AdaNCA: Neural Cellular Automata as Adaptors for More Robust Vision Transformer

Yitao Xu Tong Zhang Sabine Süsstrunk
Image and Visual Representation Lab
École polytechnique fédérale de Lausanne,Lausanne, Switzerland
{yitao.xu,tong.zhang,sabine.susstrunk}@epfl.ch

Abstract

Vision Transformers (ViTs) demonstrate remarkable performance in image classification through visual-token interaction learning, particularly when equipped with local information via region attention or convolutions. Although such architectures improve the feature aggregation from different granularities, they often fail to contribute to the robustness of the networks. Neural Cellular Automata (NCA) enables the modeling of global visual-token representations through local interactions, as its training strategies and architecture design confer strong generalization ability and robustness against noisy input. In this paper, we propose **Ada**ptor **N**eural **C**ellular **A**utomata (AdaNCA) for Vision Transformers that uses NCA as plug-and-play adaptors between ViT layers, thus enhancing ViT's performance and robustness against adversarial samples as well as out-of-distribution inputs. To overcome the large computational overhead of standard NCAs, we propose *Dynamic Interaction* for more efficient interaction learning. Using our analysis of AdaNCA placement and robustness improvement, we also develop an algorithm for identifying the most effective insertion points for AdaNCA. With less than a 3% increase in parameters, AdaNCA contributes to more than 10% of absolute improvement in accuracy under adversarial attacks on the ImageNet1K benchmark. Moreover, we demonstrate with extensive evaluations across eight robustness benchmarks and four ViT architectures that AdaNCA, as a plug-and-play module, consistently improves the robustness of ViTs.

1 Introduction

Vision Transformers (ViTs) exhibit impressive performance in image classification, through globally modeling token interactions via self-attention mechanisms [11, 14, 77]. Recent works show that integrating local information into ViTs, *e.g.,* using region attention [26, 38, 39, 54, 65, 70, 73] or convolution [8, 10, 16, 22, 23, 37, 42, 52, 68, 71, 75], further enhances the ViT's capabilities in image classification. Although advanced local structures contribute to better captures of local information, the robustness of ViTs has not increased. They remain vulnerable to noisy input such as adversarial samples [5, 9, 13, 41, 42] and out-of-distribution (OOD) inputs [17, 18, 28, 30, 67].

Recently, Neural Cellular Automata (NCA) was proposed as a lightweight architecture for modeling local cell interactions [44], where cells are represented by 1D vectors. To perform downstream tasks, similarly to the idea of token interactions in ViTs, cells in NCA interact with each other by alternating between a convolution-based *Interaction* stage and an MLP-based *Update* stage [46, 49]. The critical difference, however, is that cell interactions in NCA evolve over time by recurrent application of the two stages, whereas ViT computes the token interaction in a single step per layer. During this process, cells dynamically modulate their representations, based on the interactions with their neighbors, and they gradually enlarge their receptive fields. Unlike commonly used convolutional neural networks, NCA maintains resolution during neighborhood expansion. The recurrent update scheme enables the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

cells to explore various states, thus preventing NCA from overfitting and enhancing its generalization ability [44, 46]. NCA training involves various kinds of stochasticity [45], which enables the models to generalize to input variability and adapt to unpredictable perturbations. It is the modulation of local information and stochasticity during training that make NCA robust against noisy input [49, 48, 55, 62].

However, the original NCA has substantial computational overhead when operating in high-dimensional space, which is a common scenario in ViTs. This poses a non-trivial challenge when integrating NCA into ViTs. To reduce the dimensionality of interaction results, hence to lower the computational cost, we propose *Dynamic Interaction* to replace the standard *Interaction* stage. In this stage, tokens dynamically modify the interaction strategy, based on the observation of their environment. This adaptation to the variability of the environment contributes to the model robustness. Our modified **Ada**ptor **NCA** (AdaNCA), as a plug-and-play module, improves ViTs performances, as illustrated in Figure 1. Adding AdaNCA to different ViT architectures consistently improves their robustness to both adversarial attacks and OOD input. AdaNCA also improves clean accuracy.

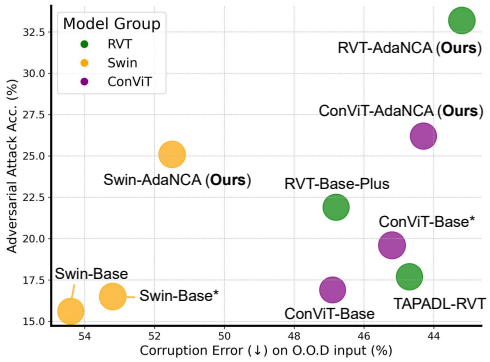

Figure 1: The accuracy under adversarial attacks (APGD-DLR [9]) versus corruption error on out-of-distribution input (ImageNet-C [28]) of various ViT models [16, 23, 38, 42]. AdaNCA improves the robustness of different ViTs against both adversarial attacks and OOD input. $\star$: the same model architecture but with more layers.

Motivated by empirical observations of the positive relationship between network redundancy and model robustness [28], we develop a dynamic programming algorithm for computing the most effective insert position for AdaNCA within a ViT, based on our proposed quantification of network redundancy. Our method results in consistent improvements across eight robustness benchmarks and four different baseline ViT architectures. Critically, we demonstrate that the improvements do not originate from the increase in parameters and FLOPS but are attributed to AdaNCA. Our contributions are as follows:

- We propose AdaNCA: It integrates **N**eural **C**ellular **A**utomata into ViTs' middle layers as lightweight **Ada**ptors for the robustness enhancement of ViTs against adversarial attacks and OOD inputs in image classification. With less than 3% more parameters, AdaNCA-extended ViTs can, under certain adversarial attacks, achieve 10% higher accuracy.
- We introduce *Dynamic Interaction* to replace the *Interaction* stage in standard NCA, thus enhancing model robustness and efficiency in terms of parameters and computation.
- We propose a method for determining the most effective insert positions of AdaNCA, for maximum robustness improvement.

2 Related Works

2.1 Local structure in Vision Transformers

Since the proposal of Vision Transformer (ViT) [14], a series of works have introduced local structures into ViTs to enhance their performance [8, 10, 16, 22, 23, 37, 42, 52, 62, 68, 70, 71, 73, 75, 76]. Here, we mention one of the earliest local structure modifications and those relevant to our work. Stand-Alone Self-Attention (SASA), as introduced by Ramachandran et al. [54], utilizes sliding window self-attention in ViTs. Following this, Liu et al. [38] develop a non-sliding window attention mechanism that partitions feature maps and computes self-attention, both within and between these partitions; it is termed Shifted Window (Swin) attention. Another method for modeling local information is convolution. D'Ascoli et al. [16] introduce a soft local-inductive bias by using gated positional self-attention, thus fusing self-attention and convolution. Despite these advancements in better modeling of local information, few methods lead to a more robust ViT architecture [42]. This leaves the models to behave subpar when encountering slightly noisy inputs or distribution shifts.

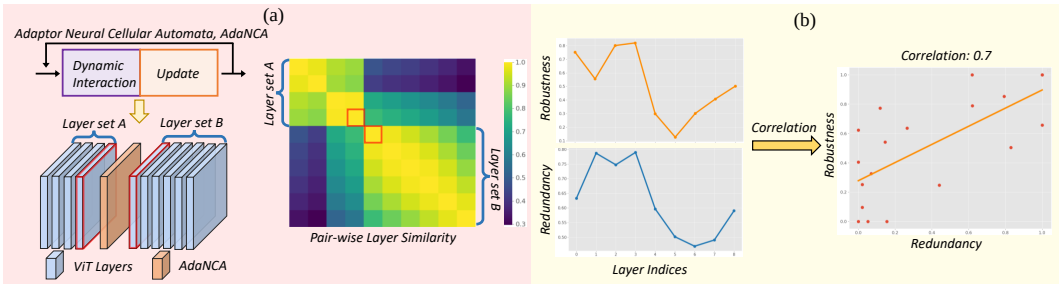

Figure 2: Method overview. (a) To improve model performance and robustness, Neural Cellular Automata (NCA) can be inserted into Vision Transformers (ViTs) as **Ada**ptors, hence termed AdaNCA. The details of AdaNCA are presented in Section 3.2. The improvement is maximized when AdaNCA is inserted between two layer sets that each consists of similar layers. (b) The robustness improvement brought by AdaNCA is highly correlated with the corresponding network redundancy quantification of the insert position introduced in Section 3.3. This supports the idea that AdaNCA should be placed between two sets of redundant layers.

2.2 Robust architecture in Vision Transformers

Researchers have developed various architectural changes for building more robust ViTs against adversarial attacks, such as FGSM [61] or PGD [41], as well as out-of-distribution (OOD) inputs, such as image corruption [28]. Zhou et al. [78] propose Full Attention Networks to boost the robustness of ViTs against OOD images. Mao et al. [42] first systematically analyze the relationship between different components in a ViT, drawing a positive relationship between convolutional components and the robustness of ViTs against adversarial samples and OOD data. By extending [42] and [78], Guo et al. [23] propose input-dependent average pooling in order to adaptively select different aggregation neighborhoods for different tokens, thus achieving the state-of-the-art robust ViTs in OOD generalization. Different interpretations of the self-attention operation can also lead to more robust architectures [24, 58]. However, the methods that introduce additional architectures [23, 24, 58, 78] are either implemented on ViTs with limited size or focus on non-adversarial robustness. On the contrary, our method introduces NCA as lightweight plug-and-play adaptors into base-level ViTs, thus enhancing their clean accuracy and robustness against both adversarial samples and OOD inputs.

2.3 Neural Cellular Automata

Gilpin [20] demonstrates that CA can be represented by convolutional neural networks. By extending [20], Mordvintsev et al. [44] propose NCA in order to mimic the biological cell interactions and model morphogenesis. Following this idea, several works apply NCA in computer vision, including texture synthesis [43, 46, 49, 48, 50], image generation [33, 47, 51, 62], and image segmentation [32, 57]. Randazzo et al. [55] propose applying NCA for modeling collective intelligence on image classification tasks, but the limitation to binary images restricts its practical application. Tesfaldet et al. [62] first establish the connection between ViT and NCA via recurrent local attention. It leads to a more robust model for handling image corruptions in image inpainting tasks. However, their application is limited to image impainting on small datasets such as MNIST [1] and CIFAR10 [36]. The NCA in [62] attempts to emulate a ViT whereas our approach distinguishes itself by not doing so. We first applies NCA in image classification on ImageNet1K with base-level ViT models. Moreover, we propose the new *Dynamic Interaction* for efficient cell interaction modeling, reducing the computational overhead, and for enhancing the model performance.

3 Method

The overview of our method is shown in Figure 2. In this section, we first review the NCA model and ViT architecture. In Section 3.1, we establish the connection between NCA and ViT, in terms of token interaction modeling. We then present the design of our AdaNCA in Section 3.2. We insert AdaNCA into the middle layers of ViT to improve its robustness. We introduce the relationship between the

insert position of AdaNCA and the relative improvements of model robustness in Section 3.3. This relationship leads to the algorithm for deciding the most effective placement for AdaNCA.

3.1 Preliminaries

Vision Transformers Vision Transformers (ViTs) operate on token maps $\mathbf{X} \in \mathbb{R}^{N \times C}$, where the number of tokens is N and each token is represented by a C-dimensional vector. ViTs learn the interaction between these tokens via self-attention [64] and compute the interaction result $\mathbf{X}_{attn}$, as described in Equation 1.

$$\mathbf{X}_{attn} = \sigma\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{D}}\right)\mathbf{V}. \tag{1}$$

$\mathbf{Q}, \mathbf{K}, \mathbf{V}$ stand for query, key, and value, respectively. They are deduced from different linear projections of the input, i.e., $\mathbf{Q} = \mathbf{X}\mathbf{W}_\mathbf{Q}, \mathbf{K} = \mathbf{X}\mathbf{W}_\mathbf{K}, \mathbf{V} = \mathbf{X}\mathbf{W}_\mathbf{V}$, where $\mathbf{W}_\mathbf{Q}, \mathbf{W}_\mathbf{K}, \mathbf{W}_\mathbf{V} \in \mathbb{R}^{C \times D}$. D is the hidden dimensionality in self-attention. σ is Softmax. After self-attention, tokens are fed into a Multilayer Perceptron (MLP) to obtain the updated representations $\mathbf{X}_{out}$:

$$\mathbf{X}_{out} = f_\theta(\mathbf{X}_{attn}). \tag{2}$$

f is the MLP and θ stands for its parameters. The self-attention and MLP form a single ViT block, and a ViT model can be built via stacking ViT blocks.

