

# Approximate Nullspace Augmented Finetuning for Robust Vision Transformers

Haoyang Liu<sup>1†</sup>, Aditya Singh<sup>2†</sup>, Yijiang Li<sup>3</sup>, Haohan Wang<sup>1</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign

<sup>2</sup>Sorted technologies

<sup>3</sup>University of California, San Diego

{hl57, haohanw}@illinois.edu

aditya.singh@sortedtech.io

yijiangli@ucsd.edu

Enhancing the robustness of deep learning models, particularly in the realm of vision transformers (ViTs), is crucial for their real-world deployment. In this work, we provide a finetuning approach to enhance the robustness of vision transformers inspired by the concept of nullspace from linear algebra. Our investigation centers on whether a vision transformer can exhibit resilience to input variations akin to the nullspace property in linear mappings, which would imply that perturbations sampled from this nullspace do not influence the model’s output when added to the input. We start from the observation that many existing ViTs satisfy this property because their patch embedding layer has a non-trivial nullspace. Then, we extend the notion of nullspace to nonlinear settings and demonstrate that it is possible to synthesize approximate nullspace elements for ViT’s encoder blocks through optimization. Finally, we propose a finetuning strategy for ViTs wherein we augment the training data with synthesized approximate nullspace noise. We find that our finetuning approach significantly improves the models’ robustness to both adversarial and natural image perturbations.<sup>1</sup>

## 1. Introduction

The field of computer vision has experienced significant advances, marked by the emergence of Vision Transformers (ViTs) [1] as a notable milestone. Following this advancement, a series of architectural refinements have been explored [2–4], paving the way for the development of vision foundation models [5, 6] through the scaling up of both the model and dataset. Despite these strides, robustness continues to be a pivotal concern for their practical deployment, as they exhibit fragility in the face of imperceptible (adversarial) and perceptible perturbations.

Adversarial samples are generated by adding imperceptible noises to the input, aiming to cause the model to produce incorrect and overly confident predictions [7–9]. Perceptible perturbations are artifacts that arise from various operations, such as JPEG compression, simulated weather effects (fog, snow), or adjustments to the image’s brightness, hue, or contrast, to name a few [10]. The semantic content of the image however, remains unchanged after perceptible or imperceptible perturbations. Hence, we expect the model to output similar predictions for perturbed and unperturbed images.

Applying transformations to the input during training, known as data augmentation, is one of the widely employed techniques for improving robustness. The underlying goal of applying augmentations is to enforce invariance (i.e., consistency) under a predefined set of perturbations. To induce adversarial robustness, worst-case adversarial perturbations are first identified through an optimization procedure and then used to train the model [11, 12]. For robustness against perceptible noise, augmentation strategies have evolved from simple transformations such as horizontal flips and rotations to more complex augmentations like MixUp [13], CutMix [14], and AugMix [15].

---

<sup>1</sup>Code is available at: <https://github.com/Liu-Hy/ns-vit>.

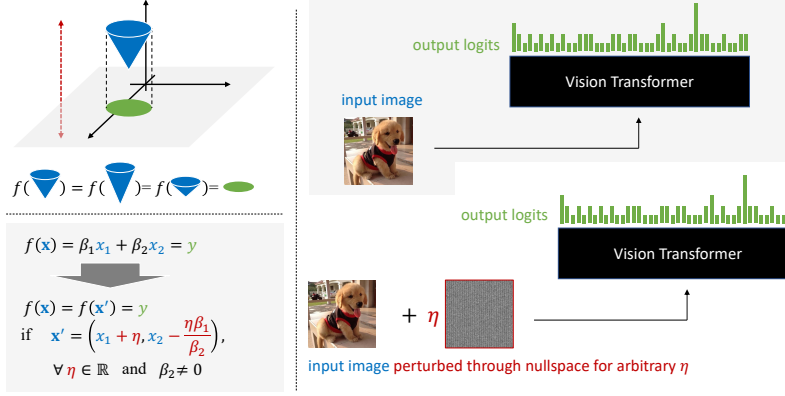


Figure 1: **An illustration of the nullspace in three cases (projection function, left top; linear function, left bottom; vision transformer, right).** For these three cases, there exists some **nullspace**, such that the **output** of the function with respect to the **input** will remain unperturbed regardless of the perturbation strength. Also, the **nullspace** is function-specific (model-specific) and will not vary for different samples.

There is an observable divide in the treatment of these two types of robustness [16]. Adversarial noises are generated via an optimization process, whereas augmentations are defined heuristically often by domain experts. It has also been observed that standard data augmentation strategies, in isolation, do not improve adversarial robustness [17–19]. Additionally, adversarial training (training with adversarial perturbations) often leads to a drop in performance on non-adversarial images [11, 12, 20].

Here, we consider robustness a property of the model and thus agnostic to the noise type. To this end, we consider the nullspace as the central theme of our study. The nullspace, a fundamental concept in linear algebra, refers to the subspace of a domain that is mapped to zero by a linear mapping. By definition, any vector from the nullspace, when added to the input of the linear mapping, does not alter its output. In Figure 1, we present the concept of nullspace from different perspectives.

This paper first identifies that most off-the-shelf pre-trained ViT models exhibit a nontrivial nullspace due to the linear patch embedding layer. Since this layer is the first block of a ViT, any invariance to it implies invariance to the entire model. Consequently, a nontrivial nullspace also exists for ViTs. To further explore robustness, we define the approximate nullspace of the transformer encoder and use optimization methods to synthesize noise vectors approximating nullspace properties for nonlinear blocks. Finally, we propose fine-tuning the model using these synthesized nullspace-like elements as additive training data augmentation. This approach enlarges the approximate nullspace, enhancing the model’s robustness. The main contributions of our paper include:

- We demonstrate connections between the robustness of vision transformers to the algebraic notion of nullspace, substantiated by experimental results showing that enlarging the approximate nullspace effectively improves the model robustness.
- We conduct comprehensive analysis on the existence of nullspace within transformer models. We establish the existence of nullspace at the patch embedding layer, and empirically identify an approximate nullspace at the nonlinear encoder level of transformers by validating their algebraic properties.
- We propose an effective data augmentation method by exploiting and enlarging the model’s approximate nullspace, which enhances model robustness without architectural modifications and only involves fine-tuning with minimal additional data. Our method is empirically validated across multiple benchmark datasets, showing significant robustness improvements against adversarial and out-of-distribution scenarios.

## 2. Related Work

**Data augmentation and Invariance:** Data augmentation enforces invariance by training models to predict consistently across different input views, offering a theoretical improvement in estimating statistical risk [21, 22]. However, incorrect augmentation choices can degrade performance [21–23]. Early image augmentations such as flipping, cropping and rotation have evolved into advanced techniques such as MixUp [13], CutMix [14], and strategies for chain augmentations, including AutoAugment for policy optimization, TrivialAugment [24], and RandAug [25]. AugMix [15] combines transformations with a consistency loss, while differentiable augmentations optimize transformations for specific tasks [26, 27]. Hounie et al. [28] frame data augmentation as an invariance-constrained learning problem, using a relaxed invariance notion to model augmentation distributions. Unlike these approaches, our work avoids reliance on pre-defined augmentations.

**Robustness in ViTs:** Research highlights Vision Transformers (ViTs) as more robust than Convolutional Neural Networks (CNNs) [29, 30], with adversarial examples that exhibit low transferability between these architectures [31], although some studies offer counterpoints [32]. ViTs demonstrate insensitivity to patch-based transformations that distort semantics, relying on robust but nonindicative features [33]. Robustness-enhancing methods for transformer-based models are often model-agnostic, using data augmentation [34–37] and regularization [36, 38, 39], consistent with broader robustness frameworks [40, 41]. For example, Xiao et al. [34] masks image patches using class activation maps and refills them with random samples, while Chen et al. [38] adopts sharpness-aware optimization for a smoother loss landscape. However, these approaches focus on external modifications or optimization, often neglecting the intrinsic properties of the model.

**Nullspace and Neural Networks:** The study of nullspaces in neural networks began with Goggin et al. [42], who explored MLPs’ universal approximation by comparing input nullspaces and outputs. Using the *learning XOR* example, they demonstrated that hidden layers enable MLPs to map inputs to targets even if the targets reside in the nullspace of the inputs. More recently, Sonoda et al. [43] mathematically analyzed nullspaces in fully connected networks.

In applications, Wang et al. [44] leveraged nullspaces in continual learning to map new tasks to the nullspace of existing ones. As a novel architecture, NullSpaceNet [45] mapped inputs from the same category to a joint nullspace rather than a feature space.

