Denoising Vision Transformers

Jiawei Yang*,†,1,⋆ Katie Z Luo*,2 Jiefeng Li[3] Congyue Deng[4]
Leonidas Guibas[4] Dilip Krishnan[5] Kilian Q Weinberger[2]
Yonglong Tian[5] Yue Wang[1]

[1]University of Southern California [2]Cornell University
[3]Shanghai Jiaotong University [4]Stanford University
[5]Google DeepMind
*equal technical contribution †project lead

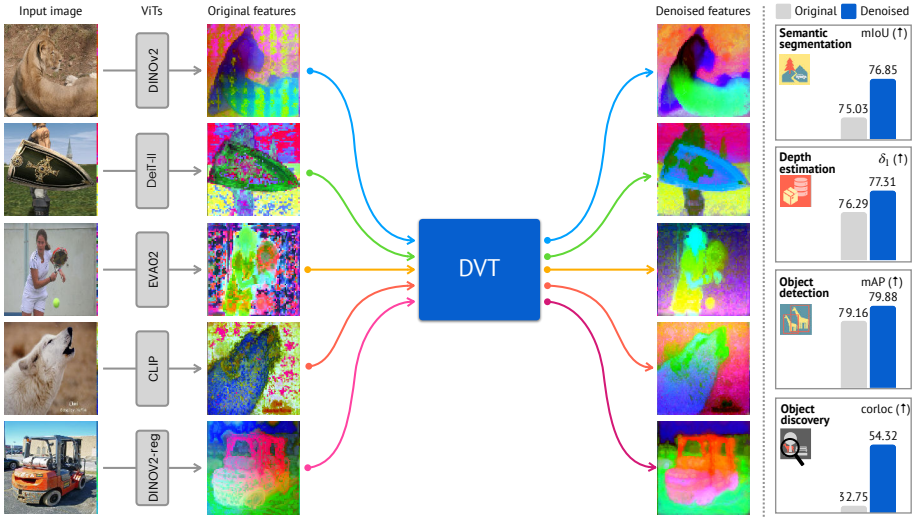


Fig. 1: **Denoising Vision Transformers (DVT)** effectively suppresses noisy artifacts in the visual features of all Vision Transformers (ViTs) we have tested and improves performance on a broad spectrum of dense prediction tasks, including semantic segmentation, depth estimation, object detection, and object discovery. Our evaluation encompasses a representative set of ViTs, including DINOv2 [25], DeiT-III [36], EVA-02 [13], CLIP [27], and DINOv2-reg [7]. We visualize the features before and after DVT, colored via principal component analysis (PCA). Best viewed in color. **Right**: We report the downstream dense prediction task performances, averaged over all models.

Abstract. We study a crucial yet often overlooked issue inherent to Vision Transformers (ViTs): feature maps of these models exhibit grid-like artifacts ("Original features" in Fig. 1), which hurt the performance of ViTs in downstream dense prediction tasks such as semantic segmentation, depth prediction, and object discovery. We trace this issue down to the positional embeddings at the input stage. To mitigate this, we

⋆ Work partially completed while interning at Google Research

propose a two-stage denoising approach, termed Denoising Vision Transformers (DVT). In the first stage, we separate the clean features from those contaminated by positional artifacts by enforcing cross-view feature consistency with neural fields on a per-image basis. This per-image optimization process extracts artifact-free features from raw ViT outputs, providing clean feature estimates for offline applications. In the second stage, we train a lightweight transformer block to predict clean features from raw ViT outputs, leveraging the derived estimates of the clean features as supervision. Our method, DVT, does not require retraining the existing pre-trained ViTs, and is immediately applicable to any Vision Transformer architecture. We evaluate our method on a variety of representative ViTs (DINO, DeiT-III, EVA02, CLIP, DINOv2, DINOv2-reg) and demonstrate that DVT consistently improves existing state-of-the-art general-purpose models in semantic and geometric tasks across multiple datasets (Fig. 1, right, Tabs. 2 to 4). We hope our study will encourage a re-evaluation of ViT design, especially regarding the naive use of positional embeddings. Our code and checkpoints are publicly available in our project page.

1 Introduction

In recent years, Transformers [38] have emerged as the universal architecture for modern foundation models across many modalities, from text [1,6,28,30] to audio [20,41], and images [2,9]. Among these, Vision Transformers (ViTs) [9] trained at scale not only achieve state-of-the-art under multiple benchmarks but also exhibit intriguing behaviors and capabilities across various tasks [3,16,25,27].

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 [3,9,13,25,27,36] (illustrated in Fig. 1 left). These artifacts not only compromise visual clarity but also hinder feature interpretability and disrupt semantic coherence. For example, Fig. 2 demonstrates that applying clustering algorithms directly on the raw ViT output results in noisy clusters, and the patch feature similarity is less reliable. Additionally, these artifacts are frequently concealed by *seemingly* impressive performance on downstream tasks, thus evading thorough examination or detection by the research community. Addressing these artifacts can unleash the potential of pre-trained ViTs and lead to substantial performance improvements (Fig. 1 right). Therefore, our work aims to answer a crucial research question: *Is it feasible to effectively denoise these artifacts in pre-trained ViTs, ideally without model retraining?*

To answer this, we first investigate the origins of these artifacts. We hypothesize that positional embeddings, a fundamental component of ViT architecture, play a pivotal role in the emergence of these artifacts. Our initial analysis supports this hypothesis: *First*, when a zero tensor is fed into a pre-trained DINOv2 model [25], the resulting output is predominantly characterized by similar noise patterns (Fig. 3-(a, 2)). *Second*, we observe the absence of such artifacts in the

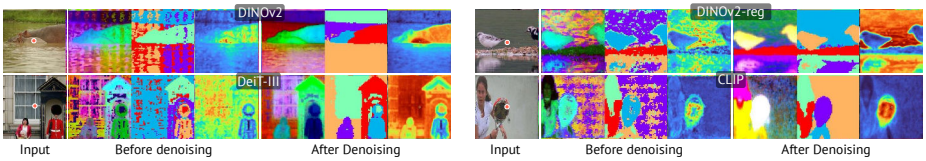


Fig. 2: **Artifacts hurt semantic coherence.** For each triplet, we show a feature map, a K-Means cluster map, and a similarity map of the central patch (red dotted) with other patches in the image. Observe how artifacts negatively impact clustering accuracy and similarity correspondences, and how our denoising mitigates these issues.

outputs of a DINOv2 model trained without positional embeddings, which contrasts sharply with the standard model outputs (Fig. 3-(a, 1) v.s. (a, 3)). *Third*, take a video with continuous frames as an example (Fig. 3-(c)). Despite the significant differences in the context of various input frames, the artifacts maintain a generally consistent relative position in the images (Fig. 3-(c), middle row).

With these insights, we present a two-stage denoising approach, Denoising Vision Transformers (DVT), to suppress artifacts in pre-trained ViTs. In the first stage, we obtain clean features from contaminated ones by enforcing cross-view feature consistency and artifact consistency with neural fields on a per-image basis. This per-image denoising process extracts noise-free features from raw output, providing these denoised 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. More importantly, this denoiser can be seamlessly integrated into pre-trained ViTs without extensive *re-training*, providing denoised features for online applications and generalizing well to unseen data.

We conduct empirical evaluations to demonstrate the efficacy of DVT on six representative ViTs: DINO [3], DINOv2 [25], DINOv2 with Register (DINOv2-reg) [7], DeiT-III [36], EVA-02 [13,14], and CLIP [27]. These evaluations demonstrate significant improvements in performance across various dense prediction vision tasks such as semantic segmentation, depth estimation, object detection, and object discovery. In summary, our contributions are:

- We identify and highlight the widespread occurrence of noise artifacts in ViT features, pinpointing positional embeddings as a crucial underlying factor. To the best of our knowledge, we are the first to provide such an analysis.
- We introduce a tailored noise model for ViTs, along with a neural field based denoising technique. This combination effectively isolates and removes noise artifacts from ViT features.
- We develop a flexible and efficient denoiser that integrates seamlessly with pre-trained ViTs, enabling real-time applications.
- Our approach results in substantial performance improvements across various ViTs and downstream dense prediction tasks (Fig. 1, right, Tabs. 2 to 4).

