SINDER: Repairing the Singular Defects of DINOv2

Haoqi Wang[1], Tong Zhang[1], and Mathieu Salzmann[1,2]

[1] School of Computer and Communication Sciences, EPFL, Switzerland
[2] Swiss Data Science Center, Switzerland
{haoqi.wang, tong.zhang, mathieu.salzmann}@epfl.ch

Abstract. Vision Transformer models trained on large-scale datasets, although effective, often exhibit artifacts in the patch token they extract. While such defects can be alleviated by re-training the entire model with additional classification tokens, the underlying reasons for the presence of these tokens remain unclear. In this paper, we conduct a thorough investigation of this phenomenon, combining theoretical analysis with empirical observations. Our findings reveal that these artifacts originate from the pre-trained network itself, specifically stemming from the leading left singular vector of the network's weights. Furthermore, to mitigate these defects, we propose a novel fine-tuning smooth regularization that rectifies structural deficiencies using only a small dataset, thereby avoiding the need for complete re-training. We validate our method on various downstream tasks, including unsupervised segmentation, classification, supervised segmentation, and depth estimation, demonstrating its effectiveness in improving model performance. Codes and checkpoints are available at `https://github.com/haoqiwang/sinder`.

Keywords: DINOv2 · Singular Defect · Unsupervised Segmentation

1 Introduction

Self-supervised learning (SSL) has emerged as a highly effective method for network pre-training [18], producing features beneficial across a wide spectrum of downstream tasks. SSL significantly accelerates large-scale training for vision models, exemplified by recent advancements such as DINOv2 [18]. While SSL models excel in image classification tasks, their use for comprehensive image understanding, e.g., segmentation, is significantly challenged by the presence of defective patch tokens, as depicted in Figure 1. These anomalies materialize as high-norm tokens in the feature maps of vision transformers. Despite research efforts to understand this phenomenon [7], explanations remain in their infancy. Current observations indicate that these flawed patches offer minimal local information, suggesting a tendency for large and deep vision transformers to recycle redundant patch tokens to store more useful information. To this day, the only approach to addressing this issue [7] requires re-training the network from scratch with additional register tokens on vast amounts of data, which is typically prohibitive and offers limited explanations of the underlying phenomenon.

In this paper, we aim to bridge the understanding gap of these defects by providing mathematical explanations. In contrast to previous work attributing the defects to the image classification token, we discover that these defects inherently exist and share high similarity across the entire dataset. To further explore and develop theoretical foundations, we linearize the weights of each network block. Our analysis reveals a strong correlation between the defects in each layer and the corresponding leading left singular vector of the linearized operations. We thus term this phenomenon *singular defects*. Importantly, our analysis evidences that such singular defects depend solely on the pre-trained network weights, and not on the inputs.

To mitigate these singular defects, we propose a method based on fine-tuning the singular values of linear layers in the network. Specifically, we impose a smoothness regularization on the detected defective tokens to rectify them and restrict the number of learnable parameters to as few as possible to retain the original feature quality. Our approach can fine-tune pre-trained large-scale models using only a small dataset without the need for labeled data. Our experimental results demonstrate that our method effectively rectifies these defects and enhances performance in semantic segmentation tasks, particularly in the unsupervised setting, while maintaining performance in classification tasks. Compared to the existing solution of [7], which requires retraining networks on private LVD-142M data [18], our method offers advantages in terms of reduced carbon emissions, memory footprint, and time consumption. Considering the limited availability of proprietary large-scale datasets, our approach offers a viable and economical-friendly solution for deploying large-scale models.

In a nutshell, our contributions can be summarized as follows:

- We unveil the correlation between the leading left singular vector and the direction of defects, enabling us to theoretically predict the direction of defects for each layer.
- We introduce a data-efficient fine-tuning technique to address the singular defects of DINOv2 without necessitating access to large-scale datasets.
- We conduct extensive experiments to study the properties of the defects and show that our proposed solution not only retains the feature quality for downstream classification tasks but also improves the pixel-level prediction tasks such as unsupervised segmentation.

2 Motivation

A prominent characteristic of the defective patch tokens in the last layer is their high norm, a phenomenon that was also observed in previous work [7]. We illustrate the norms of the patch tokens of several images in Figure 1. The norm of the defective tokens is much larger than that of normal tokens. For example, on 500 randomly selected natural images from the validation set of ImageNet-1K, their average norms are 434.0 vs. 57.6, respectively.

