denote performance changes via S&D. * indicates the use of modern training recipes, including stronger augmentation and optimization practices.

Main	S&D	ImageNet	ImageNet-C	ImageNet-$\bar{C}$	ImageNet-3DCC	ImageNetV2-C	Avg.mCE (↓)
Backbones of Neural Network							
MobileNetV2 (Sandler et al., 2018)	✗	71.4	86.2	86.4	82.8	89.8	86.3
	✔	71.6 [71.2]	84.9	86.6	82.6	89.2	85.8 (**-0.5**)
ResNet-50 (He et al., 2016)	✗	75.7	76.7	79.4	73.2	83.1	78.1
	✔	74.5 [75.6]	72.4	78.2	71.3	79.2	75.3 (**-2.8**)
VGG19 (Simonyan, 2014)	✗	73.9	81.6	84.6	78.6	86.4	82.8
	✔	75.9 [74.2]	79.7	81.1	76.9	84.9	80.7 (**-2.1**)
WideResNet-50_2 (Zagoruyko, 2016)	✗	78.3	71.7	74.4	68.9	79.2	73.6
	✔	76.3 [78.3]	68.5	73.9	67.7	76.1	71.6 (**-2.0**)
Data Augmentation & Model Regularization							
AugMix (Hendrycks et al., 2019)	✗	76.2	72.8	74.8	70.3	79.6	74.4
	✔	74.9 [76.2]	68.1	75.4	68.3	75.7	71.9 (**-2.5**)
TriAug (Müller & Hutter, 2021)	✗	76.7	70.3	75.5	67.5	77.6	72.7
	✔	75.3 [76.7]	67.7	75.7	66.3	75.4	71.3 (**-1.4**)
Label Smoothing (Müller et al., 2019)	✗	76.6	75.2	77.1	72.1	81.5	76.5
	✔	75.0 [76.6]	72.1	77.3	70.8	79.0	74.8 (**-1.7**)
Mixup (Zhang et al., 2018)	✗	76.7	71.7	71.4	70.1	78.7	73.0
	✔	74.9 [76.7]	68.1	71.1	68.8	75.6	70.9 (**-2.1**)
Advanced Training Recipe							
MobileNetV2* (Sandler et al., 2018)	✗	72.1	86.1	82.0	81.3	89.2	84.7
	✔	72.5 [72.3]	78.5	78.3	76.3	83.7	79.2 (**-5.5**)
ResNet-50* (He et al., 2016)	✗	80.7	66.5	66.1	62.3	73.7	67.2
	✔	79.5 [79.7]	61.5	65.0	60.9	69.8	64.3 (**-2.9**)
ConvNeXt-base* (Liu et al., 2022)	✗	84.0	53.6	53.1	53.1	63.5	55.8
	✔	82.9 [83.4]	50.7	50.1	51.8	61.2	53.5 (**-2.3**)

Table 2. Comparison with robust learning paradigms on ImageNet-C. We report Top-1 Accuracy (↑). * denotes methods that used either strong augmentation (S&D*) or improved training recipes (AdaSAP$^*_P$).

Baseline	**Stochastic Depth**	**AdaSAP$^*_P$**	**EWS**
39.2	38.9 (-0.3)	43.3 (+4.1)	40.6 (+1.4)
DST	**DAMP**	**DAT**	**TVM**
38.7 (-0.5)	41.4 (+2.2)	41.1 (+1.9)	39.8 (+0.6)
VOneNet	**GaborNet**	**S&D**	**S&D***
40.3 (+1.1)	37.5 (-1.7)	42.7 (+3.5)	46.3 (+7.1)

the latent space while adhering to the robust hierarchical constraints of the original group. By merging G'' with these augmented pathways, we form the optimal group G^*. Finally, G^* is integrated into the model f_θ for standard training, ensuring that the network is guided by a diverse yet stable set of computational pathways.

4. Experiments

4.1. Experimental Setup

Evaluation dataset. The evaluation uses eight robustness benchmark datasets. For **image classification**, we use ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-$\bar{C}$ (Mintun et al., 2021), and ImageNet-3DCC (Kar et al., 2022). ImageNet-C includes 15 corruption types at five severity levels, ImageNet-$\bar{C}$ adds 10 new corruptions, and ImageNet-3DCC introduces 12 3D-based corruptions mimicking real-world distortions. Additionally, we create ImageNetV2-C by applying 15 corruption types to ImageNetV2 (Recht et al., 2019). For **object detection**, we use COCO-C (Michaelis et al., 2019), a corrupted version of COCO (Lin et al., 2014). For **semantic segmentation**, we generate VOC-C, Cityscapes-C, and ADE20K-C from PASCAL VOC 2012 (Everingham et al., 2015), Cityscapes (Cordts et al., 2016), and ADE20K (Zhou et al., 2019). All datasets are used exclusively for testing.

Evaluation design. We compare S&D with: (1) Subnetwork selection (Stochastic Depth (Huang et al., 2016), AdaSAP (Bair et al., 2024)); (2) Dynamic weight evolution (EWS (Guo et al., 2022), DST (Wu et al., 2025), DAMP (Trinh et al., 2024)); (3) Consistency learning (DAT (Mao et al., 2022), TVM (Saikia et al., 2021)); and (4) Bio-inspired design (VOneNet (Dapello et al., 2020), GaborNet (Pérez et al., 2020)). To demonstrate broad applicability, we evaluate S&D across classification, detection, and segmentation using CNN and Transformer backbones, combined with data augmentations, regularization, and widely adopted training recipes (detailed in Appendix Table A.1). We further assess robustness under test-time adaptation and real-world corruptions. For fairness, all models follow standard protocols or their original settings. Metrics include clean accuracy/mAP/mIoU and mCE or average corrupted performance. Full baseline details are provided in Appendix A.3.

Implementation details. S&D is applied during the initial training phase ($N = 5$ in Equation (6)). We update the stem stage of each backbone as the updated pathway group G^t, a choice validated against deeper stages in Appendix B.6. We adopt CKA (Kornblith et al., 2019) as the default similarity metric R, which yields consistent gains across various metric alternatives (Appendix B.4). We also explore an ensemble-based strategy for evaluating the clean images (Appendix A.2). All classification models follow

Table 3. We report Top-1 Accuracy (↑) for ImageNet-100 and mean Corruption Error (mCE) for its variants (C, $\bar{\text{C}}$, 3DCC, and V2-C). Avg. mCE (↓) denotes the average error across these four benchmarks. $(+)$ and $(-)$ reflect the performance changes after applying S&D.

Main	S&D	IN-100	IN-100-C	IN-100-$\bar{C}$	IN-100-3DCC	IN-100V2-C	Avg.mCE (↓)
MobileViT_S	✗	85.4	88.8	92.5	84.8	92.9	89.8
	✔	85.7	87.3	91.6	85.1	91.6	88.9 (**-0.9**)
EfficientFormer_L1	✗	91.6	73.7	66.3	65.2	80.4	71.4
	✔	91.8	74.4	63.4	64.0	80.6	70.6 (**-0.8**)
ViT_Tiny	✗	84.5	76.2	72.7	78.2	81.7	77.2
	✔	86.3	76.1	70.5	77.8	80.3	76.2 (**-1.0**)
Mambaout_femto	✗	94.1	60.0	54.7	57.7	68.2	60.2
	✔	94.2	57.6	56.3	57.5	67.4	59.7 (**-0.5**)

Table 4. The semantic segmentation performance on ADE20K-C, Cityscapes-C and VOC-C. We report the mean Intersection over Union (mIoU ↑). **S** and **R** represent the results on clean and corrupted data, respectively.

