Denoising Vision Transformers

Jiawei Yang*,†,1 Katie Z Luo*,2 Jiefeng Li3 Kilian Q Weinberger2 Yonglong Tian4 Yue Wang1

[1]University of Southern California [2]Cornell University [3]Shanghai Jiao Tong University [4]Google Research

*equal technical contribution †project lead

Project Page and Code: https://jiawei-yang.github.io/DenoisingViT/

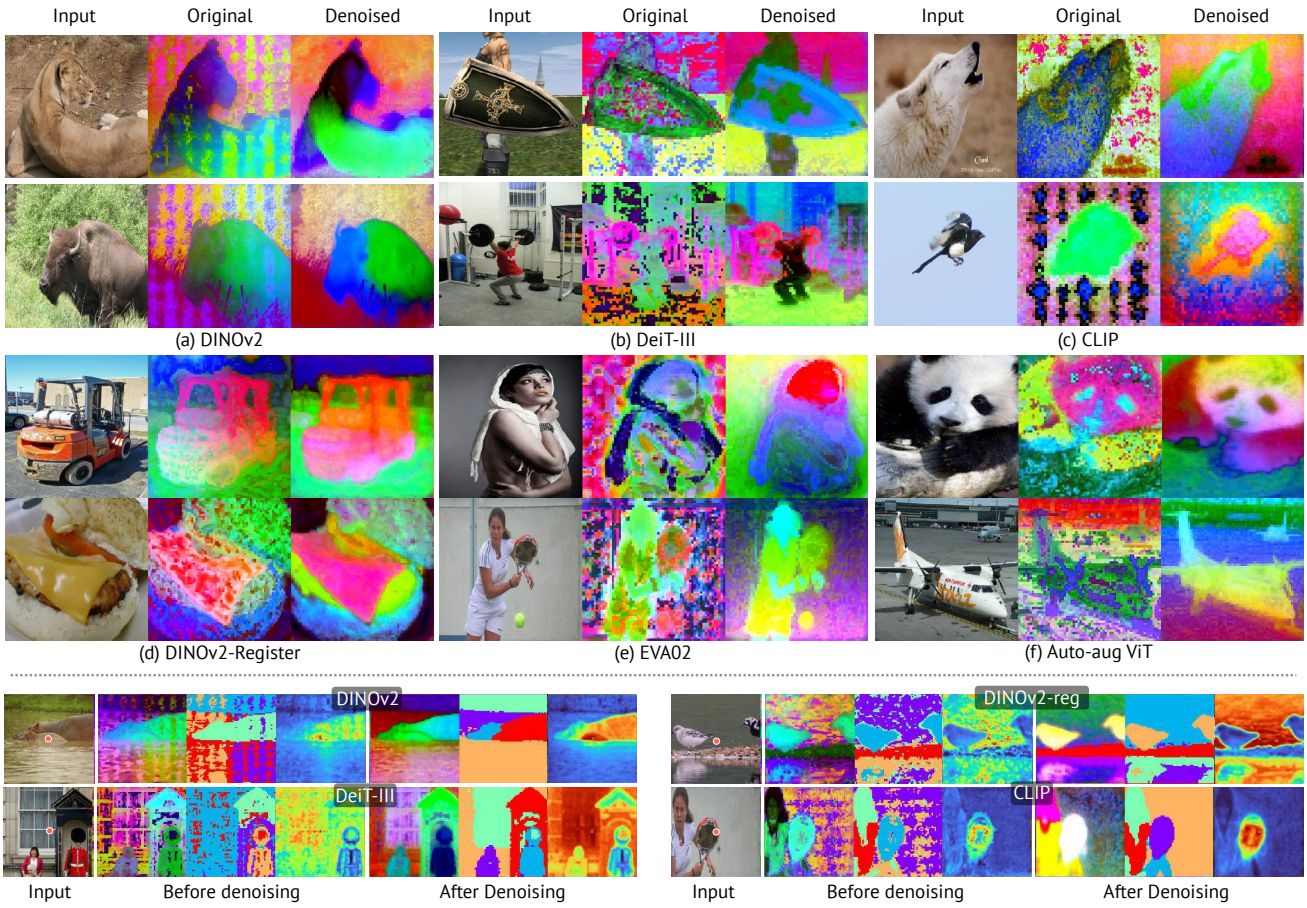

Figure 1. **Denoising Vision Transformers (DVT)** removes the noisy artifacts in visual features present in almost all Vision Transformers (ViTs). We use a representative set of ViTs as examples, including supervised (*e.g.* DeiT-III [32], Auto-aug ViT [10, 29]), reconstruction (*e.g.*, EVA-02 [13]), self-distillation (*e.g.*, DINOv2 [22], DINOv2-reg [8]), and multi-modal (*e.g.*, CLIP [26]) algorithms. **Top**: Each image triplet showcases an input image, its corresponding raw feature visualization, and the cleaned feature map denoised by DVT. **Bottom**: These triplets display, in order, a feature map, a K-Means cluster map, and a similarity map of the central patch (red dotted) with other patches in the image. Observe how the artifacts negatively impact clustering accuracy and similarity correspondences and how our DVT effectively addresses these issues. The feature colors in the visualizations are produced using principle component analysis (PCA). Best viewed in color.

Abstract

We delve into a nuanced but significant challenge inherent to Vision Transformers (ViTs): feature maps of these models exhibit grid-like artifacts ("Original" in Figure 1), which detrimentally hurt the performance of ViTs in downstream tasks. Our investigations trace this fundamental issue down to the positional embeddings at the input stage. To address this, we propose a novel noise model, which is universally ap-

plicable to all ViTs. Specifically, the noise model dissects ViT outputs into three components: a semantics term free from noise artifacts and two artifact-related terms that are conditioned on pixel locations. Such a decomposition is achieved by enforcing cross-view feature consistency with neural fields in a per-image basis. This per-image optimization process extracts artifact-free features from raw ViT outputs, providing clean features for offline applications. Expanding the scope of our solution to support online functionality, we introduce a learnable denoiser to predict artifact-free features directly from unprocessed ViT outputs, which shows remarkable generalization capabilities to novel data without the need for per-image optimization. Our two-stage approach, termed Denoising Vision Transformers (DVT), does not require re-training existing pre-trained ViTs and is immediately applicable to any Transformer-based architecture. We evaluate our method on a variety of representative ViTs (DINO, MAE, DeiT-III, EVA02, CLIP, DINOv2, DINOv2-reg). Extensive evaluations demonstrate that our DVT consistently and significantly improves existing state-of-the-art general-purpose models in semantic and geometric tasks across multiple datasets (e.g., *+3.84 mIoU). We hope our study will encourage a re-evaluation of ViT design, especially regarding the naive use of positional embeddings.*

1. Introduction

In recent years, Transformers [34] have emerged as the universal architecture for modern foundation models across many modalities, from language to audio [19, 36], text [1, 6, 24, 27], and images [2, 10]. Vision Transformers (ViTs) [10] are now the new de-facto standard in vision-related tasks. These models not only achieve state-of-the-arts under multiple benchmarks but also exhibit intriguing behaviors and capabilities across various tasks [4, 15, 22, 26].

Despite these significant strides made by ViTs, our work reveals a crucial yet often overlooked challenge: the presence of persistent noise artifacts in ViT outputs, observable across various training algorithms [4, 10, 13, 15, 22, 26, 32] (illustrated in Figure 1). These artifacts, beyond being visually annoying, hinder feature interpretability and disrupt semantic coherence. For example, the bottom row of Figure 1 demonstrates that applying clustering algorithms directly on the raw ViT outputs results in noisy clusters. This issue, prevalent across numerous existing *pre-trained* ViTs, hinders model performance in downstream tasks, underscoring the need for a complete study to mitigate these artifacts. To that end, this paper aims to answer a crucial research question: *Is it feasible to effectively denoise these artifacts in pre-trained ViTs, ideally without model re-training?*

To answer this, we first investigate the origins of these artifacts. We posit that positional embeddings, a fundamental component of ViT architecture, significantly contribute to

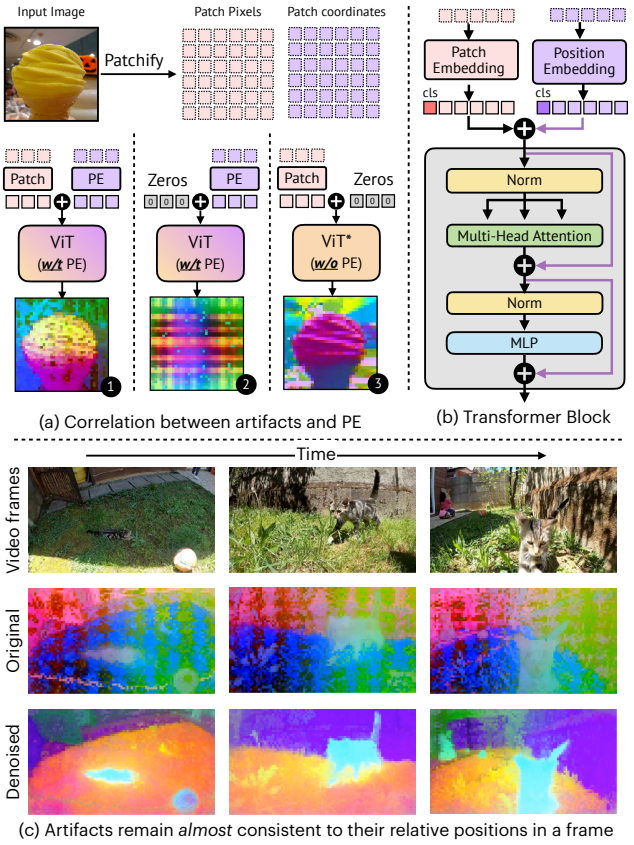

Figure 2. **Impact of positional embeddings in ViTs.** (a) Comparison between DINOv2 ViTs [22] trained with and without positional embeddings ((“ViT” v.s. “ViT*”), showcasing feature maps for: (1) a standard ViT process, (2) ViT using only positional embeddings (PE) as input, emphasizing the emergence of artifacts, and (3) a PE-free ViT* process, displaying a clear absence of these artifacts. In the figure, “Patch”: patch embedding, “PE”: position embedding. (b) Illustration of how ViT retains and propagates the positional embeddings. (c) Despite significant differences in the context of various frames, the artifacts maintain a consistent relative position in the images (central row). Our DVT effectively denoises these artifacts, as demonstrated in the final row.

this phenomenon. Our initial analysis substantially supports this hypothesis: First, when a zero-tensor (*i.e.*, no content) is fed into a pre-trained DINOv2 model [22], the resulting output is predominantly characterized by similar noise patterns (Figure 2-(a, 2)). Second, we observe a notable absence of such artifacts in the outputs of a DINOv2 model trained without positional embeddings, which contrasts sharply with the standard model outputs (Figure 2-(a, 1) v.s. (a, 3)). Finally, despite the significant differences in the context of various input frames, the artifacts maintain a consistent relative position in the images (Figure 2-(c), middle row).