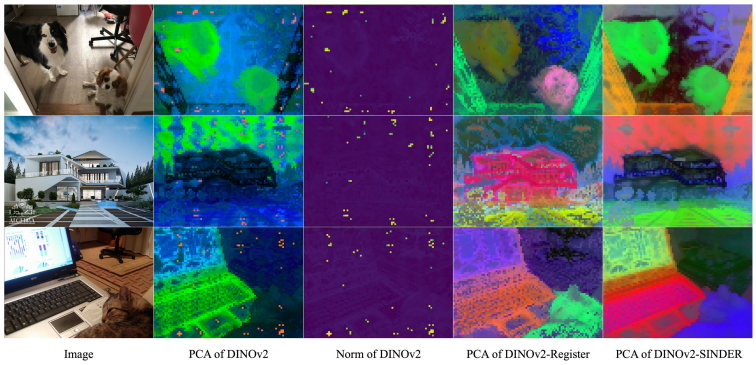


Fig. 1: Visualization of singular defects in the feature map of the last layer of DINOv2. The images are resized to have height 896 when input into the networks. The color of the PCA visualization comes from the three principal components of the patch tokens.

The second characteristic of defective patch tokens is that their directions are largely image-independent. Firstly, the feature directions of the defective patch tokens within each image are almost the same. This can be intuitively seen from Figure 1, whose second column depicts a PCA representation of the tokens. High-norm tokens in the same image have the same color, indicating that their directions are close to each other. To quantify this, we compute the average pair-wise angles between the defects within each of the 500 images. Their mean is 3.1 degrees. By contrast, the statistics of the average pair-wise angles between all patch tokens within each image is 72.8 degrees.

Secondly, the defective patch tokens are almost the same across different images. To see this, we calculate the average defective tokens for each image and then compute the average pair-wise angles between them. The mean is 5.5 degrees. This confirms that the defect direction in the last layer is in essence input image agnostic. This observation differs from that in [7], where high-norm tokens were claimed to contain image-wise global information. Based on our statistics, these high-norm tokens primarily contain no input information, regardless of local or global. It thus seems natural to ask, *can the defect directions be directly inferred from the pre-trained weights without knowing the input image?*

These two observations of defects remind us of the power method [24] in linear algebra, where a vector is recursively multiplied by a square matrix; the vector converges to the leading eigenvector, regardless of its initial direction, and its norm explodes if the largest eigenvalue is larger than 1. This motivates us to approach the problem of defects from the perspective of singular value decomposition, which will be the central topic of Section 3.

As a sanity check, we clamp the singular values of the weights of all the linear layers in DINOv2 to a smaller value and found that the high-norm defects are reduced (visualizations can be found in the Appendix). Encouraged by this, in Section 4, we design a regularization strategy to limit the magnitude of the singular values and thus the defective patch tokens.

3 Singular Defect Direction

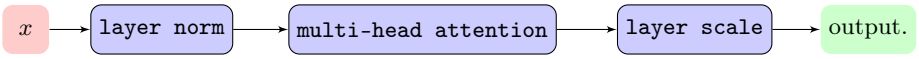
Now we delve into the origin of singular defects, focusing on the DINOv2 giant model. The DINOv2 giant is a Vision Transformer (ViT) model comprising 40 transformer layers, each containing an Attention Block and an MLP Block. These blocks act as residuals, with an identity path connecting their input and output. Our objective is to analyze the influence of each block on the defective tokens and to predict defective token directions theoretically solely from the pre-trained weights of the network, in an input-agnostic manner.

To evaluate the quality of our theoretical predictions, we manually extract the defect directions of 500 images from the ImageNet validation set. For each layer, we compute the average defect direction across these images, termed the *empirical defect direction*. Since evident defects on feature maps only manifest after the 15th layer, we focus solely on gathering defect directions from the 15th layer onwards. A good theoretical estimation of the defect direction is expected to closely align with this empirical defect direction.

It's worth noting that to facilitate tractability in our analysis, we consider the simplified scenario where there is *only one input token*. Under this assumption, the transformer layer can be approximated by linear transformations, as we will demonstrate below, rendering theoretical analysis feasible. Although the analysis is conducted in a simplified setting, we confirmed that the theoretical predictions of the singular defects are accurate.

3.1 Linear Approximation of an Attention Block

For an input token $x \in \mathbb{R}^D$, the computation of the Attention Block is



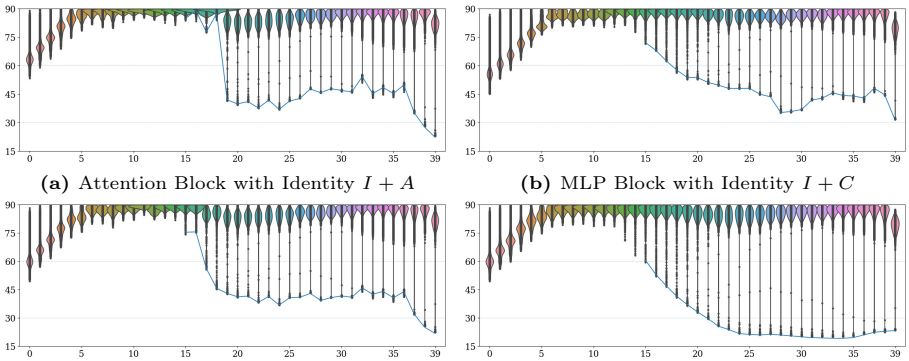
Under the single-token assumption, we will show that we can approximate these operations as linear transformations. Let us first study the `layer norm`. This operation is non-linear because of the division by the standard deviation. However, if we ignore this rescaling, the rest is linear and can be written as $A_1(A_0x) + b_1$, where $A_0 = I - \frac{1}{D}\mathbb{1}_{D\times D}$ is the centering, $A_1 = \mathrm{diag}(w)$ is the diagonal matrix of scaling parameters, and b_1 contains the bias parameters.