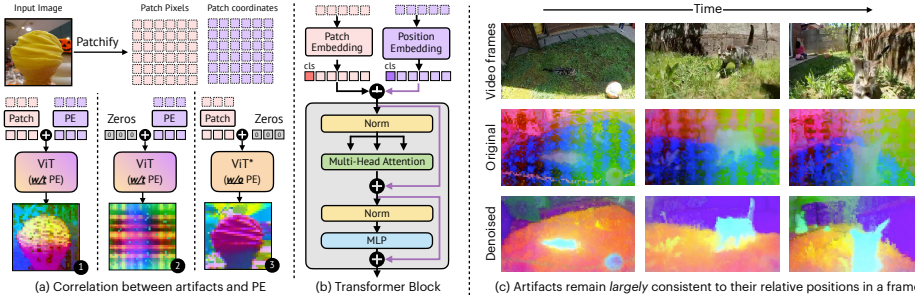


Fig. 3: **Impact of positional embeddings in ViTs.** (a) Comparison between DINOv2 ViTs [25] trained with and without positional embeddings (("ViT" v.s. "ViT*"). We show feature maps from (1) a standard ViT, (2) a ViT using only positional embeddings (PE) as input, emphasizing the emergence of artifacts, and (3) a PE-free ViT*, 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 largely maintain a consistent relative position in the images (central row). Our DVT effectively denoises these artifacts, demonstrated in the final row.

2 Related Works

General purpose features from Vision Transformers. Transformers have been used extensively across multiple domains as general-purpose feature extractors [1, 6, 8, 29, 30, 37]. Vision Transformers [9] (ViTs) pre-trained via supervised learning [18, 36, 39] or self-supervised learning [3, 16, 25, 46] have demonstrated strong generalizability to various downstream visual tasks, even without fine-tuning. However, we show that ViTs trained with diverse training objectives exhibit commonly observed noise artifacts in their output feature maps. These artifacts are often overlooked in practice because their presence cannot be simply reflected by image *classification* accuracy. Thus, our work focuses on evaluating pre-trained ViTs for *dense recognition* tasks such as segmentation, depth estimation, and object discovery. We demonstrate how these artifacts adversely affect dense recognition tasks, thereby motivating our method to mitigate them.

ViT artifacts. Our work studies the noise artifacts in ViTs, an issue that has been previously observed but often remains unexplored. These artifacts manifest as noisy attention maps in supervised ViTs (*i.e.*, ViTs do not attend to objects of interest well) [3, 5]. Concurrently with our study, two recent studies similarly have also identified artifacts in self-supervised ViTs [7, 44]. Specifically, [7] describe these as "high-norm" patches in low-informative background regions, hypothesizing 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*, as we observe similar artifacts in small or base ViTs that cannot be easily identified by extremely high feature norm values. Instead, we find a strong correlation between the presence of artifacts and the use of posi-

tional embeddings in ViTs. This finding suggests that artifacts are not strictly confined to certain model sizes or training scales but are more fundamentally linked to the inherent design of ViTs. Moreover, unlike the method proposed by [7] that retrains ViTs with register tokens [15,43] from scratch, our approach directly denoises pre-trained models without retraining. Users can *dynamically* enable or disable the plugged-in denoiser as needed. Lastly, we note that some *weak* artifacts still exist in DINOv2 models trained with registers [7] (see Fig. 1 DINOv2-reg and appendix), and our DVT can effectively denoise them, improving the performance of DINOv2-reg.

3 Preliminaries

Forward process in ViTs. Despite varying training approaches, the ViT architecture has mostly remained consistent with its original design as presented in [9] and [39]. The forward process of a ViT, depicted in Fig. 3-(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 resolution, P is the patch resolution, C represents the number of pixel channels and N is the number of total 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 $\mathbf{E}^i_{pos}$, are added to the patch embeddings. Formally, the forward process of a ViT is as follows:

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

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

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

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

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 how the *input-independent* positional embeddings function as a spatial inductive basis, intermixing with inputs and propagating throughout 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). Lastly, 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 Fig. 4.

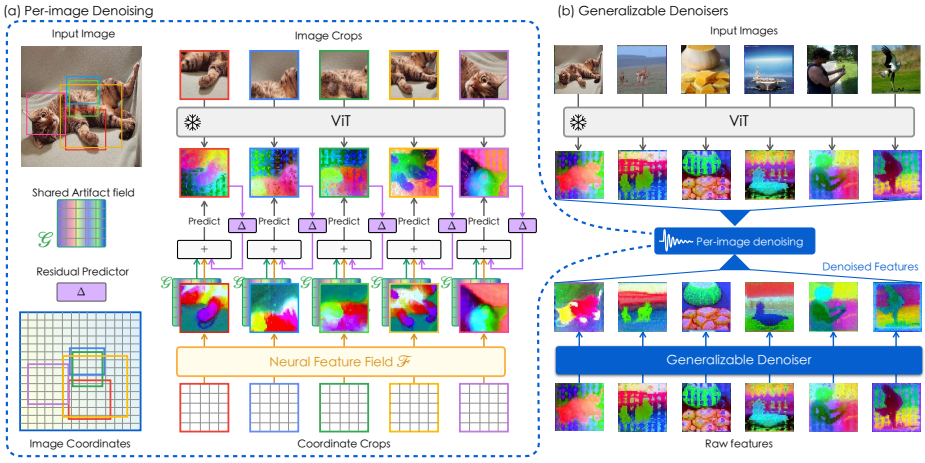


Fig. 4: **Method Overview.** DVT consists of a two-stage denoising pipeline. (a) In the first stage, our method decomposes the raw feature of an image crop into a noise-free semantics term $\mathcal{F}$, an input-independent, position-related artifact term $\mathcal{G}$, and an additional residual term Δ. (b) In the second stage, we train a generalizable denoiser to predict clean features from their original features. At inference time, only a single feedforward is needed to obtain denoised features.

4.1 Factorizing ViT Outputs

Our method is grounded in the principle that 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 Eqs. (1) to (4) and Fig. 3-(b), ViTs intermix patch embeddings with positional embeddings, thereby breaking the transformation invariance of visual features. This breach of invariance might not appear immediately problematic, but our investigations, illustrated in Fig. 3-(a) and (c), reveal a distinct correlation between the inclusion of positional embeddings and the emergence of undesirable artifacts in ViT outputs. Particularly, the middle row of Fig. 3-(c) shows that these artifacts persist with minor variation across different images, highlighting their consistency independent of the input content.

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 that accounts for the interdependence of semantics and positions $h(\mathbf{x}, \mathbf{E}_{pos})$. The decomposition is formally expressed as:

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

This factorization is universally applicable to all ViTs. For example, in scenarios where the output feature map is spatially invariant (*e.g.*, no positional

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

embedding is used), the sum of g and h becomes a constant bias term that can be merged into f.

4.2 Per-image Denoising with Neural Fields

Directly addressing the above decomposition problem within a single forward pass in a ViT is impractical due to the intertwined nature of output features. To overcome this, we exploit the consistencies in cross-view features and artifacts: (1) *Feature consistency* refers to the transformation invariance of visual features, where despite the varied spatial transformations, the 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 such that the semantic features obtained from any transformed view, $f\left(t\left(\mathbf{x}\right)\right)$, are equivalent to the transformed original semantic features, $t\left(f(\mathbf{x})\right)$; that is, $f\left(t\left(\mathbf{x}\right)\right) = t\left(f(\mathbf{x})\right)$ with $t \sim \mathcal{T}$. Next, we describe our approach for learning the different terms in Eq. (5) in conjunction 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*, alongside 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 coordinate networks, known as neural fields [17,19,23,32,35,44], to actualize $\mathcal{F}$ and $\mathcal{G}$. Specifically, we define $f(t(\mathbf{x})) = \mathcal{F}(\text{coords}(t(\mathbf{x})))$, where $\text{coords}(\cdot)$ extracts the pixel coordinates of the transformed views relative to 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. We learn the semantics field $\mathcal{F}$, the artifact field $\mathcal{G}$, and the residual term Δ by minimizing a regularized reconstruction loss:

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

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

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

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

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

Here, $\cos(\cdot,\cdot)$ denotes the cosine similarity, $\text{sg}(\cdot)$ 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 is designed to encourage $\widehat{\Delta}$ to remain

minimal by imposing a sparsity regularization, thereby allowing $\widehat{\mathbf{y}'}$ to represent as much of the ViT output as possible. The use of stop-gradient operators is to avoid trivial solutions, such as identity mapping. The reconstructed feature from our method is $\widehat{\mathbf{y}} = \mathcal{F}_\theta\left(\text{coords}\left(t\left(\mathbf{x}\right)\right)\right) + \mathcal{G}_\xi + \text{sg}\left(h_\psi\left(\text{ViT}\left(t\left(\mathbf{x}\right)\right)\right)\right)$, each term corresponding to f, g, and h as defined in Eq. (5).