Neural Cellular Automata NCA aims at modeling cell interactions. In the 2D domain, cells live on a 2D grid with size $H \times W$. Each cell is represented by a vector with dimensionality C. All cells collectively define the cell states $\mathbf{S} \in \mathbb{R}^{H \times W \times C}$. In a single step of NCA, to generate the interaction output $\mathbf{S}_\mathcal{I}$, cells first interact with their neighborhoods for an information exchange in the *Interaction* stage [49]; the interaction is typically instantiated via depth-wise convolutions [44, 46, 48]:

$$\mathbf{S}_\mathcal{I} = (\mathbf{S} \circledast [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_\mathcal{M}])_\oplus. \tag{3}$$

$\mathcal{C}_i$ is the ith convolutional kernel, and $\mathcal{M}$ denotes the total number of kernels. '$\circledast$' denotes depth-wise convolution. The kernels can either be fixed [49, 48] or learnable [44, 46]. The results of all kernels are concatenated channel-wise in the $\oplus$ operation. $\mathbf{S}_\mathcal{I}$ is then passed to an MLP in the *Update* stage [49]:

$$\mathbf{S}_{out} = f_\theta(\mathbf{S}_\mathcal{I}) \odot \mathbf{M}. \tag{4}$$

$\mathbf{S}_{out}$ is then used to update the cell states in a residual scheme. f is the MLP, and θ stands for its parameters. Typically, NCA uses the simplest MLP, with two linear layers and one activation between them. $\mathbf{M} \in \mathbb{R}^{H \times W \times C}$, sampled from $Bernoulli(p)$, is a random binary mask to introduce stochasticity in NCA; it ensures asynchronicity during the cell updates [45]. $\odot$ is point-wise multiplication. NCA learns an underlying dynamic that governs the cell behaviors [50], as depicted by the stochastic differential equation (SDE) in Equation 5:

$$\frac{\partial \mathbf{S}}{\partial t} = \mathcal{F}_\Theta(\mathbf{S}) \odot \mathbf{M} = f_\theta\left[(\mathbf{S} \circledast [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_\mathcal{M}])_\oplus\right] \odot \mathbf{M}. \tag{5}$$

$\mathcal{F}$ represents operations in *Interaction* as well as *Update* stages. Θ is the set containing trainable parameters in the two stages. Discretizing the SDE with $\Delta t = 1$ naturally results in a recurrent residual update scheme:

$$\mathbf{S}^{t+1} = \mathbf{S}^t + \mathcal{F}_\Theta(\mathbf{S}^t) \odot \mathbf{M}. \tag{6}$$

After $t = T$ steps, the cell states S^T is extracted to accomplish certain downstream tasks. The traditional NCA involves several other specific designs though, in our case, it is impractical to adapt them. We provide a discussion on this topic in Appendix E.

3.1.1 Connecting NCA and ViT

Both NCA and ViT learn interactions between a set of elements, i.e., tokens in ViT and cells in NCA. Hereafter, we refer to a cell in NCA as a token, aligning it with the concept in ViTs. The asynchronicity [45] introduced by the random mask $\mathbf{M}$ can be regarded as a cell-wise stochastic depth [31], a more fine-grained version of the sample-wise stochastic depth. In previous NCA works, stochasticity is maintained during testing [46]. Such a scheme is problematic in our case because (1) test-time stochasticity produces obfuscated gradients [3], leading to the circumvention of adversarial

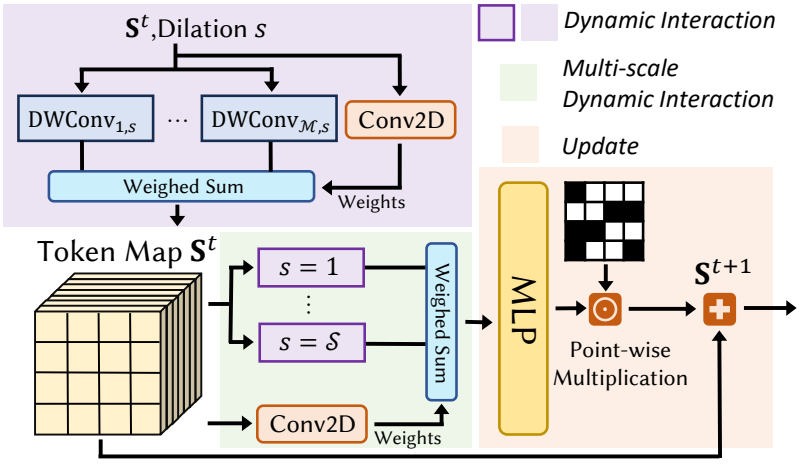

Figure 3: Overview of AdaNCA architecture. Instead of concatenating the interaction results generated by the depth-wise convolutions, our *Dynamic Interaction* conducts a point-wise weighted sum on them to improve the efficiency and enhance the performance. The weights are obtained based on the token states so that each token can dynamically adjust, according to the inputs, the interaction strategy. The *Multi-scale Dynamic Interaction* aggregates the results from *Dynamic Interaction* , where the convolutions have different dilation rates. Then, to finish one step of evolution, the output is fed into the *Update* stage .

attacks, and (2) the model can output different results given the same inputs. To this end, we adopt the strategy in dropout-like techniques [31, 59], which compensates activation values during training. Given $\mathbf{M} \sim Bernoulli(p)$, the evolution of NCA in AdaNCA is defined as:

$$\textbf{Train}: \mathbf{S}^{t+1} = \mathbf{S}^t + \frac{\mathcal{F}_\Theta(\mathbf{S}^t)}{p} \odot \mathbf{M}; \quad \textbf{Test}: \mathbf{S}^{t+1} = \mathbf{S}^t + \mathcal{F}_\Theta(\mathbf{S}^t). \tag{7}$$

We discuss the necessity of such a scheme in Section D.1 in the Appendix. Furthermore, NCA typically outputs the cell states at a random time step T, resulting in random update steps for all cells. Such randomness ensures the stability of NCA across various time steps [44]. Finally, the recurrent steps of NCA during a single training epoch enable the exploration of a wide range of cell states. In the early stage of training, the model is not adequately trained hence serves as a source of noise to itself through the recurrence. With all these components, the trained model can effectively handle the variability and unpredictability of the input thus be robust against noisy input. Our ablation studies in Section 4.3 demonstrate the effectiveness of these strategies in enhancing the model performance.

3.2 AdaNCA architecture

The architecture of AdaNCA is shown in Figure 3. It shares a similar update scheme with the standard NCA, as described in Equation 7; but, it is more computationally efficient due to the proposed *Dynamic Interaction* stage. All convolutional kernels for token interaction are trainable. In the following paragraphs, we first present the design of the *Dynamic Interaction* stage and then introduce a way for more efficient token interaction by using multi-scale *Dynamic Interaction*.

3.2.1 Dynamic Interaction

The *Interaction* stage in the original NCA performs a channel-wise concatenation of the $\mathcal{M}$ output tensors from different depth-wise convolutions. Whereas, our *Dynamic Interaction* computes a weighted sum of those results. Specifically, a weight computation network $\mathcal{W}_\mathcal{I}$ takes the token map $\mathbf{S} \in \mathbb{R}^{H\times W\times C}$ as input and outputs per-token scalar weights $\mathbf{W}_{\mathcal{I}m} \in \mathbb{R}^{H\times W\times 1}$ for each of the $\mathcal{M}$ kernels. We modify Equation 3 to Equation 8:

$$\mathbf{S}_{D\mathcal{I}} = \sum(\mathbf{S} * \mathcal{W}_\mathcal{I})(\mathbf{S} \circledast [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_\mathcal{M}]) = \sum(\mathbf{S} * \mathcal{W}_\mathcal{I})\mathbf{S}_m = \sum_{m=1}^{\mathcal{M}} \left(\sum_{c=1}^{C} \mathbf{W}_{\mathcal{I}m} \odot \mathbf{S}_{mc} \right), \tag{8}$$

where $\mathbf{S}_{mc} \in \mathbb{R}^{H\times W\times 1}, \mathbf{S}_{D\mathcal{I}}, \mathbf{S}_m \in \mathbb{R}^{H\times W\times C}$.'$*$' denotes the convolution. Recall that '$\circledast$' is the depth-wise convolution. We instantiate the weight computation module $\mathcal{W}_\mathcal{I}$ by using a two-layer convolutional network. The first layer transforms the dimensionality from C to $\mathcal{M}$, and the second layer computes the actual weights, thus producing $\mathcal{M}$ scalars for each token. Both layers use 3×3 convolutions to factor in information from both the token and its neighbors. To stabilize training, we add a batch normalization between the two convolutions. Our design of the weight computation network coincides with the one in [23]. Although, our focus is on extracting various information from the same neighborhoods rather than on aggregating data from different neighborhoods.

3.2.2 Multi-scale Dynamic Interaction

Inspired by [48], which uses multi-scale token *Interaction* to facilitate long-range token communication, we propose multi-scale *Dynamic Interaction*. Concretely, all convolutions in Equation 8 now have one more degree of freedom in dilation. Dilation s represents the current operating scale being s, and $s \in \{1, 2, ..., \mathcal{S}\}$. Hence, the original *Dynamic Interaction* is a special case where $\mathcal{S} = 1$. To increase the feature expressivity, we perform a weighted sum on the outputs of all scales, where the per-token weights $\mathbf{W}_{Ms}$ are generated by a network $\mathcal{W}_M$ as described in Equation 9.

$$\mathbf{S}_{MD\mathcal{I}} = \sum(\mathbf{S} * \mathcal{W}_M)\mathbf{S}_{D\mathcal{I}} = \sum(\mathbf{S} * \mathcal{W}_M)\mathbf{S}_{D\mathcal{I}} = \sum_{s=1}^{\mathcal{S}} \left(\sum_{c=1}^{C} \mathbf{W}_{Ms} \odot \mathbf{S}_{D\mathcal{I}sc} \right), \tag{9}$$

where $\mathbf{W}_{Ms}, \mathbf{S}_{D\mathcal{I}sc} \in \mathbb{R}^{H\times W\times 1}, \mathbf{S}_{MD\mathcal{I}}, \mathbf{S}_{D\mathcal{I}s} \in \mathbb{R}^{H\times W\times C}$. The $\mathcal{W}_\mathcal{I}$ in Equation 8 is shared across all scales. The weight computation network $\mathcal{W}_M$ mirrors that of $\mathcal{W}_\mathcal{I}$.

3.3 Insert positions of AdaNCA

Given a ViT and an AdaNCA, to maximize the robustness improvements, we need to determine where to insert AdaNCA. To this end, we first establish the correlation between the placement of AdaNCA and the robustness enhancement it brings. Motivated by the fact that the network redundancy contributes to the model robustness [28], we hypothesize that the effect of AdaNCA should correlate to the layer redundancy corresponding to the insert position. To quantify the redundancy, we propose the **Set Cohesion Index** κ. Given a trained model with L layers and two layer indices $i, j \in \{1, 2, ..., L\}$ where $i < j$, $\kappa(i, j)$ is defined in Equation 10.

$$\begin{aligned}\kappa(i,j) = \frac{1}{(j-i+1)^2} \sum_{m,n\in[i,j]} Sim(m,n) - \frac{\mathbb{1}_{i>1}}{(i-1)(j-i+1)} \sum_{m_1\in[1,i-1],n\in[i,j]} Sim(m_1,n) \\ - \frac{\mathbb{1}_{j<L}}{(L-j)(j-i+1)} \sum_{m\in[i,j],n_1\in[j+1,L]} Sim(m,n_1)\end{aligned} \tag{10}$$

$\mathbb{1}$ stands for the indicator function. $Sim(m, n)$ is the function for quantifying the output similarity between layer m and n. We choose Centered Kernel Alignment (CKA) [35], a common metric for measuring layer similarities inside or between neural networks [25]. A higher κ stands for a more cohesive layer set defined by layers from i to j. Inserting AdaNCA after layer i would partition the network into two layer sets, and we can compute the sum of κ of the layers before and after i, *i.e.*, $\mathcal{K}(i) = \kappa(1, i) + \kappa(i + 1, L)$. This serves as a quantification of the network redundancy that corresponds to position i. We assume that AdaNCA will not change the layer similarity structure because it is too small compared to a single layer in all ViTs.

In addition to the quantification of the network redundancy, the robustness improvement, brought by AdaNCA, is quantified using the relative increase in the attack failure rate of the AdaNCA-inserted models and the corresponding baseline. Specifically, if a model can achieve α clean test accuracy as well as α' accuracy under adversarial attacks, the attack failure rate is $\beta = \frac{\alpha'}{\alpha}$. The robustness improvement γ is then defined as $\gamma = \frac{\beta_{AdaNCA} - \beta_{base}}{\beta_{base}}$. In our experiments, we find that γ is significantly correlated with the network redundancy $\mathcal{K}$ (Pearson correlation $r = 0.6938, p < 0.001$). We refer readers to Appendix A for details of the experiments. The results validate our hypothesis and indicate that AdaNCA should be inserted into the position that can maximize network redundancy. We develop a dynamic programming algorithm to find these positions and refer readers to Appendix A.1 for the details.

Table 1: The performance of AdaNCA-enhanced ViTs and their corresponding baselines. We report the mCE for IM-C (lower is better) and accuracy for other benchmarks. AdaNCA consistently improves the clean accuracy as well as the robustness to adversarial and out-of-distribution inputs of the baseline models. Note that our models also outperform the larger baselines (* sign), indicating that the performance improvement does not merely originate from the increase in the number of parameters or FLOPS. The TAPADL method [23] can lead to more vulnerable models compared to the baselines (green numbers). **Bold** indicates the best model. †: the test is conducted on models pre-trained on ImageNet22K [42], see Appendix C.9.