For the `multi-head attention`, as we analyze for a single token case, the softmax over a singleton is a constant 1. Hence, we only need to consider the `value` parameters. Let the weights concatenated over all `value` heads be $A_2 \in \mathbb{R}^{D\times D}$, the concatenated biases from all heads be $b_2 \in \mathbb{R}^D$, and the weights and biases of the output projection be A_3 and b_3, respectively, then the multi-head attention can be written as $A_3(A_2x + b_2) + b_3$.

Finally, we rewrite the `layer scale` as A_4x, where $A_4 = \mathrm{diag}(w)$ is the diagonal matrix of scaling parameters.

Combining the above operations, the Attention Block can be approximated as a series of linear transformations,

$$\mathrm{Attention}(x) \approx A_4(A_3(A_2(A_1(A_0x) + b_1) + b_2) + b_3) := Ax + b. \tag{1}$$



(a) Attention Block with Identity $I + A$

(b) MLP Block with Identity $I + C$

(c) Layer: Attention and MLP Block E

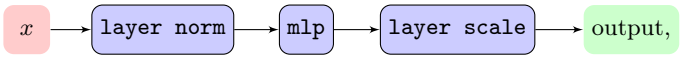
(d) Composed multiple layers G

Fig. 2: Angle between theoretical and empirical defect directions. Blue lines are the angle between the empirical defect direction and the leading left singular vector of $I + A$, $I + C$, E, G, respectively. Angles between the leading left singular vectors and all the patch tokens in each layer are shown as violin plots. The x-axis is the layer index, and the y-axis is the acute angle in degrees. The villa image in Figure 1 is used.

Drawing inspiration from the power method, we relate the empirical defect directions with the leading left singular vector corresponding to the largest singular value of $I + A$. In Figure 2a, we depict the angles between the leading left singular vector and the empirical defective tokens for each layer as a blue line. It is evident that after layer 19, these angles converge, with values consistently below 40 degrees. Additionally, we present the angles between the leading left singular vector and all patch tokens of an image as violin plots. The defective tokens are identifiable as isolated points within the violin plots (more visualizations in the Appendix). These findings suggest that the angles between the leading left singular vector and the patch tokens serve as a reliable metric for detecting defective tokens.

3.2 Linear Approximation of an MLP Block

The computation graph for an MLP Block is



where `layer norm` and `layer scale` can be processed in the same manner as in the Attention Block. However, the `mlp` layer is non-linear, expressed as $C_3(\text{silu}(W_1x + h_1) \odot (W_2x + h_2)) + d_3$, where $\odot$ is the element-wise product, C_3, W_1, W_2 are weights, and h_1, h_2, d_3 are the biases. We approximate the `mlp` using least squares. Specifically, we sample 100,000 random inputs $X \in \mathbb{R}^{D\times 100{,}000}$ and compute their outputs $Y = \text{silu}(W_1X + h_1) \odot (W_2X + h_2) \in \mathbb{R}^{M\times 100{,}000}$, where M is the output dimension of W_1. We solve the least-square problem $C_2X = Y$, and obtain the linear approximation with matrix $C_2 \in \mathbb{R}^{M\times D}$.

Ultimately, we can approximate the MLP Block as

$$\mathrm{MLP}(x) \approx C_4(C_3(C_2(C_1(C_0x) + d_1)) + d_3) := Cx + d, \tag{2}$$

where $C_0 = I - \frac{1}{D}\mathbb{1}_{D\times D}$ is the centering of the `layer norm`, C_1 is the diagonal matrix of the `layer norm` scaling weights, d_1 is the bias of the `layer norm`, and C_4 is the diagonal matrix of the `layer scale` scaling weights. The angles between the leading left singular vector of $I+C$ and the empirical defect directions are shown in Figure 2b.

3.3 Combining Attention and MLP Blocks

Based on our previous approximations, we can combine the linearized Attention Block and MLP Block as follows,

$$\mathrm{Layer}(x) \approx x + Ax + b + C(x + Ax + b) + d := Ex + f, \tag{3}$$

where the identity path is incorporated. We plot the effect of the leading left singular vector of E in Figure 2c, which resembles both Figure 2a and Figure 2b.