## 3. Nullspace and Invariance

When a mapping  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is invariant to some additive noise  $\mathbf{v}$ , it implies the following:

$$f(\mathbf{x} + \mathbf{v}) = f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}. \quad (1)$$

This invariance has interesting connections to the concept of *nullspace* in linear algebra. Formally, the nullspace of a linear mapping  $f$  is a set  $\mathcal{N}$  identified by  $\mathcal{N} = \{\mathbf{v} \in \mathcal{X} | f(\mathbf{v}) = 0\}$ . For a non-trivial nullspace  $\mathcal{N} \neq \phi$ , we have  $f(\mathbf{x} + \mathbf{v}) = f(\mathbf{x}), \forall \mathbf{v} \in \mathcal{N}, \forall \mathbf{x} \in \mathcal{X}$ . We can interpret this by saying that the linear mapping is invariant to the noise vector sampled from its nullspace. For brevity, we refer to this noise vector as **nullspace noise**.

### 3.1. Non-trivial Nullspace of the Patch Embedding Layer

Vision transformer [1] is a function  $f_\omega$  with  $\omega$  as the trainable weights. The function takes as input an image  $\mathbf{x} \in \mathcal{X}^{c \times h \times w}$  and outputs a classification response  $\mathbf{y} \in \mathcal{Y}^k$  over  $k$  categories.  $c$  is the number of channels (typically 3 for red, green, and blue),  $h, w$  correspond to height and width of the input image. This neural network function can be broken down into 3 stages, namely:

- *patch embedding stage*,  $f_\theta : \mathcal{X}^{c \times r \times r} \rightarrow \mathcal{U}^d$ . This step projects the input image patch of predetermined dimensions  $c, r$  and  $r$  to a one-dimensional embedding of length  $d$ . It is ensured that patches have no overlaps between them. Hence, the number of such non-overlapping patches generated from the input image are  $m = \frac{h \times w}{r^2}$ .



Figure 2: **An example of nullspace noise.** We show (a) sample input image, (b) noise generated by the basis vectors of the nullspace and (c) noisy image as a result of adding the nullspace noise to the input. Model’s predictions for the clean and noisy inputs are identical.

- *self-attention stage*,  $f_\phi : \mathcal{U}^{(m+1) \times d} \rightarrow \mathcal{V}^{(m+1) \times d}$ . In the next step, the generated patch embeddings are passed through layers of self-attention modules to process long range interactions amongst them. Apart from the  $m$  patch embeddings an additional embedding in form of cls token is utilised in this step.
- *classification stage*,  $f_\psi : \mathcal{V}^d \rightarrow \mathcal{Y}^k$ . The final step is to perform the  $k$ -way classification. For this, we simply keep the processed encoding corresponding to cls token and project it through a linear classification layer.

Since the first layer of the ViT is a linear mapping, according to the rank-nullity theorem, it always has a non-trivial nullspace if  $cr^2 > d$ . In practice, for many ViT-based architectures, we find that this is the case. In Table 1, we report the identified nullspace dimensions for off-the-shelf pre-trained ViT models.

Given the weights of the patch embedding layer  $f_\theta$ , finding its nullspace is a standard practice [46–48]. Let  $B_\theta = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$  be the  $k$  basis vectors for this nullspace, we can sample an element from  $\mathcal{N}_\theta$  as:

$$\mathbf{v} = \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 + \dots + \lambda_d \mathbf{b}_k. \quad (2)$$

The property of such a sample will be that the output of the patch embedding will effectively remain preserved,  $f_\theta(\mathbf{x} + \mathbf{v}) = f_\theta(\mathbf{x})$ . Since the output after the first layer remains unaffected, the final output of the classification remains unchanged. In Figure 2, we provide visualization of noise synthesized using basis vectors. This noise can be added to *any* input image with complete invariance. In Section 3.2, we explore if it possible to learn a nullspace-like counterpart for the non-linear blocks of ViTs.

### 3.2. The Generalized Nullspace from the Encoder

So far we have demonstrated that a non-trivial nullspace exists for the patch embedding layer, and hence the entire vision transformer is invariant to all perturbations in that space. We move further down the structure of ViT and investigate whether the encoder is also invariant to certain perturbations. The self-attention layer is non-linear, which means the notion of nullspace cannot be directly applied to  $f_\phi$ . However, the *invariance* property that can be implied from the nullspace of linear functions, that any vector from this set will not alter the function’s output when added to any input, is still desirable in the nonlinear case when it comes to the robustness of neural models.

Table 1: **Nullspace dimensions for pre-trained ViT models.** Nullspace is trivial (0) when embedding dimension exceeds input dimension.

Model	Patch Size	Emb. Dim.	Null Dim.
tiny	$16 \times 16$	192	576
small	$32 \times 32$	384	2688
	$16 \times 16$	384	384
base	$32 \times 32$	768	2304
	$16 \times 16$	768	2
	$8 \times 8$	768	0
large	$32 \times 32$	1024	2048
	$16 \times 16$	1024	0



In fact, data augmentation can often be formulated as a process of adding noise to the input and enforcing invariance. Therefore, to study the ViTs' inherent invariance to input perturbations, we extend the notion of nullspace to the nonlinear setting and define the *Generalized Nullspace*,  $\tilde{\mathcal{N}}_\phi$ , of the transformer encoder  $f_\phi$ , as below:

$$\tilde{\mathcal{N}}_\phi = \{\mathbf{v} | f_\phi(\mathbf{u} + \mathbf{v}) = f_\phi(\mathbf{u}) \quad \forall \mathbf{u} \in \mathcal{U}\}, \quad (3)$$

Here, we use the tilde accent  $\tilde{\cdot}$  to distinguish  $\tilde{\mathcal{N}}_\phi$  from the conventional nullspace  $\mathcal{N}_\phi$ . We term it the Generalized Nullspace because it depicts invariance in both linear and nonlinear settings, and that for a linear function  $f_\theta$  we have  $\mathcal{N}_\theta \subseteq \tilde{\mathcal{N}}_\theta$ , since any vector sampled from the conventional nullspace of a linear function satisfies this invariance property. If such a set exists, it directly implies that the transformer model is robust to certain perturbations in the input space. Our theoretical analysis established the following sufficient conditions for the existence of a nontrivial generalized nullspace. (The complete proof is given in Appendix A.)

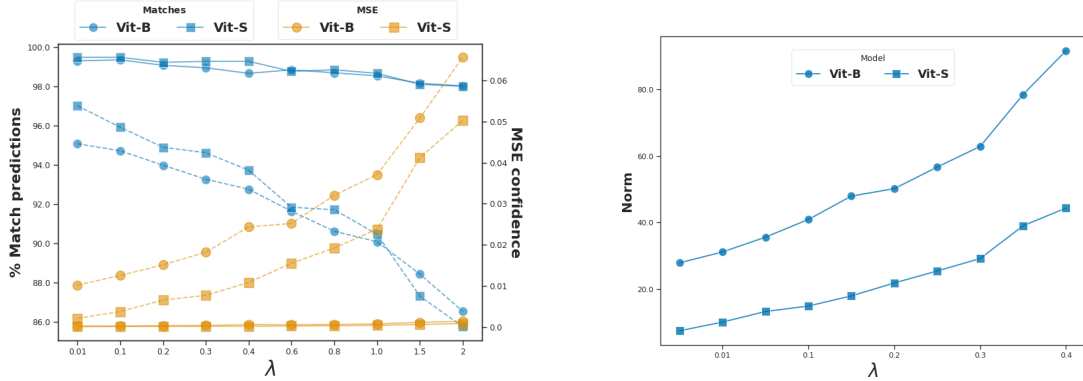
**Proposition 1.** Consider a self-attention layer with  $h$  heads and  $\{(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)\}_{i=1}^h$  as its query, key and value projection matrices. If the following conditions are met

1.  $\mathbf{Q}_i \mathbf{K}_i^\top$  is symmetric for  $i = 1, \dots, h$
2. The row space  $\mathcal{R}(\mathbf{V}_i^\top) \subseteq \mathcal{R}(\mathbf{Q}_i \mathbf{K}_i^\top)$  for  $i = 1, \dots, h$
3. for some  $m \neq n$ ,  $\mathbf{Q}_m \mathbf{K}_m^\top$  has colinearity with  $\mathbf{Q}_n \mathbf{K}_n^\top$ , i.e. for some  $k$  the  $k$ th row of  $\mathbf{Q}_m \mathbf{K}_m^\top$ , denoted as  $\mathbf{r}_{m,k}$ , satisfies  $\mathbf{r}_{m,k} \neq \mathbf{0}$  and  $\mathbf{r}_{m,k} \in \mathcal{R}(\mathbf{Q}_n \mathbf{K}_n^\top)$

then there exists at least one  $\mathbf{W}$  such that  $\mathbf{W} \neq \mathbf{0}$  and  $\text{head}_i(\mathbf{X} + \mathbf{W}) = \text{head}_i(\mathbf{X})$  for all attention head  $i$  in this layer and arbitrary  $\mathbf{X}$ .

*Remark 1.* Condition 1 can be met if  $\mathbf{Q}_i$  and  $\mathbf{K}_i$  satisfy some special relation. For example, let  $\mathbf{PDP}^{-1}$  be a diagonalization of a real symmetric matrix  $\mathbf{A}$ . If  $\mathbf{Q}_i = \mathbf{BP}$  and  $\mathbf{K}_i = \mathbf{B}(\mathbf{P}^{-1})^\top \mathbf{D}$ , then we have  $\mathbf{Q}_i \mathbf{K}_i^\top = \mathbf{BAB}^\top$  to be symmetric.