With this insight, our work develops a novel two-stage denoising approach, Denoising Vision Transformers (DVT),

specifically designed for removing position-dependent artifacts from pre-trained ViTs. In the first stage, we formulate a universal noise model for ViT outputs, which factorizes the output into three components: a noise-free semantics term and two terms associated with the undesirable position-based artifacts. This decomposition is achieved by enforcing cross-view feature consistency with neural fields in a per-image basis. The per-image denoising process extracts noise-free features from raw outputs and provides these clean ViT features for offline applications. In the second stage, we train a lightweight denoiser model, consisting of a single Transformer block, to predict the denoised features from the raw ViT outputs. This denoiser seamlessly integrates into pre-trained ViTs, provides denoised features for online applications, and generalizes well to unseen data.

We conduct empirical evaluations to demonstrate the efficacy of DVT on seven representative ViTs: DINO [4], DINOv2 [22], DINOv2 with Register [8], DeiT-III [32], MAE [15], EVA-02 [12, 13], and CLIP [26]. These evaluations showcase significant enhancements in performance across various dense vision tasks. Our contributions are:

- We identify and highlight the widespread occurrence of noise artifacts in ViT features, pinpointing positional embeddings as a crucial underlying factor.
- We introduce a novel noise model tailored for ViT outputs, paired with a neural field-based denoising technique. This combination effectively isolates and removes noise artifacts from features.
- We develop a streamlined and generalizable feature denoiser for real-time and robust inference.
- Our approach significantly improves the performance of multiple pre-trained ViTs in a range of downstream tasks, confirming its utility and effectiveness (*e.g.*, as high as a 3.84 mIoU improvement after denoising).

2. Related Works

General purpose features from Vision Transformers. Transformers have been used extensively across multiple domains as general-purpose feature extractors. Originally used primarily in language modeling, the Transformer architecture has found success through language-based self-training methods such as next word prediction [1, 6, 25, 33] or masked language modeling [9, 27], to name a few. In parallel, Vision Transformers pre-trained via supervised learning [17, 32, 35] or self-supervised learning [4, 15, 22, 41] have demonstrated strong generalizability to various downstream visual tasks, even without fine-tuning. In this work, we show that ViTs trained with diverse training objectives exhibit commonly observed noise artifacts in their outputs. By addressing this issue, we significantly enhance the quality of local features, as evidenced by improvements in semantic segmentation and depth prediction tasks.

ViT artifacts. We study the fundamental issue of noise artifacts in ViTs, a phenomenon that has been previously noticed yet often unexplored. These artifacts are noticeable as noisy attention maps in supervised ViTs (*i.e.*, ViTs do not attend to objects of interest well) [4, 5]. Concurrent to ours, two recent studies similarly discover artifacts even in self-supervised ViTs [8, 39]. Specifically, [8] describe these as "high-norm" patches in low-informative background regions, suggesting their occurrence is limited to large (*e.g.* ViT-*large* or greater) and sufficiently trained ViTs. However, our analysis indicates that this may not be the full picture. We find a strong correlation between the presence of artifacts and the use of positional embeddings in ViTs. This finding suggests their presence is not strictly confined to certain model sizes or training scales but is more fundamentally linked to the inherent design of ViTs. Moreover, unlike the method proposed by [8] that re-trains ViTs with register tokens [14, 38] from scratch, our approach directly denoises pre-trained models without re-training. Additionally, we note that artifacts still exist in DINOv2 trained with registers [8] (see Figure 1 DINOv2-reg, and Figure S13), and our DVT can effectively denoise them and improve their performance.

3. Preliminaries

Forward process in ViTs. Despite varying training approaches, the ViT architecture has largely remained consistent with its original design as presented in [10] and [35]. The forward process of a ViT, depicted in Figure 2-(b), starts by converting images into 2D patches and then embedding them, followed by a forward process of Transformer blocks. Specifically, an image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ is first divided into patches $\mathbf{x}_p \in \mathbb{R}^{N\times(P^2\cdot C)}$, where (H, W) denotes the image's resolution, P is the patch resolution, C represents the number of pixel channels, and N is the total number of patches. These patches are then mapped to D dimensions using a trainable linear projection $\mathbf{E} \in \mathbb{R}^{(P^2\cdot C)\times D}$ to generate patch embeddings. To inject spatial information, positional embeddings, which encode patch coordinates and are denoted as $\mathbf{E}^i_{pos}$, are added to the patch embeddings. Formally, the forward process of a ViT is as follows:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{cls}} + \mathbf{E}^{\text{cls}}_{pos}; \mathbf{x}^0_p\mathbf{E} + \mathbf{E}^0_{pos}; \ \cdots\ ; \mathbf{x}^{N-1}_p\mathbf{E} + \mathbf{E}^{N-1}_{pos}] \tag{1}$$

$$\mathbf{z'}_l = \text{MSA}\left(\text{LN}(\mathbf{z}_{l-1})\right) + \mathbf{z}_{l-1}, \quad l = 1\cdots L \tag{2}$$

$$\mathbf{z}_l = \text{MLP}\left(\text{LN}(\mathbf{z'}_l)\right) + \mathbf{z'}_l, \quad l = 1\cdots L \tag{3}$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L) \tag{4}$$

Here, $\mathbf{x}_{\text{cls}}$ and $\mathbf{E}^{\text{cls}}_{pos}$ represent the class token and its positional embedding, respectively, L denotes the number of layers, and LN stands for layer normalization. Multi-head self-attention layers and multi-layer perceptron layers are termed MSA and MLP, respectively. Note that the input-

independent positional embeddings operate as a spatial inductive basis and intermix with inputs, propagating through the entire ViT.

4. Denoising Vision Transformers

In this section, we start by analyzing ViT outputs to motivate our approach (§4.1). Then, we introduce our per-image denoising method, which removes artifacts and produces noise-free features (§4.2). Finally, we explain how the noise-free features are utilized as pseudo-labels to train a generalizable denoiser (§4.3). Our method pipeline is depicted in Figure 3.

4.1. Factorizing ViT Outputs

Ideal visual features should be inherently translation and reflection invariant, *i.e.*, the features of an object should remain consistent, regardless of changes in the viewing window, size, and orientation. However, as indicated in Equations (1) to (4) and Figure 2-(b), ViTs intertwine patch embeddings with positional embeddings, breaking the transformation invariance of visual features. This breach of invariance might not seem immediately problematic, but our detailed investigations, as illustrated in Figure 2-(a) and (c), establish a distinct correlation between the inclusion of positional embeddings and the emergence of undesirable artifacts in ViT outputs. Particularly, the middle row of Figure 2-(c) shows that these artifacts remain nearly consistent regardless of input content, only exhibiting small residual variation across different images.

These observations motivate us to decompose ViT outputs into three terms: (1) an input-dependent, noise-free semantics term $f(\mathbf{x})$[1]; (2) an input-independent artifact term related to spatial positions $g(\mathbf{E}_{pos})$; (3) and a residual term accounting for the co-dependency of semantics and positions $h(\mathbf{x}, \mathbf{E}_{pos})$. Accordingly, we have:

$$\mathrm{ViT}(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{E}_{pos}) + h(\mathbf{x}, \mathbf{E}_{pos}), \tag{5}$$

This factorization is universally applicable to all ViTs. For instance, in scenarios where the output feature map is spatially invariant (*e.g.*, no positional embedding is used), both g and h become zero functions [7]. Conversely, when every feature is dependent on both position and semantics, f and g turn into zero functions.

4.2. Per-image Denoising with Neural Fields

Directly addressing the above decomposition problem from a single forward pass in a ViT is impractical due to the intertwined nature of output features. To overcome this, we harness cross-view feature and artifact consistencies: (1) Feature consistency refers to the transformation invariance of visual features, wherein despite varied spatial transformations, the essential semantic content remains invariant; (2) Artifact consistency means that the input-independent artifact remains observable and constant across all transformations. Formally, consider an image $\mathbf{x}$ and a set of its randomly transformed views $T(\mathbf{x}) = \{t_0(\mathbf{x}), t_1(\mathbf{x}), \cdots\}$, where each transformation t_i is drawn from a distribution of random augmentations $\mathcal{T}$, consisting of random resizing, cropping, and flipping. Our goal is to derive a mapping f that ensures the semantic features obtained from any transformed view, $f\left(t\left(\mathbf{x}\right)\right)$, remains equivalent to the transformed original semantic features, $t\left(f(\mathbf{x})\right)$. That is $f\left(t\left(\mathbf{x}\right)\right) = t\left(f(\mathbf{x})\right), t \sim \mathcal{T}$. Next, we describe our approach for jointly learning the different terms in Equation (5) to derive f.

Neural fields as feature mappings. At the core of our approach is to have a holistic image semantics representation, $\mathcal{F}$, for *each individual image*, paired with a spatial artifact feature representation, $\mathcal{G}$, shared by *all transformed views*. The holistic image feature representation $\mathcal{F}$ is designed to capture spatially independent, artifact-free semantics, while $\mathcal{G}$ should encode position-dependent but input-independent noise. We use neural fields [16, 18, 20, 28, 31, 39] to approximate f and g. Specifically, we define $f(t(\mathbf{x})) = \mathcal{F}(\mathrm{coords}(t(\mathbf{x})))$, where $\mathrm{coords}(\cdot)$ extracts the *pixel* coordinates of the transformed views in the original image $\mathbf{x}$, and $g(\mathbf{E}^i_{pos}) = \mathcal{G}(i)$, with $i \in \{0, \cdots, N-1\}$ denoting the patch index. For simplicity, we use $\mathcal{G}$ to denote the 2D artifact feature map reshaped from the 1D ordered sequence $\{\mathcal{G}(i)\}_{i=0}^{N-1}$. We refer to $\mathcal{F}$ and $\mathcal{G}$ as the semantics field and the artifact field, respectively.

Learning the decomposition. Our goal is to learn the semantics field $\mathcal{F}$, the artifact field $\mathcal{G}$, and the residual term Δ by minimizing a regularized reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{distance}} + \alpha\mathcal{L}_{\text{residual}} + \beta\mathcal{L}_{\text{sparsity}} \tag{6}$$

$$\mathcal{L}_{\text{distance}} = 1 - \cos(\mathbf{y}, \widehat{\mathbf{y}}) + \|\mathbf{y} - \widehat{\mathbf{y}}\|_2, \tag{7}$$

$$\mathcal{L}_{\text{residual}} = \|\mathrm{sg}\left(\mathbf{y} - \widehat{\mathbf{y}'}\right) - \widehat{\Delta}\|_2, \quad \mathcal{L}_{\text{sparsity}} = \|\widehat{\Delta}\|_1 \tag{8}$$

$$\text{where } \mathbf{y} = \mathrm{sg}\left(\mathrm{ViT}\left(t\left(\mathbf{x}\right)\right)\right), \quad \widehat{\mathbf{y}} = \widehat{\mathbf{y}'} + \mathrm{sg}(\widehat{\Delta}) \tag{9}$$

$$\widehat{\mathbf{y}'} = \mathcal{F}_\theta(\mathrm{coords}(t(\mathbf{x}))) + \mathcal{G}_\xi, \quad \widehat{\Delta} = h_\psi(\mathbf{y}) \tag{10}$$

Here, $\cos(\cdot,\cdot)$ denotes the cosine similarity, $\mathrm{sg}(\cdot)$ represents the stop-gradient operation, $t(\cdot)$ is a random transformation sampled from $\mathcal{T}$, and θ, ξ and ψ are the learnable parameters. Our loss function ensures $\widehat{\Delta}$ remains minimal by imposing a sparsity regularization, thereby allowing $\widehat{\mathbf{y}'}$ to represent as much of ViT outputs as possible. The use of stop-gradient operators is crucial to avoid trivial solutions, such as identity mapping. The reconstructed feature from our method is $\widehat{\mathbf{y}} = \mathcal{F}_\theta\left(\mathrm{coords}\left(t\left(\mathbf{x}\right)\right)\right) + \mathcal{G}_\xi + \mathrm{sg}\left(h_\psi\left(\mathrm{ViT}\left(t\left(\mathbf{x}\right)\right)\right)\right)$, each term corresponding to f, g, and h as delineated in Equation (5).