3.4 Predicting Defective Token Direction for Each Layer

We can further improve the prediction of defect direction by composing the linear approximations from layer 0 to layer i, where the matrix multiplied with x is

$$G_i := E_iE_{i-1}\cdots E_0. \tag{4}$$

The result of the leading left singular vector of G_i is shown in Figure 2d. We find that after layer 20, the leading left singular vectors are very close to the empirical defect directions, and from layers 15 to 19, the result is also better than previous attempts. Thus, we define the leading left singular vector of G_i as the theoretical *singular defect direction*[3] for layer i. Figure 2d demonstrates that we can accurately predict the empirical defect direction by the singular defect direction. So we had referred to this type of defective tokens as *singular defects*.

Note that the definition of singular defect direction originates solely from the pre-trained network weights; it does not depend on the input image in inference.

4 Repairing Singular Defects

Having identified the connection between the singular defect direction and the empirical defect direction, we next aim to repair the singular defects of the network with minimum modifications to the network parameters. A key requirement of such repairs is that they should maintain the feature quality of the original network. Without the defective tokens, we expect a spatially smooth

[3] We do not differentiate between a singular defect direction and its negative direction.

and coherent feature map, thus leading to stronger performance for dense prediction downstream tasks. We identify two key aspects that contribute to this goal. First, imposing smooth regularization suffices to ensure the resulting network produces spatially smooth and coherent feature maps. Second, to maintain the feature quality, the algorithm should modify as few parameters as possible, refraining from overtraining the network.

Based on these observations, we design an algorithm called ***Sin****gular* ***D****efect* ***R****epairing* (SINDER, Algorithm 1). In essence, SINDER aims to repair the first defective layer encountered in the forward pass using a smooth regularization, by modifying a few parameters. We describe the corresponding loss in Section 4.1 and discuss the importance of limiting the number of learnable parameters in Section 4.2.

Algorithm 1 Singular Defect Repairing (SINDER)

Input: A pre-trained network, a finetune dataset, termination threshold $\rho = 25\%$, $M = 500$, skip threshold $\sigma = 3$, mask threshold $\mu = 4$, learnable layers $\lambda = 10$

Compute singular defect direction ν_i for each layer i of the pre-trained network
while more than ρ of recent M images are not clear **do**
  Sample an image
  for all layers i **do**
    Find defective tokens of layer i using ν_i ▷ See Equation (5)
    if the number of defective tokens is less than σ **then**
      continue ▷ Skip the current layer
    end if
    Compute loss using Equation (8)
    Backward and update the parameters from layer $i - \lambda$ to layer i
    break ▷ Skip remaining layers
  end for
end while

4.1 Loss Design

The core idea underlying our method is to first identify the defective tokens and then apply a spatial smoothness prior to regularizing them. Let the patch tokens of a layer be x_t, $t = 1, \ldots, T$, where $T = H \times W$ is the number of tokens, and let the singular defect direction of the ith layer of the network be ν_i. We identify the defective tokens as follows. First, define the logit l_t as the absolute value of the inner product between the normalized patch token and ν_i, *i.e.*,

$$l_t = \left| \frac{x_t}{\|x_t\|} \cdot \nu_i \right|. \tag{5}$$

Then, we take the set of defective tokens $\mathcal{D}$ to be those that deviate from the mean logit by more than the *mask threshold* $\mu = 4$ times the standard deviation.

For a defective token $x_t \in \mathcal{D}$, we define its learning target based on the weighted average of its 3×3 spatially neighboring tokens $\mathcal{N}_t$. Let token $x_{t'} \in \mathcal{N}_t$ be a neighboring token of x_t. Then, we compute the coefficient

$$c_{tt'} = \frac{\exp(-l_{t'}/\tau)}{\sum_{s \in \mathcal{N}_t} \exp(-l_s/\tau)}, \tag{6}$$

where τ is a temperature hyperparameter. Additionally, we multiply $c_{tt'}$ with a 3×3 Gaussian kernel and re-normalize the resulting coefficients. This step assigns greater weight to closer neighbors compared to farther ones. Finally, we utilize the resulting coefficients, denoted as $\tilde{c}_{tt'}$, to linearly combine the 3×3 neighboring tokens into the learning target of each defective token x_t as

$$\tilde{x}_t = \sum_{t' \in \mathcal{N}_t} \tilde{c}_{tt'} x_{t'}. \tag{7}$$

We define our loss function L as the average distance between the defective tokens and their respective learning targets, which can be expressed as

$$L = \frac{1}{|\mathcal{D}|} \sum_{t \in \mathcal{D}} \|x - \tilde{x}_t\|. \tag{8}$$

If the number of defects for every layer is less than the skip threshold σ, then we call this image *clear*.