In addition, evidence has shown that,  $\mathbf{Q}_i \mathbf{K}_i^\top$  can be empirically symmetric, especially for ViTs, when the attention heads are visualized and correlation of parameters is calculated [49].



(a) Noise influence on the model output under different regularization strengths (b)  $\ell_2$  norm of learned noise under different regularization strengths

Figure 3: **Exploratory experiments on the generalized nullspace.** (a) Solid lines (—) represents the model performance under the learned noise, and dashed lines (···) represent the performance after random permutation of the elements of the learned noise vector. (b) by changing the regularization strengths, we explore noise in the generalized nullspace at different magnitudes.

### 3.3. Synthesizing (approximate) nullspace noise

Although our theory suggests a sufficient condition for the existence of generalized nullspace, analytically finding  $\tilde{\mathcal{N}}_\phi$  or probing its existence for generic transformers is challenging. Thus, as an

exploratory experiment, we employ a numeric method: we search for individual element,  $\tilde{\mathbf{v}}_\phi$ , of this set. This element is an additive perturbation that brings minimal influence to the output of  $f_\phi$  on the data distribution. We introduce a regularization term on the norm of  $\tilde{\mathbf{v}}_\phi$  to prevent the trivial solution of  $\mathbf{0}$ .

$$\mathcal{L}_\phi(\tilde{\mathbf{v}}) = \underbrace{\mathbb{E}_{\mathbf{u} \in \mathcal{D}} \|f_\psi(f_\phi^0(\mathbf{u} + \tilde{\mathbf{v}})) - f_\psi(f_\phi^0(\mathbf{u}))\|}_{\mathfrak{I}} - \lambda \log(\|\tilde{\mathbf{v}}\|). \quad (4)$$

Here,  $\|\cdot\|$  is the  $\ell_2$  norm,  $f_\phi^0$  is the representation of the cls token output by  $f_\phi$ , and  $\lambda$  is the regularization coefficient.  $\mathfrak{I}$  resembles a weaker notion of invariance compared to Equation (1). Equation (4) minimizes the  $\ell_2$  norm between the predicted logits with and without the noise. Alongside the self-attention stage, we have also incorporated the classification stage into the loss, since the target of our study is to minimize the impact on the final output of the network. To learn the noise vector, we initialize  $\tilde{\mathbf{v}}$  by sampling from a uniform distribution, and minimize the loss with gradient descent. We use ViT-S and ViT-B models with patch size 32 for evaluation. We employ ImageNette [50] as the dataset for this experiment, which is a subset of ImageNet consisting of 10 categories. We learn  $\tilde{\mathbf{v}}$  on the training dataset ( $\approx 9500$  images) and perform evaluation on the validation set ( $\approx 4000$  images).

To quantitatively evaluate learned  $\tilde{\mathbf{v}}_\phi$ , in Figure 3 (a) we report the percentage of matching classifications with and without learned nullspace noise, and the mean squared error computed at the output probabilities (hereafter “MSE confidence”). We consider a prediction to be matched if the assigned category for input is the same with and without adding the perturbation. By varying the regularization strength, we get noise vectors of different magnitude (Figure 3 (b)), all being fairly benign to the model’s predictions. However, if we randomly reset the vectors’ direction by permuting their elements, the noise significant degrades the model’s predictions.

The experiment shows the feasibility of learning elements that approximately conform to our definition of generalized nullspace with good precision, and also indicates that at different magnitudes there are certain directions in the input space toward which the perturbation is fairly benign to the model. In Appendix E, we further empirically show that the learned noise vectors exhibit good properties under scalar multiplication and convex combinations within certain scope of parameters, similar to the closure property of a vector space.

## 4. Nullspace Noise Augmented Finetuning

In this section, we investigate the application of the synthesized nullspace noise. As we discussed previously, the model is weakly invariant to the learnt noise ( $\mathfrak{I}$  in Equation (4)) and the set as a result of this relaxed notion is an approximate nullspace. To more accurately quantify this, we define the  $\epsilon$ -Approximate Generalized Nullspace as follows (later called “ $\epsilon$ -approximate nullspace” or “approximate nullspace” for brevity):

$$\tilde{\mathcal{N}}_\phi(\epsilon) = \{\tilde{\mathbf{v}} | \mathbb{E}_{\mathbf{u} \in \mathcal{D}} \|f(\mathbf{u} + \tilde{\mathbf{v}}) - f(\mathbf{u})\| \leq \epsilon\}, \quad (5)$$

where  $f(\cdot) = \text{Softmax}(f_\psi(f_\phi^0(\cdot)))$ . It is easy to verify that  $\forall \epsilon > 0, \mathbf{0} \in \tilde{\mathcal{N}}_\phi(\epsilon)$ , and that  $\forall \epsilon_2 > \epsilon_1 > 0, \tilde{\mathcal{N}}_\phi(\epsilon_1) \subseteq \tilde{\mathcal{N}}_\phi(\epsilon_2)$ .

We believe that the existence of approximate noise vectors is a property of the model. As these vectors exhibit relaxed invariance, we also believe that they play a key role in model’s inherent robustness under a variety of distribution shifts. Hence, if we can further improve invariance on approximate nullspace elements, we can potentially make the model more robust. With this belief, **we propose to fine-tune a pre-trained ViT with the learnt nullspace noise vector as an added (encoder level) input perturbation.** The motivation behind this is to enlarge the (approximate nullspace) set of noise vectors to which the model is invariant.

Formally, we employ a bi-level optimization approach, where the inner problem finds the best noise vector and the outer problem finds the model that is the most tolerant to such noise, as shown below.

$$\min_{\phi} \mathbb{E}_{\mathbf{u} \in \mathcal{D}} \ell(f_{\psi}(f_{\phi}^0(\mathbf{u} + \tilde{\mathbf{v}}_{\phi}^*)), \mathbf{y})$$

$$\text{where } \tilde{\mathbf{v}}_{\phi}^* = \arg \max_{\tilde{\mathbf{v}}} \|\tilde{\mathbf{v}}\| \quad \text{s.t. } \tilde{\mathbf{v}} \in \mathcal{N}_{\phi}(\epsilon). \quad (6)$$

Here,  $\ell(\cdot)$  is the cross-entropy loss. While this optimization problem can also be solved in different ways, we use an efficient heuristic: we initialize the noise with a large enough sampling limit, minimize  $\mathcal{L}_{\phi}(\tilde{\mathbf{v}})$  by gradient descent according to the loss function in Equation 7, and early stop it as soon as it enters  $\mathcal{N}_{\phi}(\epsilon)$ , as shown in Equation 8.

$$\mathcal{L}_{\phi}(\tilde{\mathbf{v}}) = \mathbb{E}_{\mathbf{u} \in \mathcal{D}} \|f_{\psi}(f_{\phi}^0(\mathbf{u} + \tilde{\mathbf{v}})) - f_{\psi}^0(f_{\phi}(\mathbf{u}))\| \quad (7)$$

$$\hat{\mathbf{v}}^* = \text{SGD}(\mathcal{L}_{\phi}(\tilde{\mathbf{v}}), \tilde{\mathbf{v}}_0, \epsilon). \quad (8)$$

Here,  $\hat{\mathbf{v}}^*$  is the approximate solution for  $\tilde{\mathbf{v}}_{\phi}^*$ ,  $\text{SGD}(\mathcal{L}_{\phi}(\tilde{\mathbf{v}}), \tilde{\mathbf{v}}_0, \epsilon)$  denotes the gradient descent algorithm that minimizes the loss  $\mathcal{L}_{\phi}(\tilde{\mathbf{v}})$  starting from its initial value  $\tilde{\mathbf{v}}_0$  until it satisfies the condition  $\mathcal{L}_{\phi}(\tilde{\mathbf{v}}) < \epsilon$ . The noise norm starts from a large value and gets gradually reduced during the process. When early stopping is triggered, we obtain noise vectors that are close to the boundary of the  $\epsilon$ -approximate nullspace. For more details of our method, please refer to Algorithm 1 in Appendix B.

## 5. Experiments

### 5.1. Implementation Details

In this section, we conduct evaluation of our nullspace augmented finetuning method (Section 4) on several benchmarks. By making the model more tolerant to noise in the  $\epsilon$ -approximate nullspace, we hope to expand the nullspace itself and observe its effect on the model’s robustness under different settings.