Main	S&D	ADE20K-C		Cityscapes-C		VOC-C	
		S	R	S	R	S	R
Multi-scale Fusion-based Structure							
DeepLabV3+ (Chen et al., 2018)	✗	42.1	21.3	78.6	36.2	75.9	44.9
	✔	41.2	23.1 (**+1.8**)	77.9	39.2 (**+3.0**)	75.4	48.0 (**+3.1**)
PSPNet (Zhao et al., 2017)	✗	41.2	20.1	77.2	34.3	75.9	44.5
	✔	40.0	21.5 (**+1.4**)	76.3	37.8 (**+3.5**)	76.0	47.4 (**+2.9**)
Attention-based Structure							
Mask2Former (Cheng et al., 2022)	✗	47.2	25.2	79.3	40.8	–	–
	✔	46.4	29.2 (**+4.0**)	77.3	47.1 (**+6.3**)	–	–
ANN (Zhu et al., 2019)	✗	39.6	19.2	76.5	33.6	74.9	43.3
	✔	38.9	20.6 (**+1.4**)	75.5	35.9 (**+2.3**)	75.0	46.7 (**+3.4**)
CCNet (Huang et al., 2019)	✗	41.2	20.1	78.9	34.0	76.2	44.1
	✔	40.2	21.5 (**+1.4**)	78.0	38.0 (**+4.0**)	75.9	47.6 (**+3.5**)
DANet (Fu et al., 2019)	✗	40.1	19.9	78.0	34.5	74.5	43.9
	✔	39.4	22.0 (**+2.1**)	77.7	38.5 (**+4.0**)	74.2	46.6 (**+2.7**)
GCNet (Cao et al., 2019)	✗	40.3	19.9	76.8	33.1	76.4	43.9
	✔	39.5	21.6 (**+1.7**)	76.5	36.6 (**+3.5**)	74.8	47.5 (**+3.6**)
PSANet (Zhao et al., 2018)	✗	40.8	20.0	77.0	34.3	76.4	44.6
	✔	40.1	21.6 (**+1.6**)	75.9	36.7 (**+2.4**)	75.5	46.6 (**+2.0**)

Table 5. The object detection performance on COCO and COCO-C. We report the mean Intersection over Union (mAP ↑).

Main	S&D	COCO	COCO-C
Two-Stage Detector			
Faster RCNN (Ren et al., 2016)	✗	37.6	17.5
	✔	37.2	18.2 (**+0.7**)
Mask RCNN (He et al., 2017)	✗	38.1	18.0
	✔	38.1	18.5 (**+0.5**)
Cascade RCNN (Cai & Vasconcelos, 2018)	✗	40.5	18.9
	✔	40.5	19.5 (**+0.6**)
Cascade Mask RCNN (Cai & Vasconcelos, 2018)	✗	41.1	19.3
	✔	41.1	19.9 (**+0.6**)
One-Stage Detector			
RetinaNet (Ross & Dollár, 2017)	✗	36.3	17.1
	✔	36.6	17.6 (**+0.5**)
YOLOv5s (Reis et al., 2023)	✗	39.8	19.9
	✔	39.1	22.1 (**+2.2**)
YOLOv8s (Reis et al., 2023)	✗	46.1	25.5
	✔	45.0	27.0 (**+1.5**)

standard PyTorch recipes, while detection and segmentation tasks utilize ImageNet-pretrained ResNet-50 via MMDetection (Chen et al., 2019) and MMSegmentation (Contributors, 2020). Ablation studies regarding transformation types and hyper-parameter sensitivity are detailed in Appendix B.

4.2. Experimental Results

4.2.1. Main results on robust classification

Comparison with baselines. We evaluate S&D against diverse robust training paradigms: sub-network selection, weight evolution, consistency learning, and bio-inspired designs (Table 2). S&D consistently achieves superior robust accuracy. Without strong augmentation, S&D surpasses the second-best method, DAMP, by 1.3%. When combined with strong augmentation (*), S&D yields a 7.1% total gain over the baseline and outperforms AdaSAP$^*_P$ by 3.0%. This confirms S&D's effectiveness as a standalone strategy. For a fair comparison, results are compiled from original reports or our implementations; see Appendix A.3 for details.

Compatibility with training recipes. Table 1 shows S&D's efficacy across various backbones and configurations. We observe consistent gains of 0.4–2.5% regardless of model scale. Notably, S&D exhibits strong synergy with augmentations and regularizations, boosting robust accuracy by 2.2% (AugMix) and 1.9% (Mixup). Furthermore, under modern recipes, S&D yields substantial gains of 4.3% on MobileNetV2 and 2.0% on ConvNeXt-Base. These results confirm S&D as a versatile plug-and-play enhancement; see Appendix C.1 for extended results.

4.2.2. Versatility across architectures, tasks and scenarios

Architecture diversity. We evaluate S&D on Transformer and Mamba backbones using ImageNet-100 and its corruption variants (e.g., -C, -$\bar{C}$, -3DCC). S&D consistently enhances robustness across these diverse architectures (Table 3). While gains are most pronounced for CNNs due to their texture bias (Geirhos et al., 2018a), S&D still yields

Table 6. The segmentation performance (mIOU) on ACDC.

Main	S&D	Fog	Night	Rain	Snow	Avg.
DeepLabV3+	✗	63.3	13.3	49.2	46.0	43.0
	✔	64.1	11.0	49.5	47.8	43.1 (**+0.1**)
PSPNet	✗	62.0	9.7	45.2	39.8	39.2
	✔	62.8	10.9	46.7	42.7	40.8 (**+1.6**)
Mask2Former	✗	63.0	25.5	49.8	50.0	47.1
	✔	64.0	23.7	48.3	51.4	46.9 (**+-0.2**)
CCNet	✗	61.4	8.7	49.9	45.8	41.5
	✔	63.6	13.9	44.5	42.9	41.2 (**-0.3**)
DANet	✗	61.1	9.2	46.3	43.6	40.1
	✔	61.1	10.0	48.6	41.2	40.2 (**+0.1**)
GCNet	✗	57.7	5.6	45.1	36.8	36.3
	✔	61.5	10.7	45.6	42.7	40.1 (**+3.8**)

Table 7. Comparison of test-time adaptation Methods at the severity 5 on ImageNet-C (Without vs. With S&D). We report Top-1 Accuracy (↑).

	AdaContrast	Tent	BN	EATA
w/o S&D	34.9	37.3	31.5	42.1
w/ S&D	38.1 (+3.2)	38.3 (+1.0)	31.9 (+0.4)	42.6 (+0.5)
	SANTA	**SAR**	**RMT**	**ROTTA**
w/o S&D	39.9	37.8	42.2	32.6
w/ S&D	40.6 (+0.7)	38.6 (+0.8)	44.2 (+2.0)	34.0 (+1.4)

steady improvements for ViT and Mamba models. These results demonstrate S&D's versatility across various architectural paradigms.

Downstream tasks. To verify S&D's versatility, we extend evaluation to dense prediction tasks. For semantic segmentation (Table 4), S&D yields consistent mIoU gains across ADE20K-C (1.4–4.0%), Cityscapes-C (2.3–6.3%), and VOC-C (2.0–3.6%), with Mask2Former achieving the largest boost. In object detection (Table 5), S&D effectively enhances robust mAP for both two-stage (Faster R-CNN: 17.5→18.2) and one-stage detectors (YOLOv5s: 19.9→22.1) while maintaining competitive clean performance. These results confirm S&D's broad applicability in enhancing robustness for diverse visual applications.

Real-world scenarios. We first validate S&D under real-world adverse weather for semantic segmentation and object detection. On the ACDC dataset (Sakaridis et al., 2021), S&D improves mIoU across most backbones (Table 6), notably boosting GCNet by 3.8%, with corresponding segmentation visualizations provided in Appendix D.3. On the DWD benchmark (Wu & Deng, 2022), S&D enhances Faster R-CNN by 1.6 and 0.4 mAP in night-rainy and dusk-rainy scenarios, respectively. Qualitative results (Figure 7) confirm S&D's efficacy in localizing and identifying objects under such compound corruptions. To further assess S&D in dynamic environments, we evaluate continual test-time adaptation (TTA) on ImageNet-C. By integrating S&D with eight representative methods, including Tent (Wang et al., 2021), BN (Benz et al., 2021), AdaContrast (Chen et al., 2022), EATA (Niu et al., 2022), SANTA (Chakrabarty et al., 2023), SAR (Niu et al., 2023), RMT (Döbler et al., 2023), and ROTTA (Yuan et al., 2023), we observe consistent performance gains (Table 7) with a 3.2% peak improvement.