Optimization. We break our optimization process into two

[1]Throughout this paper, we use "noise" and "artifacts" interchangeably.

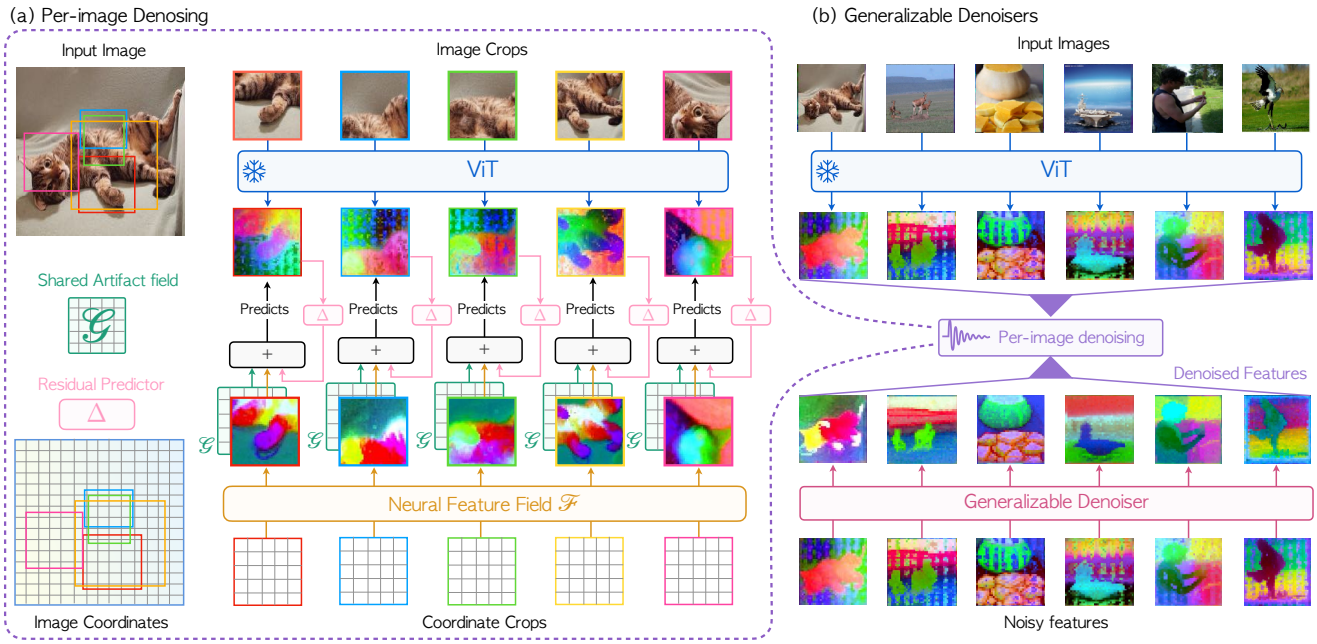

Figure 3. **Denoising Vision Transformers (DVT).** DVT consists of a two-stage denoising pipeline. In the first stage, our method decomposes the noisy features of a crop into a noise-free semantics term $\mathcal{F}$, an input-independent, position-related artifact term $\mathcal{G}$, and an additional residual term Δ (left). In the second stage, we train a generalizable denoiser with these individually optimized, clean features (right).

phases, each spanning half of the total training iterations. In the first phase, we train $\mathcal{F}_\theta$ and $\mathcal{G}_\xi$ using only $\mathcal{L}_{\text{distance}}$, allowing them to capture a significant portion of the ViT outputs. After completing half of the optimization iterations, we freeze $\mathcal{G}_\xi$ and continue to train $\mathcal{F}_\theta$ alongside h_ψ using $\mathcal{L}_{\text{recon}}$ for the rest iterations. The coefficients α and β in $\mathcal{L}_{\text{recon}}$ balance loss scales and regulate the residual term to prevent $\widehat{\Delta}$ from over-explaining the outputs.

4.3. Generalizable Denoiser

Our per-image denoising method can already effectively remove artifacts from ViT outputs, yielding visually stunning denoised feature maps, as showcased in Figure 1. The problems we are left with are run-time efficiency and distribution shifts. Specifically, the per-image approach is suboptimal for real-time applications, and individually denoised feature maps can lead to distribution shifts due to sample bias, which hampers the feature coherence across different images. To address these issues, we introduce a generalizable denoiser.

After per-image denoising, we accumulate a dataset of pairs of noisy ViT outputs $\mathbf{y}$ and their denoised counterparts $\mathcal{F}$, denoted as $\mathcal{B} = \{(\mathbf{y}_i, \mathcal{F}_i)\}|_{i=1}^{B}$. To achieve a generalizable denoising model, we distill these individually denoised samples into a denoiser network D_ζ, which is trained to predict noise-free features from raw ViT outputs. The training objective is formulated as:

$$\mathcal{L}_{\text{distance}}^{\text{DVT}} = 1 - \cos\left(D_\zeta\left(\mathbf{y}\right), \mathcal{F}\right) + \left\|D_\zeta\left(\mathbf{y}\right) - \mathcal{F}\right\|_2 \quad (11)$$

Specifically, our generalizable denoiser consists of a single Transformer block, supplemented with additional learnable positional embeddings that are applied post the forward pass of a ViT. This design aims to mitigate the input-independent artifacts. To predict denoised features, the outputs from a pre-trained ViT are added with these positional embeddings and then processed through the Transformer block. This can be efficiently implemented in a single line of code:

```
denoised_feats = self.denoiser(y + self.PE)
```

Here, `self.denoiser` refers to the single Transformer block, and `self.PE` represents the additional learnable positional embeddings, and `y` is the ViT output. Notably, this learned denoiser is lightweight, thus adding minimal latency to the original ViT. It also learns to generalize across samples, enabling real-time applications and mitigating the distribution shift issue inherent to per-image denoising.

5. Experiments

In this section, we first test our per-image denoising algorithm on ViTs trained with different objectives. Then, we evaluate the effectiveness of our generalizable denoiser on dense prediction tasks. For all experiments, we default to using ViT-base models with patch sizes of 14 or 16, depending on the availability of their implementations and model weights in PyTorch Image Models (`timm` [37]). We defer the implementation details to the supplementary material.

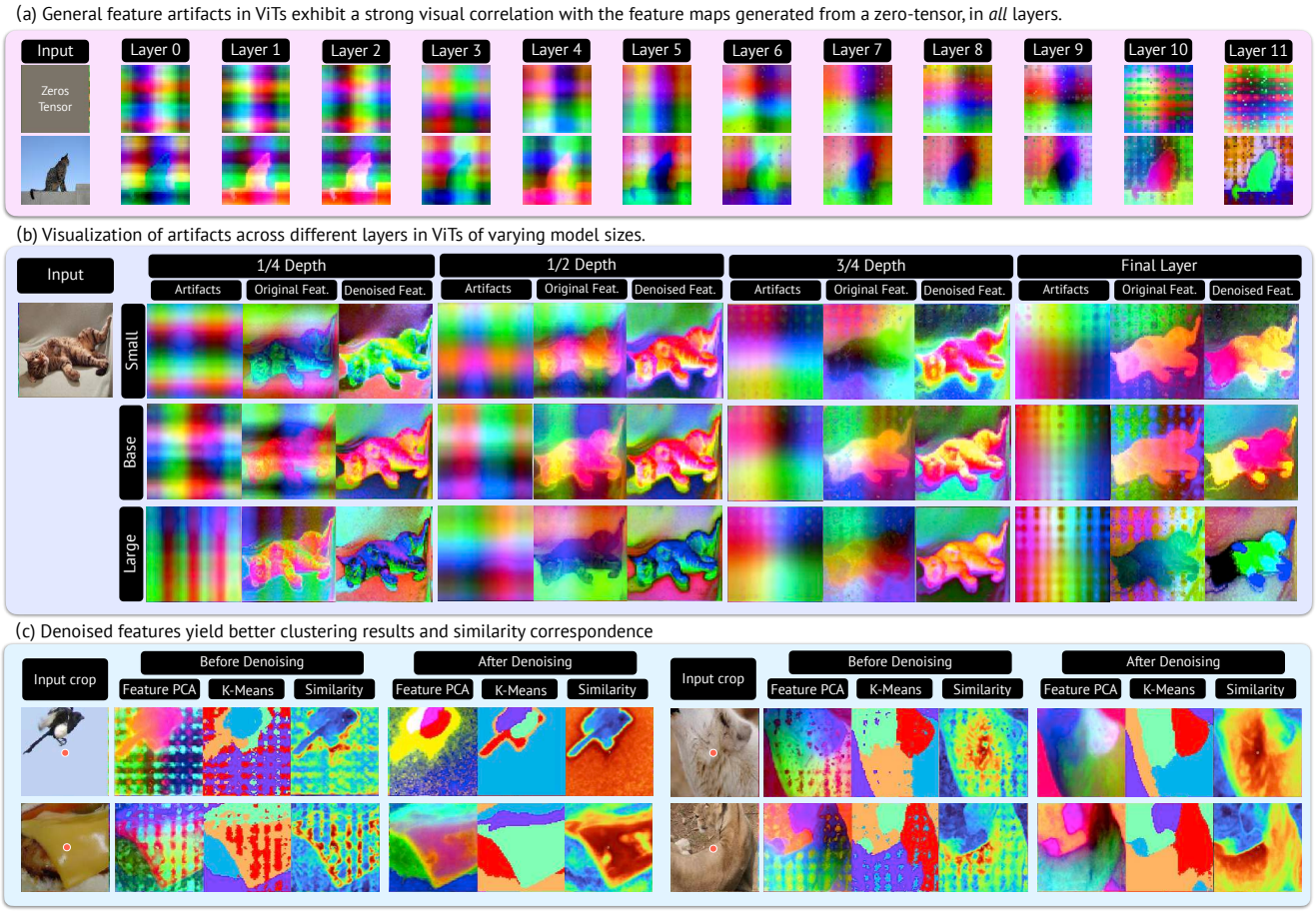

Figure 4. **Visual analysis of ViT output features and denoised features.** (a) Visualizations of the feature maps from all layers of a DINOv2 [22] ViT-base model, using an empty image and a cat image as input. The artifacts in the cat's feature maps have a strong visual correlation to empty input's feature maps. (b) Visualizations of the decomposed artifacts, the original features, and the denoised features across various layers of DINOv2 ViTs. We observe similar patterns in differently-sized ViTs. (c) Visualizations of the K-Means clustering results and the cosine similarity of the central patch (red dot) to other patches. Notice that feature maps have fewer artifacts and enhanced semantic clarity after denoising, resulting in improved clustering results and similarity correspondence.