Optimization. We break our optimization process into two 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. The problems we are left with are run-time efficiency and distribution shifts. Specifically, the per-image denoising process is suitable for offline applications but undesired for real-time applications, and individually denoised feature maps can lead to feature distribution shifts due to sample bias, which hampers the feature coherence across images. To address these issues, we introduce a generalizable denoiser.

After applying per-image denoising, we accumulate a dataset of pairs consisting of noisy ViT outputs $\mathbf{y}$ and their denoised counterparts $\mathcal{F}$, denoted as $\mathcal{B} = \{(\mathbf{y}_i, \mathcal{F}_i)\}_{i=1}^{B}$. We then train a denoiser network D_ζ to predict noise-free features from raw ViT outputs, *i.e.*, $\hat{\mathcal{F}} = D_\zeta(\mathbf{y})$. The loss function is:

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

Our generalizable denoiser is implemented as 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.

Notably, this learned denoiser is lightweight, thus adding negligible latency to the original ViT and facilitating real-time applications. It also learns to generalize across samples, mitigating the distribution shift issue in the per-image denoising process.

5 Experiments

In this section, we first explore if ViTs trained with different objectives all have artifacts. Then, we evaluate the effectiveness of our generalizable denoiser on dense prediction tasks. For all experiments, we default to using ViT-*base* models

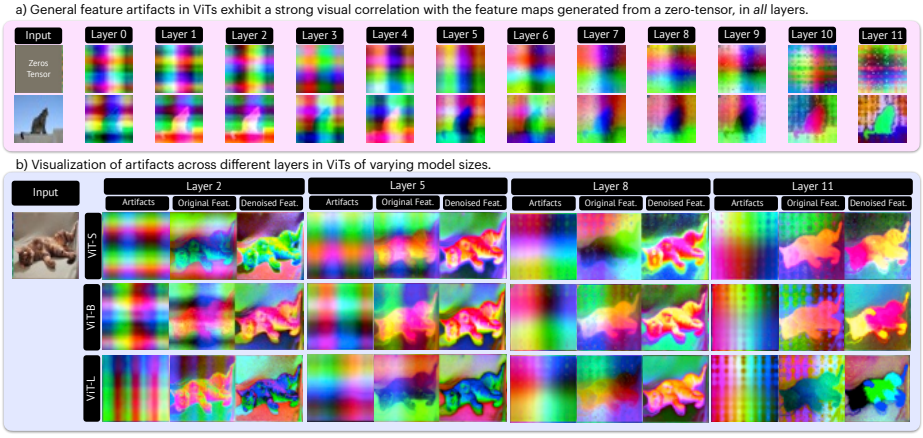


Fig. 5: **Visual analysis of ViT output features and denoised features.** (a) Visualizations of the feature maps from all layers of a DINOv2 [25] ViT-*base* model. Notably, the artifacts in the feature maps derived from the cat image exhibit a strong visual correlation with those from the zero-tensor inputs. (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 models.

with patch sizes of 14 or 16, depending on the availability of their implementations and model weights in PyTorch Image Models (timm [42]). We defer all the implementation details to the appendix.

5.1 Artifacts in ViTs

Positional artifacts in different ViTs. We visualize feature maps from differently pre-trained ViTs in Fig. 1. Among these, DINOv2 [25], a state-of-the-art vision foundation model with excellent performance on downstream tasks, displays clear position-related artifacts. Additionally, DeIT-III [36], trained with image class labels, and CLIP [27], 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. In ViTs we have tested, our proposed DVT successfully mitigates these artifacts ("Original features" *vs.* "Denoised features" in Fig. 1).

Artifacts in different layers. In Fig. 5, 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

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

	Before denoising	After denoising	
Method	Raw features	Artifacts term $\mathcal{G}$	Semantics term $\mathcal{F}$
DINOv2 [25]	0.44	0.54	0.22
DeiT-III [36]	0.34	0.32	0.06
CLIP [27]	0.11	0.14	0.08

ViT-*small* to ViT-*large*), diverging from the hypothesis in [7] that only large ViTs would display such patterns.

Correlation between artifacts and positions. Beyond visual qualitative inspection, we aim to quantitatively analyze the correlation between artifacts and their positions. Similar to [40], we use the maximal information coefficient (MIC) to measure the dependency between grid features and their normalized patch coordinates (See appendix for more details). This metric indicates how much patch features depend on their spatial positions and semantic content. As shown in Tab. 1, both the original ViT outputs and the decomposed artifacts exhibit a higher spatial correlation than the denoised semantic features, irrespective of the training methodology employed. These results support our hypothesis about the significant role of positional embeddings in the emergence of artifacts. Note that there is no "ground-truth" quantitative metric to to definitively quantify these patterns; hence, our reported numerical results should be viewed as empirical indicators, akin to the "high-norm" indicator used in [7].

5.2 Evaluation on Downstream Task Performance

We evaluate our method in dense recognition tasks, including semantic segmentation, monocular depth estimation, object detection, and object discovery. 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 generalizable denoisers.

Semantic segmentation. We follow [7, 25] to evaluate our approach in two semantic segmentation datasets: VOC2012 [12] and ADE20k [45], using a linear probing protocol, *i.e.*, a linear layer is trained to predict pixels' class from patch tokens. Tab. 2 presents the main results. We observe significant and consistent enhancements in all pre-trained ViTs across datasets. Notably, the DINOv2-*giant*, with an 83.0 mIoU on VOC2012 as reported in [25], is outperformed by our DVT-denoised DINOv2-*base* model (84.84 mIoU). This improvement is also evident in the ADE20k dataset, where the DINOv2-*giant* and DINOv2-*large* models attain mIoUs of 49.0 and 47.7, respectively, as reported in [25], while our denoised *base* model achieves a 48.66 mIoU. Remarkably, the *giant* model, which is **13**× larger than the *base* model, is outperformed by or on par with

Table 2: **Quantitative 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.

Method	VOC2012 Segmentation		ADE20k Segmentation		NYUv2 Depth Estimation	
	mIoU(↑)	mAcc(↑)	mIoU(↑)	mAcc(↑)	δ_1(↑)	abs rel(↓)
(a1) DINO [3]	63.00	76.35	31.03	40.33	73.19	**0.1701**
(a2) DINO [3] + **DVT**	**66.22**	**78.14**	**32.40**	**42.01**	**73.53**	0.1731
(b1) DeiT-III [36]	70.62	81.23	32.73	42.81	**72.16**	**0.1788**
(b2) DeiT-III [36] + **DVT**	**73.36**	**83.74**	**36.57**	**49.01**	71.36	0.1802
(c1) EVA02 [13]	71.52	82.95	37.45	49.74	63.68	0.1989
(c2) EVA02 [13] + **DVT**	**73.15**	**83.55**	**37.87**	**49.81**	**68.52**	**0.1964**
(d1) CLIP [27]	77.78	86.57	40.51	52.47	73.95	0.1679
(d2) CLIP [27] + **DVT**	**79.01**	**87.48**	**41.10**	**53.07**	**74.61**	**0.1667**
(e1) DINOv2-reg [7]	83.64	90.67	48.22	60.52	87.88	0.1190
(e2) DINOv2-reg [7] + **DVT**	**84.50**	**91.45**	**49.34**	**61.70**	**88.26**	**0.1157**
(f1) DINOv2 [25]	83.60	90.82	47.29	59.18	86.88	0.1238
(f2) DINOv2 [25] + **DVT**	**84.84**	**91.70**	**48.66**	**60.24**	**87.58**	**0.1200**

our denoised *base* model. This indicates that the performance gains primarily stem from effective artifact removal rather than the *minor* increase in model parameters of our denoiser network.