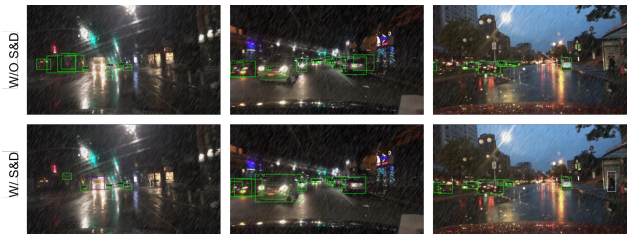


Figure 7. The detection results in night-rainy scenarios.

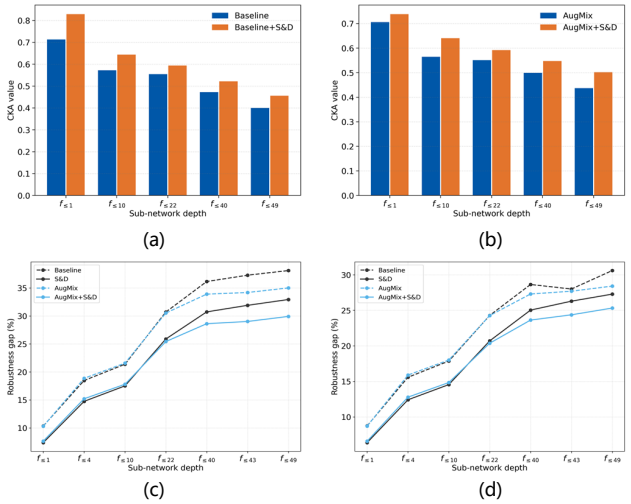


Figure 8. Comparison of (a) CKA between the baseline and S&D, (b) CKA between AugMix and AugMix+S&D, and robustness gap with and without S&D on (c) ImageNet-C and (d) ImageNet-3DCC.

These results demonstrate S&D's broad compatibility in complex, real-world degraded environments.

5. Analysis

5.1. Representation Stability

To evaluate representation stability, we compute CKA between ResNet-50 sub-network features ($f_{\leq 1}$ to $f_{\leq 49}$) under clean and corrupted inputs. As shown in Figure 8 (a, b), S&D consistently increases feature similarity across all depths. Further linear probing on ImageNet-C/3DCC using the robustness gap (lower is better) reveals consistent gap reductions in Figure 8 (c, d). These findings confirm that S&D effectively guides the network toward robust hierarchical representations in a bottom-up manner.

5.2. Training Efficiency

We measured the training-time and memory overhead of S&D relative to a standard ResNet-50 baseline. Table 8 shows that baseline training requires 2294.27 s/epoch, while MDSM adds only 0.3233 s on selected epochs (∼0.01%), and the one-time SPTS cost is 0.0057 s (<0.001%), with

Table 8. Wall-clock overhead analysis of our method. "Baseline" denotes the runtime of the original training per epoch. "MDSM" denotes the additional per-epoch overhead introduced by our method, and "SPTS" is a one-time operation. Within MDSM, "Aug" denotes corrupted input generation, "Fwd" denotes forward propagation, and "CKA" denotes CKA computation. Relative overhead is measured with respect to the baseline epoch time.

Component	Execution Frequency	Time (s)	Time Breakdown (s)	Against Baseline Epoch
Baseline training	Every epoch	2294.27	–	–
MDSM	Selected epochs	0.3233 ± 0.0015	Aug: 0.0056, Fwd: 0.0420, CKA: 0.2757	$\sim$0.01%
SPTS	Last selected epoch	0.0057 ± 0.0070	–	$< 0.001\%$

Table 9. Throughput and memory overhead analysis of MDSM compared with baseline training. "Peak Mem Diff" is computed as MDSM peak memory minus baseline peak memory. Negative values indicate that MDSM remains within the baseline training memory budget.

Baseline Throughput (img/s)	MDSM Throughput (img/s)	Baseline Peak Mem (MB)	MDSM Peak Mem (MB)	Peak Mem Diff (MB)
880.79	6003.23	21292.07	7640.79	-13651.28

CKA computation dominating the small overhead. Table 9 reports peak GPU memory and throughput: MDSM consumes 7640.79 MB versus 21292.07 MB for baseline, while throughput increases due to lighter memory usage. Overall, S&D introduces negligible wall-clock and memory overhead during training, confirming its practical efficiency without affecting test-time cost.

6. Conclusion

This paper investigates the phenomenon of robust features under image corruptions. Our analysis reveals that robust features exhibit a progressive decay across neural layers and establishes that model-wide robustness is functionally dependent on their prevalence. Building on these insights, we propose S&D, a simple refinement that suppresses non-robust pathways and diversifies robust ones during training. S&D is architecture-agnostic, parameter-free, and requires zero test-time overhead, enabling seamless plug-and-play integration. Extensive experiments demonstrate that S&D significantly enhances corruption robustness and OOD generalization, offering a new perspective on building robust vision models.

Despite these improvements, several limitations remain. S&D introduces a consistent clean–corruption trade-off, as pathways stabilized under corruptions may suppress features beneficial for clean accuracy. Evaluations have focused on moderate-scale models and datasets; scalability to larger models and full ImageNet-scale data remains to be fully explored. The current framework targets 2D/3D vision tasks, and applicability to other modalities is not yet established. Future work will focus on developing a theoretical framework for internal robust features, designing strategies to mitigate clean-accuracy degradation, and extending S&D to additional modalities such as audio and multimodal tasks.

Impact Statement

We propose Suppress & Diversify (S&D), a training-time refinement that enhances the robustness of deep neural networks under natural image corruptions. S&D improves reliability in safety-critical vision systems such as autonomous driving, surveillance, and medical imaging, without adding inference overhead. It is architecture-agnostic and compatible with diverse CNN and Transformer backbones. Potential risks include over-reliance on automated decisions in untested scenarios; users should validate models across varied environments and combine with augmentation or uncertainty estimation. S&D provides a principled approach to building reliable and generalizable visual models.

References

Bair, A., Yin, H., Shen, M., Molchanov, P., and Alvarez, J. M. Adaptive sharpness-aware pruning for robust sparse networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Benz, P., Zhang, C., Karjauv, A., and Kweon, I. S. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 494–503, 2021.

Cai, Z. and Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Cao, Y., Xu, J., Lin, S., Wei, F., and Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

Chakrabarty, G., Sreenivas, M., and Biswas, S. Santa: Source anchoring network and target alignment for con-

tinual test time adaptation. *Transactions on Machine Learning Research*, 2023.

Chen, D., Wang, D., Darrell, T., and Ebrahimi, S. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pp. 854–863. PMLR, 2017.

Contributors, M. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Dapello, J., Marques, T., Schrimpf, M., Geiger, F., Cox, D., and DiCarlo, J. J. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *Advances in Neural Information Processing Systems*, 33:13073–13087, 2020.

Diffenderfer, J., Bartoldson, B., Chaganti, S., Zhang, J., and Kailkhura, B. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *Advances in neural information processing systems*, 34:664–676, 2021.

Döbler, M., Marsden, R. A., and Yang, B. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7704–7714, 2023.

Döbler, M., Marsden, R. A., Raichle, T., and Yang, B. A lost opportunity for vision-language models: a comparative study of online test-time adaptation for vision-language models. In *European Conference on Computer Vision*, pp. 117–133. Springer, 2024.

Dong, Y., Kang, C., Zhang, J., Zhu, Z., Wang, Y., Yang, X., Su, H., Wei, X., and Zhu, J. Benchmarking robustness of 3d object detection to common corruptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1022–1032, 2023.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

Fang, A., Ilharco, G., Wortsman, M., Wan, Y., Shankar, V., Dave, A., and Schmidt, L. Data determines distributional robustness in contrastive language image pre-training (clip). In *International conference on machine learning*, pp. 6216–6234. PMLR, 2022.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., and Lu, H. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3146–3154, 2019.

Gavrikov, P. and Keuper, J. Can biases in imagenet models explain generalization? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22184–22194, 2024.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018a.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018b.

Guo, Y., Stutz, D., and Schiele, B. Improving robustness by enhancing weak subnets. In *European Conference on Computer Vision*, pp. 320–338. Springer, 2022.