Starting from a pretrained model, we use the  $\epsilon$ -approximate nullspace noise as data augmentation to fine-tune the model. The noise is generated every 40 training steps according to Equation (8) with an  $\epsilon$  of 0.03. The experiment was done within one epoch of training on the ImageNet-1k [51] dataset. We used the vanilla ViT-small and ViT-base models, and ViT-base (DAT) which is the current SOTA on ImageNet-C dataset on the EasyRobust benchmark<sup>2</sup>, trained using Discrete Adversarial Training [52]. We evaluated the model performance in a wide range of settings to test its performance on the i.i.d dataset, under adversarial attacks and distribution shifts. For adversarial attacks we utilize FGSM [7], DamageNet [53], PatchFool [54] and CW [55]. Among them, FGSM and CW are gradient-based white-box attacks, DamageNet consists of pre-generated adversarial examples, and PatchFool targets localized, adversarial patches of an image. For distribution shift we employ ImageNet-C [10], ImageNet-A [56], ImageNet-V2 [57], ImageNet-R [58], ImageNet-Sketch [59] and Stylized-Imagenet [60]. ImageNet-C consists of validation images modified by applying corruptions in the form of weather effects, noises, etc. ImageNet-A applies the imagenet objects in hard contexts. ImageNet-R and ImageNet-Sketch consist of imagenet categories in different art forms. ImageNet-Stylized applies texture transfer onto the ImageNet validation images to create shape-texture contradictions.

We use the EasyRobust library [61] for code implementation and the checkpoints of ViT-base (DAT). For more implementation details please see our supplementary document.

### 5.2. Experiment: Robustness Evaluation

We evaluated the effect of nullspace finetuning to improve the robustness of vision transformers under different settings. We used the official mCE score as the evaluation metric for ImageNet-C, where a lower mCE indicates better robustness, and we used the accuracy score for all other settings. We used 100 – mCE before taking the average in all settings.

<sup>2</sup><https://github.com/alibaba/easyrobust>

Table 2: Effect of our nullspace augmented finetuning (NS) method on different models evaluated on multiple benchmark datasets. Excluding DAT, vanilla ViT-S and ViT-B, the values for the baselines are directly reported from the corresponding papers. For DAT, we report the reproduced results following their evaluation setting.

Methods	Clean	Adversarial Robustness				Out of Distribution Robustness						Average
		PatchFool	CW	FGSM	DamageNet	A	C↓	V2	R	Sketch	Stylized	
ViT-S	74.19	0.68	4.63	13.79	29.82	16.35	71.13	62.51	34.67	14.26	12.15	26.54
ViT-S + NS (ours)	<b>77.47</b>	<b>19.10</b>	<b>9.37</b>	<b>25.95</b>	<b>32.43</b>	<b>20.77</b>	<b>55.98</b>	<b>66.5</b>	<b>41.61</b>	<b>25.67</b>	<b>16.02</b>	<b>34.45</b>
ViT-B	77.68	15.92	12.54	25.65	38.69	23.88	62.16	66.05	41.63	16.31	17.97	34.01
ViT-B + MixUp [13]	77.80	-	-	-	-	12.20	61.80	-	34.90	-	-	-
ViT-B + RandAugment [25]	79.10	-	-	-	-	-	43.60	-	23.00	-	-	-
ViT-B + PR [33]	78.20	-	-	-	-	-	47.60	-	21.40	-	-	-
ViT-B + RandAugment + PR	79.30	-	-	-	-	-	43.60	-	23.80	-	-	-
ViT-B + AugMix [15]	78.80	-	-	-	-	-	42.20	-	24.90	-	-	-
ViT-B + AugMix + PR	79.30	-	-	-	-	-	<b>41.60</b>	-	25.70	-	-	-
ViT-B + SAM [38]	79.90	-	-	-	-	-	43.50	67.50	26.40	-	-	-
RobustViT-B [62]	80.40	-	-	-	-	23.00	-	69.80	35.40	<b>35.80</b>	-	-
ViT-B + NS	<b>81.42</b>	<b>23.52</b>	<b>14.23</b>	<b>36.50</b>	<b>40.44</b>	<b>24.55</b>	<b>47.82</b>	<b>70.25</b>	<b>44.85</b>	<b>26.35</b>	<b>19.02</b>	<b>39.39</b>
ViT-B + DAT[52]	<b>81.47</b>	22.64	23.59	48.80	43.31	23.83	45.95	<b>70.24</b>	<b>48.68</b>	36.94	<b>23.99</b>	43.41
ViT-B + DAT + NS	81.33	<b>24.14</b>	<b>23.61</b>	<b>48.98</b>	<b>43.67</b>	<b>24.22</b>	<b>45.91</b>	70.14	48.48	<b>37.25</b>	23.87	<b>43.61</b>

Table 3: Comparison of our NS method with PGD-based adversarial robustness methods of Madry and TRADES. We report the performance for a ViT-S model.

Method	clean	FGSM	DamageNet	A	C (↓)	V2	R	Sketch	Stylized
ViT-S	74.19	13.79	29.82	16.35	71.13	62.51	34.67	14.26	12.15
Madry	70.53	<b>39.37</b>	<b>49.91</b>	9.37	81.74	58.88	39.04	21.36	10.76
TRADES	74.02	38.85	36.28	16.53	73.11	63.37	40.86	<b>26.43</b>	13.22
NS	<b>77.47</b>	25.95	32.43	<b>20.77</b>	<b>55.98</b>	<b>66.5</b>	<b>41.61</b>	25.67	<b>16.02</b>

The result in Table 2 shows that our nullspace finetuning method consistently improves the robustness of models under distribution shifts and adversarial attacks, yielding a large gain in average performance for the vanilla ViT-small and ViT-base model, and slightly outperforms various baselines consistently while also slightly outperforming DAT. This not only shows that our nullspace finetuning method is effective but also validates our previous hypothesis about the connection between the tolerance to nullspace and the robustness of transformer models.

### 5.3. Experiment: Adversarial Finetuning

In this experiment, we compare our method with fine-tuning using two PGD adversarial training methods, Madry [63] and TRADES [12] on the ViT-S model. TRADES, in each training iteration, generates adversarial examples using PGD and updates the model’s parameters to minimize the worst-case loss on these adversarial examples while also minimizing the standard classification loss on clean data. Madry, on the other hand, focuses exclusively on minimizing the worst-case loss on adversarial examples. In Table 3, we observe that Madry and TRADES provide better performance for adversarial evaluation. This is expected as the methods are catered for improving adversarial robustness. However, this exclusivity leads to relatively poorer performance in a wider benchmark evaluation. Compared to our method, Madry and TRADES perform considerably lower in the natural OOD setting.

### 5.4. Enlarged Approximate Nullspace

To gain more insight about the dynamics of our nullspace finetuning method, we monitor the  $l_2$  norm of the learned noise and various performance metrics during the training, as shown in Fig. 4. Before the nullspace finetuning, it was hard to optimize the noise into the  $\epsilon$  region even with increased training, so the norm started with a high value. As the training starts, we find that the noise was always able to enter the  $\epsilon$  region. In Appendix C, we show the MSE probability of the learned noise vectors after each round of noise learning, which were all smaller than  $\epsilon$ . More importantly, the norm

Table 4: **Impact of  $\epsilon$  on the final performance.** Moreover, we also compare our NS method against random  $\epsilon$  noise based finetuning.

$\epsilon$	Finetuning	FGSM	DamageNet	A	C ( $\downarrow$ )	V2	R	Sketch	Stylized
0.01	NS	<b>26.04</b>	<b>33.65</b>	<b>20.45</b>	56.26	<b>66.47</b>	<b>41.4</b>	<b>23.34</b>	<b>15.85</b>
	Random	21.54	28.81	17.07	<b>55.13</b>	61.98	34.97	14.43	12.14
0.03	NS	<b>25.95</b>	<b>32.43</b>	<b>20.77</b>	55.98	<b>66.5</b>	<b>41.61</b>	<b>25.67</b>	<b>16.02</b>
	Random	23.18	29.61	16.91	<b>54.68</b>	62.2	35.05	14.77	12.34
0.1	NS	<b>25.38</b>	<b>33.09</b>	<b>20.16</b>	56.41	<b>66.47</b>	<b>40.42</b>	<b>22.66</b>	<b>15.78</b>
	Random	23.93	30.56	16.47	<b>54.52</b>	62.48	34.66	14.99	12.35

of the learned noise gradually increases along the process of model fine-tuning. The fluctuation may have mainly resulted from the randomness in mini-batches and the optimization dynamics. The model allows for noises with larger and larger norms to be within  $\epsilon$ -approximate, which informally suggests an enlarging  $\epsilon$ -approximate nullspace. Accompanied by the trend is the increase in robustness scores in both OOD and adversarial settings, which corroborates our findings.

## 5.5. Ablation Study

We conduct an extensive study to analyse the performance of our method under choice of  $\epsilon$ . Furthermore, we also compare our approach with a simple baseline of using an  $\epsilon$  noise sampled from a Gaussian distribution.

From Table 4, we can infer that the nullspace noise based finetuning is relatively robust to the choice of  $\epsilon$ . Moreover, compared to using randomly generated  $\epsilon$ -noise, our nullspace based training provides significant performance boost. This observation stands across different values of  $\epsilon$ .