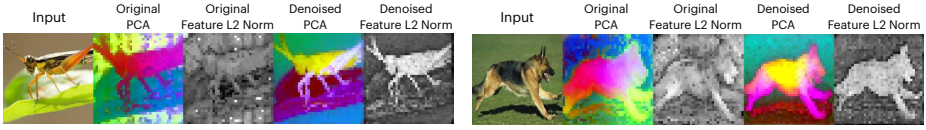
Our DVT also increases the performance of the concurrent DINOv2-reg model [7], where a ViT is trained with dummy learnable register tokens. As evidenced in Tab. 2, our DVT enhances the performance of both DINOv2 ((f1) *vs.* (f2)) and DINOv2-reg ((e1) *vs.* (e2)). When applying DVT only, DINOv2 shows more improvements compared to using registers ((f2) *vs.* (e1)); 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 achieves 83.64 mIoU and 48.22 mIoU on the respective benchmarks. Furthermore, DVT can further enhance the performance of DINOv2-reg ((e1) *vs.* (e2)) on both datasets (+0.86 in VOC2012 and +1.12 in ADE20k). In addition, DINOv2-reg [7] requires retraining entire models from scratch using 142M images, while our approach requires training a single Transformer block using 10k denoised samples.

Depth estimation. Following [25], we evaluate our method on the NYUv2-Depth dataset [24] using a linear evaluation protocol (more details in appendix). As shown in Tab. 2, our method clearly enhances the performance of most pre-trained ViTs. For context, the DINOv2-*large* model exhibits a 0.01 RMSE improvement over the DINOv2-*base* model with 3.5× more parameters. Our denoiser achieves similar performance gains with 0.08× the parameters of the base model. These results highlight our method's efficiency, achieving marked performance gains with minimal increases in parameter count.

Object detection. In this experiment, we train ViTDet detectors [21] on the frozen features following the Faster RCNN framework [31] (more details in appendix). We train all models on the VOC trainval07+12 subset and report their mAP metrics on the test2007 subset. Results are reported in Tab. 3. Our approach

Table 3: **Object detection with frozen features.** We report the mAP metric on the VOC object detection benchmark.

Method	DINOv2 [25]	DINOv2-reg [7]	CLIP [27]	DeiT-III [36]	DINO [3]	EVA02 [14]
baseline	81.4	80.9	80.9	75.8	76.4	79.4
baseline + **DVT**	**81.9** (+0.5)	**81.4** (+0.5)	**81.7** (+0.9)	**77.0** (+1.2)	**77.1** (+0.7)	**80.2** (+0.8)



(a) DINOv2 (b) DINOv2-reg

Fig. 6: **Emerged object discovery ability.** The features denoised by our DVT show higher feature norms on objects of interest.

Table 4: **Unsupervised object discovery using LOST [33].** We report the corloc score across three datasets. Our DVT significantly improves existing models. †: results quoted from [7]; these models are ViT-*large* trained on the ImageNet-22k dataset while our reported results are based on the publicly available ViT-*base*.

Method	VOC2007	VOC2012	COCO20k
(a) †DINOv2 [25]	35.3	40.2	26.9
(b) †DINOv2-reg [7]	55.4	60.0	42.0
(c) DINOv2-reg	38.0	41.5	26.9
(d) DINOv2-reg + **DVT**	**56.1** (+18.1)	**59.3** (+17.8)	**45.5** (+18.6)
(e) DINOv2	30.8	35.9	23.4
(f) DINOv2 + **DVT**	**58.0** (+27.2)	**60.3** (+24.4)	**46.7** (+23.3)

shows consistent improvements over the studied ViTs. Notably, DINOv2-reg [7] shows a slight decrease in object detection performance when compared to the original DINOv2 [25], while our approach improves it.

Object discovery. Unsupervised object discovery has been a long-standing problem of interest. An intriguing finding from our experiments is the emerging capability of object discovery in denoised ViTs. Fig. 6 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 from the feature norm values. This enhancement in object clarity is *not* a goal of DVT but emerges as the outcome of our method.

To quantitatively assess these enhancements, we follow [7] to use LOST [33] for evaluating object discovery efficacy before and after applying our DVT. We use feature norms as an indicator of object prominence. We conduct object discovery experiments on PASCAL VOC 2007 [11] and 2012 [12] and COCO20k datasets [22]. Tab. 4 presents the results. Our DVT significantly improves both DINOv2 [25] and DINOv2-reg [7] in all the evaluated datasets. In particular, while the publicly available DINOv2-reg shows some improvements ((c) *vs.* (e)), we find that it falls short of the performance levels reported in [7] ((c) *vs.* (b)).

Despite this, our DVT achieves more substantial enhancements in object discovery capabilities, even compared to the numbers reported in [7] ((f) *vs.* (b)).

Table 5: Ablation study on per-image denoising using KNN segmentation protocol on VOC12val.

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 6: 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

5.3 Ablation Study

In this section, we provide ablation studies to understand the importance of different components in our proposed DVT.

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. Tab. 5 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 Tab. 6. 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 multilayer convolution denoiser with a Conv1x1-ReLu-Conv1x1-ReLu-Conv1x1 structure, and 4) a multilayer convolution denoiser with a Conv3x3-ReLu-Conv3x3-ReLu-Conv3x3 structure. We observe that 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 embeddings achieves the best result (d), and removing the learnable position embeddings obtains very similar numerical performance (e). We empirically find that the design of (d) leads to better qualitative visualizations, and thus we use this setting.

Scaling behaviors. We study how DVT scales with model sizes and data scales in Fig. 7. In (a), we see DVT boosts differently-sized ViTs, even allowing ViT-*base* to match or exceed ViT-*giant* performance in semantic segmentation. Overall, DVT's scaling behaviors closely align with those of baseline models. In (b), we

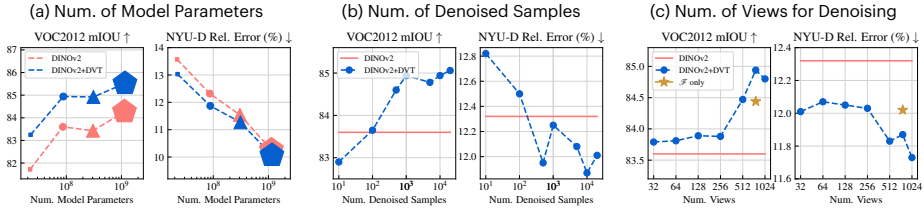


Fig. 7: **DVT's Scaling Behaviors.** We study the generalizable denoiser's performance for (a) different model sizes, (b) the number of denoised samples used for training denoisers, and (c) the number of views used when performing per-image denoising.

study the impact of the number of denoised training samples on task performance, where DVT shows promising results even with limited training samples (*e.g.*, 100∼1000). Note that our denoiser never sees ADE20k and NYU-depth datasets during training, yet generalizes effectively. In (c), we plot the task performance *vs*. the number of views used for the denoising. DVT benefits from more views in first-stage denoising. When training neural fields, more views enhance performance, while fewer views lead to overfitting. In particular, aggregating views is itself an approach to denoising, which still *aligns with* our motivation. We also demonstrate that a denoiser trained on samples denoised solely by aggregating views via neural fields ($\mathcal{F}$-only in (c)) surpasses baselines but underperforms the full DVT, which further confirms the effectiveness of our proposed denoising procedure.

6 Discussion and Future Works

Denoising Vision Transformers (DVT) introduces a robust method leveraging neural fields to eliminate feature artifacts from ViTs. This work additionally pinpoint positional embeddings as the primary source of these artifacts, despite their importance in various vision tasks. Using a neural field optimization process, DVT efficiently extracts clean features from the noise-riddled feature maps of existing ViTs. And using a scalable feature denoiser model, DVT eliminates the need for individual image optimizations. When learned from individually denoised samples, our denoiser generalizes well to unseen data and improves pre-trained ViTs by large margins in dense vision tasks. More broadly, our research suggests several avenues for future exploration: (1) understanding the role of positional embeddings in ViT could inform the design of next-generation deep learning architectures, and (2) redefining positional embeddings within ViTs and transformers is also an imperative problem. Lastly, combining the insights from our work and those of [7] could lead to a *more complete* picture of how these artifacts emerge. We hope that the results presented in this work contribute to a deeper understanding of artifacts in vision transformers and beyond.