Guo, Y., Stutz, D., and Schiele, B. Improving robustness of vision transformers by reducing sensitivity to patch corruptions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4108–4118, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *European conference on computer vision*, pp. 646–661. Springer, 2016.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 603–612, 2019.

Iandola, F. N. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Kamann, C. and Rother, C. Benchmarking the robustness of semantic segmentation models with respect to common corruptions. *International journal of computer vision*, 129(2):462–483, 2021.

Kar, O. F., Yeo, T., Atanov, A., and Zamir, A. 3d common corruptions and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18963–18974, 2022.

Kim, W. J., Cho, Y., Jung, J., and Yoon, S.-E. Feature separation and recalibration for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8183–8192, 2023.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., and Finn, C. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., and Ren, J. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022.

Li, Z., Brendel, W., Walker, E., Cobos, E., Muhammad, T., Reimer, J., Bethge, M., Sinz, F., Pitkow, Z., and Tolias, A. Learning from brains how to regularize machines. *Advances in neural information processing systems*, 32, 2019.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Madaan, D., Shin, J., and Hwang, S. J. Adversarial neural pruning with latent vulnerability suppression. In *International Conference on Machine Learning*, pp. 6575–6585. PMLR, 2020.

Mao, X., Chen, Y., Duan, R., Zhu, Y., Qi, G., Li, X., Zhang, R., Xue, H., et al. Enhance the visual representation via discrete adversarial training. *Advances in Neural Information Processing Systems*, 35:7520–7533, 2022.

Mehta, S. and Rastegari, M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., and Brendel, W. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

Mintun, E., Kirillov, A., and Xie, S. On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, 34:3571–3583, 2021.

Modas, A., Rade, R., Ortiz-Jiménez, G., Moosavi-Dezfooli, S.-M., and Frossard, P. Prime: A few primitives can boost robustness to common corruptions. In *European Conference on Computer Vision*, pp. 623–640. Springer, 2022.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Müller, S. G. and Hutter, F. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 774–782, 2021.

Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., and Tan, M. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., and Tan, M. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Park, J., Chai, J. C. L., Yoon, J., and Teoh, A. B. J. Understanding the feature norm for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1557–1567, 2023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Pérez, J. C., Alfarra, M., Jeanneret, G., Bibi, A., Thabet, A., Ghanem, B., and Arbeláez, P. Gabor layers enhance network robustness. In *European Conference on Computer Vision*, pp. 450–466. Springer, 2020.

Qin, Y., Zhang, C., Chen, T., Lakshminarayanan, B., Beutel, A., and Wang, X. Understanding and improving robustness of vision transformers through patch-based negative augmentation. *Advances in Neural Information Processing Systems*, 35:16276–16289, 2022.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Reis, D., Kupec, J., Hong, J., and Daoudi, A. Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*, 2023.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

Ross, T.-Y. and Dollár, G. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2980–2988, 2017.

Safarani, S., Nix, A., Willeke, K., Cadena, S., Restivo, K., Denfield, G., Tolias, A., and Sinz, F. Towards robust vision by multi-task learning on monkey visual cortex. *Advances in Neural Information Processing Systems*, 34: 739–751, 2021.

Saikia, T., Schmid, C., and Brox, T. Improving robustness against common corruptions with frequency biased models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10211–10220, 2021.

Sakaridis, C., Dai, D., and Van Gool, L. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10765–10775, 2021.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Simonyan, K. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tao, L., Feng, L., Wei, H., Yi, J., Huang, S.-J., and Chen, S. Can adversarial training be manipulated by non-robust features? *Advances in Neural Information Processing Systems*, 35:26504–26518, 2022.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

Teti, M., Kenyon, G., Migliori, B., and Moore, J. Lcanets: Lateral competition improves robustness against corruption and attack. In *International Conference on Machine Learning*, pp. 21232–21252. PMLR, 2022.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Trinh, Q. T., Heinonen, M., Acerbi, L., and Kaski, S. Improving robustness to corruptions with multiplicative weight perturbations. *Advances in Neural Information Processing Systems*, 37:35492–35516, 2024.

Tu, W., Deng, W., and Gedeon, T. Toward a holistic evaluation of robustness in clip models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Vaish, P., Wang, S., and Strisciuglio, N. Fourier-basis functions to bridge augmentation gap: Rethinking frequency augmentation in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17763–17772, 2024.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Wang, H., Deng, Y., Yoo, S., and Lin, Y. Exploring robust features for improving adversarial robustness. *IEEE Transactions on Cybernetics*, 2024.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Wightman, R., Touvron, H., and Jégou, H. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

Wu, A. and Deng, C. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 847–856, 2022.

Wu, B., Xiao, Q., Wang, S., Strisciuglio, N., Pechenizkiy, M., van Keulen, M., Mocanu, D. C., and Mocanu, E. Dynamic sparse training versus dense training: The unexpected winner in image corruption robustness. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Yan, H., Zhang, J., Niu, G., Feng, J., Tan, V., and Sugiyama, M. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *International Conference on Machine Learning*, pp. 11693–11703. PMLR, 2021.

Yang, Y., Khanna, R., Yu, Y., Gholami, A., Keutzer, K., Gonzalez, J. E., Ramchandran, K., and Mahoney, M. W. Boundary thickness and robustness in learning models. *Advances in Neural Information Processing Systems*, 33: 6223–6234, 2020.

Yin, D., Gontijo Lopes, R., Shlens, J., Cubuk, E. D., and Gilmer, J. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, 2019.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Yu, W. and Wang, X. Mambaout: Do we really need mamba for vision? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4484–4496, 2025.

Yuan, L., Xie, B., and Li, S. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Yucel, M. K., Cinbis, R. G., and Duygulu, P. Hybridaugment++: Unified frequency spectra perturbations for model robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5718–5728, 2023.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zagoruyko, S. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Zeng, R., Chen, X., Liang, J., Wu, H., Cao, G., and Guo, Y. Benchmarking the robustness of temporal action detection models against temporal corruptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18263–18274, 2024.

Zhang, C., Liu, A., Liu, X., Xu, Y., Yu, H., Ma, Y., and Li, T. Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity. *IEEE Transactions on Image Processing*, 30:1291–1304, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C. C., Lin, D., and Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 267–283, 2018.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., and Torralba, A. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Mixstyle neural networks for domain generalization and adaptation. *International Journal of Computer Vision*, 132(3):822–836, 2024.

Zhu, Z., Xu, M., Bai, S., Huang, T., and Bai, X. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 593–602, 2019.

A. Implementation Details

A.1. The Tweaking Operations

The tweaking operations include four types, `Channel shuffling`, `Horizontal flipping`, `Vertical flipping`, `Matrix transpose`, abbreviated as `op1`, `op2`, `op3` and `op4`. Let $A \in \mathbb{R}^{C\times W\times H}$ be the weight matrix, and c, w, h are the corresponding index, respectively. The tweaking operations can be formulated as:

$$op_i(x) = \begin{cases} A_{\sigma(c),w,h} & \text{if } i = 1 \\ A_{c,W-w+1,h} & \text{if } i = 2 \\ A_{c,w,H-h+1} & \text{if } i = 3 \\ A_{c,h,w} & \text{if } i = 4, \end{cases} \tag{A.1}$$

where σ is a random permutation that rearranges the C channels.

A.2. Ensemble-based Strategy

We additionally test a simple ensemble-based strategy on ImageNet. Specifically, for Model A, we select the ImageNet-pretrained model provided by PyTorch, while Model B uses the same architecture but incorporates S&D, such as ResNet-50 and ResNet-50+S&D. Inspired by (Park et al., 2023), we find that the l_2 norm of the output at the first layer can serve as a confidence value to distinguish between non-corrupted and corrupted data. Given an unknown sample x, its corresponding feature is denoted as z. We select the threshold value as 0.5. The prediction $\hat{y}$ can be computed:

$$\hat{y} = \begin{cases} \text{Model A}(x) & \text{if } ||z||_2 < 0.5 \\ \text{Model B}(x) & \text{otherwise} \end{cases} \tag{A.2}$$

The results of Tab. 1 in the main text demonstrate that this ensemble-based approach is effective across most network architectures.