5.1. Artifacts in ViTs

First, we explore if ViTs trained with different objectives exhibit similar artifacts. To this end, we test with a few representative ViTs, categorizing them into two groups based on the severity of observed artifacts: one with strong artifacts and the other with mild artifacts.

Algorithms producing strong artifacts. We highlight several ViT training algorithms that result in pronounced feature artifacts, as observed in Figure 1 (except for (d)). Among these, DINOv2 [22], a state-of-the-art vision foundation model with excellent performance on downstream tasks, displays clear position-related artifacts. Additionally, DeIT-III [32], trained with class labels, and CLIP [26], trained by text-image alignment, also exhibit noticeable artifacts. Furthermore, EVA02 [13], which distills local patch features from a pre-trained CLIP model using masked image modeling, also has clear feature artifacts. Our proposed method successfully mitigates these artifacts in all the tested ViTs (compare "Original" and "Denoised" in Figure 1).

Algorithms producing mild artifacts. Conversely, certain models demonstrate only weak artifacts. Specifically, DINO [4] and MAE [15] tend to exhibit low-frequency patterns that are less visually noticeable in individual images[2]. Intriguingly, while DINOv2 [22] trained with register tokens (DINOv2-reg [8]) initially appears to be free from artifacts in [8], our DVT uncovers their existence (Figure 1-(d), and its bottom row). Although DINOv2-reg shows fewer artifacts compared to the standard DINOv2, it still displays more artifacts than DINO and MAE. We recognize Regis-

[2]These patterns are more prominent in videos.

Table 1. **Comparison of features correlation to spatial positions.** We report the maximal information coefficient (MIC) between grid features and their normalized patch coordinates.

	Before denoising	After denoising	
	Original	Artifacts	Semantics
DINOv2 [22]	0.44	0.54	0.22
DeiT-III [32]	0.34	0.32	0.06
CLIP [26]	0.11	0.14	0.08

ter as an improved ViT training technique, but it does not fundamentally eliminate the artifacts.

Correlation between artifacts and positions. Beyond qualitative analyses, we quantitatively investigate the correlation between artifacts and patch positions. Specifically, we compute the maximal information coefficient (MIC) between grid features and their normalized patch coordinates (elaborated in the Appendix). This metric indicates the correlation extent between features and spatial positions. Table 1 presents the results. We observe that both the original ViT outputs and the decomposed artifacts exhibit a stronger spatial correlation than the denoised semantic features, regardless of the training approach. This confirms the link between positional embeddings and the emergence of undesirable artifacts.

5.2. Evaluation on Downstream Task Performance

Setup. We follow [8, 22] to assess our denoiser across several benchmarks: semantic segmentation tasks on VOC2012 [11] and ADE20k [40], and the depth prediction task on the NYU-depth benchmark [21], using a linear probing protocol. It is important to note that there is no direct competitor for these tasks in our study. Instead, our focus is on comparing the performance of pre-trained ViTs before and after applying our DVT. For all the models in the main experiments, we use 10k denoised samples randomly selected from the VOC2012 and the VOC2007 datasets, excluding their validation samples, to train the second-stage denoiser.

Results. Table 2 presents the main results. We observe significant and consistent enhancements in nearly all pre-trained ViTs across various dense prediction tasks post-denoising. These improvements are achieved without expensive re-training of ViTs at scale, unlike Register [8]; our DVT uses just a single Transformer block for denoising. Notably, the DINOv2-*giant* model, with an 83.0 mIoU on VOC2012 as reported in [22], is significantly outperformed by our DVT-denoised DINOv2-*base* model (84.84 mIoU). This improvement extends to the ADE20k dataset, where the DINOv2-*giant* and DINOv2-*large* models yield mIoUs of 49.0 and 47.7, respectively as in [22], while our denoised base model achieves a 48.66 mIoU. These results suggest that the performance enhancement is primarily due to effective artifact removal, rather than the *tiny* increase in the number of parameters of our denoiser network.

Enhancement of DINOv2 with register tokens. Our DVT also boosts the performance of the recently introduced DINOv2-reg model [8], where a ViT is trained with dummy learnable register tokens. As shown in Table 2, our DVT significantly enhances the performance of both DINOv2 [22] and DINOv2-reg [8]. When applying DVT only, DINOv2 witnesses more improvements compared to using registers; for instance, DINOv2 denoised by DVT achieves 84.84 mIoU in VOC2012 and 48.66 mIoU in ADE20k, surpassing the performance of DINOv2-reg, which attains 83.64 mIoU and 48.22 mIoU on the respective benchmarks. Additionally, DVT can further enhance the performance of DINOv2-reg [8] by a substantial margin on both datasets (+0.86 in VOC2012 and +1.12 in ADE20k). These findings suggest that DVT is more adept at addressing the artifact issue inherent in ViTs. In addition, DINOv2-reg [8] requires training ViTs from scratch using 142M images, while our approach only requires training a single Transformer block using 10k denoised samples.

5.3. Qualitative results

Visual analysis of ViTs. In Figure 4, we present a visual analysis of the artifact decomposition across various layers of DINOv2 ViTs of different sizes (b), alongside feature maps generated using only zero-tensors as input (a). Notably, the artifacts decomposed by our DVT show a strong visual resemblance to these zero-tensor-input feature maps. In addition, we observe that the artifacts vary across layers: the shallower layers predominantly exhibit low-frequency patterns, whereas the deeper layers are characterized by high-frequency patterns. Importantly, these patterns are consistent across ViTs of different sizes (*e.g.*, from ViT-*small* to ViT-*large*), contradicting the suggestion in [8] that only large and sufficiently trained ViTs would display such patterns. Further, Figure 4-(c) showcases the enhanced similarity of central patches compared to other patches post-denoising. Lastly, we see that the artifacts in feature maps will hurt the K-means clustering accuracy significantly and our DVT addresses this issue. These factors are particularly important for dense prediction tasks.

Emerged object discovery ability. An intriguing finding from our experiments is the emerging capability of object discovery in denoised ViTs. Figure 5 illustrates this through PCA visualizations and $L2$ norms of the feature maps. Post-denoising, not only are the artifacts removed, but also the objects of interest become more distinctly visible. This enhancement in object clarity is not an original goal of DVT but emerges as the outcome of our method. It is noteworthy

Table 2. **Qualitative performance of DVT.** DVT improves differently pre-trained ViTs for dense prediction tasks. We report performance on semantic segmentation (VOC2012, ADE20K) and depth prediction (NYUd) tasks. The best results are **bolded**.

		VOC2012 [11]			**ADE20k** [40]			**NYUd** [21]	
		mIoU (↑)	aAcc (↑)	mAcc (↑)	mIoU (↑)	aAcc (↑)	mAcc (↑)	RMSE (↓)	Rel (↓)
WEAK ARTIFACTS	MAE [15]	50.24	88.02	63.15	23.60	68.54	**31.49**	**0.6695**	**0.2334**
	MAE [15] + DVT	**50.53**	**88.06**	**63.29**	**23.62**	**68.58**	31.25	0.7080	0.2560
	DINO [4]	63.00	91.38	76.35	31.03	73.56	40.33	0.5832	**0.1701**
	DINO [4] + DVT	**66.22**	**92.41**	**78.14**	**32.40**	**74.53**	**42.01**	**0.5780**	0.1731
	DINOv2-reg [8]	83.64	96.31	90.67	48.22	81.11	60.52	0.3959	0.1190
	DINOv2-reg [8] + DVT	**84.50**	**96.56**	**91.45**	**49.34**	**81.94**	**61.70**	**0.3880**	**0.1157**
STRONG ARTIFACTS	DeiT-III [32]	70.62	92.69	81.23	32.73	72.61	42.81	**0.5880**	**0.1788**
	DeiT-III [32] + DVT	**73.36**	**93.34**	**83.74**	**36.57**	**74.44**	**49.01**	0.5891	0.1802
	EVA02 [13]	71.52	92.76	82.95	37.45	72.78	49.74	0.6446	0.1989
	EVA02 [13] + DVT	**73.15**	**93.43**	**83.55**	**37.87**	**75.02**	**49.81**	**0.6243**	**0.1964**
	CLIP [26]	77.78	94.74	86.57	40.51	76.44	52.47	0.5598	0.1679
	CLIP [26] + DVT	**79.01**	**95.13**	**87.48**	**41.10**	**77.41**	**53.07**	**0.5591**	**0.1667**
	DINOv2 [22] (reprod.)	83.60	96.30	90.82	47.29	80.84	59.18	0.4034	0.1238
	DINOv2 [22] + DVT	**84.84**	**96.67**	**91.70**	**48.66**	**81.89**	**60.24**	**0.3943**	**0.1200**

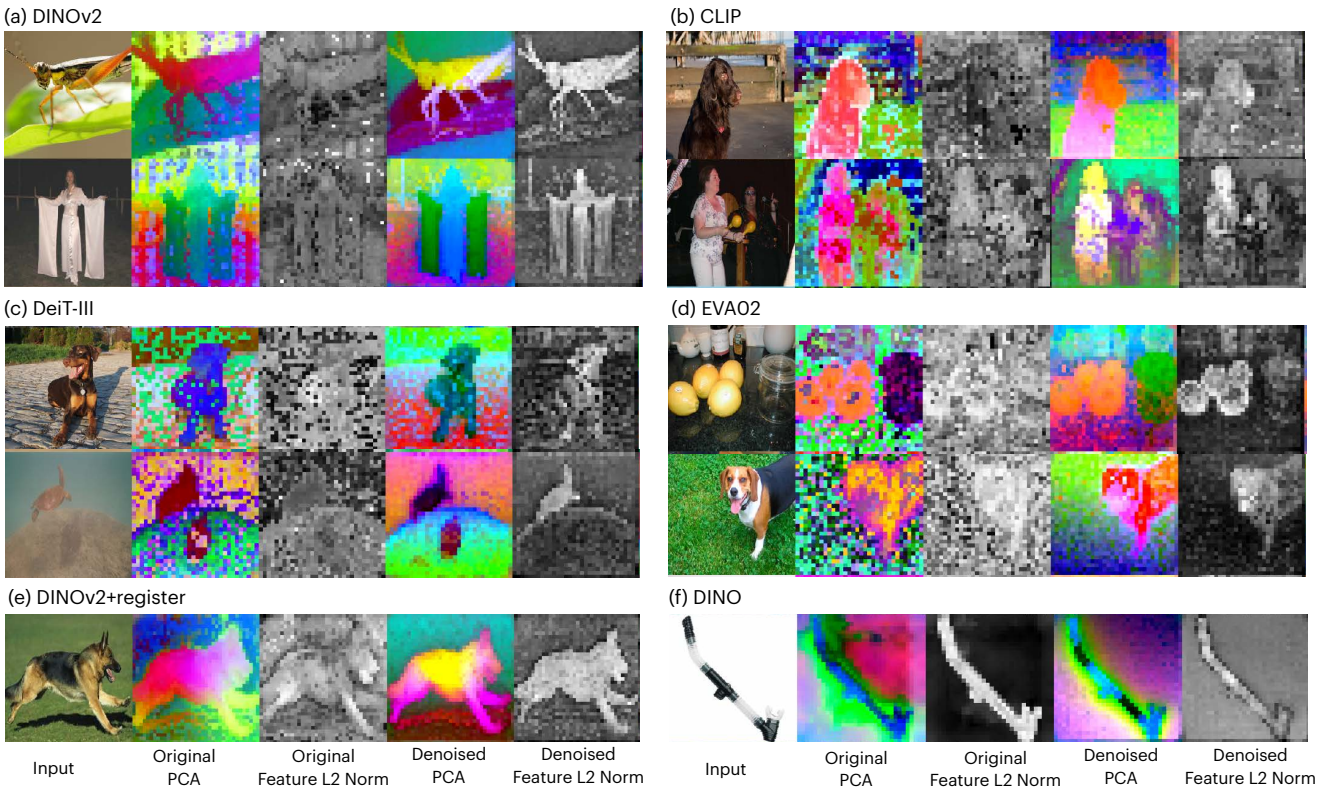

Figure 5. **Emerged object discovery ability.** We present qualitative results for DVT's learned denoiser outputs. Features are visualized using PCA and $L2$ feature norms, comparing original ViT features with our denoised features across different algorithms. Noticeably, DVT denoised features show higher feature norms on objects of interest and reduced high- (see a, b) and low-norm artifacts (see c, d).