Acknowledgements

We are grateful to many friends, including Jiageng Mao, Junjie Ye, Justin Lovelace, Varsha Kishore, and Christian Belardi, for their fruitful discussions on this work and follow-ups. Katie Luo is supported by an Nvidia Graduate Fellowship. Leonidas Guibas acknowledges the support from a Vannevar Bush Faculty Fellowship. We also acknowledge an unrestricted gift from Google in support of this project.

References

1. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020) 2, 4
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers, p. 213–229. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-58452-8_13, http://dx.doi.org/10.1007/978-3-030-58452-8_13 2
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) 2, 3, 4, 11, 12, 22, 26, 32
4. Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE (Oct 2021). https://doi.org/10.1109/iccv48922.2021.00951, http://dx.doi.org/10.1109/ICCV48922.2021.00951 26
5. Chen, X., Hsieh, C.J., Gong, B.: When vision transformers outperform resnets without pre-training or strong data augmentations. arXiv preprint arXiv:2106.01548 (2021) 4
6. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways (2022) 2, 4
7. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers (2023) 1, 3, 4, 5, 10, 11, 12, 13, 14, 22, 25, 31, 34
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019) 4

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 (2021) 2, 4, 5
10. El-Nouby, A., Klein, M., Zhai, S., Bautista, M.A., Toshev, A., Shankar, V., Susskind, J.M., Joulin, A.: Scalable pre-training of large autoregressive image models. arXiv preprint arXiv:2401.08541 (2024) 25
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html 12
12. 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 10, 12
13. Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva-02: A visual representation for neon genesis. arXiv preprint arXiv:2303.11331 (2023) 1, 2, 3, 9, 11, 22, 26, 29
14. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale (2022) 3, 12
15. Goyal, S., Ji, Z., Rawat, A.S., Menon, A.K., Kumar, S., Nagarajan, V.: Think before you speak: Training language models with pause tokens (2023) 5
16. 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) 2, 4, 22, 26, 33
17. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields (2023) 7
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023) 4
19. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation (2022) 7
20. Li, N., Liu, S., Liu, Y., Zhao, S., Liu, M., Zhou, M.: Close to human quality tts with transformer. arXiv preprint arXiv:1809.08895 (2018) 2
21. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: European Conference on Computer Vision. pp. 280–296. Springer (2022) 11, 24
22. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014) 12
23. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022) 7, 19
24. Nathan Silberman, Derek Hoiem, P.K., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 11
25. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023) 1, 2, 3, 4, 9, 10, 11, 12, 22, 24, 26, 27

26. Press, O., Smith, N.A., Lewis, M.: Train short, test long: Attention with linear biases enables input length extrapolation (2022) 26
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021) 1, 2, 3, 9, 10, 11, 12, 22, 26, 28
28. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018) 2
29. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019) 4
30. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer (2023) 2, 4, 26
31. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015) 11, 24
32. Shen, W., Yang, G., Yu, A., Wong, J., Kaelbling, L.P., Isola, P.: Distilled feature fields enable few-shot language-guided manipulation. arXiv preprint arXiv:2308.07931 (2023) 7
33. Siméoni, O., Puy, G., Vo, H.V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., Ponce, J.: Localizing objects with self-supervised transformers and no labels. arXiv preprint arXiv:2109.14279 (2021) 12, 24, 25
34. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding (2023) 26
35. Tancik, M., Srinivasan, P.P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J.T., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. NeurIPS (2020) 7
36. Touvron, H., Cord, M., Jégou, H.: Deit iii: Revenge of the vit (2022) 1, 2, 3, 4, 9, 10, 11, 12, 22, 26, 30
37. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models (2023) 4
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 2
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023) 4, 5
40. Voita, E., Ferrando, J., Nalmpantis, C.: Neurons in large language models: Dead, n-gram, positional. arXiv preprint arXiv:2309.04827 (2023) 10
41. Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., Saurous, R.A.: Tacotron: Towards end-to-end speech synthesis. In: Interspeech 2017. ISCA (2017). https://doi.org/10.21437/interspeech.2017-1452, http://dx.doi.org/10.21437/Interspeech.2017-1452 2
42. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861 9
43. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks (2023) 5
44. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., et al.: Emernerf: Emergent spatial-temporal scene decom-

position via self-supervision. In: The Twelfth International Conference on Learning Representations (2024) 4, 7
45. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset (2018) 10
46. 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) 4

Supplementary Material: Denoising Vision Transformers

In the appendix, we provide detailed implementation details in §A, elaborate on evaluation protocols and additional results in §B, and discuss the understanding of position embeddings in ViT in §C. Lastly, we discuss the limitations of this work and propose a few avenues for future work in §D.

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 feature 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 [23], 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 [23] 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, which corresponds to the feature dimension of the ViT being studied (*e.g.*, 768 for a ViT-*base* and 1024 for a ViT-*large*).

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 bilinearly interpolated to fit any arbitrary image resolution. We specifically choose these K values to minimize the need for run-time interpolation during training, thus improving denoising efficiency.

Residual predictor 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 $N = 768$ views (crops) from each image, applying random augmentations, which include random flipping

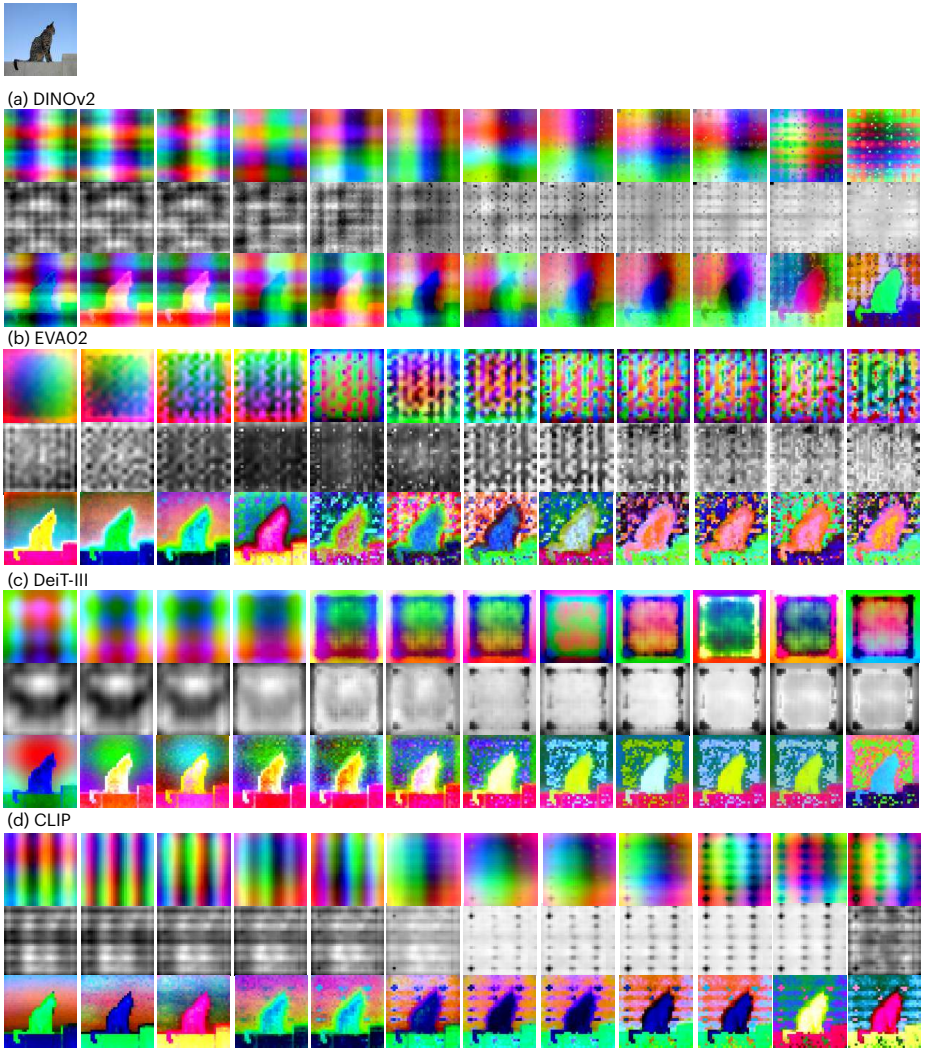


Fig. S1: **Feature map visualizations: positional embeddings (PE) and a cat image in different ViTs.** We visualize the feature maps across different layers (1 to 12) of various pre-trained ViT-base models, displayed sequentially from left to right. For each panel, the top row shows the feature maps generated by inputting a zero tensor, highlighting the influence of PE alone. The middle row presents 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.