A.3. Baseline Methods

We integrate S&D in representative methods to validate its effectiveness across image classification, object detection and semantic segmentation tasks.

Robustness-related paradigms. We compare S&D against four representative categories of robustness-enhancing methods: (1) Sub-network Selection, including AdaSAP (Bair et al., 2024) (results cited from the original paper) and Stochastic Depth (Huang et al., 2016) (reproduced following the original protocols); (2) Dynamic Weight Evolution, such as EWS (Guo et al., 2022), DST (Wu et al., 2025), and DAMP (Trinh et al., 2024), where results are cited from their respective publications; (3) Consistency Learning, featuring DAT (Mao et al., 2022) (evaluated using the official checkpoint) and TVM (Saikia et al., 2021) (reproduced following original settings); and (4) Bio-inspired Designs, such as VOneNet (Dapello et al., 2020) and GaborNet (Pérez et al., 2020), both of which are strictly reproduced.

Image classification. We evaluate four aspects: (1) neural network backbones, including convolutional architectures such as AlexNet (Krizhevsky et al., 2012), MobileNetV2 (Sandler et al., 2018), SqueezeNet (Iandola, 2016), VGG (Simonyan, 2014), ResNet (He et al., 2016), ResNeXt (Xie et al., 2017), WideResNet (Zagoruyko, 2016), and ConvNeXt (Liu et al., 2022), as well as attention- and state-space-based models including Vision Transformer (ViT) (Dosovitskiy, 2020), MobileViT (Mehta & Rastegari, 2021), Efficientformer (Li et al., 2022), and MambaOut (Yu & Wang, 2025); detailed results for ViT and Mamba-based architectures are reported in Table 3; (2) data augmentation (e.g., AugMix (Hendrycks et al., 2019), AutoAug (Cubuk et al., 2018), RandAug (Cubuk et al., 2020), TriAug (Müller & Hutter, 2021), Random Erasing (Zhong et al., 2020)); (3) model regularization (e.g., CutMix (Yun et al., 2019), Mixup (Zhang et al., 2018), and Label smoothing (Müller et al., 2019)) and (4) a modern Pytorch training recipe (Paszke et al., 2019) combining diverse augmentations, regularization, and extended training schedules.

Object detection&semantic segmentation. We test both two-stage detectors (e.g., Faster RCNN (Ren et al., 2016), Mask RCNN (He et al., 2017), Cascade RCNN, and Cascade Mask RCNN (Cai & Vasconcelos, 2018))and one-stage detectors (RetinaNet (Ross & Dollár, 2017), YOLOv5 and YOLOv8 (Reis et al., 2023)). For semantic segmentation, we evaluate on multi-scale fusion-based (e.g., DeepLabV3+ (Chen et al., 2018), PSPNet (Zhao et al., 2017)) and attention-based architectures (e.g., Mask2Former (Cheng et al., 2022), ANN (Zhu et al., 2019), CCNet (Huang et al., 2019), DANet (Fu et al., 2019), GCNet (Cao et al., 2019), and PSANet (Zhao et al., 2018)).

Test-time adaptation methods. To further evaluate the compatibility and effectiveness of S&D under distribution shifts, we integrate it with representative test-time adaptation (TTA) methods, including AdaContrast (Chen et al., 2022), BN statistics update (Benz et al., 2021), Tent (Wang et al., 2021), EATA (Niu et al., 2022), SANTA (Chakrabarty et al., 2023), SAR (Niu et al., 2023), RMT (Döbler et al., 2023) and RoTTA (Yuan et al., 2023), under the continual test-time adaptation setting. In this setting, models adapt sequentially to streaming corrupted data without access to the original training distribution. As shown in Tab. 5 of the main work, S&D consistently improves performance across various TTA methods and under diverse corruption types.

Table A.1. Hyperparameters of the baseline training recipe and advanced training recipe.

	Baseline	Advanced		Baseline	Advanced
Train Res	224	176	Label smoothing ϵ	✗	0.1
Test Res	224	232	Repeated Aug	✗	4
Epochs	90	600	H. flip	✔	✔
Batch size	256	1024	RRC	✔	✔
Optimizer	SGD-M	SGD-M	Tri Augment	✗	✔
LR	0.1	0.5	Mixup alpha	✗	0.2
LR Decay	step	cosine	Cutmix alpha	✗	1.0
decay rate	0.1	-	Erasing prob.	✗	0.1
decay epochs	30	-	ColorJitter	✔	✔
Weight decay	10^{-4}	2×10^{-5}	EMA	✗	✔
Warmup Epochs	✗	5	CEloss	✔	✔

A.4. Training Configurations

Neural network backbone. Convolutional architectures are trained using the official PyTorch training recipes (Paszke et al., 2019). For the Vision Transformer (ViT), we follow the training protocol of DeiT (Touvron et al., 2021), trained from scratch without knowledge distillation. Other Transformer-based and Mamba-based models are trained from scratch according to the original procedures described in their respective papers, using implementations from the timm library (Wightman, 2019).

Data augmentation and model regularization. We consistently use ResNet-50 with the following configuration: 90 epochs, batch size of 256, initial learning rate of 0.1, SGD optimizer, and a StepLR scheduler that decays the learning rate by a factor of 0.1 every 30 epochs. All techniques are implemented using standard PyTorch modules, with hyperparameters listed in Table A.2.

Table A.2. Hyperparameter settings for data augmentation and model regularization strategies.

AugMix	**RandAug**	**Random Erasing**
severity=3	magnitude=9	probability=0.1
CutMix	**Mixup**	**Label Smoothing**
α=0.1	α=0.1	ϵ=0.1

Advanced training recipe. We adopt a new PyTorch training protocol for validation and conduct experiments on MobileNetV2, ResNet-50, ResNeXt-50 and ConvNeXt. The comparison with the baseline is shown in Table A.1.

Test-time adaptation method. We evaluate performance under continual test-time adaptation on ImageNet-C at severity level 5, using the same open-source codebase (Döbler et al., 2024), with the batch size set to 64 for all methods.

B. Ablation Studies and Analysis

B.1. Ablation of MDSM and SPTS

We conduct ablation studies on the two core components of S&D. As shown in Table B.1, applying only MDSM slightly improves ImageNet-C robustness (33.5 vs. 32.9), while combining MDSM with SPTS yields the best corruption robustness (34.8), with minimal impact on clean accuracy.

Table B.1. Ablation results of S&D using ResNet-18 on ImageNet and ImageNet-C. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

MDSM	SPTS	ImageNet	ImageNet-C
✗	✗	69.2	32.9
✔	✗	**69.3**	33.5
✔	✔	68.6	**34.8**

B.2. Choice of Tweaking Operations

In SPTS, we adopt GeoDiversify, a geometric diversification strategy based on permutation and spatial transformations, to enhance robust feature diversity while preserving structural consistency. See Section A.1 for details. We also explore several other structure-consistent tweaking operations, including **WeightMix**, which performs weighted mixing between robust and non-robust pathways; **GaussPerturb**, which adds Gaussian noise to the weights of robust pathways; **FreqPerturb**, which applies perturbations in the frequency domain; **LowRankPerturb**, which applies perturbations along the smallest singular vector after SVD; and **UniPerturb**, which injects uniform noise into robust pathway weights. Results in Table B.2 show that **GeoDiversify** achieves the best trade-off between robustness gain and structural fidelity, and thus we use it as the default operation in our main experiments.

B.3. Choice of Corruption-generating Transforms

As mentioned in Sec. 3.3.2, we synthesize corrupted inputs $\hat{X}$ from clean images X using simple transformations to

Table B.2. The results of different tweaking operations using ResNet-18 for ImageNet, ImageNet-C, and ImageNetV2-C. The highest accuracy is indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	**69.2**	32.9	24.4
WeightMix	69.1	34.9	26.3
GaussPerturb	68.6	35.3	26.6
UniPerturb	68.5	35.5	26.7
FreqPerturb	68.6	35.5	**26.8**
LowRankPerturb	68.4	35.3	26.6
GeoDiversify	68.4	**35.6**	**26.8**

guide robust pathway selection in MDSM. We evaluate several corruption strategies: **Contrast+Noise** (Saikia et al., 2021), combining contrast adjustment and additive noise; **ColorJitter**, randomly perturbing brightness, contrast, saturation, and hue; **LowFreq**, adding structured low-frequency perturbations; **MixedFreq**, injecting noise in both low- and high-frequency components; **Solarize**, randomly inverting pixels above a threshold; **AutoAugment** (AutoAug), applying ImageNet-optimized policies; and **RandAugment** (RandAug), using randomized augmentations with uniform magnitude. Results in Table B.3 show that while all methods provide useful supervision for MDSM, **Contrast+Noise** achieves the best trade-off between simplicity and effectiveness, and is thus adopted in our main experiments.