Table 3. Ablation study on per-image denoising using KNN segmentation evaluation protocol on the VOC2012 validation set.

Representations	mIoU
(a) DINOv2	65.35
(b) $\mathcal{F}$	67.81
(c) $\mathcal{F} + \mathcal{G}$	70.82
(d) $\mathcal{F} + \mathcal{G} + \hat{\Delta}$	**70.94**

Table 4. Ablation study on the architectural design of generalizable denoiser. We report the mIoU of the VOC2012 validation set.

Denoiser architectures	mIoU
(a) DINOv2 (reproduced)	83.60
(b) conv1x1	82.15
(c) conv3x3	83.27
(d) Single Transformer Block + PE.	**84.84**
(e) Single Transformer Block	84.81

that not all pre-trained ViTs initially demonstrate this object discovery ability, as seen in Figure 5-(b,c,d) "Original PCA"; however, this capability is remarkably evident after the denoising process. It intriguingly implies an intrinsic property of denoised ViTs — finding salient objects.

5.4. Ablation Study

In this section, we provide ablation studies to understand the importance of different components in our proposed DVT. We use DINOv2-base [22] for the experiments here.

Factorization. We ablate our per-image denoising method using a K-Nearest-Neighbor (KNN) pixel segmentation evaluation protocol on the VOC2012 dataset. Specifically, we collect class centroids from each training image by masked pooling to construct a memory bank using ground truth annotations. Then, for each pixel in a validation image, we classify it based on its 20 nearest neighbors in the memory bank. We report the mIoU on the validation set. Table 3 shows the results. We observe that combining the artifact field $\mathcal{G}$ and the residual term $\hat{\Delta}$ yields the best result (d). Omitting both these elements reduces our approach to merely utilizing a neural field $\mathcal{F}$ to learn multi-crop ensembled image features, without addressing artifacts (b). While this variant shows improvement, it falls behind our proposed method by a large margin, underscoring the importance of removing artifacts.

Generalizable denoiser. We explore alternative architectural designs for our generalizable denoiser in Table 4. We study four variations: 1) our default setting, which incorporates a single Transformer Block with new learnable position embeddings; 2) our default setting but without position embeddings; 3) a multi-layer convolution denoiser with a `Conv1x1-ReLu-Conv1x1-ReLu-Conv1x1` structure, and 4) a multi-layer convolution denoiser with a `Conv3x3-ReLu-Conv3x3-ReLu-Conv3x3` structure. We observe that the denoisers based on convolutional structures (b, c) do not yield good results, with the conv1x1 setting performing the worst (c). Moreover, we note that our default setting with a Transformer block and learnable positional embedding achieves the best result (d), and removing learnable position embeddings obtains similar numerical performance (e), but we find that our default setting (Transformer Bloack + PE.) is more sensitive to local details such as text and watermark, as shown in Figure S7. Additionally, qualitative comparisons in Figure S7 highlight that convolution-based denoisers typically struggle with removing artifacts.

6. Discussion and Future Works

Our work has introduced DVT, a robust method leveraging neural fields to eliminate feature artifacts from ViTs. We pinpoint positional embeddings as the primary source of these artifacts, despite their importance in various vision tasks. Utilizing a neural-field optimization process, DVT efficiently extracts clean features from the noise-riddled feature maps of existing ViTs. Building upon this, we propose a scalable feature denoiser, eliminating the need for individual image optimizations. When learned from a few denoised samples, our denoiser generalizes well to unseen data, and improves pre-trained ViTs by large margins in dense vision tasks. Furthermore, our research suggests several avenues for future exploration: Understanding the role of positional embeddings in ViTs could inform the design of next-generation deep learning architectures. Redefining positional embeddings within ViTs and Transformers is also an imperative problem. Finally, devising a method to denoise pre-trained ViT features without additional training presents a fascinating challenge.

Acknowledgements We are grateful to many friends, including Congyue Deng, Jiageng Mao, Junjie Ye Justin Lovelace, Varsha Kishore, and Christian Belardi, for their fruitful discussions on this work and follow-ups. We acknowledge an unrestricted gift from Google in support of this project.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya

Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 2, 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. *End-to-End Object Detection with Transformers*, page 213–229. Springer International Publishing, 2020. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021. 15

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2, 3, 6, 8, 14, 15, 19

[5] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021. 3

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. 2, 3

[7] Christopher Clapham, James Nicholson, and James R Nicholson. *The concise Oxford dictionary of mathematics*. Oxford University Press, USA, 2014. 4

[8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023. 1, 3, 6, 7, 8, 14, 15, 19

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 3

[10] 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, 2021. 1, 2, 3

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 7, 8

[12] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale, 2022. 3

[13] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 1, 2, 3, 6, 8, 14, 15, 18

[14] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens, 2023. 3

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 2, 3, 6, 8, 14, 15, 20

[16] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields, 2023. 4

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 3

[18] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation, 2022. 4

[19] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou. Close to human quality tts with transformer. *arXiv preprint arXiv:1809.08895*, 2018. 2

[20] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 4, 12

[21] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7, 8

[22] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 6, 7, 8, 9, 14, 15, 17

[23] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022. 16

[24] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2

[25] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 3

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual

models from natural language supervision, 2021. 1, 2, 3, 6, 7, 8, 14, 15, 17

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. 2, 3, 16

[28] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023. 4

[29] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 1

[30] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. 15

[31] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 4

[32] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit, 2022. 1, 2, 3, 6, 7, 8, 14, 15, 18

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 3

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 3

[36] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *Interspeech 2017*. ISCA, 2017. 2

[37] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 5

[38] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2023. 3

[39] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision, 2023. 3, 4

[40] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018. 7, 8

[41] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 3

Denoising Vision Transformers

Supplementary Material

In the appendix, we include comprehensive implementation details (§A) as well as discussions on the understanding of ViTs (§B), focusing specifically on the nuances of position embeddings. Following this, we discuss the limitations of this work and propose avenues for future exploration (§C).

A. Implementation Details

A.1. Denosing with Neural Fields

Recall that we decompose the output feature map from a pre-trained ViT into three components: $\mathbf{y} \approx \mathcal{F}(\mathcal{A}) + \mathcal{G} + \mathbf{h}(\mathbf{y})$, where $\mathcal{F}$ is a s semantic field, $\mathcal{G}$ is an artifact field, and $\mathbf{h}$ is a residual predictor. We describe their implementation details below.

Neural field $\mathcal{F}$. To facilitate efficient learning, we use InstantNGP [20], a type of compact and fast coordinate network, parameterized by learnable multi-level hash grids $\mathcal{H}$ and a lightweight MLP $\phi(\cdot)$, to learn $\mathcal{F}$. It takes as input a normalized 2D coordinate (i, j), within the range of [0, 1], and outputs its corresponding feature vector, *i.e.*, $\mathcal{F}(i, j) = \phi(\mathcal{H}(i, j))$. We refer readers to [20] for a more detailed understanding of the learnable hash grids. In our implementation, we use a hash encoding resolution that spans from 2^4 to 2^{10} with 16 levels. Each hash entry has a channel size of 8. The maximum number of hash entries of each resolution is 2^{20}. For the lightweight MLP, we use a two-layer `Linear-ReLu-Linear` structure. The hidden dimension of this MLP is half the size of the output feature dimension, and the output feature dimension itself corresponds to the feature dimension of the ViT being studied (*e.g.* 768 for a ViT-B and 1024 for a ViT-L).

Artifact field $\mathcal{G}$. For all experiments, we use a 2D learnable feature map of size $C \times K \times K$ to learn the input-independent noise, where C corresponds to the feature dimension of the studied ViT, and K is the spatial size. We compute K by $(H-P)/S+1$, where H is the height&width of input images (which we resize to be square), P is the patch size, and S is the stride size used in the model. To accommodate ViTs with different patch sizes, we set H to 518 for those trained with a patch size of 14, and 512 for ViTs with a patch size of 16, resulting in K values of 37 and 32, respectively. Note that this feature map, $\mathcal{G}$, can be interpolated to fit any arbitrary image size. We specifically choose these K values to minimize the need for interpolation during training, thus enhancing training efficiency.

Residual predictor $\mathbf{h}$. The residual predictor is structured as a 3-layer MLP with `ReLU` activation after the hidden layers. The hidden dimension is set to be one-quarter of the channel dimension of the ViT being studied.

Optimization. In our implementation, we extract $N = 768$ crops from each image, applying random augmentations, which include random flipping with a probability of 0.5, and random resizing and cropping, where the size of the crop is scaled between 0.1 to 0.5 of the original image size and the aspect ratio is maintained between 3/4 and 4/3.

The coefficients in our loss function (Equation (6)) are set as $\alpha = 0.1$ and $\beta = 0.02$. We use Adam optimizer, with a learning rate of 0.01 and a `LinearLR` decay strategy. Our models are trained for 20,000 iterations. Each iteration will process 2048 randomly sampled pixels from the pre-extracted feature maps. Note that due to the efficient implementation of $\mathcal{F}$ and the pre-extraction of patch features, our denoising typically takes about 100-160 seconds to finish (including the feature extraction time). This rapid optimization process allows us to easily amortize the denoising cost with parallel computes, thereby ensuring the practicality and applicability of our method in various scenarios.

We use the same hyperparameters for all experiments without any specific tuning. See Figures S9 to S15 for visualizations of some examples of our per-image denoising outputs.

A.2. Generalizable Denoiser

Optimization To train the denoiser, we optimize the loss function defined in Equation (11). Note that our approach does not necessitate re-training ViTs; instead, it only optimizes the newly initialized parameters. The denoiser is trained over 10 epochs with a batch size of 64, utilizing the AdamW optimizer with a learning rate of 2e-4 and a cosine learning rate scheduler. The denoiser training typically takes about 2 hours on 8 GPUs.

A.3. ViT Models

Model identifiers. We provide the `timm` model identifiers of the ViTs studied in this paper in Table S5. For experiments with large input image sizes (*e.g.* using the 512-sized images as input to a model trained with 224-image-resolution), we always resize the position embeddings using bicubic interpolation to accommodate the increased size.