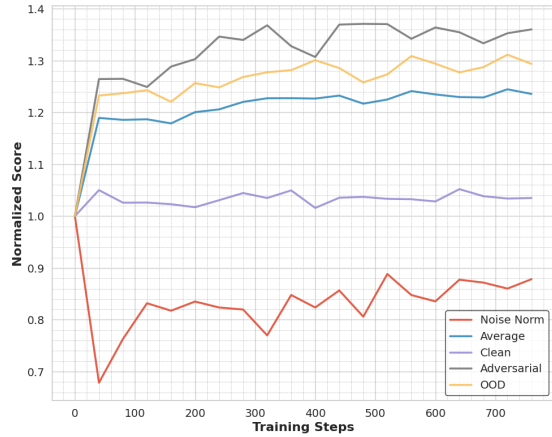


Figure 4: **Change trend of multiple metrics with training steps.** "Adversarial" is the average performance of the 4 adv. robustness settings, "OOD" is the average score on the six OOD datasets, and "avg" is the total average. All values are divided by their initial values to show the trend more clearly.

## 6. Discussion

**Applications in Model Patenting** In addition to the applications we discussed, we consider another potential usage of our findings is to patent a ViT after a model is trained, as the nullspace will be unique property of any set of weights of certain ViT architectures. Different from the existing line of research in model watermarking [64–66], the patenting through nullspace will not require any additional steps during training, although will face limited usage scenarios in comparison.

**Applications in Image Watermarking** Using the nullspace noise, it is possible to apply signatures onto input images without harming the output or operability of the networks. In the supplementary document, we present the cases where certain marks in form of nullspace noise can be superimposed on any desired input image.

**Potential Limitation about the Nullspace Approximation** Different from the nullspace defined in linear algebra, the nullspace of the entire ViT can only be approximated because of the non-linearity in the network architecture. However, it is worthy mentioning that we can still calculate the exact nullspace of ViT if we only consider the patch embedding layer, through which, our results will qualitatively deliver the same message, but with quantitative differences.



## 7. Conclusion

In this work, we have explored the concept of nullspace in Vision Transformers (ViTs) to understand their robustness. Our findings demonstrate that a non-trivial nullspace indeed exists for Vision Transformers, a direct consequence of the patch embedding layer. This discovery implies that there are elements that, when added to an input, do not affect the output of the network, potentially offering an explanation for the robustness exhibited by ViTs. Moreover, we have extended the definition of nullspace, preserving a property that implies invariance of a mapping’s output to input perturbations, and empirically identified a space that exhibits such property within the input space of the non-linear transformer encoder. By linking the presence of nullspace with our extended definition to the general robustness of a network, we were able to devise a new approach to improve the robustness of ViTs. Our empirical results suggest that fine-tuning ViTs with the learnt nullspace noise can significantly enhance their robustness to a variety of robustness benchmarks.

This study offers a new perspective to the robustness of vision transformers. We believe these findings can assist in furthering the robustness of ViTs, potentially facilitating advancements in the development of more resilient models. Looking forward, there is more to explore in this direction. Future research could focus on the development of efficient algorithms for learning nullspace and investigate its presence in other architectures and layers of deep neural networks.

## References

- [1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [2] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.
- [3] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual transformer networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [4] 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 (ICCV)*, 2021.
- [5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [6] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once, 2023.
- [7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6572>.
- [8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016. doi: 10.1109/CVPR.2016.282.
- [9] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015. doi: 10.1109/CVPR.2015.7298640.
- [10] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.
- [11] 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. URL <https://openreview.net/forum?id=rJzIBfZAb>.
- [12] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pages 7472–7482. PMLR, 2019.
- [13] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. URL <https://openreview.net/forum?id=r1Ddp1-Rb>.
- [14] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.
- [15] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple method to improve robustness and uncertainty under data shift. In *ICLR*, 2020. URL <https://openreview.net/forum?id=SigmrxFvB>.

- [16] Haoyang Liu, Maheep Chaudhary, and Haohan Wang. Towards trustworthy and aligned machine learning: A data-centric survey with causality perspectives. *arXiv preprint arXiv:2307.16851*, 2023.
- [17] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples, 2021.
- [18] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/rice20a.html>.
- [19] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, volume 34, pages 29935–29948. Curran Associates, Inc., 2021. URL [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/fb4c48608ce8825b558ccf07169a3421-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/fb4c48608ce8825b558ccf07169a3421-Paper.pdf).
- [20] Jacob Clarysse, Julia Hörmann, and Fanny Yang. Why adversarial training can hurt robust accuracy. In *ICLR*, 2023. URL <https://openreview.net/forum?id=-CA8yFkPc70>.
- [21] Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation. *The Journal of Machine Learning Research*, 21(1):9885–9955, 2020.
- [22] Han Shao, Omar Montasser, and Avrim Blum. A theory of pac learnability under transformation invariances. *Advances in Neural Information Processing Systems*, 35:13989–14001, 2022.
- [23] Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks. *arXiv preprint arXiv:2005.00178*, 2020.
- [24] Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *ICCV*, pages 774–782, October 2021.
- [25] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hassel, M.F. Balcan, and H. Lin, editors, *NeurIPS*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020. URL [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/d85b63ef0ccb114d0a3bb7b7d808028f-Paper.pdf).
- [26] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy M. Hospedales, Neil Martin Robertson, and Yongxin Yang. DADA: differentiable automatic data augmentation. 2020.
- [27] Evangelos Chatzipantazis, Stefanos Pertigkiozoglou, Kostas Daniilidis, and Edgar Dobriban. Learning augmentation distributions using transformed risk minimization. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=LRytNj8Xw0>.
- [28] Ignacio Hounie, Luiz F. O. Chamon, and Alejandro Ribeiro. Automatic data augmentation via invariance-constrained learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 13410–13433. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/hounie23a.html>.
- [29] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021.
- [30] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
- [31] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021.

- [32] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26831–26843. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/e19347e1c3ca0c0b97de5fb3b690855a-Paper.pdf>.
- [33] Yao Qin, Chiyuan Zhang, Ting Chen, Balaji Lakshminarayanan, Alex Beutel, and Xuezhi Wang. Understanding and improving robustness of vision transformers through patch-based negative augmentation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *NeurIPS*, pages 16276–16289, 2022.
- [34] Yao Xiao, Ziyi Tang, Pengxu Wei, Cong Liu, and Liang Lin. Masked images are counterfactual samples for robust fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20301–20310, 2023.
- [35] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.
- [36] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=4nPswr1KcP>.
- [37] Jihao Liu, Boxiao Liu, Hang Zhou, Hongsheng Li, and Yu Liu. Tokenmix: Rethinking image mixing for data augmentation in vision transformers. In *European Conference on Computer Vision*, pages 455–471. Springer, 2022.
- [38] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *ICLR*, 2022. URL <https://openreview.net/forum?id=LtKcMgG0eLt>.
- [39] Hila Chefer, Idan Schwartz, and Lior Wolf. Optimizing relevance maps of vision transformers improves robustness. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL [https://openreview.net/forum?id=upuYKQiyxa\\_](https://openreview.net/forum?id=upuYKQiyxa_).
- [40] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- [41] Haoyang Liu, Maheep Chaudhary, and Haohan Wang. Towards trustworthy and aligned machine learning: A data-centric survey with causality perspectives, 2023.
- [42] Shelly D. D. Goggin, Karl E. Gustafson, and Kristina M. Johnson. Accessing the null space with nonlinear multilayer neural networks. In Dennis W. Ruck, editor, *Science of Artificial Neural Networks*, volume 1710, pages 308 – 316. International Society for Optics and Photonics, SPIE, 1992. doi: 10.1117/12.140097. URL <https://doi.org/10.1117/12.140097>.
- [43] Sho Sonoda, Isao Ishikawa, and Masahiro Ikeda. Ghosts in neural networks: Existence, structure and role of infinite-dimensional null space. *CoRR*, abs/2106.04770, 2021. URL <https://arxiv.org/abs/2106.04770>.
- [44] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 184–193, June 2021.
- [45] Mohamed H. Abdelpakey and Mohamed S. Shehata. Nullspacenet: Nullspace convolutional neural network with differentiable loss function. *CoRR*, abs/2004.12058, 2020. URL <https://arxiv.org/abs/2004.12058>.