Table B.3. The results of ResNet-18 under different corruption-generating transforms on ImageNet, ImageNet-C, and ImageNetV2-C. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	**69.2**	32.9	24.4
ColorJitter	68.8	33.2	24.9
LowFreq	68.0	34.8	26.2
MixedFreq	67.8	**34.9**	26.2
Solarize	68.8	33.8	25.4
AutoAug	68.8	34.2	25.6
RandAug	68.5	34.7	26.0
Contrast+Noise	68.4	**34.9**	**26.3**

B.4. Choice of Similarity Metrics

We evaluate the effectiveness of various similarity metrics used in Eq. (6), including the l_1 norm, l_2 norm, SVCCA (Raghu et al., 2017), and CKA (Kornblith et al., 2019). As shown in Table B.4, all metrics improve corruption robustness compared to the baseline, demonstrating the flexibility of our method with respect to the choice of similarity measure. While l_1 and l_2 norms yield slightly higher robustness on ImageNet-C, they also lead to a more significant drop in clean accuracy on ImageNet. In contrast, CKA achieves a better trade-off between robustness and clean performance. Therefore, we adopt CKA as the similarity metric in our main experiments to balance accuracy on both clean and corrupted data.

Table B.4. The results of different similarity metrics using ResNet-18 for ImageNet, ImageNet-C, and ImageNetV2-C. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	69.2	32.9	24.4
l_1	67.4	**36.8**	**27.8**
l_2	67.3	36.4	27.6
SVCCA	**69.6**	33.2	24.8
CKA	68.6	34.8	26.2

B.5. Choice of TopK

We conduct an ablation study on the TopK selection ratio in Eq. (7), evaluating settings of 5%, 10%, 15%, 30%, 40%, and 50%. As shown in Table B.5, selecting TopK in the range of 30% to 50% yields a better trade-off between clean accuracy and corruption robustness, with 50% (i.e., retaining half the pathways) serving as a balanced and effective default in our framework.

Table B.5. The results of different settings on Top-K using ResNet-18 for ImageNet, ImageNet-C and ImageNetV2-C. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	**69.2**	32.9	24.4
5%	59.0	28.2	21.3
10%	62.4	31.1	23.2
15%	66.0	34.6	26.1
30%	68.4	35.3	26.6
40%	68.6	**35.4**	**26.7**
50%	69.1	34.3	25.9

B.6. Position for Applying S&D

As highlighted in Sec. 3 of our main work, the S&D refinement can be flexibly applied to any sub-network of a deep model. To investigate how its placement affects performance, we integrate S&D at different depths of ResNet-18, specifically after the sub-networks defined by $f_{l\leq 1}$, $f_{l\leq 3}$, $f_{l\leq 5}$, $f_{l\leq 7}$, and $f_{l\leq 9}$. We evaluate these variants on ImageNet, ImageNet-C, and ImageNetV2-C, with results reported in Table B.6. The findings show that applying S&D at the shallowest position ($f_{l\leq 1}$) yields the highest corruption robustness, while positions $f_{l\leq 3}$ and $f_{l\leq 5}$ still provide meaningful improvements. However, as S&D is placed deeper in the network, such as at $f_{l\leq 7}$ or $f_{l\leq 9}$, the robustness gain gradually diminishes. This trend demonstrates that refining pathways early, before non-robust features propagate through subsequent layers, is essential for achieving optimal robustness.

Table B.6. Accuracy of ResNet-18 when applying S&D at varying sub-network depths on ImageNet, ImageNet-C, and ImageNetV2-C. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	**69.2**	32.9	24.4
$f_{l\leq 1}$	68.6	**34.8**	**26.2**
$f_{l\leq 3}$	69.1	33.2	24.7
$f_{l\leq 5}$	68.6	33.0	24.8
$f_{l\leq 7}$	67.7	32.1	24.0
$f_{l\leq 9}$	66.5	30.6	22.7

Table B.8. The results of different learning rates using ResNet-18+S&D for ImageNet, ImageNet-C, and ImageNetV2-C. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	**69.2**	32.9	24.4
0.0	68.6	34.8	**26.2**
0.001	67.6	**35.0**	**26.2**
0.01	67.9	34.3	25.6
0.1	68.4	34.4	25.6
1	68.8	34.5	25.6

B.7. Sensitivity to Random Seeds

To assess the stability of S&D, we repeat the experiments across multiple random seeds. The results, summarized in Table B.7, show consistent improvements in corruption robustness regardless of the seed, with minimal variance in both clean and corrupted accuracy. This indicates that the performance gains introduced by S&D are stable and not attributable to favorable random initialization.

Table B.7. The results of different random seeds using ResNet-18+S&D for ImageNet, ImageNet-C, and ImageNetV2-C. # denotes the seed index. Results are reported in top-1 accuracy (%), with the highest values indicated in **bold**.

	ImageNet	ImageNet-C	ImageNetV2-C
Baseline	**69.2**	32.9	24.4
#1	68.5	34.2	25.8
#2	68.4	35.1	26.5
#3	68.2	34.9	26.3
#4	68.5	35.3	26.6
#5	68.3	35.4	26.6
#6	68.2	35.0	26.4
#7	68.2	35.8	27.0
#8	68.5	**37.6**	**28.9**

B.8. Sensitivity to Learning Rate

As discussed in Sec. 3.3.3 of our main work, the optimal pathway group G^* produced by S&D is integrated into the neural network for standard training. Here, we evaluate different learning rates applied to G^* during this phase. Results in Table B.8 show that, compared to the baseline, all tested learning rates improve corruption robustness, albeit with a slight drop in clean accuracy. In our main experiments, we fix the learning rate for G^* at 0.0 to preserve the refined robust features without further adaptation.

C. Extended Experimental Results

C.1. Extended Results on Image Classification

We present additional image classification results in Table D.1 and Table 3. Table D.1 reports performance on corruption benchmarks using various convolutional neural networks not included in the main text, showing that S&D consistently improves robustness across various settings.

C.2. Fine-Grained Corruption Robustness

We provide a per-corruption breakdown of ImageNet-C performance for MobileNetV2, ResNet-50, and ConvNeXt-base in Table D.2. This complements the averaged robustness results and highlights how S&D performs across individual corruption types. S&D consistently improves robustness across most corruptions: all 15/15 corruption types on MobileNetV2, 12/15 on ResNet-50, and 13/15 on ConvNeXt-base, totaling 40 out of 45 architecture-corruption pairs. Gains span noise, blur, weather, and digital corruptions, confirming the method is not overfitting to specific corruptions. Some exceptions exist, such as Snow, Fog, and Brightness on ResNet-50, and Elastic and Pixelate on ConvNeXt-base, indicating areas for further investigation. Overall, this analysis demonstrates that S&D provides broad, architecture-agnostic improvements in corruption robustness, while also identifying specific corruptions where improvements are limited.

D. Additional Visualizations

D.1. Extended Visualizations of Robust and Non-robust Features

In Sec. 3.2.2, we present the cumulative distributions of γ-robust features across sub-networks under two common corruptions, including Gaussian Noise and Fog. Here, we extend this analysis by providing visualization results for an additional 13 corruption types, as shown in Figure D.1. Notably, the patterns observed under these 13 corruptions align closely with those reported in the main work, further corroborating our findings.

D.2. Fine-grained Analysis of Robust vs. Non-robust Feature Pathways

In Sec 3.2.3, we analyze the impact of selectively retaining γ-robust versus non-γ-robust pathways on model robustness under image corruptions, as summarized in Fig. 3 of the main work. Here, we provide extended results for this anal-

ysis across all 15 corruption types, as shown in Figure D.2. The consistent advantage of γ-robust pathways in enhancing effective robustness is evident across all corruption types, further reinforcing the critical role of robust feature aggregation in the robustness of downstream sub-networks and the full model.