A.4. Correlation

In the main text, we mention the correlation between artifacts and their positions in images without detailed context, which

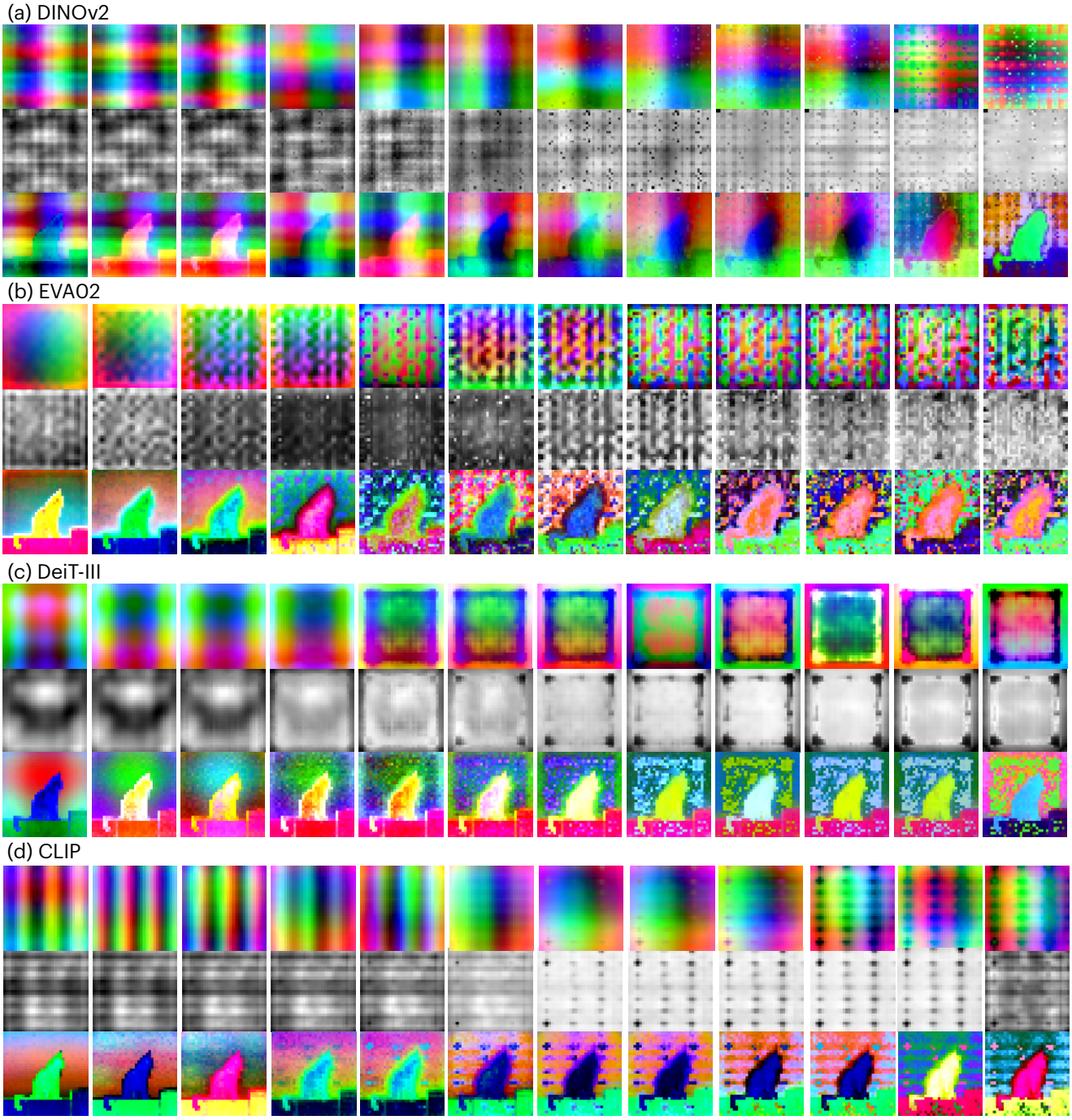

Figure S6. **Feature map visualizations: positional embeddings (PE) and a cat image in different ViTs.** We visualize the feature maps across different layers (1 to 12) of various pre-trained ViT-base models, displayed sequentially from left to right. For each panel, the top row shows the feature maps generated by inputting zero-tensors, highlighting the influence of PE alone. The middle row showcases the feature norm of the PE feature map. The bottom row presents the feature map for a sample cat image, allowing for a comparison that reveals visual correlations between the artifacts in general image feature maps and the PE feature map.

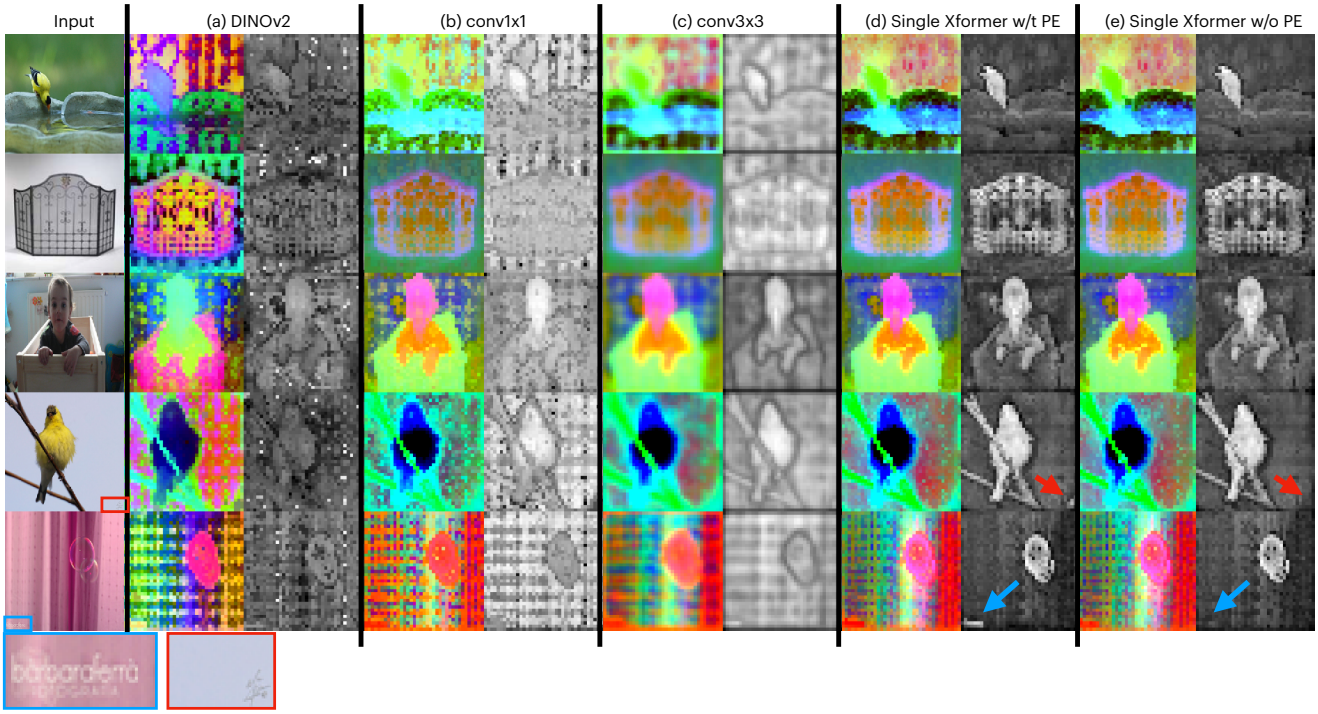

Figure S7. **Qualitative comparison of different denoiser architecture designs**. Convolution-based denoisers typically do not yield good performance (b, c). We empirically find that the denoiser with learnable new positional embeddings (PE) is sensitive to subtle details (see the blue and red rectangles and arrows). "Xformer": Transformer block.

Table S5. **`timm` model indentifiers.**

Model	Model identifier
DINOv2 [22]	`vit_base_patch14_dinov2.lvd142m`
Register [8]	`vit_base_patch14_reg4_dinov2.lvd142m`
DINO [4]	`vit_base_patch16_224.dino`
MAE [15]	`vit_base_patch16_224.mae`
EVA02 [13]	`eva02_base_patch16_clip_224.merged2b`
CLIP [26]	`vit_base_patch16_clip_384.laion2b_ft_in12k_in1k`
DeiT-III [32]	`deit3_base_patch16_224.fb_in1k`

we now provide. Our focus is on quantifying the correlation between different features and their positions within an image. To analyze this correlation, we employ the maximal information coefficient (MIC), a metric originally used for measuring the strength of linear or non-linear associations between two scalar variables. To adapt MIC for our purpose, we compute the association between high-dimensional features $\mathbf{f}$ and their positions. We calculate this by taking the maximal MIC across all channels of $\mathbf{f}$ and averaging the MICs of the x and y coordinates:

$$\frac{\max_{c\in\mathcal{C}} \mathrm{MIC}(\mathbf{f}(x,:),x) + \max_{c\in\mathcal{C}} \mathrm{MIC}(\mathbf{f}(:,y),y)}{2}, \tag{S12}$$

where $\mathbf{f}(x,:)$ denotes the feature vector on the x-coordinate, $\mathbf{f}(:,y)$ at the y-coordinate, and $\mathcal{C}$ is the channel size of $\mathbf{f}$. For hyperparameters of scalar MIC, we set $B = (H \times W)^{0.6}$:

$$\mathrm{MIC}(\mathbf{X};\mathbf{Y}) = \max_{|\mathbf{X}||\mathbf{Y}|<B} \frac{I[\mathbf{X};\mathbf{Y}]}{\log_2\left(\min\left(|\mathbf{X}|,|\mathbf{Y}|\right)\right)}, \tag{S13}$$

where $I[\mathbf{X};\mathbf{Y}]$ denotes the mutual information between two random variables $\mathbf{X}$ and $\mathbf{Y}$. We compute this metric from 100 randomly selected samples from the ImageNet dataset.

Our analysis includes a comparison of MIC values for the decomposed noise map, the original noisy ViT features, and the denoised, artifact-free features. The results, present in Table 1 of the main paper, reveal that the decomposed noise map exhibits the highest correlation with image positions.

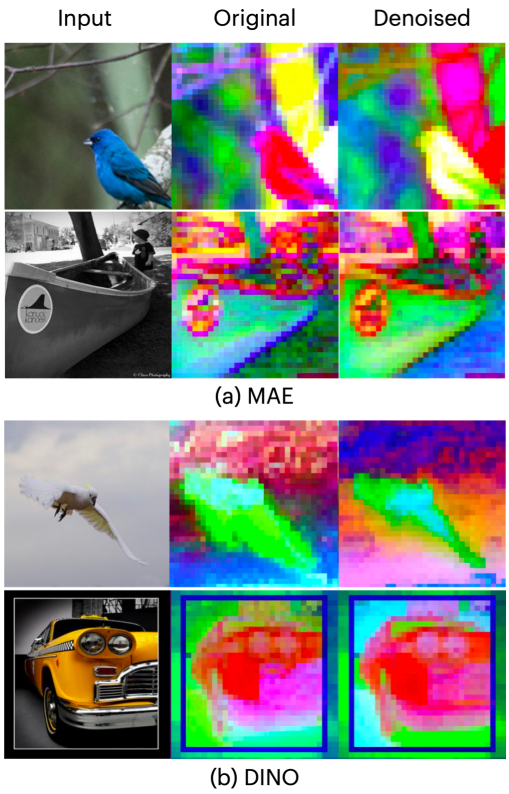

Figure S8. **Features from Weak Artifact Algorithms.**

The noisy features, entangled with noise artifacts originating from the position embeddings, display the second-highest positional correlation. In contrast, the noise-free features denoised by our method show the lowest correlation with positions, demonstrating the effectiveness of our decomposition approach in removing such artifacts.