4.2 Limiting the Number of Learnable Parameters

Given the fact that we fine-tune the model with significantly fewer images compared to the original training set, it becomes imperative to control the number of trainable parameters to avoid compromising the model's generalization ability in downstream tasks. Our observation of a profound connection between the leading left singular vector of network operations and the empirical defect directions serves as the foundation for our approach. Based on the intuition from the power method, the high norm of defects is related to the corresponding leading singular value. We thus propose to constrain learning to singular values only. Specifically, we decompose the weight of every linear layer in DINOv2 as USV^T using SVD and freeze the parameters U and V during fine-tuning. This greatly reduces the number of learnable parameters.

Furthermore, our experiments reveal that further restricting the number of learnable parameters benefits feature quality preservation. Consequently, we opt to completely freeze most layers during fine-tuning. As illustrated in line 10 of Algorithm 1, only the 10 layers preceding the first defective layer are trainable in each iteration. The effectiveness of this approach will be validated in Section 5.5.

5 Experiments

In this section, we first demonstrate the improvement resulting from our approach in the downstream task of unsupervised segmentation (Section 5.1). We

Table 1: Results on unsupervised segmentation using STEGO. Backbones are frozen. The unsupervised results are shown in the Cluster columns. The Linear probe results are supervised and used for reference only.

Backbone for STEGO	Cityscapes				Potsdam-3			
	Cluster		Linear		**Cluster**		Linear	
	mIoU	Acc	mIoU	Acc	mIoU	Acc	mIoU	Acc
DINOv2	19.38	72.54	43.00	91.69	67.01	80.52	76.47	86.72
DINOv2-Register	18.62	67.00	**43.97**	91.66	61.03	75.69	**80.03**	**88.97**
DINOv2-SINDER	**21.77**	**77.39**	43.06	**92.05**	**70.26**	**82.40**	77.39	87.31

use two representative unsupervised segmentation methods, STEGO [14] and CAUSE [17]. The results of unsupervised segmentation demonstrate the importance of the repaired spatially smooth feature map in dense downstream tasks. Then, we verify that our repaired DINOv2 retains feature quality. To this end, we test classification performance on ImageNet-1K [8], which ensures the quality of the `cls_token` (Section 5.2), as well as supervised segmentation on ADE20k [27] and VOC2012 [10] and depth estimation on NYUd [20], which ensures the quality of the patch tokens (Section 5.3). Finally, we study our design choices and hyperparameter settings (Section 5.5).

In these experiments, we compare 3 models: The official release of DINOv2 giant, the DINOv2 giant model trained with registers [7], and our repaired DINOv2 giant model based on the original DINOv2 without registers.

Training Setting. We randomly select 30k images from the training set of ImageNet-1K to fine-tune for one epoch. We use SGD with momentum 0.9 and weight decay 0. The batch size is 1, and the learning rate is 0.005. All input images are center-cropped and resized to 518×518. The training procedure follows Algorithm 1, which takes about six hours on a V100 GPU for fine-tuning. We limit the number of learnable layers to $\lambda = 10$ in each iteration. The loss Equation (8) is computed on the first layer that has no less than $\sigma = 3$ defects. Training stops if, during the latest $M = 500$ iterations, no loss was produced in more than $1 - \rho = 75\%$ of the iterations. The resulting network checkpoint is benchmarked on various datasets such as Cityscapes [6], Potsdam-3 [16], VOC2012 [10], ADE20k [27], *etc.* in the following sections.

5.1 Unsupervised Segmentation

As shown by the PCA visualization in Figure 1, the advantage of the repair is a spatially smooth feature map. Speculatively, our repaired DINOv2 can thus benefit dense prediction tasks such as unsupervised segmentation because the new feature map has clearer boundaries and more coherent semantics. To verify this intuition, we compare our repaired DINOv2 with the original DINOv2 and DINOv2-Register using two representative unsupervised segmentation methods,

Table 2: Results on unsupervised segmentation using CAUSE. Backbones are frozen.

Backbone for CAUSE	Cityscapes						VOC2012					
	Without CRF			With CRF			Without CRF			With CRF		
	mIoU	mAP	Acc	mIoU	mAP	Acc	mIoU	mAP	Acc	mIoU	mAP	Acc
DINOv2	31.4	45.2	85.2	31.5	57.6	89.8	55.8	71.3	91.7	57.5	79.0	93.1
DINOv2-Register	33.3	51.2	87.6	35.3	71.6	**90.7**	48.9	74.8	90.9	51.1	78.8	92.0
DINOv2-SINDER	**35.6**	**54.6**	**88.4**	**35.9**	**72.9**	**90.7**	**62.9**	**85.6**	**93.6**	**63.8**	**88.3**	**94.1**

Table 3: Results on ImageNet-1K classification. Backbones are frozen.