- [46] Jin Ho Kwak and Sungpyo Hong. *Linear Algebra*. Birkhäuser, Boston, MA, 2004. doi: 10.1007/978-0-8176-8194-4.
- [47] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, MA, fourth edition, 2009. ISBN 9780980232714 0980232716 9780980232721 0980232724 9788175968110 8175968117.
- [48] Gilbert Strang. *Introduction to Linear Algebra, Fourth Edition*. Wellesley Cambridge Press, February 2009. ISBN 0980232716. URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0980232716>.
- [49] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. Attentionviz: A global view of transformer attention. *arXiv preprint arXiv:2305.03210*, 2023.
- [50] Jeremy Howard. Imagenette, 2018. URL <https://github.com/fastai/imagenette/>.
- [51] 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.
- [52] Xiaofeng Mao, YueFeng Chen, Ranjie Duan, Yao Zhu, Gege Qi, Shaokai Ye, Xiaodan Li, Rong Zhang, and Hui Xue’. Enhance the visual representation via discrete adversarial training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=qtZac7A3-F>.
- [53] Sizhe Chen, Zhengbao He, Chengjin Sun, Jie Yang, and Xiaolin Huang. Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2188–2197, 2022. doi: 10.1109/TPAMI.2020.3033291.
- [54] Yonggan Fu, Shunyao Zhang, Shang Wu, Cheng Wan, and Yingyan Lin. Patch-fool: Are vision transformers always robust against adversarial perturbations? In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=28ib9tf6zhr>.
- [55] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, Los Alamitos, CA, USA, may 2017. IEEE Computer Society. doi: 10.1109/SP.2017.49. URL <https://doi.ieeecomputersociety.org/10.1109/SP.2017.49>.
- [56] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.
- [57] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.
- [58] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8349, October 2021.
- [59] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.
- [60] 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.
- [61] Xiaofeng Mao, Yuefeng Chen, Xiaodan Li, Gege Qi, Ranjie Duan, Rong Zhang, and Hui Xue. Easyrobust: A comprehensive and easy-to-use toolkit for robust computer vision. <https://github.com/alibaba/easyrobust>, 2022.



- [62] Hila Chefer, Idan Schwartz, and Lior Wolf. Optimizing relevance maps of vision transformers improves robustness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *NeurIPS*, volume 35, pages 33618–33632. Curran Associates, Inc., 2022. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/d9fa720cf96f7c18ac4d9e04270f0bbf-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/d9fa720cf96f7c18ac4d9e04270f0bbf-Paper-Conference.pdf).
- [63] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=rJzIBfZAb>.
- [64] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdoor. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631, 2018.
- [65] Bitar Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 485–497, 2019.
- [66] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244, 2020.
- [67] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [68] Raymond B. Wolfgang and Edward J. Delp. A watermark for digital images. *Proceedings of 3rd IEEE International Conference on Image Processing*, 3:219–222 vol.3, 1996.
- [69] Vidyasagar M Potdar, Song Han, and Elizabeth Chang. A survey of digital image watermarking techniques. In *INDIN’05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pages 709–716. IEEE, 2005.
- [70] Ali Al-Haj. Combined dwt-dct digital image watermarking. *Journal of computer science*, 3(9): 740–746, 2007.
- [71] Vivekananda Bhat, Indranil Sengupta, and Abhijit Das. An adaptive audio watermarking based on the singular value decomposition in the wavelet domain. *Digital Signal Processing*, 20(6): 1547–1558, 2010.
- [72] Hal Berghel. Digital watermarking makes its mark. *Networker*, 2(4):30–39, 1998.
- [73] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32, 2019.
- [74] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.
- [75] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems*, 32, 2019.
- [76] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, June 2021.
- [77] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626. IEEE Computer Society, 2017. ISBN 978-1-5386-1032-9. URL <http://dblp.uni-trier.de/db/conf/iccv/iccv2017.html#SelvarajuCDVPB17>.
- [78] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.385. URL

<https://aclanthology.org/2020.acl-main.385>.

## Supplementary Material

### A. Proof of Proposition 1

Let  $d$  be the hidden dimension of the attention layer.  $\mathbf{Q}_i, \mathbf{K}_i \in \mathbb{R}^{d \times d_k}$  where  $d_k = d/h$ .  $\text{rank}(\mathbf{Q}_i \mathbf{K}_i^\top) \leq \text{rank}(\mathbf{K}_i^\top) \leq d_k$ . Consider the sum of row spaces  $S = \text{R}(\mathbf{Q}_1 \mathbf{K}_1^\top) + \text{R}(\mathbf{Q}_2 \mathbf{K}_2^\top) + \dots + \text{R}(\mathbf{Q}_h \mathbf{K}_h^\top)$ .  $S$  is a subspace of  $\mathbb{R}^d$ . For  $i = 1, \dots, h$ , choose a basis for  $\text{R}(\mathbf{Q}_i \mathbf{K}_i^\top)$ , denoted as  $B_i = \{\mathbf{b}_1, \dots, \mathbf{b}_{n_i}\}$ ,  $|B_i| = n_i \leq d_k$ . Without loss of generality, let  $\mathbf{r}_{m,k} \in B_m$ .

$S = \text{span}(\bigcup_{i=1}^h B_i)$ , so

$$\begin{aligned} \dim(S) &= \dim\left(\text{span}\left(\bigcup_{i=1}^h B_i\right)\right) = \dim\left(\text{span}\left(\left(\bigcup_{\substack{i=1 \\ i \neq m}}^h B_i\right) \cup (B_m \setminus \{\mathbf{r}_{m,k}\})\right)\right) \\ &\leq \left|\left(\bigcup_{\substack{i=1 \\ i \neq m}}^h B_i\right) \cup (B_m \setminus \{\mathbf{r}_{m,k}\})\right| \leq (h-1)d_k + (d_k - 1) = d - 1. \end{aligned} \quad (9)$$

So,  $\exists \mathbf{w} \in \mathbb{R}^d$ ,  $\mathbf{w} \neq \mathbf{0}$  and  $\mathbf{w} \in S^\perp$ . This means for  $i = 1, \dots, h$ ,  $\mathbf{w} \in (\text{R}(\mathbf{Q}_i \mathbf{K}_i^\top))^\perp$ ,  $\mathbf{w} \in \text{N}(\mathbf{Q}_i \mathbf{K}_i^\top)$ . By condition 2,  $\text{N}(\mathbf{V}_i) \supseteq \text{N}(\mathbf{Q}_i \mathbf{K}_i^\top)$ , so  $\mathbf{w} \in \text{N}(\mathbf{Q}_i \mathbf{K}_i^\top) \cap \text{N}(\mathbf{V}_i^\top)$ .

Then, we can choose  $\mathbf{W}$  wherein each row is a multiple of  $\mathbf{w}$ . We have  $\mathbf{WV}_i = \mathbf{0}$ , and for any input to the encoder  $\mathbf{X} \in \mathbb{R}^{n \times d}$ ,

$$\mathbf{WQ}_i \mathbf{K}_i^\top \mathbf{X}^\top + \mathbf{XQ}_i \mathbf{K}_i^\top \mathbf{W}^\top + \mathbf{WQ}_i \mathbf{K}_i^\top \mathbf{W}^\top = \mathbf{0}. \quad (10)$$

Consider the output of attention head,

$$\begin{aligned} \text{head}_i(\mathbf{X} + \mathbf{W}) &= \text{Softmax}\left(\frac{(\mathbf{X} + \mathbf{W}) \mathbf{Q}_i \mathbf{K}_i^\top (\mathbf{X} + \mathbf{W})^\top}{\sqrt{d_k}}\right) (\mathbf{X} + \mathbf{W}) \mathbf{V}_i \\ &= \text{Softmax}\left(\frac{\mathbf{XQ}_i \mathbf{K}_i^\top \mathbf{X}^\top + \mathbf{WQ}_i \mathbf{K}_i^\top \mathbf{X}^\top + \mathbf{XQ}_i \mathbf{K}_i^\top \mathbf{W}^\top + \mathbf{WQ}_i \mathbf{K}_i^\top \mathbf{W}^\top}{\sqrt{d_k}}\right) \mathbf{XV}_i \\ &= \text{Softmax}\left(\frac{\mathbf{XQ}_i \mathbf{K}_i^\top \mathbf{X}^\top}{\sqrt{d_k}}\right) \mathbf{XV}_i = \text{head}_i(\mathbf{X}). \end{aligned} \quad (11)$$

Adding the noise  $\mathbf{W}$  does not change the output of any attention head for arbitrary input  $\mathbf{X}$ , which completes our proof.

## B. Algorithm and implementation details

We present the algorithm of our data augmentation with nullspace noise in Algorithm 1.