Model	Params (M)	FLOPS (G)	ImageNet	Adversarial Inputs					OOD inputs		
			Clean Acc.	PGD [41]	CW [5]	APGD-DLR [9]	APGD-CE [9]	IM-A [13]	IM-C (↓) [28]	IM-R [30]	IM-SK [67]
RVT-B [42]	**88.5**	**17.7**	82.7	29.9	21.5†	21.9†	31.4†	28.5	46.8	48.7	36.0
TAPADL-RVT [23]	89.4	17.9	83.1	27.6	19.3	17.7	26.8	**32.7**	44.7	50.2	38.6
RVT-B-AdaNCA	91.0	19.0	**83.3**	**36.7**	**30.2**	**33.2**	**36.2**	31.9	**43.2**	**51.7**	**39.0**
FAN-B [78]	**50.4**	**11.7**	83.9	15.0	7.6	10.4	13.1	39.6	46.1	52.7	40.8
TAPADL-FAN [23]	50.7	11.8	**84.3**	18.6	9.2	13.5	16.9	42.3	**43.7**	**54.6**	40.7
FAN-B-AdaNCA	51.7	12.4	84.1	**20.3**	**10.6**	**14.1**	**19.1**	**42.9**	44.7	53.4	**41.0**
Swin-B [38]	**87.8**	**15.4**	83.4	21.3	13.4	15.6	23.1	35.8	54.3	46.6	32.4
Swin-B* [38]	94.1	16.7	83.3	22.8	14.6	15.9	23.8	35.2	53.2	46.9	33.7
Swin-B-AdaNCA	90.7	16.3	**83.7**	**24.1**	**20.5**	**25.1**	**24.8**	**36.0**	**51.5**	**48.2**	**35.5**
ConViT-B [16]	**86.5**	**17.7**	**82.4**	21.2	8.9	16.9	20.3	29.0	46.9	48.4	35.7
ConViT-B* [16]	93.6	19.2	82.7	24.1	10.0	20.5	23.9	30.1	45.2	49.9	37.8
ConViT-B-AdaNCA	89.0	19.0	**83.2**	**29.2**	**20.1**	**26.3**	**28.4**	**33.0**	**44.3**	**51.1**	**39.1**

Table 2: Comparisons of corruption error on each corruption type of ImageNet-C. AdaNCA consistently improves the performance of the baseline model in all categories (Swin-B), and can achieve better results in most categories compared to the SOTA method (TAPADL-RVT). **Bold** indicates the best model. Lower is better.

Model	mCE	Noise			Blur				Weather				Digital			
		Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG
Swin-B [38]	54.3	43.8	45.7	43.8	61.6	73.2	55.5	66.5	50.0	47.4	47.9	42.1	38.7	68.8	66.4	62.6
Swin-B-AdaNCA	**51.5**	**39.5**	**40.9**	**39.8**	**59.5**	**69.5**	**55.0**	**65.6**	**49.4**	**47.1**	**40.8**	**40.6**	**37.0**	**66.3**	**59.9**	**61.1**
TAPADL-RVT [23]	44.7	34.6	36.1	34.0	53.9	62.3	52.2	**60.1**	**41.9**	44.9	35.8	37.4	31.9	57.1	41.1	47.2
RVT-B-AdaNCA	**43.2**	**33.9**	**36.1**	**33.9**	**52.7**	**59.1**	**47.8**	60.5	42.6	**39.0**	**33.9**	**36.8**	**31.7**	**53.4**	**40.0**	**46.2**

4 Experiments

We use four ViT models as the baseline: Swin-base (Swin-B) [38], FAN-Base-Hybrid (FAN-B) [78], RVT-Base-Plus (RVT-B) [42], and ConViT-Base (ConViT-B) [16]. They include a hierarchical model (Swin), a convolution-attention hybrid model (FAN), and two regular models in which all layers share the same structure (RVT and ConViT). RVT is specifically built to be a robust model, whereas ConViT is not. All four models are equipped with different kinds of local structures. We use two SOTA models in terms of robustness against out-of-distribution (OOD) data, TAPADL-RVT and TAPADL-FAN [23] for comparison. Note that the SOTA method involves training with an additional loss (ADL) and that, for completeness, we keep the results from the TAPADL models trained with such a loss. However, as our focus is on the effect of architectural changes, we do not incorporate the ADL loss in training the AdaNCA-enhanced ViTs. We follow the training scheme for each model, respectively, to train the AdaNCA-equipped model from scratch on ImageNet1K. We conduct the analysis presented in Section 3.3 to decide the optimal position to insert multiple AdaNCA modules, in which the ImageNet1K pre-trained weights for analysis are obtained from the PyTorch Image Models library [69]. To balance between the computational cost and robustness improvement, we limit the number of AdaNCA to two or three, depending on the model architecture. The recurrent time step of all AdaNCA is chosen from $\{[2, 2], [2, 4], [3, 5]\}$, and we follow the design principle of ViT; it is to put more computation in the middle or high layers [38]. **We fix the random steps during testing to a single integer chosen from the range** to achieve precise results and ensure non-stochasticity during adversarial attacks. All the activation functions used in the MLP in AdaNCA are GELU [29], and the input, hidden, and output dimensionalities of the MLP are all the same. We refer readers to Appendix C.6 for the details of the training and insert scheme of AdaNCA. All of our experiments are performed on four Nvidia A100 GPUs.

4.1 Results on Image Classification

We test all models on the ImageNet1K validation set for clean accuracy. For the adversarial robustness evaluation, we choose common adversarial attack methods PGD [41], CW [5], APGD-DLR [9], and APGD-CE [9]. Moreover, we also include the natural adversarial examples in ImageNet-A

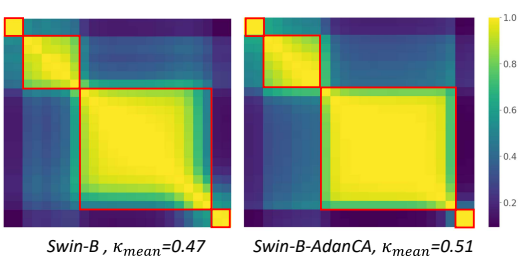

Figure 4: Pair-wise layer similarities. Layer sets are marked in red boxes. Swin-B-AdanCA has a clearer stage partition, which might be attributed to AdaNCA acting as an information transmitter between different layer sets.

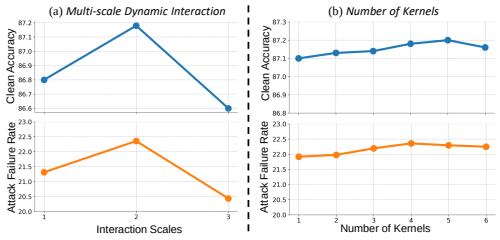

Figure 5: Ablation on the (a) scales and (b) number of kernels used in our multi-scale *Dynamic Interaction*. Overly large scales can undermine the performance and so do too many or too few kernels. We choose $\mathcal{S} = 2$, $\mathcal{M} = 4$ to balance between the clean accuracy and robustness.

(IM-A) [13]. For the PGD attack, we align with the settings used in [42]: max magnitude $\epsilon = 1$, step size $\alpha = 0.5$, steps $t = 5$. We refer readers to the Appendix C.7 for details of the other attacks. For testing the OOD generalization, we use ImageNet-C (IM-C) [28], ImageNet-R (IM-R) [30], and ImageNet-Sketch (IM-SK) [67]. We report the mean corruption error (mCE) on ImageNet-C and accuracy on all other kinds of robustness benchmarks in Table 1. Our results highlight that AdaNCA-enhanced ViTs consistently outperform corresponding baselines in various robustness tests as well as in terms of clean accuracy. Importantly, the enlarged baseline models ($\star$ sign) do not bring comparable improvements to AdaNCA, suggesting that the enhancements do not merely stem from the increase in computational budgets. However, the existing method [23] that introduces local structure into ViTs can potentially undermine the adversarial robustness of the baselines. In Table 2, we conduct an in-depth study on the corruption errors of the different categories of common corruptions in ImageNet-C. The results show that AdaNCA enhances robustness without the trade-off seen in methods that ignore texture information [17, 53]. While these methods may improve mCE for non-Blur noise, they often worsen mCE for Blur noise [53]. In contrast, AdaNCA consistently improves robustness across most categories. We refer readers to Appendix C for more results.

4.2 Layer similarity structure

Our key assumption in Section 3.3 is that AdaNCA will not change the layer similarity structure due to its small size, and that is why we use **pre-trained** networks to conduct this analysis. Here, we examine the pair-wise layer similarities in Swin-B [38] and Swin-B-AdaNCA in Figure 4. κ_{mean} is the mean of κ from all layer sets. We refer readers to Appendix C.14 for more results. AdaNCA not only preserves the original layer similarity structure but also contributes to a clearer stage partition, validating our assumption in Section 3.3. The results might be attributed to the fact that AdaNCA transmits information between different layer sets and thus layers inside each set do not bother adapting to the layers outside the set.

4.3 Ablation studies

We conduct ablation studies on ImageNet100, a 100-class subset of ImageNet1K. Previous studies [7, 15, 72, 74] have shown that ImageNet100 serves as a representative subset of ImageNet1K. Hence, we can obtain representative results for the self-evaluation of the model while efficiently using the computational resources. All the ablation experiments are based on a Swin-tiny [38] model. We insert it after the fourth layer to obtain the best robustness improvement, according to our analysis in Section 3.3 and Appendix A. First, we ablate on two hyperparameters, the number of convolutional kernels used in the *Dynamic Interaction* stage ($\mathcal{M}$) and the maximum scale ($\mathcal{S}$) used in the multi-scale *Dynamic Interaction* stage. The Clean Accuracy and Attack Failure Rate are shown in Figure 5. The attack failure rate is quantified using the same method as in Section 3.3 and Appendix A. Multi-scale interaction can contribute to the performance while overly large scale can lose local information and complicate the process of selecting the interaction neighbors. This issue is also observed in a previous work [62]. Increasing the number of kernels benefits the performance while too many

Table 3: Ablation studies on design choices. Each of the AdaNCA design choices contributes to improved performance and robustness. Baseline is a Swin-tiny [38] model trained on ImageNet-100. **Bold** indicates the best model.

Exp. Type	Recur	StocU	RandS	DynIn	Params (M)	FLOPS (G)	Accuracy (↑)	Attack Failure Rate (↓)
Baseline	✗	✗	✗	✗	27.59	4.5	86.56	12.29
Ablation	✗	✓	✗	✓	28.97	4.7	**87.36**	19.04
	✓	✗	✓	✓	27.94	4.7	86.92	19.56
	✓	✓	✗	✓	27.94	4.7	87.12	19.34
	✓	✓	✓	✗	27.93	4.7	86.72	21.98
Ours	✓	✓	✓	✓	27.94	4.7	87.18	**22.35**

kernels undermine the robustness. According to the results, we choose $\mathcal{S} = 2$, $\mathcal{M} = 4$. We then perform ablations on several design choices:

- **Recurrent update (Recur).** Ablation on unrolling the recurrence with average time step T in AdaNCA into T independent AdaNCA with time step being 1.
- **Stochastic update (StocU).** Ablate the stochastic update during training, leading to globally synchronized update of all tokens [45].
- **Random step (RandS).** Change the recurrence time step from a randomly chosen integer in range $[T_1, T_2]$ to $\lceil (T_1 + T_2)/2 \rceil$. It cannot be turned on without recurrence.
- **Dynamic interaction (DynIn).** Ablate the *Dynamic Interaction* so that the interaction results are simply summed together. The number of kernels remains the same.

As shown in Table 3, the highest robustness improvement is achieved with all components. Among them, turning off the recurrence leads to the largest drop in the robustness, as recurrence allows the model to explore more cell states than finishing the update in a single step. While it achieves the highest clean accuracy, it uses ∼4x more parameters than our method, and the improvement of the clean performance is likely due to more parameters. Without any of the two sources of randomness, stochastic update and random steps, the model cannot adapt to the variability of the inputs and thus exhibits vulnerabilities against adversarial attacks. Finally, turning off our Dynamic Interaction will cause drops in both clean accuracy and robustness, as tokens cannot decide their unique interaction weights and thus cannot generalize to noisy inputs. We provide extended ablation studies in Section D.2 in the Appendix.

4.4 Noise sensitivity examination

A drawback of adversarially robust models is their increased sensitivity to noise in specific frequency bands [60]. For instance, while adversarially trained models are robust against adversarial attacks, they can be sensitive to noise from a larger frequency-band range compared to standard models [60]. Here, we use the method and data from [60] to examine the noise sensitivity of AdaNCA-enhanced ViTs. Specifically, we evaluate the classification performance on a set of images that are mixed with noise of varying magnitudes and frequencies. The noise is sampled from Gaussian distribution with zero mean and the standard deviation indicates its magnitude. Then, it is filtered within various spatial-frequency bands, resulting in different frequencies of noise. Higher classification accuracy on a specific noise type indicates that the model is less sensitive to that noise. Figure 6 presents the results, including human data sourced from [60]. Our findings demonstrate that AdaNCA enhances ViTs by reducing the sensitivity to noise with certain frequency components, equipping ViTs with more human-like noise-resilient abilities. Crucially, AdaNCA improves the model robustness differently than adversarial training since the AdaNCA-enhanced ViTs do not exhibit increased sensitivity to the noise. We quantitatively validate the conclusion and refer readers to Appendix C.16 for the details.