Backbone	KNN		Linear	
	Top1	Top5	Top1	Top5
DINOv2	83.53	94.01	86.53	97.65
DINOv2-Register	**83.69**	93.12	**87.10**	**97.95**
DINOv2-SINDER	83.51	**94.15**	86.29	97.61

namely, STEGO and CAUSE. We follow the training settings and the processing of benchmark datasets in their respective papers. Detailed hyper-parameters and configurations can be found in the Appendix. From Table 1, we observe that, compared with DINOv2, our DINOv2-SINDER improves the mIoU of STEGO on Cityscapes [6] by +2.39%, and on Potsdam-3 [16] by +3.25% in the unsupervised cluster setting. The performance of the supervised linear setting is used for reference only. From Table 2, we observe that, compared with DINOv2, our DINOv2-SINDER improves the mIoU of CAUSE on Cityscapes by +4.2% and +4.4% in the without/with CRF settings, respectively. On the VOC2012 [10] dataset, the improvements are +7.1% and +6.3%, respectively. The tables also show that our DINOv2-SINDER outperforms the DINOv2-Register. These results confirm that our proposed SINDER is effective on the dense downstream task of unsupervised segmentation.

5.2 Classification

We test the classification performance of our repaired DINOv2-SINDER on ImageNet-1K. We follow the evaluation protocol of [18]. Specifically, we test KNN and linear probe on frozen backbones. The top-1 and top-5 accuracies of the three compared models are provided in Table 3. The top-1 and top-5 accuracies of DINOv2-SINDER are on par with the original DINOv2, for both KNN and linear probe. Compared with DINOv2-Register, the top-1 accuracy of KNN is -0.18% lower, but the top-5 accuracy of KNN is +1.03% higher. The top-1 and top-5 accuracies for the linear probe are -0.81% and -0.34% lower than the DINOv2-Register, which is similar to those of the original DINOv2. Note that DINOv2-Register requires full retraining from scratch, whereas our fine-tuning

Table 4: Results on supervised segmentation. Backbones are frozen.

Backbone	ADE20k				VOC2012			
	Linear		Multiscale		Linear		Multiscale	
	mIoU	aAcc	mIoU	aAcc	mIoU	aAcc	mIoU	aAcc
DINOv2	48.83	81.46	53.24	84.00	83.05	96.17	86.01	97.01
DINOv2-Register	49.03	81.09	53.62	83.90	83.27	96.15	86.54	97.12
DINOv2-SINDER	**51.11**	**82.70**	**54.78**	**84.75**	**84.63**	**96.57**	**86.94**	**97.25**

Table 5: Results on NYUd depth estimation. Backbones are frozen.

Backbone	Linear 1	Linear 4	DPT
DINOv2	0.370	0.309	0.242
DINOv2-Register	0.367	0.302	**0.234**
DINOv2-SINDER	**0.337**	**0.294**	0.249

uses substantially fewer resources. This comparison validates that our fine-tuned model maintains the feature quality of the `cls_token`.

5.3 Supervised Segmentation

To verify that the feature quality of the patch tokens is at least equally good, we perform supervised segmentation on ADE20k and VOC2012 using the linear probe with frozen backbones. Two training settings are tested. The linear setting only uses the last feature map, while the multi-scale setting uses the feature maps of the last four layers. The results are provided in Table 4. Compared with the original DINOv2, our repaired version improves the mIoU by +2.28% and +1.54%, respectively, for the linear and multi-scale settings on ADE20k, and +1.58% and +0.93% on VOC2012. This shows the superiority of our method. Compared with DINOv2-Register, our DINOv2-SINDER improves the mIoU by +2.08% and +1.16% on ADE20k for the linear and multi-scale settings respectively, and +1.36% and +0.40% on VOC2012. This is surprising considering that DINOv2-Register has mitigated the high-norm defects at the cost of full retraining. Although DINOv2-Register is not as performant as our DINOv2-SINDER, it is still better than DINOv2. This comparison demonstrates that our method not only retains the quality of the patch tokens but also improves the dense prediction downstream task in the supervised setting.

5.4 Depth Estimation

We evaluate the patch features using depth estimation on the NYU Depth v2 dataset, following the testing protocol in [18]. There are three settings. (1) `Linear 1` uses the last layer feature map from the frozen backbone and concatenates the `cls_token` to patch tokens. The feature map is bilinear resized to

Table 6: Constrained parameter fine-tuning with gradually stronger constraints.

Setting	KNN (ImageNet)		Seg. (ADE20k)	
	Top1	Top5	mIoU	aAcc
Singular Value and Bias	6.64	16.03	13.77	61.91
Singular Value except QK	80.12	92.82	45.53	80.62
Singular Value except QK in 15 Layers	82.81	92.88	49.85	82.51
Singular Value except QK in 10 Layers	83.51	**94.15**	**51.11**	**82.70**
Singular Value except QK in 5 Layers	**83.53**	93.15	50.61	82.65

the original resolution of the input image. The depth range is uniformly divided into 256 bins and a linear layer predicts which bin the pixel should belong to. (2) `Linear 4` is similar to `Linear 1`, except that it concatenates the tokens from layers $9, 19, 29, 39$. (3) `DPT` uses the DPT decoder on features of the frozen backbone. Regression losses are used in the setting. The results are shown in Table 5. We see that SINDER outperforms DINOv2 and DINOv2-Register in the two linear settings, showing that removing defective tokens has a greater benefit for simple head structures. For complicated head DPT, the performance is on par with DINOv2 and slightly worse than DINOv2-Register.