---

**Algorithm 1:** Data augmentation with nullspace noise

---

```

1 Input: transformer model with patch embedding layer  $f_e$ , encoder  $f_\phi$  and linear classifier  $f_\psi$ 
   parameterized by  $e, \phi, \psi$  respectively; training data  $\mathcal{T}$ ; batch size  $B$ ; sampling limit  $A$ ; noise
   nullity threshold  $\epsilon$ ; noise learning rate  $\eta_v$ ; model learning rate  $\eta_f$ ; number of outer iterations
    $K$ , noise training step  $T$ , model training step  $S$ 
2 for  $k = 0, \dots, K - 1$  do
3   Sample initial noise  $\mathbf{v} \sim \mathcal{U}(-\text{lim}, \text{lim})$ 
4   for  $t = 0, \dots, T - 1$  do
5     Sample a minibatch  $(\mathbf{X}, \mathbf{y}) \sim \mathcal{T}$ 
6     Compute  $\mathbf{U} \leftarrow f_e(\mathbf{X})$ 
7     Compute logits  $\mathbf{Z} \leftarrow f_\psi(f_\phi^0(\mathbf{U}))$ ,  $\mathbf{Z}' \leftarrow f_\psi(f_\phi^0(\mathbf{U} + [\mathbf{v}]))$  # "[v]" means
       broadcasting the noise  $\mathbf{v}$  along the sample dimension
8     Compute  $\delta \leftarrow \frac{1}{B} \sum_{i=1}^B \|\text{Softmax}(\mathbf{z}_i') - \text{Softmax}(\mathbf{z}_i)\|^2$  #  $\mathbf{z}_i$  is sample logit
9     if  $\delta < \epsilon$  then
10      | break
11    end
12    Calculate  $\ell \leftarrow \frac{1}{B} \sum_{i=1}^B \|\mathbf{z}_i' - \mathbf{z}_i\|^2$ 
13    Update  $\mathbf{v} \leftarrow \mathbf{v} - \nabla_{\mathbf{v}} \ell$ 
14  end
15  for  $s = 0, \dots, S - 1$  do
16    Sample a minibatch  $(\mathbf{X}, \mathbf{y}) \sim \mathcal{T}$ 
17    Compute  $\mathbf{U} \leftarrow f_e(\mathbf{X})$ 
18    Compute logits  $\mathbf{Z} \leftarrow f_\psi(f_\phi^0(\mathbf{U}))$ ,  $\mathbf{Z}' \leftarrow f_\psi(f_\phi^0(\mathbf{U} + [\mathbf{v}]))$ 
19    Compute loss  $\mathcal{L} \leftarrow \frac{1}{B} \sum_{i=1}^B (\ell(\mathbf{z}_i, y_i) + \ell(\mathbf{z}_i', y_i))$ , where  $\ell$  is the cross-entropy loss
20    Update model parameters  $(\psi, \phi, e) \leftarrow (\psi, \phi, e) - \nabla_{(\psi, \phi, e)} \mathcal{L}$ 
21  end
22 end
23 Output: model weight  $(\psi, \phi, e)$ 

```

---

**Hyperparameters** We fine-tuned the ViT model for  $K = 20$  rounds in all settings. In each round, we initialized the noise with sampling limit  $A = 3$ , optimized it with learning rate  $\eta_v = 0.1$  and set a threshold of  $\epsilon = 0.03$ . We empirically found that  $T = 3000$  is enough to trigger early stopping so that the learned noise satisfies the  $\epsilon$  threshold. We used  $\eta_f = 10^{-5}$  to fine-tune the model for  $S = 40$  iterations in each round. We set batch size  $B = 128$ , and slightly different from the vanilla SGD in Alg 1, we used the AdamW optimizer [67] and cosine learning rate scheduler with default hyperparameters for both the noise and the model training.

The original ViT-B + DAT model [52] used the Exponential Moving Average (EMA) for evaluation<sup>3</sup>, so we also used EMA to evaluate the performance of ViT-B + DAT fine-tuned with our method. For all the other settings, we used single model without ensemble for evaluation. We used  $\epsilon = 1/255$  for the FGSM attack consistent with [52].

**Computation time** The experiments were conducted on a combination of A100, V100 GPUs and a 3090 GPU, depending on the availability. Although we only used about 10% of the ImageNet-1k [51] training data to fine-tune the model, the main computation time is on training the nullspace noise. One run of our experiment (20 rounds) takes the time roughly equivalent to 8 epochs of standard training on ImageNet-1k.

---

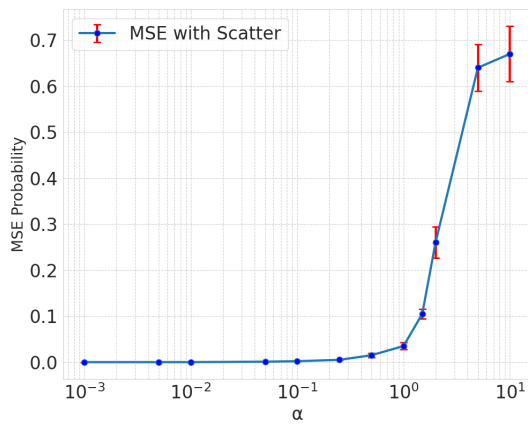
<sup>3</sup><https://github.com/alibaba/easyrobust>

## C. Change trend of the noise influence with the fine-tuning steps

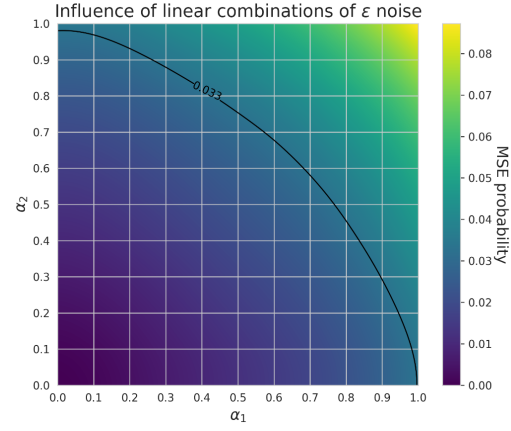
Beside the trend of noise norm and performance metrics in Fig. 4, we also keep track of the influence of the learned noise in terms of MSE probability (3.2) at every 80 steps of the model fine-tuning. As shown in Table 5, the noise influence is always lower than  $\epsilon = 0.03$ , which means early stopping is triggered and the model enters the  $\epsilon$  region.

Table 5: MSE probability of the noise at different fine-tuning steps.

Fine-Tuning Step	40	120	160	280	360	440	520	600	680	760
MSE Probability	0.028	0.027	0.026	0.029	0.028	0.029	0.027	0.025	0.028	0.026



(a) Influence of  $\epsilon$  noise under multiplication with different  $\alpha$



(b) Influence of  $\epsilon$  noise under convex combination with different  $\alpha_1, \alpha_2$

Figure 5: Validation of the properties of the  $\epsilon$ -approximate nullspace.

## D. Approximate Nullspace Properties

To explore the property of the  $\epsilon$ -approximate nullspace, we conduct an experiment to observe the behavior of the learned noise vectors under scalar multiplication and convex combination. For this, we first construct a set of  $m$   $\epsilon$ -approximate nullspace vectors  $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^m$  starting from different random initializations using  $\epsilon = 0.033$ . For scalar multiplication, we vary the scaling factor  $\alpha$  and report the mean influence of  $\alpha\mathbf{v}$  on the model's predictions in terms of MSE probabilities (Figure 5(a)). For convex combination, we sample  $n$  different pairs of nullspace vectors from  $\mathbf{V}$ , denoted as  $\mathbf{P} = \{(\mathbf{v}_{\mathcal{J}_{k,1}}, \mathbf{v}_{\mathcal{J}_{k,2}})\}_{k=1}^n$ , where  $\mathcal{J}_{k,1}, \mathcal{J}_{k,2} \in \{1, 2, \dots, m\}, \forall k \in \{1, \dots, n\}$ . Then, we vary  $\alpha_1$  and  $\alpha_2$  between  $[0, 1]$  with a grid size of 0.1, and for each combination of  $(\alpha_1, \alpha_2)$ , we evaluate the influence of the convex combination  $\alpha_1\mathbf{v}_{\mathcal{J}_{k,1}} + \alpha_2\mathbf{v}_{\mathcal{J}_{k,2}}$  on the model's prediction in MSE probability, averaged over all values of  $k$ . In practice we set  $m = 100, n = 10$ . The influence of the linear combined noise at each point of the grid is visualized as a heatmap as shown in Figure 5(b).

The results in Figure 5 show that the approximate nullspace has similar property to vector space in terms of closure under addition and scalar multiplication within a certain range of coefficients. When the scaling factor  $\alpha < 1$ , we see a clear trend that the MSE probability of the scaled noise is less than  $\alpha\epsilon$ . In the linear combination case, the line  $\alpha_1 + \alpha_2 = 1$  is well within the contour line of MSE probability being 0.033, showing that the convex combination of a pair of  $\epsilon$  noise vector is still  $\epsilon$ -approximate.



## E. Watermarking Images

Watermarking as image, usually used to convey ownership information or verify content of the data, has been studied extensively [68–71]. A watermark can be either imperceptible or perceptible. and perceptible watermarking applies a noticeable marker to convey the protected nature of the data [72]. In this section, we explore to utilize nullspace noise to apply a perceptible watermark on the image which does not alter the test-time forward process.

Figure 6 shows an example watermarking approach using the nullspace noise. Here, we emboss the ICML logo onto the natural images. The resulting modified image, attains the final predictions close to the original image. (100% match in the final output prediction and  $10^{-4}$  difference in the predicted confidence value of the assigned class.)

**Method details:** With basis vectors of the nullspace, we can construct a watermark to be overlaid on the original image without affecting the output of the network. Given a source (user’s image) and a target image (watermark), we simply need to estimate the scalar parameters corresponding to the basis vectors to satisfy  $\sum_{i=0}^{i < m} \mathbf{e}_i \lambda_i = \mathbf{v}_\theta \approx \Delta \mathbf{x}_j$ .