5 Limitation

AdaNCA has certain limitations. First, the AdaNCA-equipped ViTs cannot adapt to unseen recurrent steps of AdaNCA, which limits the generalization ability. For example, if the training step range of AdaNCA is [3,5], it cannot produce meaningful results with the test step being 6. AdaNCA

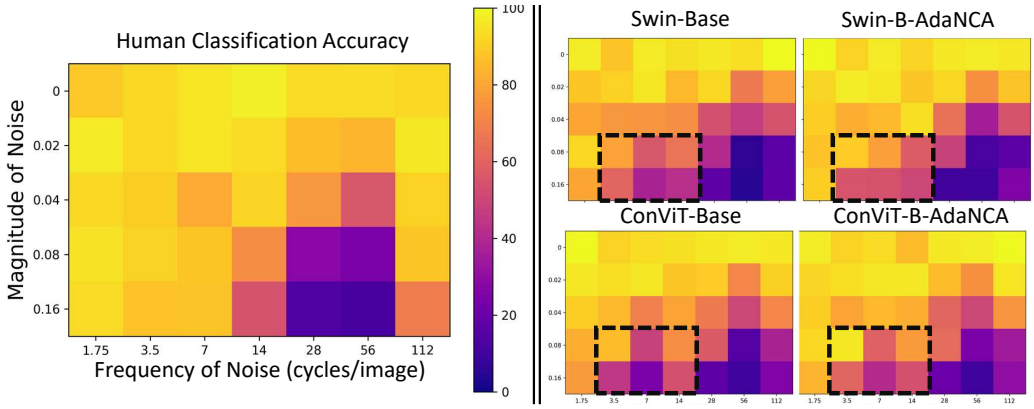

Figure 6: Classification accuracy of humans and ViTs on noisy images. The images are perturbed by Gaussian noise with different standard deviations (Magnitude of Noise) and are filtered with different spatial-frequency bands (Frequency of Noise). AdaNCA improves the accuracy on images with certain types of noise (**dotted black boxes**), indicating that it makes ViTs less sensitive to them. Quantitative results are presented in Appendix C.16.

introduces a non-negligible computation into the original architecture. Our experiments are conducted on ImageNet1K with the image size of 224×224. Whether AdaNCA can lead to impressive improvements on larger-scale problems, *e.g.,* ImageNet22K, remains a question. The size of the images can also affect the efficiency of token interaction.

6 Broader Impact

AdaNCA contributes to more robust ViTs, facilitating their usage in real-world scenarios. We bridge two powerful models, NCA and ViT, on large-scale image classification, potentially encouraging research dedicated to their synergistic combination in more practical settings. Our findings on AdaNCA improving network redundancy can stimulate more works on architectural robustness in deep learning that involves increasing the redundancy to enhance robustness. In the context of this paper, we believe that AdaNCA does not introduce any significant negative implications.

7 Conclusion

We have proposed AdaNCA, an efficient Neural Cellular Automata (NCA) that, when inserted between their middle layers, improves ViT performances and robustness against adversarial attacks as well as out-of-distribution inputs. We design our model by connecting NCA and ViT, in terms of token interaction modeling, and we propose *Dynamic Interaction* to improve the computational efficiency of standard NCA. Exploiting the training strategies and design choices in NCA, *i.e.,* stochastic update, random steps, and multi-scale interaction, we further improve the AdaNCA-enhanced ViTs' clean accuracy and robustness. To decide the placement of AdaNCA, we propose the Set Cohesion Index that quantifies the network redundancy via layer similarity and conclude that AdaNCA should be inserted between two layer sets that consist of redundant layers. Our results demonstrate that AdaNCA consistently improves ViTs performances and robustness. Evidence suggests that the mechanism by which we obtain improvement reduces the sensitivity of ViTs to certain types of noise and makes the noise-resilient ability of ViTs similar to that of humans.

References

[1] The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*.

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018.

[4] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10231–10241, 2021.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[6] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[7] Yixin Cheng, Grigorios Chrysos, Markos Georgopoulos, and Volkan Cevher. Multilinear operator networks. In *The Twelfth International Conference on Learning Representations*, 2023.

[8] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[9] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020.

[10] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. CoAtNet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

[11] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[13] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, et al. On robustness and transferability of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16458–16468, 2021.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[15] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022.

[16] Stéphane D'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. ConViT: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021.

[17] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018.

[18] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[19] Jacob Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

[20] William Gilpin. Cellular automata as convolutional neural networks. *Physical Review E*, 100 (3):032402, 2019.

[21] Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024.

[22] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. CMT: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022.

[23] Yong Guo, David Stutz, and Bernt Schiele. Robustifying token attention for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17557–17568, 2023.

[24] Xing Han, Tongzheng Ren, Tan Nguyen, Khai Nguyen, Joydeep Ghosh, and Nhat Ho. Designing robust transformers using robust kernel density estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

[25] Yena Han, Tomaso A Poggio, and Brian Cheung. System identification of neural systems: If we got it right, would we know? In *International Conference on Machine Learning*, pages 12430–12444. PMLR, 2023.

[26] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

[29] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[30] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, 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*, pages 8340–8349, 2021.

[31] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016.

[32] John Kalkhof, Camila González, and Anirban Mukhopadhyay. Med-NCA: Robust and lightweight segmentation with neural cellular automata. In *International Conference on Information Processing in Medical Imaging*, pages 705–716. Springer, 2023.

[33] John Kalkhof, Arlene Kühn, Yannik Frisch, and Anirban Mukhopadhyay. Frequency-time diffusion with neural cellular automata. *arXiv preprint arXiv:2401.06291*, 2024.

[34] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

[35] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[36] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[37] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, Michele Magno, Luca Benini, and Luc Van Goo. LocalViT: Analyzing locality in vision transformers. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9598–9605. IEEE, 2023.

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[39] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin Transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[41] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[42] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022.

[43] Alexander Mordvintsev and Eyvind Niklasson. μ-NCA: Texture generation with ultra-compact neural cellular automata. *arXiv preprint arXiv:2111.13545*, 2021.

[44] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 2020. doi: 10.23915/distill.00023. https://distill.pub/2020/growing-ca.

[45] Eyvind Niklasson, Alexander Mordvintsev, and Ettore Randazzo. Asynchronicity in neural cellular automata. In *Artificial Life Conference Proceedings 33*, volume 2021, page 116. MIT Press One Rogers Street, Cambridge, MA 02142-1209, 2021.

[46] Eyvind Niklasson, Alexander Mordvintsev, Ettore Randazzo, and Michael Levin. Self-organising textures. *Distill*, 6(2):e00027–003, 2021.

[47] Maximilian Otte, Quentin Delfosse, Johannes Czech, and Kristian Kersting. Generative adversarial neural cellular automata. *arXiv preprint arXiv:2108.04328*, 2021.

[48] Ehsan Pajouheshgar, Yitao Xu, Tong Zhang, and Sabine Süsstrunk. DyNCA: Real-time dynamic texture synthesis using neural cellular automata. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20742–20751, 2023.

[49] Ehsan Pajouheshgar, Yitao Xu, Alexander Mordvintsev, Eyvind Niklasson, Tong Zhang, and Sabine Süsstrunk. Mesh neural cellular automata. *ACM Trans. Graph.*, 2024. doi: 10.1145/3658127. URL `https://doi.org/10.1145/3658127`.

[50] Ehsan Pajouheshgar, Yitao Xu, and Sabine Süsstrunk. NoiseNCA: Noisy seed improves spatio-temporal continuity of neural cellular automata. *arXiv preprint arXiv:2404.06279*, 2024.

[51] Rasmus Berg Palm, Miguel Gonzalez Duque, Shyam Sudhakaran, and Sebastian Risi. Variational neural cellular automata. In *International Conference on Learning Representations 2022*, 2022.

[52] Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 367–376, 2021.

[53] Xinkuan Qiu, Meina Kan, Yongbin Zhou, Yanchao Bi, and Shiguang Shan. Shape-biased cnns are not always superior in out-of-distribution robustness. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2326–2335, 2024.

[54] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019.

[55] Ettore Randazzo, Alexander Mordvintsev, Eyvind Niklasson, Michael Levin, and Sam Greydanus. Self-classifying mnist digits. *Distill*, 2020. doi: 10.23915/distill.00027.002. https://distill.pub/2020/selforg/mnist.

[56] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[57] Mark Sandler, Andrey Zhmoginov, Liangcheng Luo, Alexander Mordvintsev, Ettore Randazzo, et al. Image segmentation via cellular automata. *arXiv preprint arXiv:2008.04965*, 2020.

[58] Baifeng Shi, Yale Song, Neel Joshi, Trevor Darrell, and Xin Wang. Visual attention emerges from recurrent sparse reconstruction. In *International Conference on Machine Learning*, pages 20041–20056. PMLR, 2022.

[59] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[60] Ajay Subramanian, Elena Sizikova, Najib Majaj, and Denis Pelli. Spatial-frequency channels, shape bias, and adversarial robustness. *Advances in Neural Information Processing Systems*, 36, 2024.

[61] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[62] Mattie Tesfaldet, Derek Nowrouzezahrai, and Chris Pal. Attention-based neural cellular automata. *Advances in Neural Information Processing Systems*, 35:8174–8186, 2022.

[63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[65] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021.

[66] vtddggg. Code repo for: RVT: Towards robust vision transformer. `https://github.com/vtddggg/Robust-Vision-Transformer`, 2021.

[67] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

[68] Yujing Wang, Yaming Yang, Jiangang Bai, Mingliang Zhang, Jing Bai, Jing Yu, Ce Zhang, Gao Huang, and Yunhai Tong. Evolving attention with residual convolutions. In *International Conference on Machine Learning*, pages 10971–10980. PMLR, 2021.

[69] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[70] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[71] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021.

[72] Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[73] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 30008–30022. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/fc1a36821b02abbd2503fd949bfc9131-Paper.pdf`.

[74] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S2-mlp: Spatial-shift mlp architecture for vision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 297–306, 2022.

[75] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021.

[76] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

[77] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[78] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pages 27378–27394. PMLR, 2022.

Appendix: Table of Contents

	RVT-S-AdaNCA	FAN-S-AdaNCA	Swin-T-AdaNCA
Learning rate	2.5e-4	2.5e-4	2.5e-4
Batch size	256	256	256
Model EMA decay	0.99992	0.9999	/
Stochastic depth	0.1	0.25	0.1
Random erase prob	0.25	0.3	0.25
Gradient clip	None	None	5.0
MixUp	0.8	0.8	0.8
Label smoothing	0.1	0.1	0.1
Min learning rate	1e-5	1e-6	1e-5
Weight decay	0.05	0.05	0.05

Table 4: Hyperparameters used in AdaNCA-equipped ViT training for analysis in Section 3.3 on ImageNet100.

	PGD [41]	CW [5]	APGD-DLR [9]	APGD-CE [9]
Max magnitude	0.5	/	0.5	0.5
Steps	5	20	5	5
Step size	0.25	/	/	/
κ	/	0.0	/	/
c	/	0.25	/	/
ρ	/	/	0.75	0.75
EoT	/	/	1	1

Table 5: Hyperparameters in different adversarial attacks used in the analysis in Section 3.3 on ImageNet100 as well as our ablation studies.

A Establish the correlation between AdaNCA placement and robustness improvement

Our motivation for investigating the relationship between AdaNCA and robustness stems from an empirical observation, namely that making neural networks more monolithic contributes to their robustness [28]. One of these monolithic ideas is redundancy in networks. It has been observed that several consecutive layers output similar results [4], hence building what we call "a layer set." We hypothesize that AdaNCA, as adaptors inside ViTs, should connect the different sets inside a ViT and transmit information between them. In this way, layers inside a set will no longer bother adapting to other layers outside the set, improving the redundancy and thus robustness. To this end, we propose the layer redundancy quantification κ and network redundancy quantification $\mathcal{K}$, as well as the robustness improvement measurement γ in Section 3.3.

For the robustness improvement, we insert AdaNCA into all possible positions of 3 ViT models, Swin-tiny [38], FAN-small-hybrid [78], and RVT-small-plus [42], and train them along with the baseline models on the ImageNet100 dataset [15, 72, 74], a representative 100-class subset of ImageNet1K. The total amount of models is 34, including 3 baselines and 31 AdaNCA models. All AdaNCA-models have a 1%-2% parameter increase and 5% more FLOPS. The training hyperparameters are given in Table 4. All models are trained for 300 epochs with the Cosine scheduler. Here, we only consider adversarial robustness for simplification, as it is shown that adversarial robustness is correlated with image corruption robustness within one backbone architecture [42]. We use a white-box version of AutoAttack [9] in the AdaNCA placement analysis in Section 3.3, which comprises PGD [41], CW [5], APGD-DLR [9], and APGD-CE [9]. The hyperparameters of the four attacks are in Table 5.