5.5 Ablation Study

Constrained Parameter Fine-tuning. To repair the singular defects while keeping feature quality, we need to strictly constrain the freedom of parameter learning. To show the importance of constrained parameter fine-tuning, we compare five settings with gradually stronger constraints. 1. Learning the singular values of the weight matrices together with the biases of all linear layers. 2. Only learning the singular values of the weight matrices of the linear layers, except for the query matrices and K matrices in the attention. 3. Further constraining the learnable layers in the second setting to 15 layers in each iteration. 4. Constraining the learnable layers to 10 in each iteration. 5. Constraining the learnable layers to 5 in each iteration. The results are shown in Table 6. A general trend is that the fewer learnable parameters, the more the classification accuracy and segmentation performance are preserved. However, in the extreme case of too few learnable parameters, there is no room left for improvement in segmentation. According to this study, we choose the balanced setting of restricting 10 layers.

Dynamic Layer Loss. To decide which layer to apply the loss to, we experimented with different hyper-parameter values for the skip threshold σ and the logit mask threshold μ. The results are shown in Table 7. A general trend is that if it is easier to skip layers, then stronger KNN performance is preserved. However, the improvement in segmentation is then limited. This is because more skipped layers cause earlier termination according to line 2 of Algorithm 1. For the mask threshold, we find that the value $\mu = 4$ works well for DINOv2.

Table 7: Comparison of different values of skip threshold σ and mask threshold μ.

Setting	KNN (ImageNet)		Seg. (ADE20k)	
	Top1	Top5	mIoU	aAcc
Skip Less than $\sigma = 0$	83.33	94.12	**51.19**	**82.80**
Skip Less than $\sigma = 3$	83.51	**94.15**	51.11	82.70
Skip Less than $\sigma = 5$	**83.52**	93.11	50.71	82.53
Mask Threshold $\mu = 3.5$	83.33	93.10	50.93	**82.72**
Mask Threshold $\mu = 4$	**83.51**	**94.15**	**51.11**	82.70
Mask Threshold $\mu = 4.5$	83.50	93.09	50.50	82.51

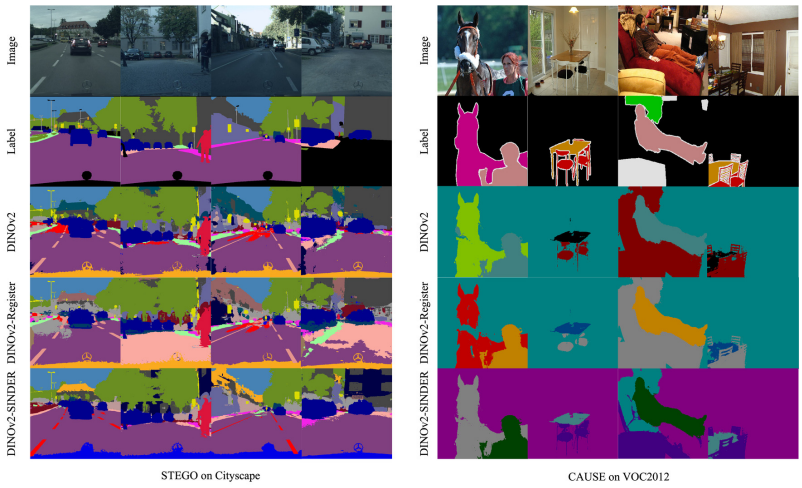


Fig. 3: Visualization of unsupervised segmentation on Cityscapes using STEGO.

6 Related Work

Self-supervised Models. The recent surge in SSL methodologies began with the application of the contrastive loss [4] or the cosine loss [13] on Siamese convolutional neural networks. Owing to their multi-modality friendly nature, Vision Transformers (ViTs) [9] are swiftly replacing CNNs as the mainstream backbone. Leveraging their global attention property, numerous SSL methods have been proposed for pre-training these networks. For instance, MoCov3 [5] follows the contrastive setting, MAE [15] reconstructs masked patch tokens, while iBoT [28] and I-JPEA [1] learn to predict feature vectors of masked or nearby regions. Despite achieving promising performance across various downstream tasks, DINOv2 [18] advances this direction further by combining the advantages of prior arts such as iBoT and DINO [3] in the loss, improving the data curation, and adopting other dedicated engineering efforts. Trained on the large-scale dataset LVD-142M [18], DINOv2 exhibits impressive performance and robust zero-shot ability, heralding a new era for training foundational vision models.