$\mathbf{e}_i$  are the basis vectors for the nullspace,  $\lambda_i$  are their corresponding scalar co-efficients which are to be determined and  $\Delta \mathbf{x}_j$  is the changed required to convert  $j^{\text{th}}$  original image patch to  $j^{\text{th}}$  watermark image patch. This can be achieved through a constrained optimisation of the following form:

$$\min \|\Delta \mathbf{x}_j - \sum_{i=0}^{i < m} \mathbf{e}_i \lambda_i\|_p. \quad (12)$$

Here,  $\Delta \mathbf{x}_j$  is the difference between the  $j^{\text{th}}$  channel of a source and target image and  $\lambda_i$  is the  $i^{\text{th}}$  nullspace basis vector of the patch embedding layer with the corresponding variable scalar  $e_i$ . We use a least-square solver to solve for the solution (Available readily with packages such as Numpy).

## F. Targeted Nullspace Noise

Due to the dimension reduction effect of the patch embedding layer in most ViTs, we can transfer an image to be visually similar to another image by human perception, without changing the output of the original image perceived by the model. This differs from adversarial examples in the following aspects:

1. The working direction to construct an adversarial example is the other way around. If the transformed image is to be considered an adversarial example, then our source becomes the target for adversarial training and our target becomes the source.
2. Generating targeted nullspace noise requires no backpropagation through the network
3. Not only does the final prediction on the transformed image matches the source image, the saliency maps also match. This is displayed in Fig. 7

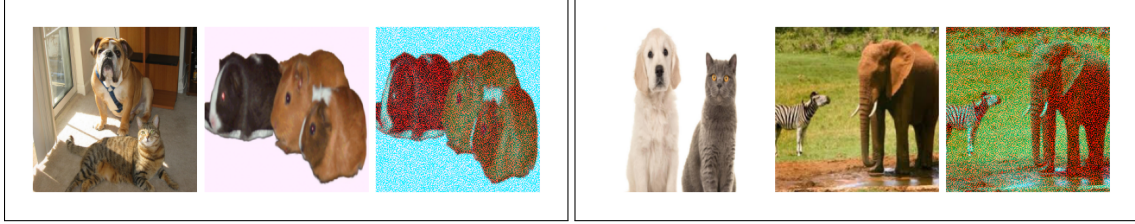
Though the transformation is not perfect, we can spot that the transformed images are visually similar to target images rather than source images. Even with this difference in the input space, transformed images and source images are classified into the same category with roughly the same confidence.

As recent studies have shown, fooling can also be extended to the interpretability methods (XAI) Dombrowski et al. [73] partially due the limitations exposed by recent studies [73–75]. However, in contrast to these works aiming to fool specific XAI method, our nullspace noise only depends on the model, not the XAI method.

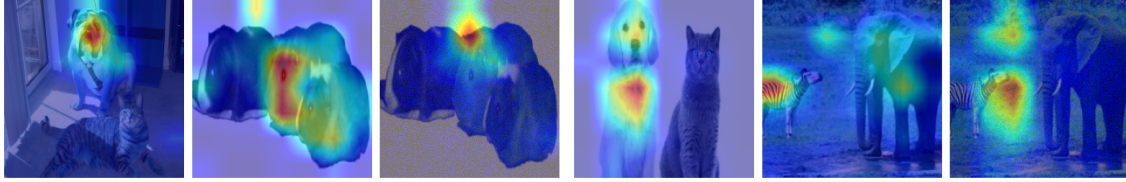
In Fig. 7(b), we show the interpretability maps as generated by LRP [76]. From the figure, we can observe that the heatmaps generated by source and transformed images are identical whereas, the



Figure 6: Watermark superposition using the nullspace basis vectors.



(a) Triplet of Source, target and transformed images



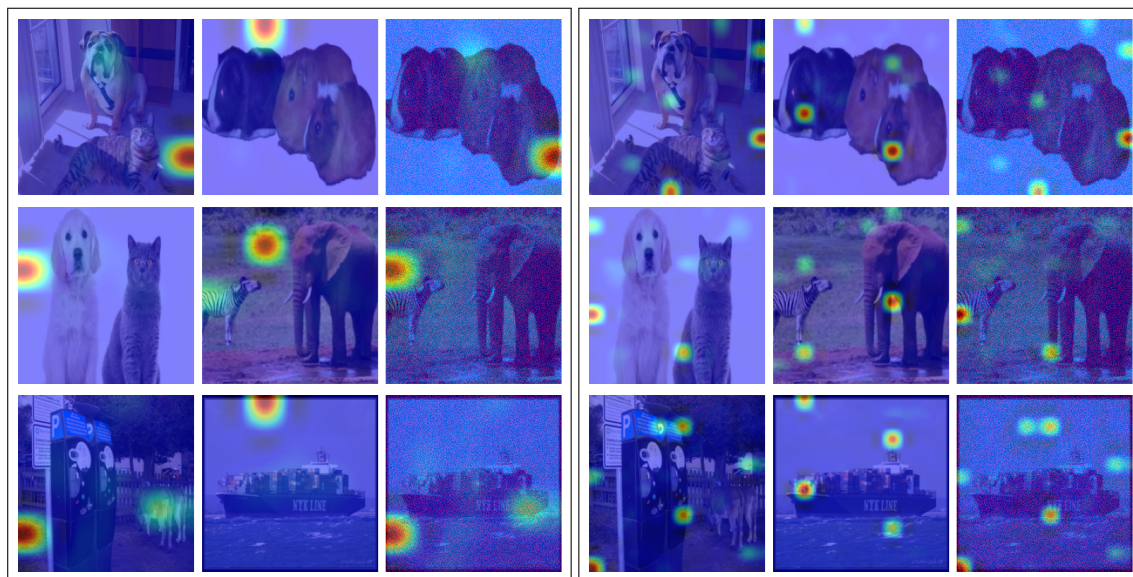
(b) Saliency maps for the corresponding images from the row above.

Figure 7: **Targeted nullspace noise.** Transformed images appear visually as target images but are interpreted as source images by the model. The equivalence between source and transformed images is not only in terms of the final predictions but also in the interpretability maps depicted in (b).

transformed image heatmaps substantially differ from target images'. Though only reported for LRP, we observed that a similar observation holds across different interpretability approaches. Here, we only presented the results on LRP, as in the context of ViTs, we found the heatmaps from other methods to be lacking (also pointed out by authors of LRP).

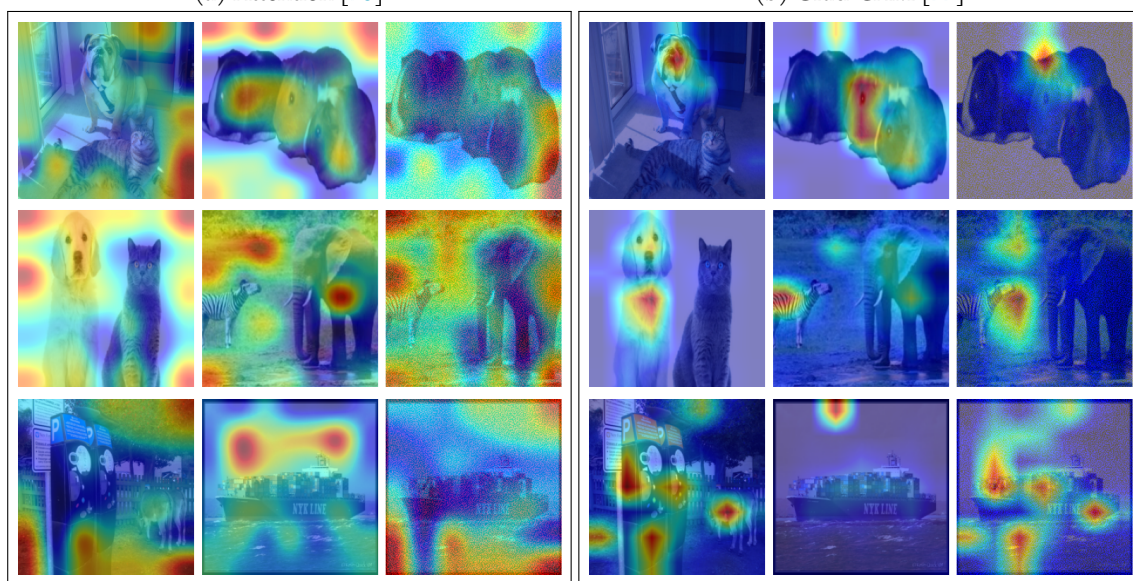
In Fig. 8 we show the saliency maps generated by different XAI methods. Even though the maps generated by methods other than LRP are poor (hard to interpret), we see that the source and transformed respond similarly to these methods.





(a) Attention [76]

(b) Grad-CAM [77]



(c) Rollout [78]

(d) LRP [76]

Figure 8: Interpretability maps generated via different methods for (source, target, transformed) images