D.3. The Visualization on Semantic Segmentation

We further test S&D on the ACDC (Sakaridis et al., 2021) dataset, a semantic segmentation under real-world adverse weather conditions, including Fog, Rain, Night, and Snow. The visualization of segmentation results using GCNet (Cao et al., 2019) (with and without S&D) is shown in Figure D.3. S&D reduces the segmentation errors across different adverse weather scenarios.

Table D.1. We report Top-1 Accuracy (ACC) for ImageNet and mean Corruption Error (mCE) for its corrupted variants (C, $\bar{\text{C}}$, 3DCC, and V2-C). Avg.mCE averages these four error rates. $[\cdot]$ indicates ensemble-based results; $(+)$ and $(-)$ denote performance changes via S&D. * indicates advanced training recipes.

Main	S&D	ImageNet	ImageNet-C	ImageNet-$\bar{C}$	ImageNet-3DCC	ImageNetV2-C	Avg.mCE (↓)
Backbones of Neural Network							
AlexNet	✗	55.6	100.0	100.0	100.0	100.0	100.0
	✔	55.1 [55.4]	98.2	99.8	99.4	98.5	99.0 (**-1.0**)
SqueezeNet1_0	✗	57.1	103.5	101.6	100.4	102.2	101.9
	✔	57.9 [57.3]	100.1	99.7	98.1	99.6	99.4 (**-1.5**)
SqueezeNet1_1	✗	57.3	104.4	101.5	100.3	102.7	102.2
	✔	58.3 [57.9]	102.2	100.7	99.0	101.1	100.8 (**-1.4**)
ResNet-18	✗	69.2	84.7	87.0	82.1	88.9	85.7
	✔	68.6 [69.2]	82.4	86.1	81.4	86.8	84.2 (**-1.5**)
ResNet-34	✗	72.8	77.9	81.4	76.3	83.8	79.9
	✔	71.9 [72.8]	74.6	80.1	74.7	81.0	77.6 (**-2.3**)
ResNet-101	✗	77.0	70.4	74.1	68.6	77.8	72.7
	✔	76.0 [77.0]	66.7	74.3	67.0	74.7	70.7 (**-2.0**)
VGG16	✗	73.3	84.6	85.8	80.7	88.7	85.0
	✔	75.1 [73.9]	82.9	84.0	79.2	87.5	83.4 (**-1.6**)
ResNeXt-50	✗	77.4	72.3	74.6	69.8	79.6	74.1
	✔	76.1 [77.4]	70.8	75.3	70.0	77.9	73.5 (**-0.6**)
ResNeXt-101	✗	79.1	66.7	69.4	65.1	74.9	69.0
	✔	77.4 [79.1]	63.0	70.2	64.2	71.7	67.3 (**-1.7**)
WideResNet-101_2	✗	78.8	67.7	71.8	66.1	76.2	70.5
	✔	77.1 [77.8]	65.7	72.0	65.7	74.0	69.4 (**-1.1**)
Data Augmentation & Model Regularization							
Random Erasing	✗	76.4	76.6	78.3	72.4	82.4	77.4
	✔	74.8 [76.4]	72.9	78.3	71.6	79.6	75.6 (**-1.8**)
AutoAug	✗	76.4	73.2	76.9	69.6	79.8	74.9
	✔	74.9 [76.4]	68.6	76.7	67.9	76.1	72.3 (**-2.6**)
RandAug	✗	76.5	73.7	75.6	70.3	80.1	74.9
	✔	75.1 [76.3]	68.8	76.5	68.1	76.4	72.5 (**-2.4**)
CutMix	✗	76.9	76.9	76.4	72.5	82.5	77.1
	✔	75.1 [76.9]	72.8	76.4	71.0	79.2	74.9 (**-2.2**)
Advanced Training Recipe							
ResNeXt-50*	✗	81.1	62.6	62.4	59.8	71.3	64.0
	✔	78.7 [79.0]	60.1	63.5	61.5	69.8	63.7 (**-0.3**)
ConvNeXt-tiny*	✗	82.5	60.9	58.3	59.8	69.5	62.1
	✔	81.8 [82.1]	57.0	57.0	57.8	66.7	59.6 (**-2.5**)
ConvNeXt-small*	✗	83.5	56.2	55.7	56.0	65.8	58.4
	✔	82.6 [82.9]	53.5	54.0	55.0	63.8	56.6 (**-1.8**)

Table D.2. We report Top-1 Accuracy (ACC) for ImageNet and mean Corruption Error (mCE) for its corrupted variants (ImageNet-C with 15 corruption types). Avg.mCE averages these 15 error rates. $[\cdot]$ indicates ensemble-based results; $(+)$ and $(-)$ denote performance changes via S&D. * indicates advanced training recipes.

Main	S&D	ImageNet	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
MobileNetV2*	✗	72.1	89.7	89.9	86.5	87.4	96.9	86.7	88.5	82.3	83.3	67.8	65.7	64.7	93.8	115.6	92.6
	✔	72.5 [72.3]	76.8	77.2	78.2	82.5	92.7	81.7	87.1	80.6	75.3	65.9	65.0	61.4	90.8	81.1	81.2
ResNet-50*	✗	80.7	67.5	68.3	68.7	67.5	85.9	73.8	72.0	64.6	62.0	48.8	46.4	46.3	79.5	80.7	65.4
	✔	79.5 [79.7]	57.0	57.1	57.4	65.1	78.5	69.0	71.2	67.0	58.2	52.5	50.2	45.9	75.5	54.7	63.7
ConvNeXt-base*	✗	84.0	50.0	51.6	49.6	60.1	72.6	53.6	63.3	50.2	46.8	50.8	39.4	39.4	63.0	58.6	54.6
	✔	82.9 [83.4]	44.8	45.0	44.2	57.4	69.0	51.9	60.6	47.0	43.1	44.3	38.4	37.0	64.6	60.1	53.8