Analyzing Self-supervised Models and Transformers. Given the significance of SSL in various applications, researchers have dedicated efforts to understand its mechanisms. Some studies [22, 23] interpret contrastive learning by dissecting the loss function into several interpretable terms, while others analyze SSL from an augmentation perspective [21,25,26]. Recently, it has been observed that the object features of iBoT and I-JPEA exhibit coupling [19], thereby impeding their ability to distinguish different objects. Similarly, DINOv2 has been found to possess defective tokens [7], which undermine its performance on dense prediction downstream tasks. While proposing re-training ViTs with more tokens, these studies do not explain the underlying phenomenon.

The pairwise positive relationships between training samples using spectral methods are investigated in [2]. Notably, they utilize Singular Value Decomposition (SVD) in their analysis, although the decomposition is applied to the representation matrix composed of feature maps. Our analysis diverges from theirs as our SVD is applied to the network parameters rather than the features themselves. The work of [12] analyzes transformers based on their interpretation as interacting particle systems. Specifically, they observe the emergence of clusters over time. However, their analysis is limited to a simplified ideal transformer architecture, which disregards the MLP block and multi-head attention. Moreover, their focus is primarily on the case where $Q = K = V = I_d$, which is not realistic. The work of [11] assumes a fixed V, which is restrictive as it does not account for the varying semantics learned across different layers. By contrast, our analysis directly examines the pre-trained weights, including the learned parameters of Q, K, and V. Furthermore, we consider the multi-head structure as well as the MLP block in our analysis.

7 Limitation and Social Impact

This work focuses on repairing existing pre-trained networks. How to avoid singular defects from training is left for future work. We primarily focus on the study of DINOv2, and we hope our treatment could motivate more research on the understanding of more transformer-based networks such as GPTs. Our method of repairing existing networks requires substantially fewer computation resources and data consumption, which reduces the carbon emissions and human labor in curating data, compared to the existing approach of fully retraining.

8 Conclusion

In this paper, we have introduced a principled way to connect the high-norm defective tokens in DINOv2 with the leading left singular vector of the pre-trained weights. Based on this finding, we propose to repair DINOv2 by fine-tuning using a smooth prior loss optimized on a restricted number of parameters. Our experiments have shown that our singular defect direction prediction aligns well with the empirical defect direction, and our repaired DINOv2 improves unsupervised pixel-level prediction downstream tasks while retaining feature quality.

Acknowledgements

This work was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5-180359.

References

1. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15619–15629 (2023)
2. Balestriero, R., LeCun, Y.: Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. Advances in Neural Information Processing Systems **35**, 26671–26685 (2022)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)
5. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. in 2021 ieee. In: CVF International Conference on Computer Vision (ICCV). pp. 9620–9629
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers (2024)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
11. Geshkovski, B., Letrouit, C., Polyanskiy, Y., Rigollet, P.: The emergence of clusters in self-attention dynamics. arXiv preprint arXiv:2305.05465 (2023)
12. Geshkovski, B., Letrouit, C., Polyanskiy, Y., Rigollet, P.: A mathematical perspective on transformers. arXiv preprint arXiv:2312.10794 (2023)
13. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
14. Hamilton, M., Zhang, Z., Hariharan, B., Snavely, N., Freeman, W.T.: Unsupervised semantic segmentation by distilling feature correspondences. arXiv preprint arXiv:2203.08414 (2022)

15. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
16. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9865–9874 (2019)
17. Kim, J., Lee, B.K., Ro, Y.M.: Causal unsupervised semantic segmentation. arXiv preprint arXiv:2310.07379 (2023)
18. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2023)
19. Qiu, C., Zhang, T., Wu, Y., Ke, W., Salzmann, M., Süsstrunk, S.: Mind your augmentation: The key to decoupling dense self-supervised learning. In: The Twelfth International Conference on Learning Representations (2023)
20. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision – ECCV 2012. pp. 746–760. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
21. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? arXiv preprint arXiv:2005.10243 (2020)
22. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International Conference on Machine Learning. pp. 9929–9939. PMLR (2020)
23. Wang, Y., Zhang, Q., Wang, Y., Yang, J., Lin, Z.: Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. arXiv preprint arXiv:2203.13457 (2022)
24. Wikipedia: Power iteration — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Power%20iteration&oldid=1188380344` (2024), [Online; accessed 06-March-2024]
25. Xiao, T., Wang, X., Efros, A.A., Darrell, T.: What should not be contrastive in contrastive learning. arXiv preprint arXiv:2008.05659 (2020)
26. Zhang, T., Qiu, C., Ke, W., Süsstrunk, S., Salzmann, M.: Leverage your local and global representations: A new self-supervised learning strategy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16580–16589 (2022)
27. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
28. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. arXiv preprint arXiv:2111.07832 (2021)