We show the qualitative results of our experiments on the Set Cohesion Index and robustness improvement in Figure 7 and the quantitative results in Figure 8. Qualitatively, γ generally follows the trend of $\mathcal{K}$ except for the top 3 layers and the last layer. Specifically, γ for the top three layers consistently exhibits a pattern where it decreases in the second layer and increases in the third layer, regardless of the trend in $\mathcal{K}$, which always increases. We hypothesize that the proximity of the top 3 layers to the input image causes AdaNCA to similarly impact the model's robustness by adapting the

input to the subsequent layers. Furthermore, the position immediately before the final layer likely serves as a transitional stage for output, adapting to different output strategies (e.g., FAN has an additional class attention head while Swin and RVT use average pooling on all tokens to generate features for classification). The distinct behavior of the last layer compared to other layers has also been noted in previous research [21]. Hence, we exclude the models (AdaNCA applied in the top three layers and before the final layer) from our analysis. The resulting number of models is 19. To conduct a cross-model quantitative comparison, we normalize all γ as well as all $\mathcal{K}$ in a single type of model. Each of the 3 models thus has a set of $\gamma_{norm} \in [0, 1]$ as well as $\mathcal{K}_{norm} \in [0, 1]$. We plot all sets of γ_{norm} and $\mathcal{K}_{norm}$ from 3 models in Figure 8. Note that the coordinate (1.0,1.0) has two overlapping points (Swin-tiny and FAN-small-hybrid), hence only 18 points are visible in the figure. γ_{norm} is significantly correlated with $\mathcal{K}_{norm}$ ($r = 0.6938, p = 9.8e-4$), as we report in Section 3.3.

A.1 Dynamic programming for AdaNCA placement

Given $\mathcal{U}$ AdaNCAs, the ViT is expected to be partitioned into $\mathcal{U}+1$ sets. The dynamic programming involves filling an array $\mathcal{D}$, where $\mathcal{D}[i][u]$ represent the maximum value of $\mathcal{K}$ achievable by partitioning the first i layers into u sets. The $\mathcal{D}$ can be obtained via Equation 11.

$$\mathcal{D}[i][u] = \max_{j<i}(\mathcal{D}[j][u-1] + \kappa(j, i-1)) \tag{11}$$

The boundary conditions are: 1) $\mathcal{D}[i][1] = \kappa(1, i)$, which is partitioning the first i layers into 1 set; 2) $\mathcal{D}[0][u] = 0$. By recording the partition position, we can find $\mathcal{U}$ partition points for inserting the AdaNCA. The pseudo-code is presented in Algorithm 1.

Algorithm 1 find_optimal_partition(Similarity Matrix S, Number of Stages $\mathcal{U}$)

$n \leftarrow$ length of S
Initialize $\mathbf{dp}$ with dimensions $(n+1, \mathcal{U}+1)$, filled with $-\infty$
Initialize $\mathbf{partition}$ with dimensions $(n+1, \mathcal{U}+1)$, filled with zeros
$\mathbf{dp}[0][0] \leftarrow 0$ {Base case}
for each i from 1 to n **do**
  for each u from 1 to $\min(i, \mathcal{U})$ **do**
    for each j from 0 to $i-1$ **do**
      current_value $\leftarrow \mathbf{dp}[j][u-1] + \kappa(j, i-1)$
      if current_value $> \mathbf{dp}[i][u]$ **then**
        $\mathbf{dp}[i][u] \leftarrow$ current_value
        $\mathbf{partition}[i][u] \leftarrow j$
      end if
    end for
  end for
end for
Initialize stages as an empty list
current_layer $\leftarrow n$
current_stage $\leftarrow \mathcal{U}$
while current_stage > 0 **do**
  start_layer $\leftarrow \mathbf{partition}$[current_layer][current_stage] $+ 1$
  Append (start_layer, current_layer) to stages
  current_layer $\leftarrow \mathbf{partition}$[current_layer][current_stage]
  current_stage $\leftarrow$ current_stage $- 1$
end while
stages = stages[::-1]
return stages

Note that here the layer index starts from 0, and so does the input requirement of κ.

B Notes on statistical significance

We do not report the error bar for training models with different seeds on ImageNet1K, as it would be very expensive (Table 10). However, we follow the seed used in the released code for each model.

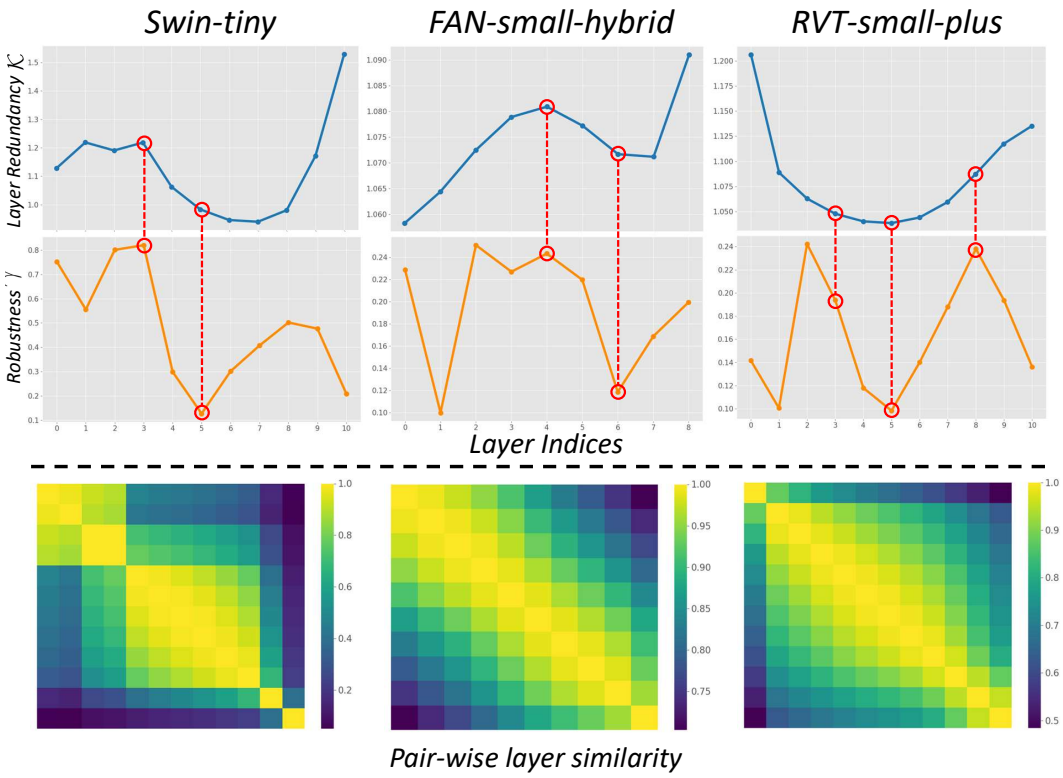

Figure 7: Top: Visualization of layer redundancy ($\mathcal{K}(i) = \kappa(1, i+1)+\kappa(i+2, L)$) and corresponding robustness improvement γ. Bottom: Visualization of pair-wise layer similarity of the 3 ViTs.

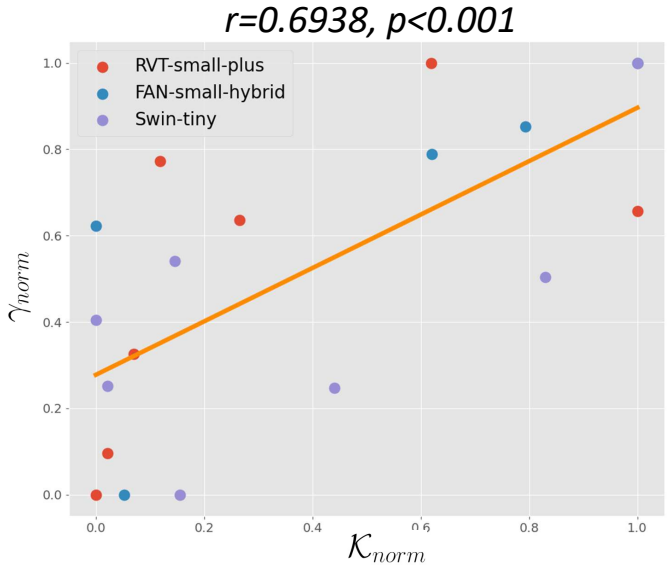

Figure 8: The relationship between the layer redundancy $\mathcal{K}$ and robustness improvement γ by inserting AdaNCA in the corresponding position. γ_{norm} and $\mathcal{K}_{norm}$ are obtained by normalizing γ and $\mathcal{K}$ within the results of each model. Note that the coordinate (1.0,1.0) has two overlapping points (Swin-tiny and FAN-small-hybrid). Quantitatively, γ_{norm} is significantly correlated with $\mathcal{K}_{norm}$ ($r = 0.6938, p < 0.001$). The linear fit is shown in orange.

Table 6: AdaNCA settings for each ViT.

	# Layers	AdaNCA positions (after the index)	Training steps	Test steps	Stochastic update
RVT-B-AdaNCA	12	4,8	[2,4],[3,5]	3,3	0.1
FAN-B-AdaNCA	16	0,6,9	[2,2],[3,5],[2,4]	2,4,3	0.1
Swin-B-AdaNCA	24	2,8,22	[2,2],[3,5],[2,4]	2,4,3	0.1
ConViT-B-AdaNCA	12	5,9	[3,5],[2,4]	3,3	0.1

Table 7: Increasing the number of recurrent steps of AdaNCA during testing can contribute to the model performance.

	# Params (M)	FLOPS (G)	Clean Acc. (↑)	Attack Failure Rate (↑)
Step=4	27.94	4.7	87.18	22.35
Step=5	27.94	4.8	87.22	23.46

We test the adversarial attack using 5 seeds and find the difference between different seeds negligible (standard deviation: PGD 0.08, CW 0.01, APGD-DLR 0.01, APGD-CE 0.02). As all our results on adversarial attacks have differences larger than 0.1, they are statistically significant.

For ImageNet100 experiments in the ablation study, we train the model 3 times and find the clean accuracy changes within a small range (standard deviation 0.08), and the robustness test results barely change (standard deviation 0.02). We also test the robustness improvement within one model using 5 seeds and the standard deviation is 0.008.

For the AdaNCA placement analysis, our correlation result is statistically significant ($p < 0.001$).

C ImageNet1K experiments

C.1 AdaNCA settings for each model

Based on Algorithm 1, we insert AdaNCA into the four chosen baseline models. The specific settings for each AdaNCA are shown in Table 6. Note that insert position 0 means before the network, since FAN-B model uses a complex patch embedding layer, we can treat it as a unique layer that encodes semantic information rather than the simple convolution patch embedding used in other models. In practice, we find inserting AdaNCA after it can contribute to the model performance. Moreover, we add the drop path operation after each AdaNCA, and the drop path rate follows the one of the layer that AdaNCA follows. Hence, the regularization is pretty strong even without the stochastic update. However, as shown in our ablation studies in Section 4.3, the stochastic update indeed contributes to the model performance. Our FLOPS computation for AdaNCA models are based on Test steps.

C.1.1 Discussion on the choice of AdaNCA steps

Our choice of AdaNCA steps differs from all previous NCA models, which typically iterate for hundreds of steps. We make this compromise as the computational costs increase linearly as the step grows. We aim to minimize the increase in the number of parameters and FLOPS for scalability and we do not want the source of improvement to merely stem from the increase in the size and computation of the models. We show that more NCA steps will indeed contribute to the model's performance in Table 7, while it introduces extra FLOPS. The model is the same as in Section 4.3 which is trained using a range of steps being $[3, 5]$. We want to underscore that it is the architecture and evolution scheme that defines an NCA as introduced in Section 3.1, instead of the number of its recurrent steps. Admittedly, fewer steps will lead to a coarser path to the target state, and can potentially undermine the model's performance. Notably, with our scheme, AdaNCA has achieved generally strong robustness improvements compared to the baselines. It indicates that our choice of the recurrent steps of AdaNCA is a good balance between computational costs and performance.

C.2 Discussion on the model choices

In our experiments, we choose RVT [42], FAN [78], Swin [38], and ConViT [16]. Our choice not only includes different types of ViTs (Regular, Hybrid, Hierarchical) and robust as well as non-robust architectures (RVT, FAN and Swin, ConViT), but also covers other aspects of ViT characteristics. First, RVT and Swin use average pooling on the final token maps to obtain the 1D feature for classification, while FAN and ConViT adopt a separate class token. It has been proved that those two strategies of classification can have a significant influence on the model robustness [42]. Despite this, AdaNCA are effective in both ways for classification. Moreover, all the architectures already contain certain kinds of local structures. RVT contains depth-wise convolution within MLP and a convolutional patch embedding layer. FAN has a ConvNeXt [40] head for patch embedding. Swin has shifted window attention. ConViT has convolution-attention coupled gated positional self-attention layers. Local structures have been proven effective in improving model robustness [42]. Therefore, we partially eliminate the possibility that AdaNCA is effective simply because of introducing local information in ViTs without any local inductive bias, such as the original ViT [14] or DeiT [63]. As shown in our ablation studies in Section 4.3, AdaNCA not only contributes to more parameter-efficient models but also improves the model robustness compared to not using the training strategies or design from NCA. Furthermore, in our choices, Swin does not conduct weight exponential moving average (EMA), while the other 3 models perform. It indicates that AdaNCA can be effective with or without model EMA.

C.3 Discussion on the hyperparameters of MLP inside AdaNCA

In AdaNCA, we set the hidden dimensionality of the MLP to the same one as the input dimension. It is different from the common design choice of MLP in ViT that uses a 4-time larger hidden layer than the input layer. Moreover, it also deviates from the design of the previous NCA that uses more than 8 times larger hidden layer size than the input one. Instead, we set the hidden dimensionality to be the same as the input one. The reason is that AdaNCA introduces additional parameters and computation in ViTs and we want the extra computational overhead to be as low as possible. The lowest dimensionality that will not cause inevitable information loss is the same one as the input dimension. Therefore, we design the MLP in AdaNCA as a non-compression-non-expand structure.