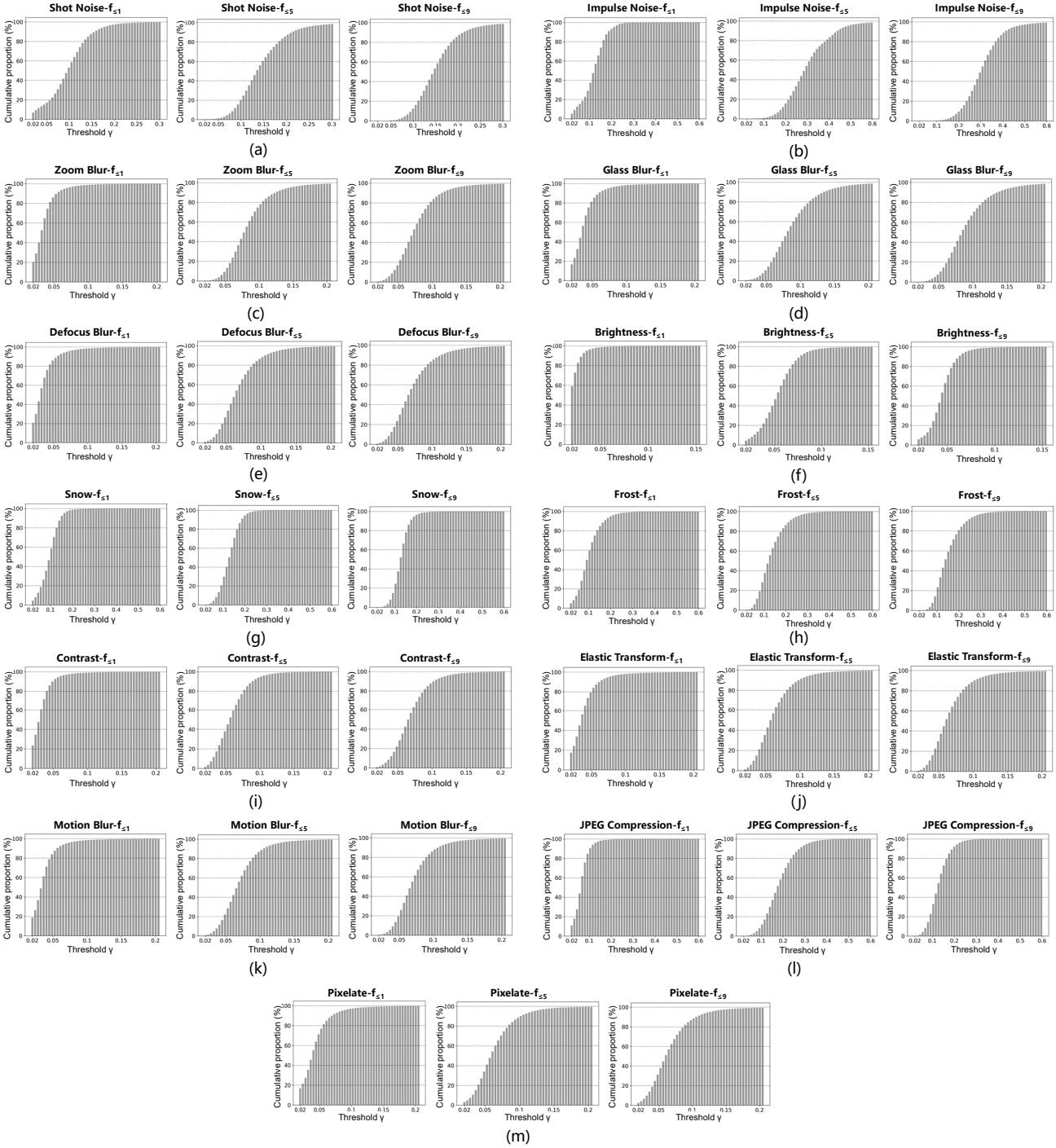


Figure D.1. The cumulative distribution of γ-robust features at different sub-networks under the corruption of (a) Shot Noise, (b) Impulse Noise, (c) Zoom Blur, (d) Glass Blur, (e) Defocus Blur, (f) Brightness, (g) Snow, (h) Frost, (i) Contrast, (j) Elastic Transform and (k) Pixelate.

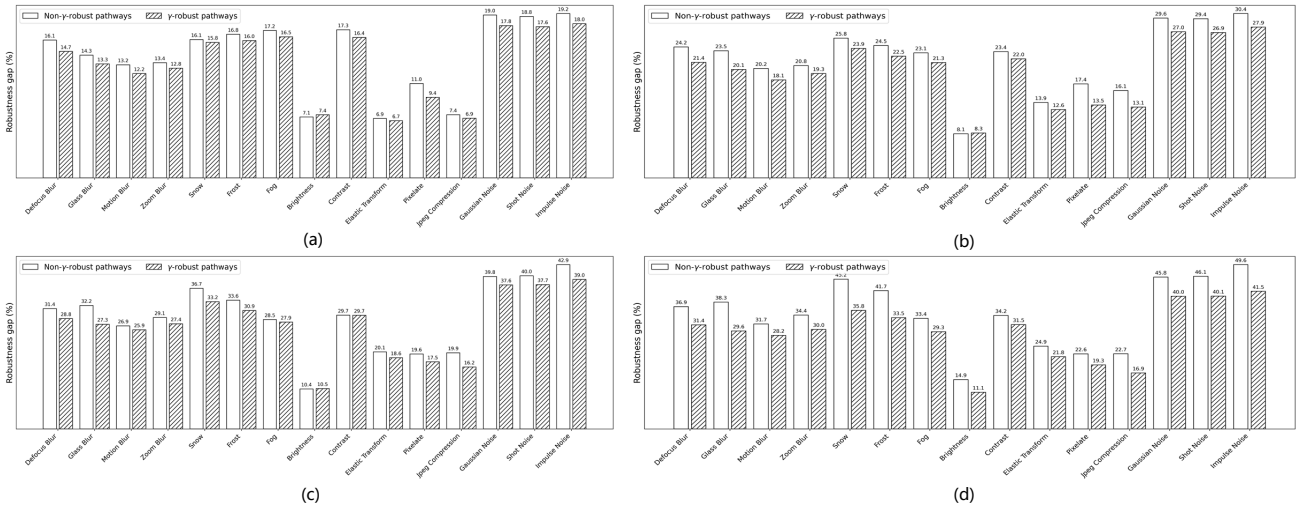


Figure D.2. Effective robustness of sub-networks (a) $f_{\leq 1}$, (b) $f_{\leq 5}$, (c) $f_{\leq 9}$, and (d) the full model f_θ when preserving either γ-robust or non-γ-robust pathways across 15 corruption types. Lower effective robustness indicates better performance.

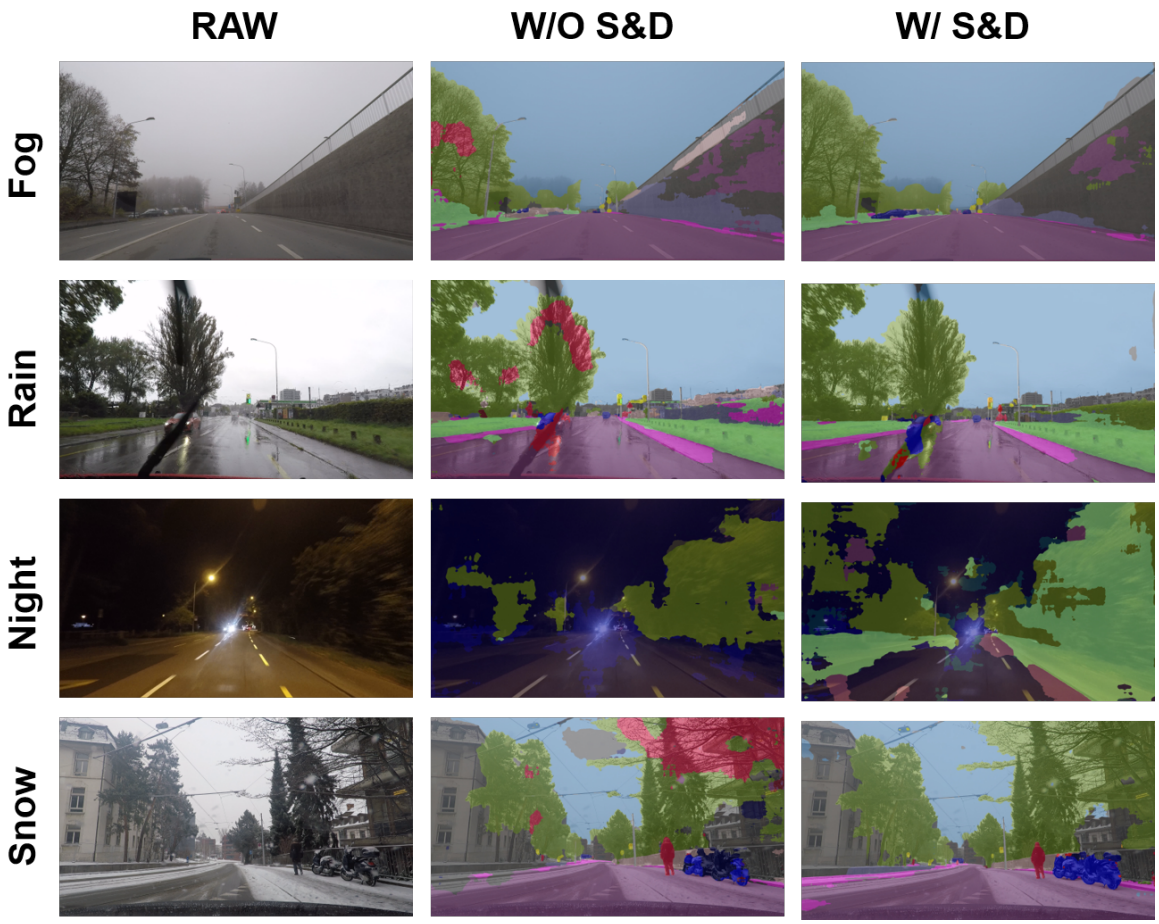


Figure D.3. The visualization of segmentation results for the ACDC dataset.