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. For understanding how the number of views used for training affects the DVT's performance, please refer to Fig. 7 in the main text (our default setting is $N = 768$).

The coefficients in our loss function ((Eq. (6)) of the main text) 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 Figs. S4 to S10 for visualizations of some examples of our per-image denoising output.

A.2 Generalizable Denoiser

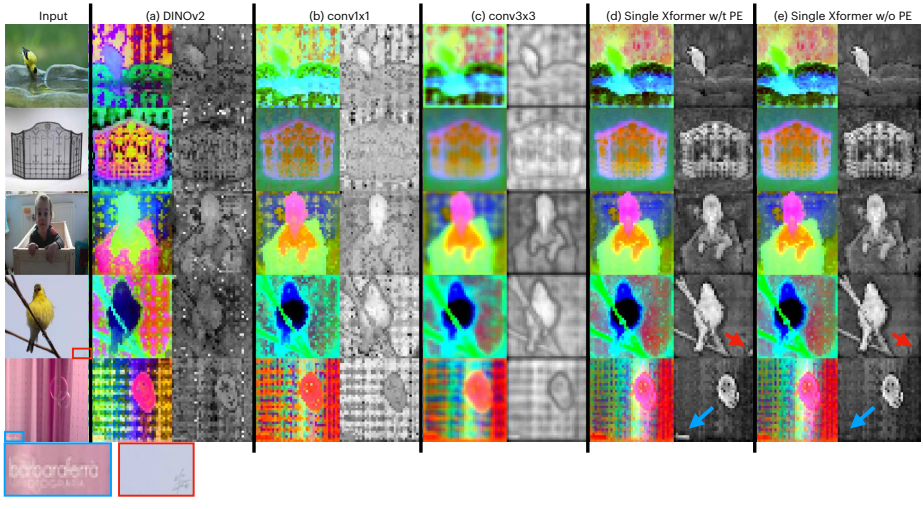


Fig. S2: **Qualitative comparison of different denoiser architecture designs**. Convolution-based denoisers typically do not yield good performance (b, c). We empirically find that the denoiser with learnable new positional embeddings (PE) is sensitive to subtle details (see the blue and red rectangles and arrows). "Xformer": Transformer block.

Optimization. To train the denoiser, we optimize the loss function defined in Eq. (11) of the main text. Note that our approach does not necessitate re-training ViTs; instead, it only optimizes the smaller denoisier network, which constitutes only 8% of the original model's size. The denoiser is trained for 10 epochs with a batch size of 64, using 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 Tab. S1. 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.

Table S1: **timm model identifiers**

Model	Model identifier
DINOv2 [25]	vit_base_patch14_dinov2.lvd142m
Register [7]	vit_base_patch14_reg4_dinov2.lvd142m
DINO [3]	vit_base_patch16_224.dino
MAE [16]	vit_base_patch16_224.mae
EVA02 [13]	eva02_base_patch16_clip_224.merged2b
CLIP [27]	vit_base_patch16_clip_384.laion2b_ft_in12k_in1k
DeiT-III [36]	deit3_base_patch16_224.fb_in1k

A.4 Correlation

In the main text, we mention the correlation between artifacts and their positions in images without a detailed context, which 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 nonlinear 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 coordinates x and y.

$$\frac{\max_{c\in\mathcal{C}} \mathrm{MIC}(\mathbf{f}(x,:),x) + \max_{c\in\mathcal{C}} \mathrm{MIC}(\mathbf{f}(:,y),y)}{2}, \tag{S1}$$

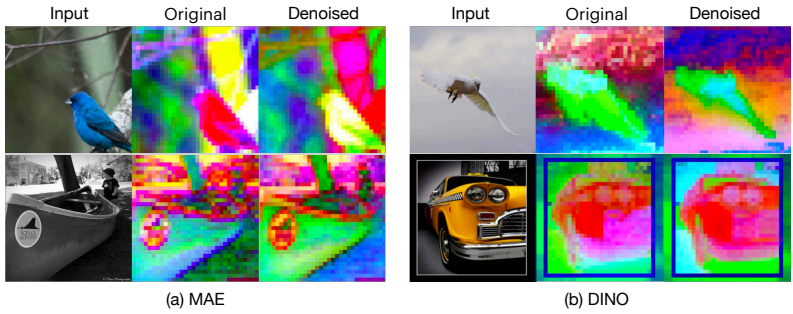


Fig. S3: **Features from Weak Artifact Algorithms.**

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 $B = (H \times W)^{0.6}$:

$$\mathrm{MIC}(\mathbf{X};\mathbf{Y}) = \max_{|\mathbf{X}||\mathbf{Y}|<B} \frac{I[\mathbf{X};\mathbf{Y}]}{\log_2\left(\min\left(|\mathbf{X}|,|\mathbf{Y}|\right)\right)}, \tag{S2}$$

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 the MIC values for the decomposed noise map, the original noisy ViT features, and the denoised, artifact-free features. The results, presented in Tab. 1 of the main paper, reveal that the decomposed noise map exhibits the highest correlation with image positions. The noisy features, which are 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 Fig. S3. We empirically observe that ViTs trained using both MAE and DINO exhibit very few visible artifacts in their feature (center column). Figs. S9 and S10 shows additional visualizations of the decomposed noise map and the learned residual terms of DINO and MAE, 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 Fig. S1. Observe that the artifact is present in almost all layers of the models. See Figs. S4 to S10 for more visualizations.

B Evaluation Protocols

We introduce our evaluation protocols here, mostly following [25]. We have released our code, checkpoints, and logs for reproducibility at https://jiawei-yang.github.io/DenoisingViT/.

Semantic Segmentation. We use a linear evaluation setting. In detail, we extract the final feature maps from the frozen backbone and pass them through the denoisers if there are any. Following this, feature maps are resized back to their original resolution. Then, a single learnable linear layer is trained to predict the semantic segmentation from these resized feature maps. The training and testing image resolutions are 518×518, following [25]. We train this linear head for $40,000$ iterations for both VOC and ADE20k datasets. We report the mean intersection over union (mIoU) metric for all experiments.

Depth estimation. We extract the final feature maps from the frozen backbone and pass them through the denoisers, if applicable. Then, we follow [25] to append the cls token to every patch token to enrich feature representations. We bilinearly upsample these features by a factor of 4 and train a linear layer using classification loss to divide the prediction into 256 uniformly distributed bins. Unlike [25], we slightly decrease the learning rate from 1e-4 to 5e-3, as we find that this modification improves most of the methods, including baselines, in our early experiments. We report our results on the commonly used metrics: AbsRel (absolute relative error $|d^* - d|/d$) and δ_1 (percentage of pixels where $\max(d^*/d,\ d/d^*) < 1.25$).

Object detection. To evaluate the object detection task, we utilize the ViTDet detector [21] to infer object bounding boxes based on feature maps extracted either from original ViTs or denoisers. The detection framework is FasterRCNN [31]. The input image resolution for training and testing is 518×518, the same as the semantic segmentation task. We train all models for 24k iterations, where we decay the learning rate at steps 20k and 22k.

Our initial attempts at directly learning an object detection head from the denoised features did not achieve superior performance. This led us to speculate that the omission of relative positional information, which was largely removed during denoising, might be important for accurately predicting the relative box coordinates of objects within the full image context. This requirement is almost *unique* to the bounding box prediction task. To counteract this, we re-add fixed sinusoidal positional embeddings into the feature maps produced by the denoisers. This adjustment, adding no additional learnable parameters, is found to enhance the detection performance. We believe that the disentanglement between positional features and semantic features would be an interesting direction to study. We also apply this method to the baseline models, and the results are shown in Tab. S2. We see that adding this step to the baselines does not yield consistent performance gains. Consequently, we apply this step only to our denoisers.

Object discovery. We use LOST [33] to evaluate the object discovery performance. LOST leverages the activation features of a pre-trained ViT for auto-

Table S2: **Object detection with frozen features.** We report the mAP metric on the VOC object detection benchmark. "fixed PE": fixed sinusoidal positional embeddings.