C.4 The effect of AdaNCA placement on robustness on ImageNet1K

Our algorithm for deciding the placement of AdaNCA is based on the correlation between the robustness improvement and the Set Cohesion Index in Section 3.3. All the experiments are conducted on ImageNet100 to efficiently use the computational resources. Here, we compare two schemes for deciding the placement of AdaNCA on ImageNet1K on Swin-B [38] and FAN-B [78] to demonstrate our result validity. The first scheme is **No-Prior**, which does not involve the knowledge of the layer similarity and performs the most reasonable placement choice. For Swin-B, the choice for placing 3 AdaNCA is to insert them between each stage pair defined by the transition between different embedding dimensionalities, namely after layer 2, layer 4, and layer 22. Coincidentally, this choice aligns with what we obtain from the dynamic programming algorithm, which is to place AdaNCA after layer 2, layer **8**, and layer 22. For FAN-B, although it is a hybrid ViT, we only consider inserting AdaNCA into its main network where all layers share the same structure. In this case, the **No-Prior** chooses to uniformly insert AdaNCA between the 16 layers, that is after layers 5 and 10. Our algorithm decides the placement should instead be after layers 6 and 9. Table 8 shows the results of two Swin-B-AdaNCA models. Inserting AdaNCA with the **No-Prior** scheme can still contribute to the model performance and robustness, but not as effective as using our proposed method.

C.5 Details of architecture change in Swin-Base* and ConViT-Base*

For Swin-Base*, we add two extra layers in stage 3 with embedding dimensionality being 512, since in the original design most of the computational budgets are dedicated to stage 3, and two layers form a complete Swin operation. For ConViT-Base*, we add an extra Gated Positional Self-Attention (GPSA) layer.

Table 8: Comparison between different schemes of AdaNCA placement. **No-Prior**: insert AdaNCA without knowledge of layer similarity structure but making the most reasonable choice, that is to insert them between the architectural stage transition where the embedding dimensionality changes for Hierarchical ViT like Swin [38], or insert them uniformly between layers that have the same structure like the main network in FAN [78]. AdaNCA consistently improves the model performance and robustness compared to the baseline models, while the improvement can be maximized by using our AdaNCA placement scheme based on Algorithm 1.

Model	**Insert Scheme**	Params (M)	FLOPS (G)	**ImageNet** Top-1	**Robustness Benchmarks** PGD	CW	APGD-DLR	APGD-CE	IM-C (↓)	IM-A	IM-R	IM-SK
Swin-B [38]	/	87.8	15.4	83.4	21.3	13.4	15.6	23.1	54.2	35.8	46.6	32.4
Swin-B-AdaNCA	No-Prior	90.7	16.3	83.6	23.1	19.4	24.8	24.5	52.0	**36.0**	47.7	34.8
Swin-B-AdaNCA	*Ours*	90.7	16.3	**83.7**	**24.1**	**20.5**	**25.1**	**24.8**	**51.5**	**36.0**	**48.2**	**35.5**
FAN-B [78]	/	50.4	11.7	83.9	15.0	7.6	10.4	13.1	46.1	39.6	52.7	40.8
FAN-B-AdaNCA	No-Prior	51.7	12.4	**84.1**	17.3	10.2	13.5	16.3	**44.7**	42.8	**53.5**	**41.0**
FAN-B-AdaNCA	*Ours*	51.7	12.4	**84.1**	**20.3**	**10.6**	**14.1**	**19.1**	**44.7**	**42.9**	53.4	**41.0**

Table 9: Hyperparameters used in AdaNCA-equipped ViT training.

	RVT-B-AdaNCA	FAN-B-AdaNCA	Swin-B-AdaNCA	ConViT-B-AdaNCA
Learning rate	1e-3	2e-3	1e-3	1e-3
Batch size	1024	2048	1024	1024
Model EMA decay	0.99996	0.99992	/	0.99996
Stochastic depth	0.1	0.35	0.5	0.1
Random erase prob	0.25	0.3	0.25	0.25
Gradient clip	None	None	5.0	None
MixUp	0.8	0.8	0.8	0.8
Label smoothing	0.1	0.1	0.1	0.1
Min learning rate	1e-5	1e-6	1e-5	1e-5
Weight decay	0.05	0.05	0.05	0.05

C.6 Training details

We use different training schemes for the four selected baseline models, closely following the parameter settings for each of them (link to reference configuration files: RVT-B,FAN-B,Swin-B,ConViT-B,). Most training settings are in line with DeiT [63] since all models used build the training upon the scheme of training a DeiT but with minor changes. Some specific training settings are listed in Table 9.

All models are trained for 300 epochs. Our ablation studies on the size of the model (Swin-B-abl, ConViT-B-abl) also follow the same training settings. The detailed time consumption for training each model using 4 Nvidia-A100 80G GPUs is in Table 10. The excessive time of RVT is because of the patch-wise augmentation [42].

C.7 Adversarial attacks

We test all AdaNCA-equipped ViTs using four adversarial attacks based on the code [34]. The detailed setting of each attack is given in Table 11. We try setting the Expectation Over Transformation (EOT) to 3 and the results do not change significantly (For Swin-B, APGD-DLR, EOT=1: 25.124, EOT=3: 25.121). Hence, we fix the EOT to 1.

C.7.1 Choice of adversarial attacks

We choose PGD [41] since it is the most popular method for examining model adversarial robustness after architectural changes [58, 42]. However, it is relatively easy to overcome the PGD attack

Table 10: Training time of each AdaNCA-equipped ViT.

	RVT-B-AdaNCA	FAN-B-AdaNCA	Swin-B-AdaNCA	ConViT-B-AdaNCA
Training time (hours)	400	180	120	100

Table 11: Hyperparameters in different adversarial attacks.

	PGD [41]	CW [5]	APGD-DLR [9]	APGD-CE [9]
Max magnitude	1.0	/	1.0	1.0
Steps	5	20	5	5
Step size	0.5	/	/	/
κ	/	0.0	/	/
c	/	0.5	/	/
ρ	/	/	0.75	0.75
EOT	/	/	1	1

Table 12: Results of ViTs and AdaNCA models under Square [2] attack on ImageNet100.

	Placement (after)	Clean Acc.	Square	Attack Failure Rate
Swin-T [38]	/	86.56	16.18	18.69
Swin-T-AdaNCA	4	**87.18**	**31.74**	**36.41**
FAN-S [78]	/	87.30	29.44	33.72
FAN-S-AdaNCA	5	**88.22**	**42.24**	**47.88**
RVT-S [42]	/	87.18	33.14	38.01
RVT-S-AdaNCA	9	**87.50**	**39.12**	**44.71**

using obfuscated gradients through methods such as random inference, noisy architecture, or non-differentiable components [3]. Our AdaNCA does not fulfill the requirement for producing obfuscated gradients since it is fully differentiable and we turn off all randomness during test. However, we still want to examine whether recurrence would result in corrupted gradient information due to the drawback of cross entropy loss [9]. Moreover, the step size in PGD can largely affect the result. Hence, we choose Auto-PGD family [9] to automatically decide the step size and incorporate the new Difference of Logits Ratio (DLR) loss, resulting in APGD-CE and APGD-DLR, respectively. We also want to include an optimization-based adversarial attack and thus select the CW [5] attack.

C.7.2 Black-box attack

We also consider the black-box attack, specifically Square [2]. However, due to the extreme computational cost of performing Square on ImageNet1K, we instead conduct Square attack to three models, Swin-tiny (Swin-T) [38], FAN-small-hybrid (FAN-S) [78], and RVT-small-plus (RVT-S), on ImageNet100. The three models are used in our AdaNCA analysis in Section 3.3. The maximum magnitude $\epsilon = 6/255$ and the number of queries is 1000. Results are shown in Table 12. The AdaNCA placements are in line with the highest robustness improvement placements demonstrated in Figure 7.

C.8 The effect of the range of the steps in random step training

Our random stap training strategy contributes to the model robustness as shown in Table 3. In all of our experiments, we adopt a range of 2 in the random step setting. Here, we examine the effect of increasing the range. We use the same setting as described in Section 4.3. Results are given in Table 13. We can observe that more choices of the recurrent steps worsen the model performance. This might stem from too much noise introduced into the training process which leads to underfitting. Therefore, we only adopt a range of 2 in our experiments.

C.9 Reason for using ImageNet22K model for RVT

We use the ImageNet22K-pretrained model for RVT-Base-plus in our main results. This is because we do not find an official release of the ImageNet1K-trained weights. Moreover, our trainable parameter count on the model in the released code differs from the one reported in the original paper (ours: 88.5M, in-paper: 91.8M) [42, 66]. Hence, self-training might deviate from the results in the official paper. We tried to contact the author for an ImageNet1K version of the model but did not succeed.

Table 13: Results of Swin-Tiny and the AdaNCA-enhanced models with different ranges of random steps on ImageNet100.

	Range	Clean Acc.	AutoAttack	Attack Failure Rate
Swin-T [38]	/	86.56	10.64	12.29
Swin-T-AdaNCA	[3,5]	**87.18**	**19.52**	**22.35**
	[2,6]	87.02	19.02	21.85
	[1,7]	86.84	16.20	18.65

Table 14: Integrating AdaNCA with a pre-trained ViT on ImageNet1K. All integration schemes perform worse than training Swin-B-AdaNCA from scratch.

	Clean Acc. (↑)	IM-A (↑)	APGD-DLR (↑)	IM-R (↑)	IM-SK (↑)
Swin-B	83.4	35.8	15.6	46.6	32.4
Swin-B-AdaNCA	**83.7**	**36.0**	**25.1**	**48.2**	**35.5**
Freeze All	83.1	30.9	11.5	46.2	32.5
Boundary Training	83.1	32.8	19.5	47.1	33.9
Finetune All	83.3	34.9	19.9	46.9	33.7

We test the result of the released model (ImageNet22K version) on PGD attack and get an accuracy of 30.47. In the paper, the authors report this number to be 29.9 on the ImageNet1K model. Hence, we assume that the results of adversarial attacks can be used as a representative, though might be slightly optimistic. Importantly, **this does not falsify our claim that TAPADL modification undermines the adversarial robustness of RVT**, since TAPADL-RVT already falls short in PGD attack when compared to the ImageNet1K version of RVT-Base-plus. However, for the O.O.D test, ImageNet22K model can differ a lot from the ImageNet1K models, as most data would be counted as in-distribution w.r.t ImageNet22K. Therefore, we use the official values reported in the original paper [42] on those benchmarks. Moreover, for the comparison other than adversarial attacks, we use TAPADL-RVT [23] as a proxy of the RVT model.

C.10 Visualization of attention maps

We aim to qualitatively show that AdaNCA helps ViTs perform correct classification when facing noise. Here, we use GradCAM++ [6] to visualize the attention map of Swin-Base [38] and Swin-B-AdaNCA on the clean images as well as images containing adversarial noise. We use APGD-DLR to generate the adversarial images. Results are shown in Figure 9. We can observe that AdaNCA-enhanced Swin model focuses more on the objects, while the baseline model attends to areas unrelated to the object on the images when facing adversarial noise.

C.11 Integrating AdaNCA into pre-trained ViT models

In our experiments, we train all models from scratch. Implementing AdaNCA as a plug-and-play module for pre-trained ViT models would certainly improve the training speed. To explore this, we experiment by inserting AdaNCA into a pre-trained Swin-base model on ImageNet1K by 1) freezing all ViT layers; 2) training only the boundary layers, where boundary layers are the layers before and after the insert position of AdaNCA; 3) Finetuning all layers. Results are given in Table 14. None of the schemes perform as well as training from scratch. The results indicate that the current NCA is able to adapt to such a scheme. However, it may struggle to effectively transmit information between two pre-trained ViT layers, as these layers have already established strong connections. In contrast, training the model from scratch allows NCA and ViT to synergistically adapt to feature variability, resulting in better overall performance. However, it is worth exploring in the future since the fine-tuning scheme can contribute to the performance.