A.5. Feature Qualitative Results

Algorithms producing mild artifacts. We additionally visualize the features for algorithms with weak artifacts in Figure S8. We empirically observe that ViTs trained using both MAE and DINO exhibit very few visible artifacts in their feature (center column). Figures S14 and S15 show additional visualizations of the decomposed noise map and the learned residual terms of MAE and DINO, respectively. We note that decomposed noise maps from these two models typically manifest low-frequency patterns and the residual terms do not yield pronounced patterns.

Additional visualizations. Additional visualizations of the feature maps at all layers of ViT models are shown in Figure S6. Observe that the artifact is present in almost all layers of the models. See Figures S9 to S15 for more visualizations.

B. Further Discussion into ViT Understanding

High-norm *vs.* Low-norm patterns. The concurrent research [8] identifies artifacts in ViTs by examining the feature norm. However, our findings, as illustrated in Figure 5 "Original Feature L2 Norm" columns (*e.g.*, there are many empty dark patches in the DEiT-III visualization (c)), reveals the existence of "low-norm" patterns that also behave as artifacts, especially in models like CLIP [26] and DeiT-III [32]. In addition, the research [8] concludes that the "high-norm" patterns are particularly pronounced in *large and sufficiently trained* ViTs, a trend that is not observed in small models, but our analysis showcases the presence of artifacts in almost all ViTs. These discoveries suggest that solely assessing artifacts based on feature norms does not provide a comprehensive understanding. Consequently, this calls for more in-depth research to fully understand the nature of artifacts in ViTs and how they impact model performance and behavior. Such insights are crucial for developing more robust, next-generation ViTs, particularly in the context of handling and interpreting features in ViTs.

Different positional embeddings. The models studied in this paper cover three major types of position embeddings (PEs) — fixed sinusoidal PE (*e.g.*, MAE [15]), learnable additive PE (*e.g.*, DINO [4], DINOv2 [22], CLIP [26], DeiT-III [32]), and learnable Rotary PE (*e.g.* EVA02 [13]). Intriguingly, our observations reveal that, regardless of the type of PE employed, artifacts are present in all the studied ViTs, though with varying extents. The emergence of artifacts seems to be a common characteristic across different PE types. Although the fundamental underlying reason behind this property remains unclear, our work identifies this issue and proposes a denoising method to rectify these artifacts.

Alternative approaches for position embeddings. A key component of our hypothesis as to why artifacts exist in ViT features is the use of positional embeddings. Currently, all ViTs leverage either fixed [15] or learned [3, 22, 32] positional embeddings that are *added to the input tokens* of the Transformer model. Alternatively, Rotary Positional Embeddings [30], which were originally proposed in the language domain for better sequence length scaling, does not directly add anything to the input tokens. Instead, this method encodes the absolute position with a rotation matrix and crucially incorporates the explicit relative position dependency in the computation of the attention values. While EVA02 [13] does leverage this kind of positional embedding, the training process involves distilling from the already-noisy features from CLIP. Indeed, the noisy artifacts of the EVA02 model bear semblance to those from CLIP models, especially in the later layers (Figure S6). Thus, while the positional embedding selection is promising, more research should be

done towards ViTs that leverage these Rotary PE for artifact reduction. Similarly, the positional embedding used in the T5 language model [27] does not add a positional embedding directly to the input; instead, it learns a bias that is added to the key-query dot-product in the self-attention step and does not include explicit position information into the self-attention value vectors. ALiBi [23], used in many large language models (LLM), also does not do so, and instead adds a static bias to the query-key dot product. These methods eliminate the input-independent portion of the final output feature while retaining the benefits of the position embedding. For future work, we suggest further exploration into adapting other such positional embedding paradigms specifically for the image domain.

C. Discussion on Limitations

Our work serves as one of the initial studies into understanding the position-based artifacts present in the features of ViT models. We explore the presence of such artifacts and propose a method to denoise such artifacts. However, the underlying reason for why such artifacts exist, and in which way, remains elusive. In particular, the severity of the artifacts depends on the algorithm that it is trained on, *i.e*. DINOv2 has more exaggerated artifacts while MAE has weaker artifacts. Thus, one direction of exploration is investigating the training paradigm including the supervision —*i.e*. local *vs*. global — as well as the loss-induced parameter landscape —*i.e*. sharp *vs*. smooth Hessians. Furthermore, a better architectural design—*e.g*. new positional embeddings—may diminish the severity of the feature artifacts. In this work, we do not explore modifying the ViT's design; however, more study into its positional embeddings and the effect on downstream features should prove interesting. Ultimately, we believe our findings are intriguing to the community and further research is needed to better understand this fundamental problem.

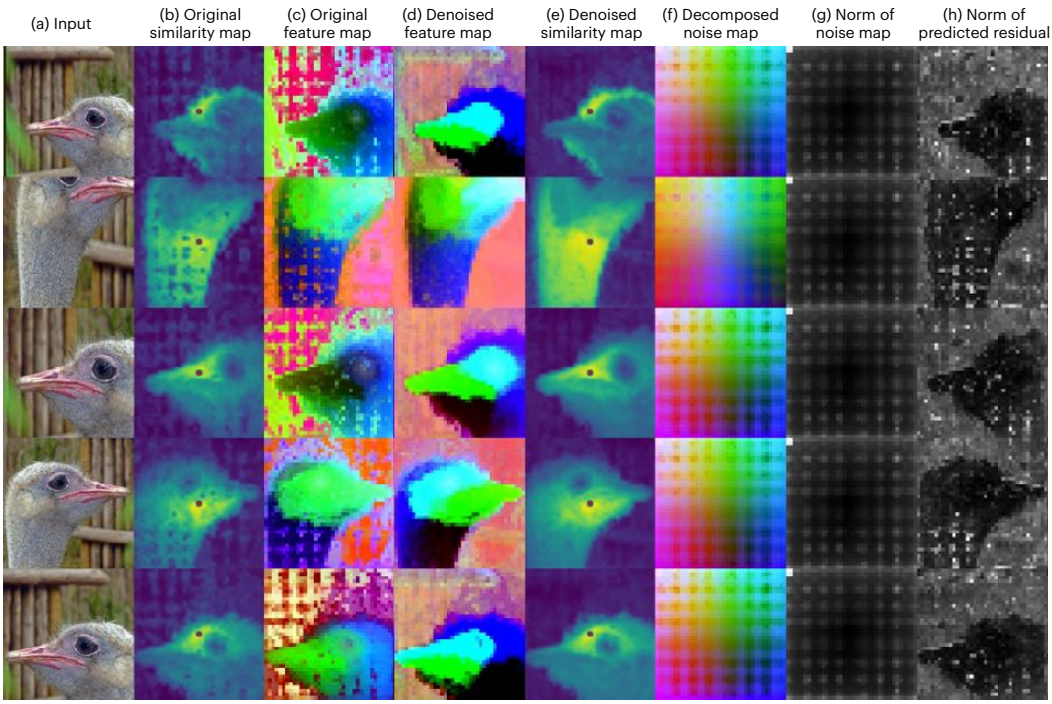

Figure S9. **Visualization of DINOv2 [22] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.

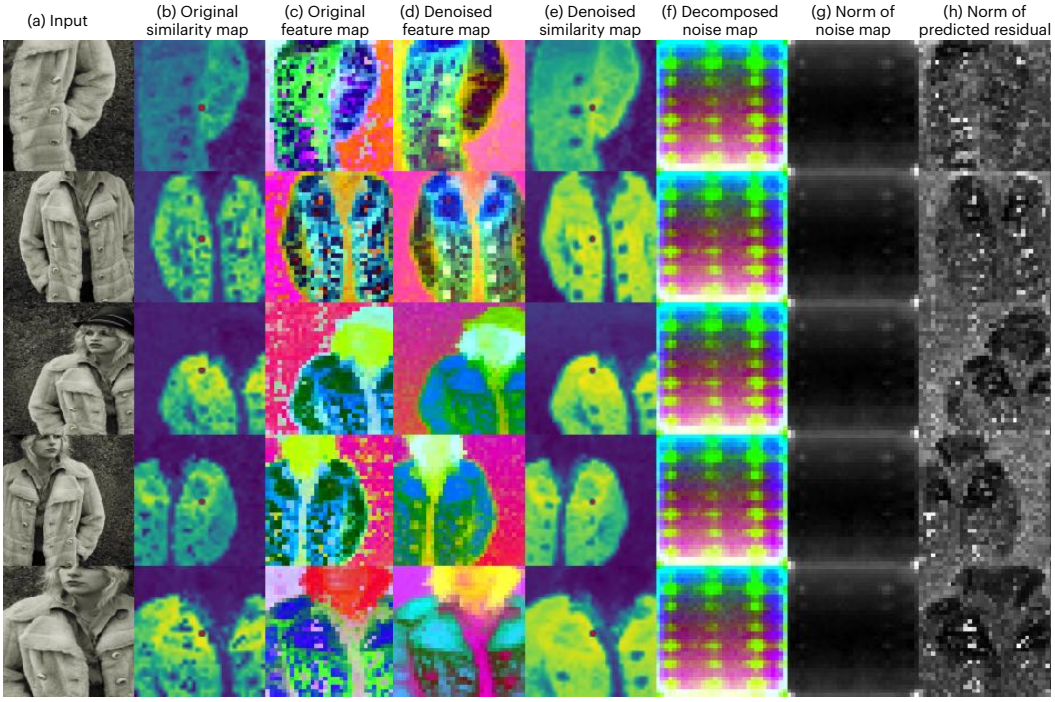

Figure S10. **Visualization of CLIP [26] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.

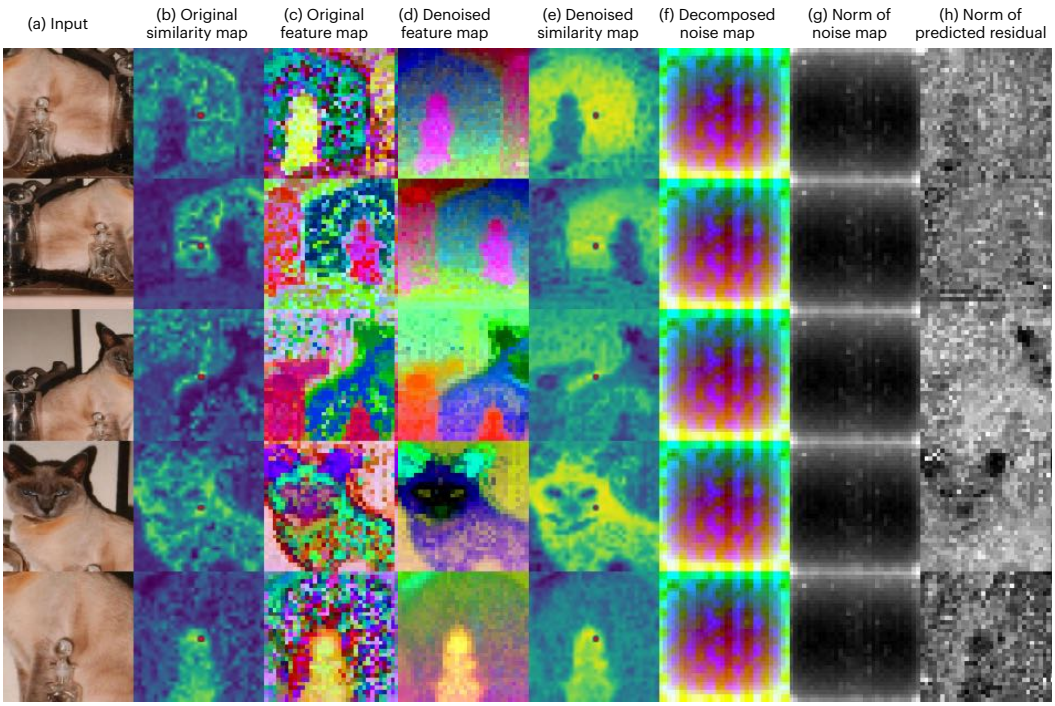

Figure S11. **Visualization of EVA02 [13] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.