Method	DINOv2	DINOv2-reg	DeiT-III	CLIP	DINO	EVA02	Avg
baseline	81.4	80.9	80.9	75.8	76.4	79.4	79.2
+fixed PE	81.5 (+0.1)	81.2 (+0.3)	80.9	75.7 (-0.1)	75.8 (-0.6)	78.7 (-0.7)	79.0 (-0.2)
+DVT	**81.9** (+0.5)	**81.4** (+0.5)	**81.7** (+0.9)	**77.0** (+1.2)	**77.1** (+0.7)	**80.2** (+0.8)	**79.9** (+0.7)

Table S3: **ImageNet Classification Accuracy using Attentive Probing.**

Method	DINOv2	DINOv2-reg	DeiT-III	CLIP	DINO	EVA02
baseline	81.2%	81.6%	81.6%	83.4%	77.0%	80.3%
+DVT	**81.8**% (+0.6)	**81.8**% (+0.2)	**82.1**% (+0.5)	**83.5**% (+0.1)	77.0%	**80.5**% (+0.2)

mated object discovery. Specifically, it uses the components of the last attention layer for computing the similarities between the different patches to discover and identify the object connected components. To use LOST, one has to manually sweep between query, key, value, or other intermediate model outputs as the indictor of objects' prominence. Through our qualitative analysis, we find that the feature norm is a good candidate to indict object prominence (See Fig. 6). We report our results on the CorLoc metric (percentage of predicted box with an IoU greater than 0.5 with one of the labeled object bounding boxes) as in [7,33].

Classification. Although the global-level classification task is beyond the scope of our approach, our DVT demonstrates improved performance over its baselines through the use of an attentive probe protocol. Following the methodology described in AIM [10], we conduct an "attentive probe" on both the original and denoised patch tokens, omitting the CLS token, which our approach does not process during training. This probe employs attention mechanisms to maximize the extraction of information from each patch token. The backbones and the denoisers are frozen during our evaluation, and we train the attentive layer for 10 epochs. The results, presented in Tab. S3, suggest that DVT can potentially improve over its baselines, even though the denoising objective is orthogonal to classification. We believe integrating the CLS token into the denoising process represents a promising avenue for future research to enhance classification performance.

Additionally, we underscore the versatility of the denoiser in our DVT as a *plug-in-and-play* module, which can be optionally activated or deactivated to support various functionalities without compromising *any* properties of the original models. In essence, by leveraging the original class tokens before the denoiser, one can always recover the original models' classification performance.

C Further Discussion into ViT Understanding

Different positional embeddings. The models studied in this paper cover three major types of position embeddings (PEs) — fixed sinusoidal PE (*e.g.*,

MAE [16]), learnable additive PE (*e.g.*, DINO [3], DINOv2 [25], CLIP [27], DeiT-III [36]), 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 [16] or learned [4, 25, 36] positional embeddings that are *added to the input tokens* of the Transformer model. Alternatively, Rotary Positional Embeddings [34], 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. Although EVA02 [13] does leverage this kind of positional embedding, the training process involves distilling from the already-noisy features of CLIP. Indeed, the noisy artifacts of the EVA02 model resemble those of CLIP models, especially in the later layers (Fig. S1). 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 [30] 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 in the self-attention value vectors. ALiBi [26], 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.

D Discussion on Limitations

Limitations. Our approach faces some practical and theoretical challenges. On the practical front, although our method leverages parallel computing to amortize the denoising process, the time required to denoise a single image, such as one with a resolution of 518×518, remains high — approximately 100 seconds. This duration may be impractical for commercial or personal users with limited access to parallel computing resources, despite the fact that we can finish denoising 10k samples within hours. Additionally, our generalizable denoisesr, trained on the last layer features of pretrained ViTs, does not remove noise in intermediate outputs. Users requiring denoised features from multiple layers might need to train distinct denoisers for different layers. From the theoretical perspective, the reasons behind the presence of these artifacts remain unclear. Integrating

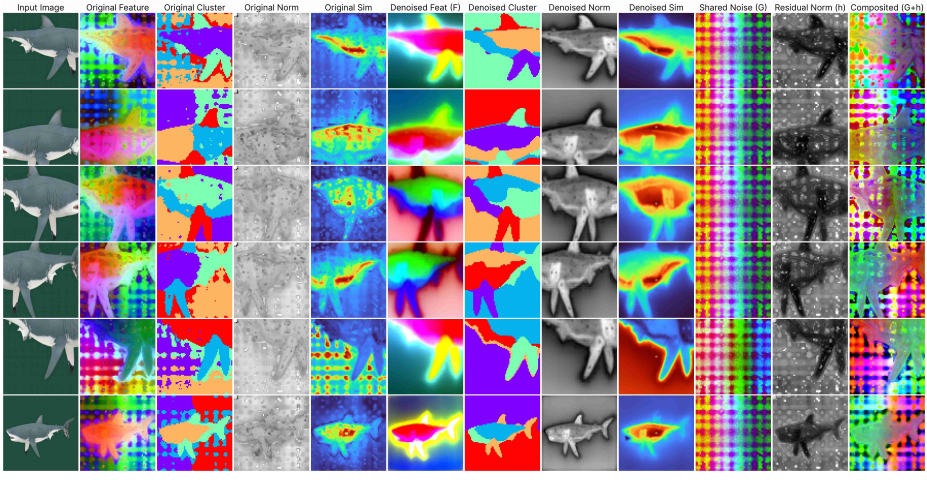


Fig. S4: **Visualization of DINOv2 [25] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise $(\mathcal{G} + \mathbf{h})$.

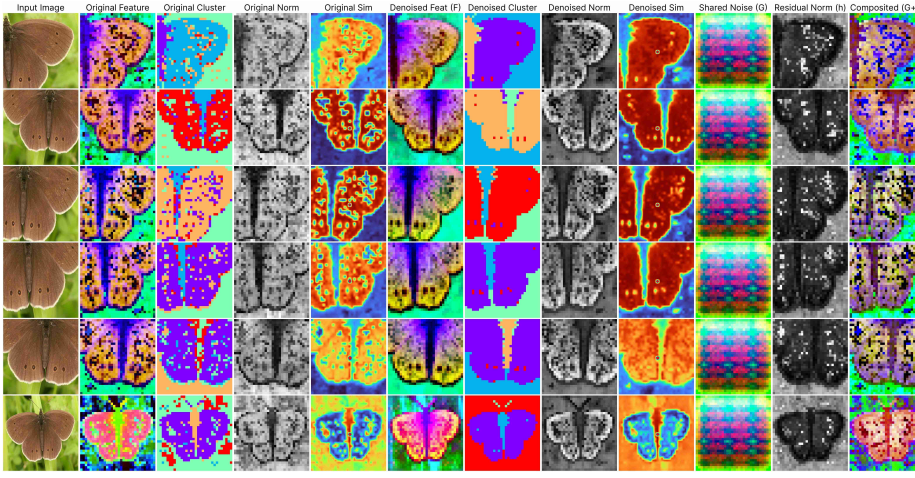


Fig. S5: **Visualization of CLIP [27] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise ($\mathcal{G} + \mathbf{h}$).

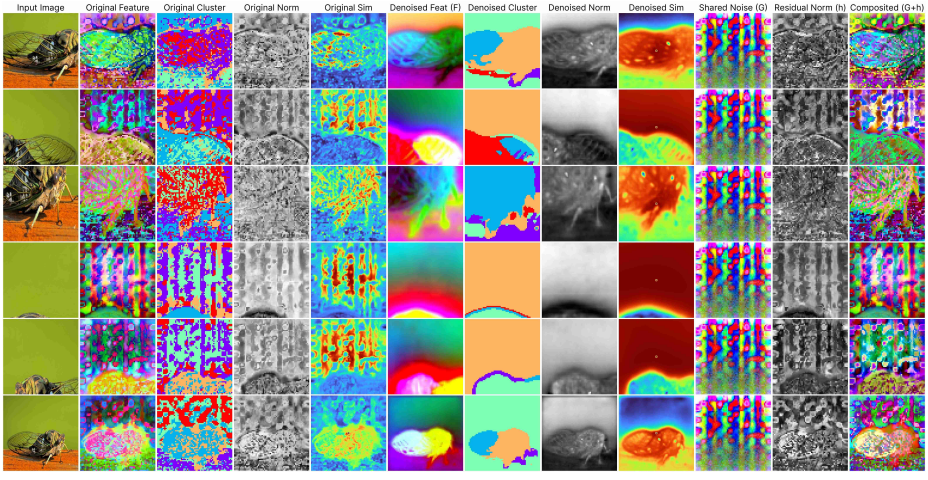


Fig. S6: **Visualization of EVA02 [13] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise $(\mathcal{G} + \mathbf{h})$.

