Vision Transformers Need More Than Registers

Cheng Shi Sibei Yang

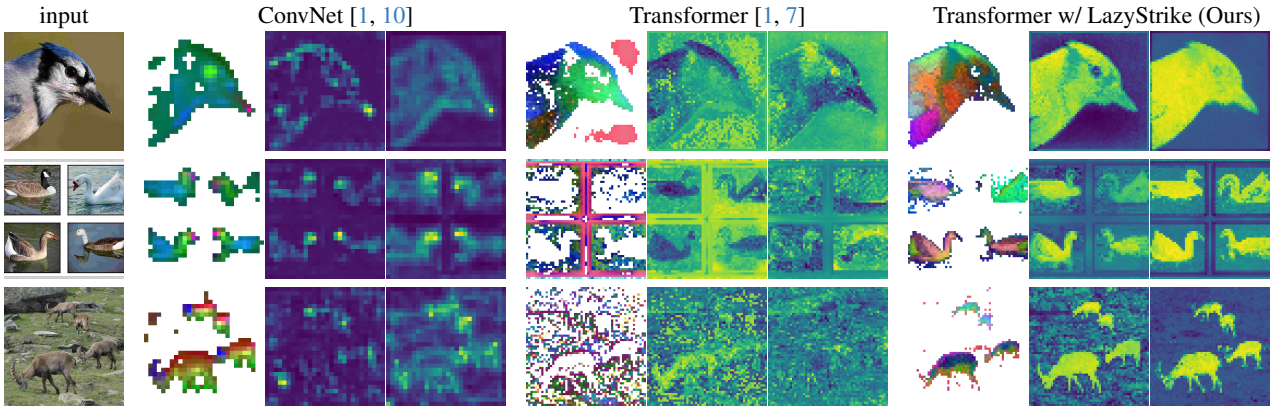

Figure 1. For the first time, LazyStrike provides a unified framework for the systematic analysis and effective mitigation of diverse image artifacts across various supervision settings for vision transformer. For each method, we show the patch score (middle: full supervision; right: self-supervision) and PCA visualization (left: full supervision). Patch score is the similarity between `CLS` token and patch features.

Abstract

Vision Transformer, pre-trained on large-scale datasets, provides general-purpose feature representations for a broad range of downstream tasks. However, artifacts in ViTs are widely observed across different supervision and downstream tasks. Through systematic analysis of artifacts in ViTs, we find that existing approaches have not sufficiently elucidated the fundamental mechanisms underlying these artifacts. In this paper, we reveal their origin in ***lazy behavior of ViT*** *where semantically irrelevant local patches act as shortcuts to represent global image semantics, influenced by the global dependencies of attention and the sparsity of annotations. Our solution—discarding more than half of the image patches to strike out ViT's lazy tendency—eliminates artifacts and mitigates associated distortion, delivering consistent performance improvements across 12 benchmarks under label, text, or self-supervision. We hope our work offers a new perspective on ViT's behavior. All the code and weights will be made publicly available.*

1. Introduction

> *"A problem well-stated is a problem half-solved."*
> – Charles Kettering

Vision transformer [7] (ViT) has become the *de-facto* standard for image recognition [6]. More importantly, it serves as a general-purpose feature extractor across various specific vision tasks [4, 11], functioning as a frozen foundation model pre-trained on large-scale data to embed images into feature representations, enabled by its scalability in data and model size. More broadly, this generic ViT feature extractor can adapt to various supervision methods during pre-training, with different approaches exhibiting characteristics particularly suited to diverse downstream tasks. Specifically, ***supervised methods***—such as training ViTs with fully-supervised classification labels or weakly-supervised image-text pairs (*e.g.*, in models like CLIP [22])—produce dense features for open-vocabulary tasks, and function as visual encoders for large vision-language models (LVLMs) [17]. Alternatively, ***self-supervised methods*** [1, 3, 20], particularly the DINO [1] model trained solely on images, demonstrate the potential for object and part discovery, making them applicable to unsupervised segmentation tasks [24].

However, recent studies uncover perplexing phenomenas in ViTs when applied to downstream tasks requiring dense features. For instance, DINO [35] demonstrates that label-supervised ViTs suffer from an **attention** deficit [21], while CLIPSelf [30] observes that text-supervised ViTs fail to produce dense image **features** that are accurately aligned with textual cues in open-vocabulary tasks. Meanwhile, Register [5] reveals that self-supervised ViTs generate artifacts in the **attention** maps, commonly referred to as high-norm

tokens, which adversely affect object localization tasks [24].