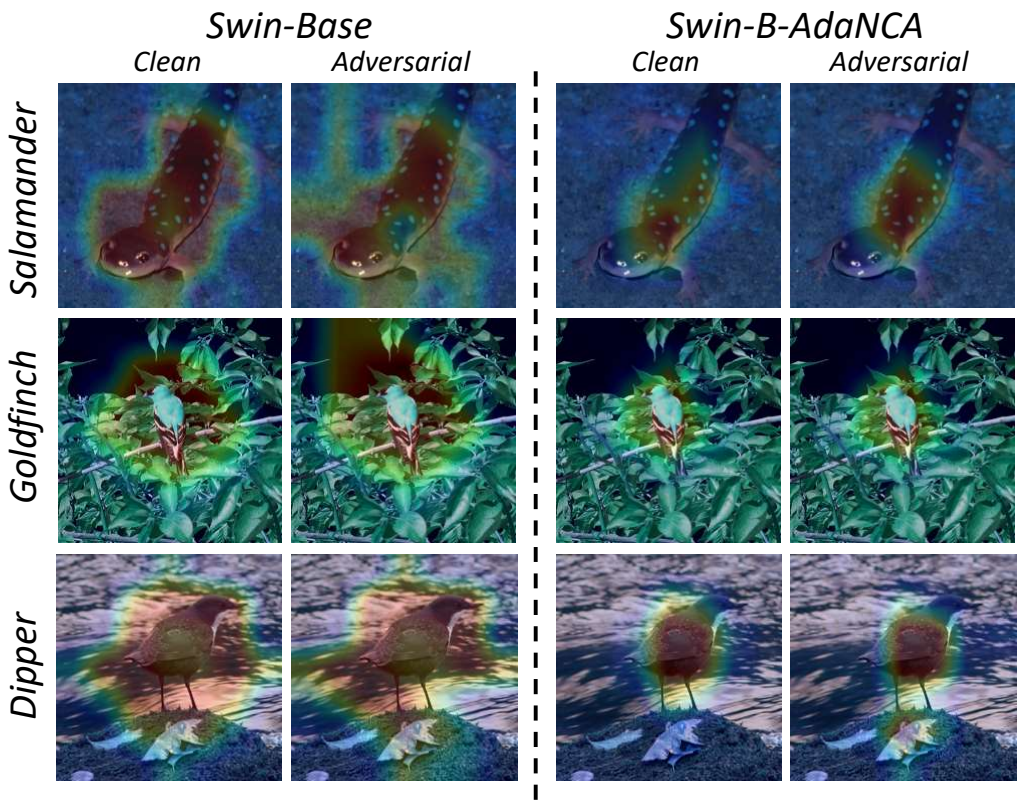

Figure 9: Visualization of attention maps of Swin-Base [38] and Swin-B-AdaNCA using GradCAM++ [6] on clean images and images with adversarial noise. AdaNCA helps ViTs focus more on the object when facing noise.

Table 15: Comparison with ViTCA. ViTCA performs worse in clean accuracy and is on-par with AdaNCA in robustness. Note that ViTCA has more parameters than our scheme Dynamic Interaction. The training setting is the same as in Section 4.3 in the main paper.

	# Params (M)	FLOPS (G)	Clean Acc.(↑)	Attack Failure Rate (↑)
Dynamic Interaction	**27.94**	**4.7**	**87.18**	**22.35**
ViTCA, scale=1	28.48	4.8	86.78	19.47
ViTCA, scale=2	29.08	4.9	86.72	22.28

C.12 Additional results on comparison with current methods

We have presented the comparison between our AdaNCA and the current SOTA method, TAPADL models [23], in Section 4.1. Here, we provide an additional comparison with the ViTCA model [62]. In ViTCA, tokens interact with each other through local self-attention. We aim to compare our proposed Dynamic Interaction module with the interaction learning scheme in ViTCA. Results are given in Table 15. Despite having more parameters, the ViTCA-like scheme performs worse in clean accuracy and is on-par with AdaNCA in robustness. This indicates that it is promising to explore the possibility of incorporating ViTCA into our framework.

C.13 Additional results on same model architecture but with more layers ($\star$ models)

Due to the computational cost of RVT [42] and FAN [78] (Table 10), we do not train them with more layers and thus do not have a $\star$ version of these two models. Instead, we train the smaller version of models on ImageNet100. We choose RVT-small-plus (RVT-S) and FAN-small-hybrid (FAN-S) as two baselines, which is in line with our other experiments on ImageNet100. We add one extra

Table 16: Results of different ViTs on ImageNet100. $\star$ means the same model design but with more layers. The improvement AdaNCA brings does not merely stem from the increase in the number of parameters or FLOPS.

	Params (M)	FLOPS (G)	Placement (after)	Clean Acc.	Acc. AutoAttack	Attack Failure Rate
FAN-S [78]	25.8	6.7	/	87.30	18.70	21.42
FAN-S* [78]	27.7	7.1	/	87.46	19.40	22.88
FAN-S-AdaNCA	26.1	6.9	5	**88.22**	**27.76**	**31.47**
RVT-S [42]	23.0	4.7	/	87.18	29.80	34.18
RVT-S* [42]	24.8	5.1	/	**87.52**	32.14	36.72
RVT-S-AdaNCA	23.3	4.9	9	87.50	**36.90**	**42.17**

Table 17: Comparisons of corruption error on each corruption type of ImageNet-C. AdaNCA consistently improves the performance of different ViT models.

Model	mCE	Noise			Blur				Weather				Digital			
		Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG
Swin-B [38]	54.3	43.8	45.7	43.8	61.6	73.2	55.5	66.5	50.0	47.4	47.9	42.1	38.7	68.8	66.4	62.6
Swin-B-AdaNCA	**51.5**	**39.5**	**40.9**	**39.8**	**59.5**	**69.5**	**55.0**	**65.6**	**49.4**	**47.1**	**40.8**	**40.6**	**37.0**	**66.3**	**59.9**	**61.1**
TAPADL-RVT [23]	44.7	34.6	36.1	34.0	53.9	62.3	52.2	**60.1**	**41.9**	44.9	35.8	37.4	31.9	57.1	41.1	47.2
RVT-B-AdaNCA	**43.2**	**33.9**	**36.1**	**33.9**	**52.7**	**59.1**	**47.8**	60.5	42.6	**39.0**	**33.9**	**36.8**	**31.7**	**53.4**	**40.0**	**46.2**
ConViT-B [16]	46.9	36.0	37.6	35.5	56.5	64.4	51.5	65.0	42.6	40.5	39.3	39.1	38.0	61.7	43.6	51.8
ConViT-B-AdaNCA	**44.3**	**34.7**	**35.9**	**34.2**	**53.1**	**61.0**	**48.6**	**60.4**	**40.5**	**40.1**	**37.2**	**37.2**	**33.5**	**57.1**	**42.4**	**49.4**
FAN-B [78]	46.1	36.5	36.8	34.7	52.8	65.9	47.7	56.9	40.6	44.4	37.9	37.4	34.3	62.8	52.8	50.6
FAN-B-AdaNCA	**44.7**	**35.5**	**36.0**	**33.7**	**51.9**	**65.7**	**45.3**	**55.6**	**39.7**	**43.3**	**37.3**	**37.0**	**33.1**	**62.0**	**46.8**	**47.9**

layer to RVT-small-plus and one extra layer to the FAN network in FAN-small-hybrid, resulting in RVT-S* and FAN-S*. We compare it with our AdaNCA-equipped model where AdaNCA is after layer 9 in RVT-S and after layer 5 in the FAN-S, aligning with Table 12. We use the same AutoAttack as described in Section A to measure the robustness improvement and attack failure rate. Results are shown in Table 16. We can observe that the increase in the number of parameters and FLOPS does not account for the improvement in robustness.

C.14 Additional results on layer similarity structure

Here, we show the layer similarity structure for FAN-B-Hybrid [78] and ConViT-B [16] with the ImageNet1K weights and the weights after training with AdaNCA. For RVT-Base-plus, we train it with the released code [66] but with a different number of parameters, as discussed in Section C.9. This training is only for obtaining the layer similarity structure, and we find it stable after 30 epochs. The results are shown in Figure 10. In practice, we ignore the stage partition found by the dynamic programming algorithm in Section A.1 that is after the first layer or before the last layer since we want a stage to contain more than 1 layer. The results further validate our assumption that AdaNCA is used as an adaptor between stages and transmits information between them, making them more and more different from each other. Interestingly, not all AdaNCA contributes to drastic clearer stage partitions. For example, the first AdaNCA in FAN-B is not effective in making the stage as clear as the second one. However, the overall Set Cohesion Index increases, and the layer similarity structures in all 4 networks do not change. We believe the deep learning architecture research can benefit from our results, in that developing more architectural changes which contribute to stage partition in the network and examine the effect on network robustness and generalization. More importantly, we use a single non-parametric metric for quantifying the layer similarities. Introducing advanced output similarity quantification methods might lead to new findings and other fascinating results.

C.15 Additional results on category-wise mCE on ImageNet-C

We provide additional results on category-wise mCE on ImageNet-C in Table 17. Our test follows the data transformation used in [42].

C.16 Additional results on noise sensitivity examination

We use the data from [60]. In the experiment, human subjects are asked to classify the noise-contaminated images, and the classification accuracy is recorded. Then, the images are fed into the

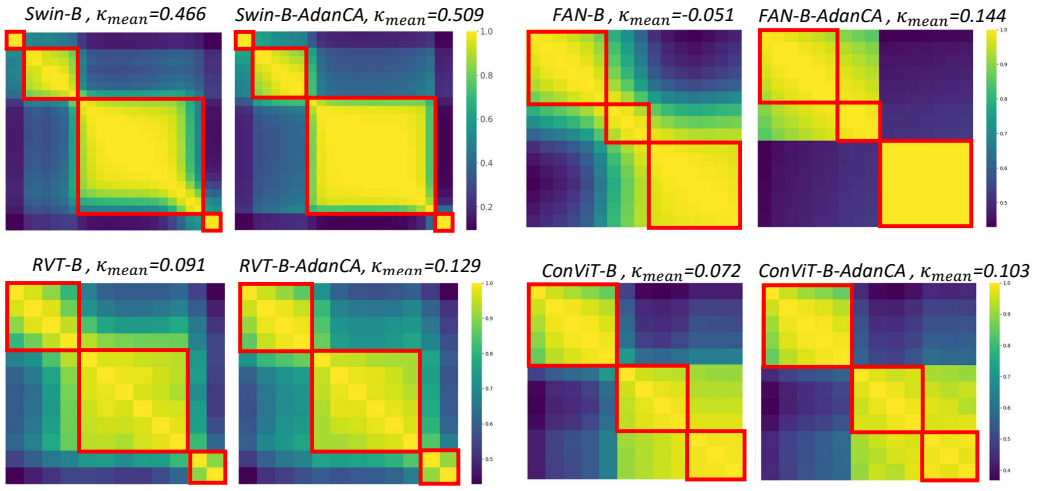

Figure 10: Layer similarity structures of different ViTs. Layer set marked in red boxes. AdaNCA is inserted between each pair of stages. κ_{mean} is the mean of κ of all stages marked out.

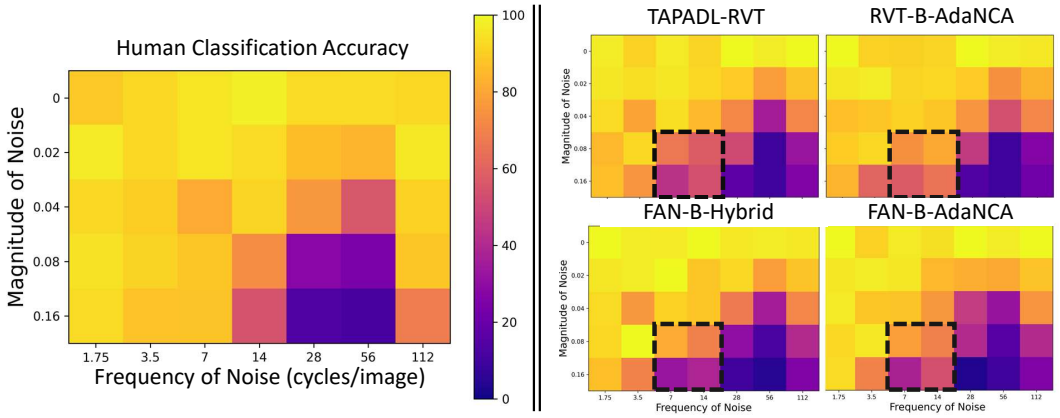

Figure 11: Noise sensitivity examination on humans and ViTs. AdaNCA contributes to more human-like noise-resilient ability and reduces the sensitivity of ViTs on certain types of noise (yellow boxes).

models trained on ImageNet1K, whose outputs are then transformed into a 16-way ImageNet label [56] and compared against the ground truth labels. Their classification accuracy is also recorded and used for comparison with human results. We present the additional results on FAN-B-Hybrid [78] and the SOTA model TAPADL-RVT [23] compared to the AdaNCA-equipped model in Figure 11.

Moreover, we adopt a quantitative metric on the test. In the original experiment [60], a Gaussian curve is fit based on the discretized accuracy map. However, such a method ignores the continuous change in the classification pattern. Hence, we propose a simple metric that examines the similarity between the accuracy map of humans and of models. Specifically, given the ground truth accuracy map (human performance) $\mathcal{A}_{gt}$, and model accuracy map $\mathcal{A}_m$, the similarity Γ is defined as:

$$\Gamma = 100 - \frac{1}{N}\sum_{\epsilon,f} |\mathcal{A}_m(\epsilon, f) - \mathcal{A}_{gt}(\epsilon, f)| \tag{12}$$

ϵ, f represent the magnitude and frequency of the noise, respectively. A higher Γ indicates that the model can achieve more similar accuracy maps with humans averaged across all noisy images. N is the total number of types of noise added. Since all models as well as humans perform similarly on low-magnitude noise, we ignore the first two noise levels ($\epsilon = 0, 0.02$). Hence, in our experiments,

Table 18: Quantitative results on the frequency preference examination experiments. We report the similarity of accuracy maps between model results and humans.