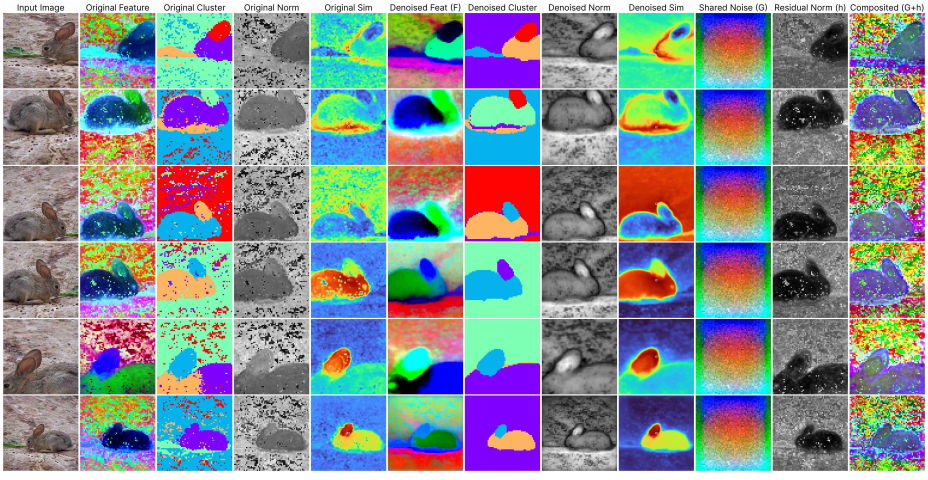


Fig. S7: **Visualization of DeiT-III [36] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise $(\mathcal{G} + \mathbf{h})$.

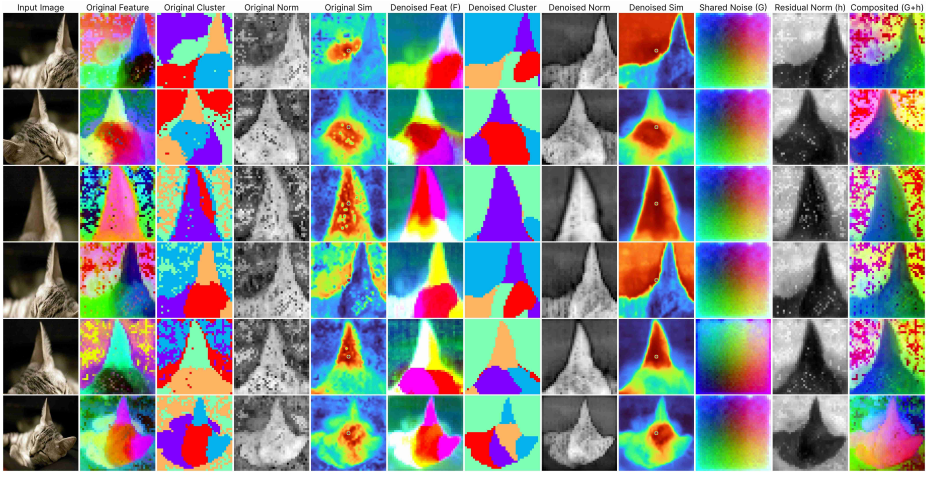


Fig. S8: **Visualization of DINOv2 with Registers [7] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise ($\mathcal{G} + \mathbf{h}$).

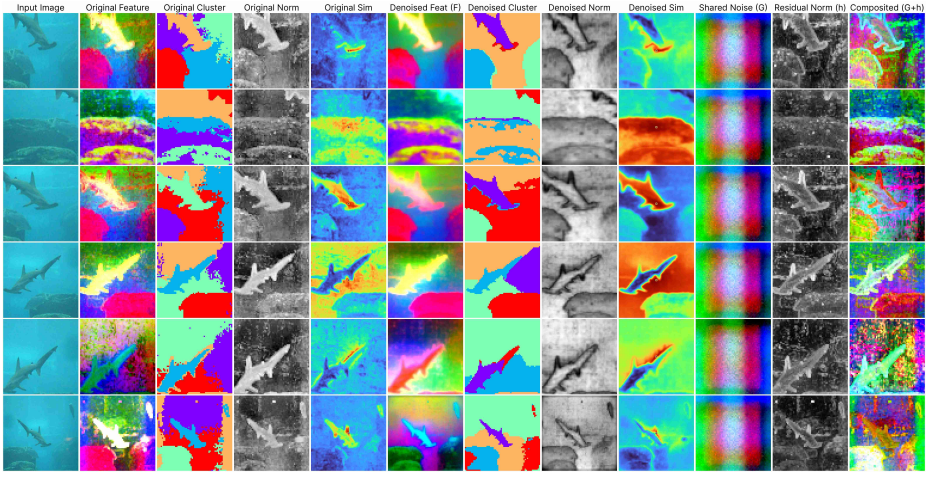


Fig. S9: **Visualization of DINO [3] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise $(\mathcal{G} + \mathbf{h})$.

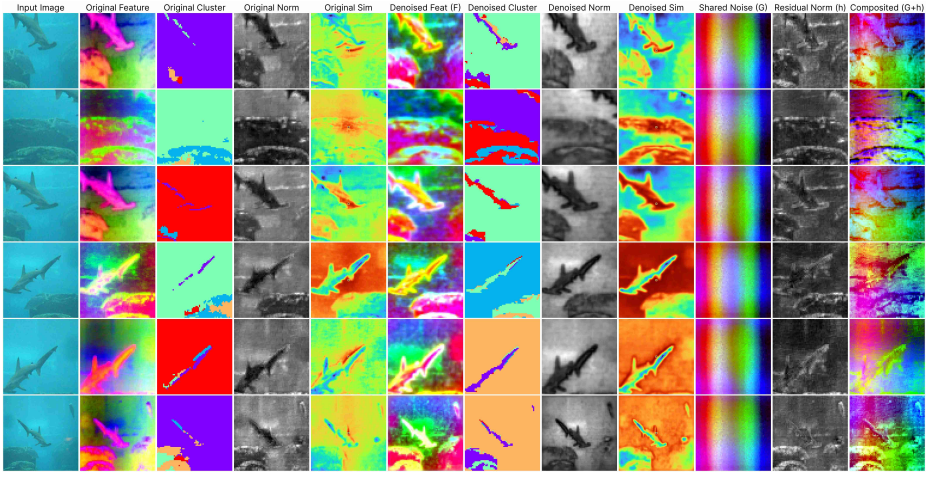


Fig. S10: **Visualization of MAE [16] per-image denoising**. We visualize all components of the per-image denoising stage. From left to right: In the first 5 columns we visualize the input image, the original noisy feature map from the model, the K-Means clusters on the original features, the L2 norm on the original features, and the similarity between the central red patch and other patches. In the next 4 columns we visualize the the denoised feature map using DVT, the denoised features' K-means clusters, the denoised features' L2 norms, and their similarity post-denoising. In the last 3 columns we visualize the decomposed shared noise term $\mathcal{G}$, the L2 norm of the predicted residual term $\mathbf{h}$, and the composite noise $(\mathcal{G} + \mathbf{h})$.

insights from Registers [7] with our findings could yield a more comprehensive understanding of these phenomena.

Broader Impact. Our work serves as one of the initial studies to understand the position-based artifacts present in the features of ViT models. We identify and propose methods to mitigate these artifacts, yet the root causes and characteristics of these artifacts are not fully understood. The severity of artifacts varies with the training algorithms; for instance, DINOv2 exhibits more pronounced artifacts compared to MAE, which shows subtler discrepancies. Thus, one direction of exploration is to investigate the training paradigm that includes 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 more research is needed to better understand this fundamental problem.