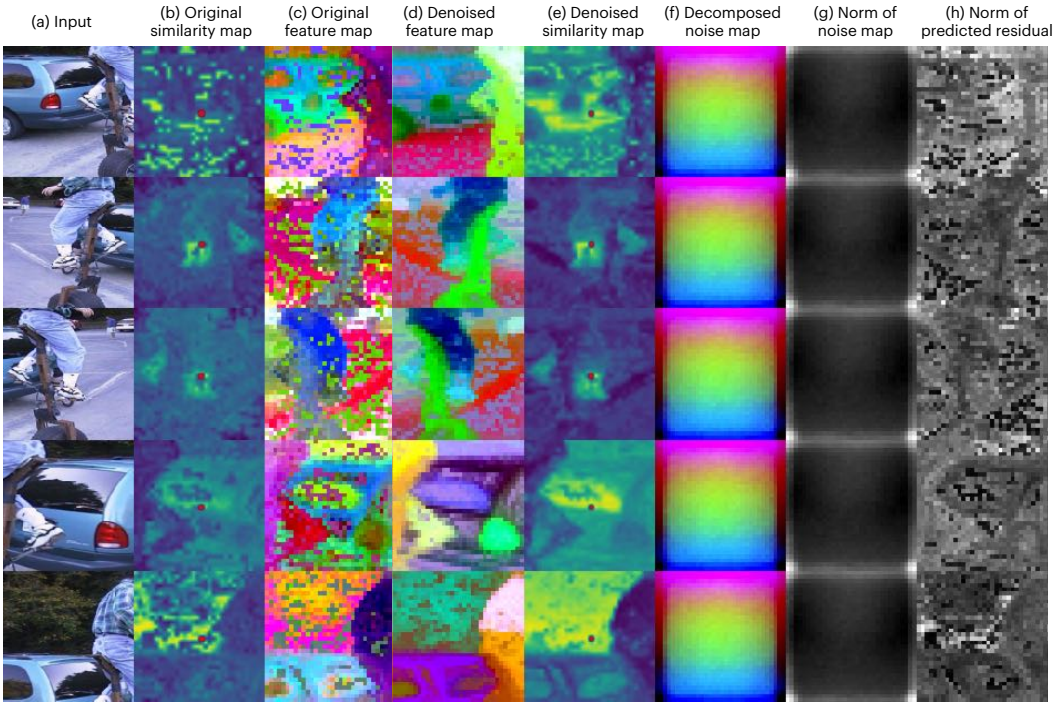

Figure S12. **Visualization of DeiT-III [32] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.

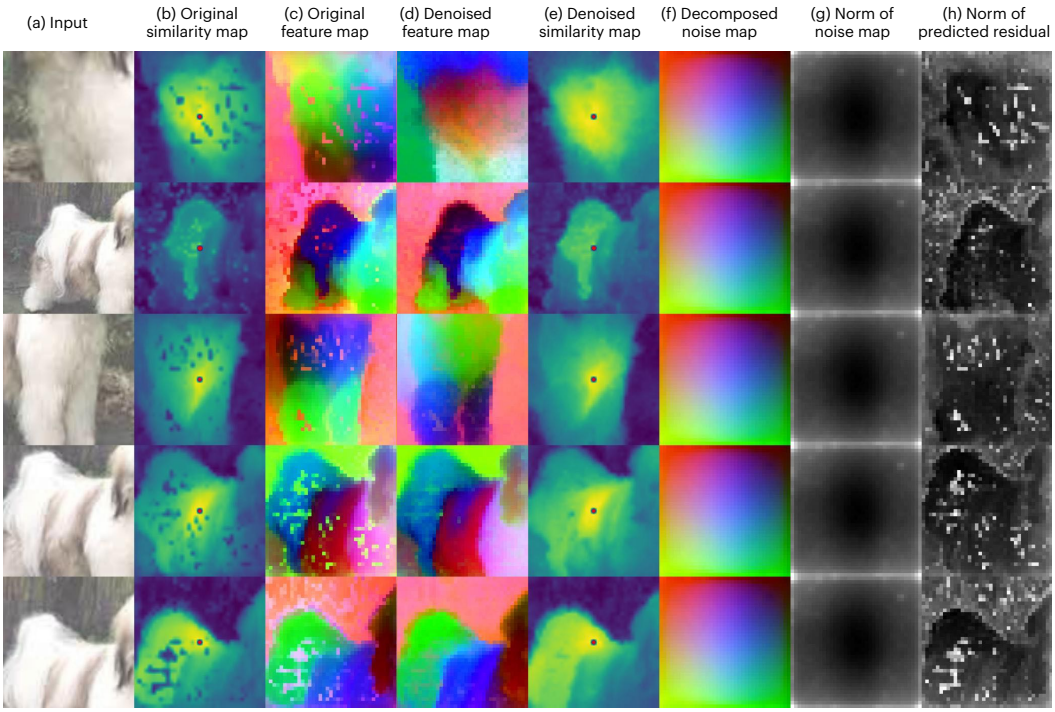

Figure S13. **Visualization of DINOv2 with Registers [8] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.

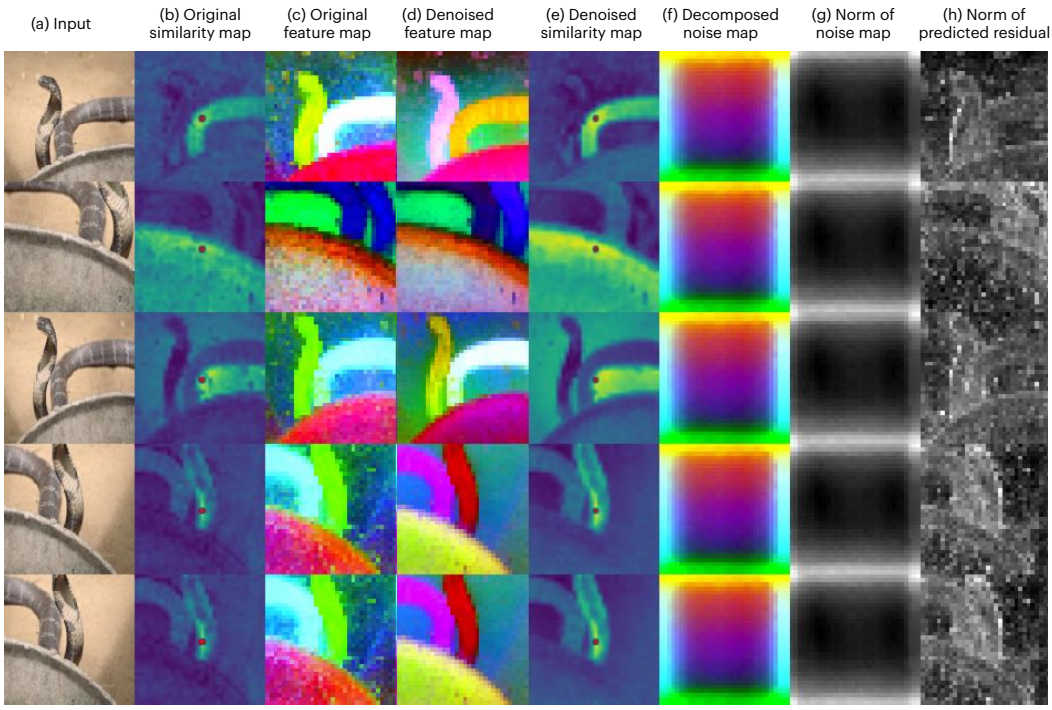

Figure S14. **Visualization of DINO [4] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.

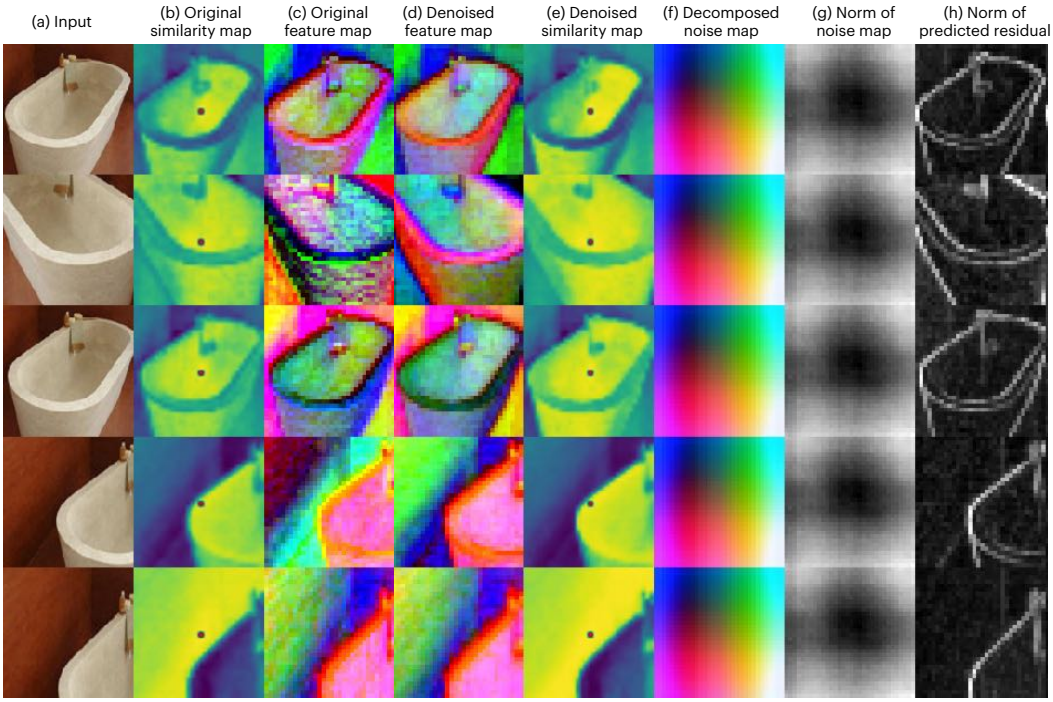

Figure S15. **Visualization of MAE [15] per-image denoising**. We visualize (a) the input image, (b) the similarity between the central red patch and other patches, (c) the original noisy feature map, (d) the denoised feature map, and (e) the similarity post-denoising. Additionally, we show the (f) decomposed noise map $\mathcal{G}$ and (g) its L2 norm as well as (h) the L2 norm of the predicted residual term $\mathbf{h}$.