	Accuracy map similarity to ground truth Γ
ResNet50 [27]	72.23
ResNet50-Adv [60]	52.12
TAPADL-RVT [23]	81.91
RVT-B-AdaNCA	**82.17**
FAN-B-Hybrid [78]	83.52
FAN-B-AdaNCA	**84.75**
Swin-B [38]	77.07
Swin-B-AdaNCA	**81.34**
ConViT-B [16]	79.69
ConViT-B-AdaNCA	**81.56**

$N = 21$. We report Γ in Table 18. We also include the results of ResNet50 [27] as well as the L2-adversarially-trained version [60] (ResNet50-Adv, trained with L2-bounded adversarial noise where the maximum magnitude of noise is 0.25), and present the visualizations of the accuracy maps of those two models in Figure. Critically, our results indicate that the adversarially trained model cannot lead to improved accuracy similarities, resulting in a less human-like noise-resilient ability. This is in line with the conclusion of the original work [60], validating our proposed metric. Contrary to adversarial training, we obtain more human-like decision patterns than the compared models, validating our claim in Section 4.4.

C.17 Datasets information and license

- ImageNet1K [12]. This dataset contains 1.28M training images and 50000 images for validation. We report the top1 accuracy on the 50000 validation images. License: Custom (research, non-commercial).
- ImageNet-C [28]. This dataset contains 15 types of 2D image corruption types that are generated by different algorithms. The metric on this dataset is mean corruption error, whose lower value represents a more robust model against those corrupted images. License: CC BY 4.0.
- ImageNet-A [13]. This dataset contains naturally existing adversarial examples that can drastically decrease the accuracy of ImageNet1K-trained CNNs. It is a 200-class subset of the ImageNet1K dataset. License: MIT license.
- ImageNet-R [30]. This dataset contains different artistic renditions of 200 classes from the original ImageNet object classes. The original ImageNet dataset discourages non-real-world images, and thus the artistic renditions render the images to be O.O.D. License: MIT license.
- ImageNet-SK [67]. This dataset contains 50000 images with 1000 classes that match the validation set of the original ImageNet dataset. All the images are black-and-white sketches instead of real-world photographs of the object class. License: MIT license.

C.18 Model and code license

- Swin-B [38]: MIT License.
- FAN-B [78]: Nvidia Source Code License-NC.
- ConViT-B [16]: Apache 2.0 License.
- RVT [42]: Apache 2.0 License.
- TAPADL [23]: MIT License.
- Code for adversarial attacks [34]: MIT License.
- PyTorch Image Model [69]: Apache 2.0 License.
- Code for GradCAM++ [19]: MIT License.

Table 19: Stochasticity during testing will give a false sense of adversarial robustness. We conduct the same experiments twice, which result in Trials 1 and 2. The training setting is the same as in Section 4.3 in the main paper.

	Clean Acc. (↑)	CW Acc. (↑)
Test-time Synchronous	**87.18**	9.48
Test-time Stochasticity, Trial 1	87.10	**10.72**
Test-time Stochasticity, Trial 2	87.06	10.60

Table 20: Removing $\frac{1}{p}$ in Equation 7 during training and directly testing with synchronized update. Our dropout-like compensation scheme improves AdaNCA's performance.

	Clean Acc. (↑)	Attack Failure Rate (↑)
Original	**87.18**	**22.35**
No $\frac{1}{p}$	86.84	21.94

D Extended ablation studies

In this section, we discuss the design of our AdaNCA and provide additional ablation studies on the designs of AdaNCA.

D.1 The dropout-like strategy

In Section 3.1.1, we introduce our approach of using the dropout-like strategy to perform the stochastic update during training and testing. While previous NCA works claim that the stochasticity can be preserved [44, 46] or can be simply switched off [62] during testing, the proposed dropout-like scheme is necessary in our case. First, We highlight that stochasticity during testing can hinder the evaluation of adversarial robustness by producing obfuscated gradients [3], leading to the circumvention of adversarial attacks. Table 19 illustrates this issue, where we test the classification accuracy under CW attack. Stochasticity also results in inconsistent outputs. This is problematic for practical image classification tasks where reliable output is critical, unlike applications focusing on visual effects [46, 62] or collective behaviors [55]. The change in clean accuracy in Table 19 indicates that given the same image, the stochasticity can lead to different decisions, hindering the deployment of the trained models in real-world scenarios. To further demonstrate the necessity of our scheme, we remove the compensation of the output magnitude during training, *i.e.*, we remove the $\frac{1}{p}$ in Equation 7 during training and directly test the model with a synchronized update. The results are given in Table 20. The drop in clean accuracy and robustness suggests that downstream ViT layers struggle with varying NCA output magnitudes, indicating the usefulness of our proposed dropout-like method.

D.2 Dynamic Interaction

Our motivation for developing the Dynamic Interaction module is the high dimensionality of feature vectors in modern ViT models. We underscore that any operations involving linear transformations of the concatenated interaction results will lead to drastic increases in computational costs, as shown in Table 21. Note that the concatenation scheme nearly doubles the FLOPS for RVT compared to the baseline, leading to difficulties in training. Such an increase renders scalability a challenge. Admittedly, more parameters can contribute to better performance, as shown in Table 22. Note, however, that our Dynamic Interaction achieves 70% of the improvements of the original NCA concatenation scheme (10.06/14.62) in robustness improvement and 60% improvement in the clean accuracy (0.62/1.06), with merely 10% of the parameters and FLOPS (0.35/2.99 for # Params and 0.2/2.3 for FLOPS). Therefore, our Dynamic Interaction scheme provides a good trade-off between the performance and computational costs. It is capable of scaling up, allowing us to insert AdaNCA into even larger models, such as current vision-language models where the token dimensionality is even higher. Moreover, we qualitatively showcase in Figure 12, that Dynamic Interaction helps robustify AdaNCA when facing noisy inputs.

Table 21: The concatenation scheme of the original NCA results in a large increase in the computational costs when the token dimensionality is high, leaving scaling up to large datasets and model sizes a challenge.

	# Params (M)	FLOPS (G)
Swin-B	87.8	15.4
Swin-B-AdaNCA	**90.7**	**16.3**
Swin-B-AdaNCA, Concat	115.5	24.7
RVT-B	88.5	17.7
RVT-B-AdaNCA	**91.0**	**19.0**
RVT-B-AdaNCA, Concat	112.3	34.0

Table 22: Performance of using our Dynamic Interaction and the concatenation scheme of original NCA on ImageNet100. Numbers in parentheses indicate performance improvement compared to the Baseline. Our method achieves a large portion of the performance improvement (60% 70%) brought by the concatenation scheme with much fewer computational costs (10%).

	# Params (M)	FLOPS (G)	Clean Acc.(↑)	Attack Failure Rate (↑)
Baseline	27.59	4.5	86.56	12.29
Dynamic Interaction	**27.94 (+0.35)**	**4.7 (+0.2)**	87.18 (+0.62)	22.35 (+10.06)
Concat	30.58 (+2.99)	6.8 (+2.3)	**87.62 (+1.06)**	**26.91 (+14.62)**

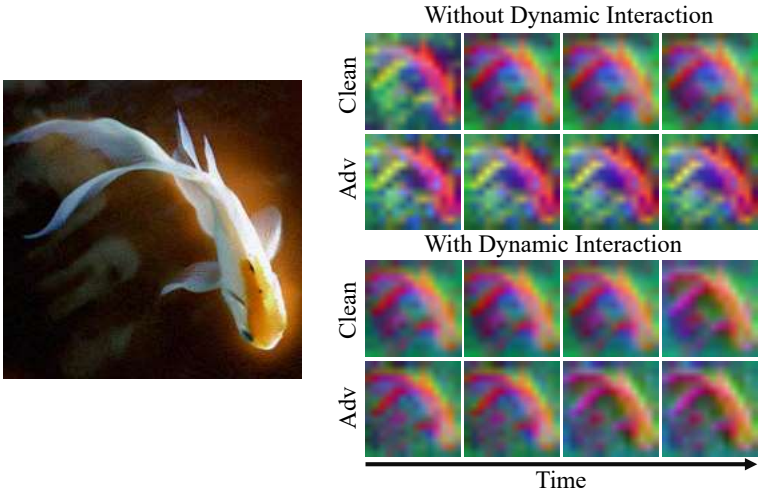

Figure 12: Visualization of the developing token maps of AdaNCA using PCA. Without Dynamic Interaction, AdaNCA cannot recover from the noisy adversarial inputs. Equipped with it, the model can stick to the evolution path similar to the clean inputs.

Table 23: Overly large neighborhood size during token interaction will undermine the model performance as it complicates the process of figuring out which neighbors to interact with and adds too much noise during the process. While increasing the model capacity can alleviate such an issue, AdaNCA still struggles to handle too-noisy information when the scale is too high. Note that the larger filter scheme achieves on-par results with our method when the scale is 2, indicating that it does not provide additional useful information for token interaction at this scale.

	# Params (M)	FLOPS (G)	Clean Acc.(↑)	Attack Failure Rate (↑)
Original, scale=3	**27.94**	**4.7**	86.60	20.44
Larger filter,scale=3	28.04	**4.7**	**86.74**	**21.13**
Larger $\mathcal{W}_M$ receptive field,scale=3	27.98	**4.7**	86.66	20.32
Original, scale=2	**27.94**	**4.7**	**87.18**	**22.35**
Larger filter,scale=2	27.96	**4.7**	87.14	22.31

D.3 Multi-scale Dynamic Interaction

Our motivation for developing multi-scale interaction is to perform more efficient token interaction learning over local scales since ViT already contains global information. Enlarging the neighborhood size will: 1) Complicate the process of selecting the neighbors to interact with; 2) Introduce excessive noise, making it difficult for tokens to accurately acquire neighbor information. 3) Repeatedly acquire the global information provided by ViT. The recurrence further amplifies the noise. In our ablation study in Section 4.3, we notice that increasing the number of scales to 3 will undermine the performance. Such an issue is also observed in previous work [62] where local self-attention is used for interaction learning. While in theory both Dynamic Interaction and self-attention can discount far-away information as the tokens can decide their own interaction neighborhood, overly large neighborhood size can lead to the problems mentioned above. Note that our model gains performance when $\mathcal{S} = 2$ (5×5 neighborhood), indicating the usefulness of our multi-scale module. We further explore whether the multi-scale issue can be solved by increasing the model capacity. Specifically, we develop two additional schemes. The first is to replace the dilation in Dynamic Interaction by simply using larger filters. The other one enlarges the receptive field of the weight computation network $\mathcal{W}_M$ in Equation 9. Instead of using two 3×3 convolution layers, we change the first layer to be a 5×5 convolution. As a result, the receptive field of $\mathcal{W}_M$ becomes 7×7, matching the one when $\mathcal{S} = 3$. We use the same training setting as in Section 4.3. Results are given in Table 23. Although increasing model capacity might alleviate the noise issues, AdaNCA struggles with overly large neighborhoods.

E Differences between traditional NCA and AdaNCA

While AdaNCA is inspired by NCA, it does not fully exploit the designs of traditional NCA. We discuss three key differences between AdaNCA and NCA below:

1. Different initial states. NCA in image generation starts from either constant or randomly initialized cell states, typically referred to as the seed states, while AdaNCA receives the outputs from previous ViT layers. Hence, AdaNCA handles structured inputs at the very beginning.
2. Different usage of cell states. Traditional NCA does not use all cell states for accomplishing the downstream tasks. After certain steps of evolution, a subset of the cell states is extracted to perform the given task. The unused cell states, termed hidden states, facilitate cell communications as they can be used to store additional information [44]. The dimensionality of the hidden states is typically several times that of the input. When the cell dimensionality is high, as is the case in ViT, adding too many hidden states will bring too much computational costs. Moreover, cells can store effective enough information when they have high dimensionality. Therefore, we do not adopt the hidden states design in AdaNCA.
3. No pooling strategy. A critical component of traditional NCA is the pooling strategy. Instead of starting from the seed states in every epoch, NCA fetches the starting states from a pool, where the output states from previous epochs are stored. Through this strategy, NCA can explore much longer time steps without suffering from gradients or memory issues. While it is tempting to incorporate the pooling trick into our method, two major concerns hinder its

practical implementation: 1) The previous NCA models start from their own outputs in the pooling strategy. In other words, the cell states in the pool are generated by the model itself. However, AdaNCA starts from the outputs of a ViT layer. Such a difference renders the pool trick hard to implement. 2) Taking a step back, even if there is a well-designed pooling strategy for AdaNCA, it will bring too much computational costs during testing. A single step of AdaNCA introduces non-trivial FLOPS during testing, as shown in Table 1. The ultimate goal of the pooling strategy is to ensure the stability of NCA in large time steps, while large time steps will slow down the inference.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claim on AdaNCA improving ViT's robustness is validated in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation of our work is discussed in Section 5.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

Justification: We do not have theory assumptions and proofs in the paper.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: We disclose all architecture and training details in Section 3.2, 4, C.1 and C.6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We are preparing the code release and will publish the code soon.

Guidelines:

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

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We disclose all training details in Section 4, C.1 and C.6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: Please see Section B in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We provide the hardware and time consumption for our experiments in Section C.6 in the Appendix.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and strictly follow it during the research.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: Please see Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper does not pose such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Please see Section C.17 and C.18 in the Appendix.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [No]

    Justification: We do not include code or model release in the paper, but we are preparing it for a full publication.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not have crowdsourcing and we do not conduct research with human subjects.

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

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

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

    Answer: [NA]

    Justification: The paper does not have crowdsourcing and we do not conduct research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.