Intuitively, shouldn't these phenomena reflect a common underlying issue in ViTs, merely manifesting differently under various supervision paradigms [1, 12, 26] ? But, unfortunately, previous methods have focused on addressing the issues that manifest (even developing distinct solution approaches for different downstream tasks), yet the fundamental origination mechanisms remain underexplored in prior research. Therefore, in our preliminary exploration, we found that no single method [5, 29, 36] could comprehensively address those perplexing phenomena. This result suggests that our understanding of ViT remains incomplete, despite its existence [7] for half a decade. Given that all issues stem from a singular underlying cause, there must exist a unified solution. **In this paper, we undertake an investigation rooted in first principles thinking, systematically defining, analyzing, and ultimately resolving the problem of different types of artifacts observed in Vision Transformers, from their most fundamental elements.**

To establish a unified definition for those perplexing phenomena in ViT across different paradigms, we introduce the **Patch Score**—a metric that quantifies the similarity between patch features and the `CLS` token, which encapsulates an image's global semantics—thereby assessing local semantic consistency relative to the global representation, independent of the training paradigm. The intuition behind patch score is that for ViT under different supervision, the training objective aims to align the `CLS` feature with supervisory signals (*e.g.*, labels or text), any misalignment in dense features or high-norm token results in increased patch scores in non-foreground regions, as shown in Fig. 1. To quantitatively assess artifacts in patch scores, we propose the **Point-in-Box benchmark**, which evaluates whether the patch with the highest score is located within the annotated object bounding boxes. As shown in Fig. 1 and Tab. 1, we find that across different supervision settings, ViTs exhibit disordered patch scores and much lower point-in-box score compared with ConvNet [10]. For clarity and simplicity, unless explicitly stated otherwise, the term "artifact" in the following text specifically refers to semantically irrelevant background tokens that erroneously yield high patch scores.

Based on patch score and point-in-box score, we conduct an in-depth analysis into ViT's behavior, we propose a new hypothesis aimed at better explaining the artifacts in ViTs:

- *The sparsity of annotations results in redundant, semantically irrelevant image patches.* For example, we find that masking the top 50% highest-scoring patches in a pre-trained ViT does not harm image recognition performance on ImageNet (Fig. 2).
- *ViT's global dependencies allow it to exploit redundant local patches as shortcuts to represent global semantics.* In the absence of patch-level annotations, models may adopt a "lazy behavior" by diffusing small foreground semantics to background at the beginning of the training (Fig. 3). We validate that reducing ViT's global dependencies indeed mitigates artifact phenomena (Tab. 2).

Method	High Norm	Point-in-Box (%)
ResNet [10]	✗	68.4
ViT [7]	✓	42.7
+Register [5]	✗	41.5
DINO-ResNet [1]	✗	71.1
DINO-ViT [1]	✗	45.3
OpenCLIP-ResNet‡ [22]	✗	53.9
OpenCLIP-ViT [22]	✓	39.8
+Register [5]	✗	37.6

Table 1. Point-in-Box score across different supervision methods. We find that Register indeed effectively avoids high-norm phenomena but high-norm token is not the root cause of artifacts. ‡: CLIP-ResNet adopts attention-pool.

Building on this interpretation, we further validate our hypothesis by proposing a straightforward solution to eliminate these artifacts: *By discarding certain background patches during pre-training, we enforce ViTs to lock foreground semantics.* Specifically, we enable the model to learn to identify redundant tokens (Sec. 5.2) and to drop redundant patches at varying ratios during aggregation. As shown in Fig. 5, ViTs automatically shift their attention to foreground objects, aligning high-scoring patches with the foreground as these ratios are appropriately increased. After the **La**zy behavior is **St**ruck away, our approach, termed LaSt-ViT, eliminates artifacts in our patch scores across all supervision methods, effectively addressing both high-norm issues and feature misalignment. More importantly, for the first time, **ViTs exhibit emergent properties [35] across different supervision settings.**

In summary, our main contributions are as follows:

- We systematically analyze the root cause of different types of artifacts in ViTs, namely lazy behavior, providing a unified metric and a comprehensive, in-depth explanation.
- We propose a simple yet effective solution, drop more than half of the image patches, which eliminates artifacts for both supervised and self-supervised ViT.
- We show that the emergent properties of ViTs across different supervision settings can be achieved by LaSt-ViT and explained by our hypothesis, providing a new perspective on ViT behavior.
- By solely eliminating artifacts, LaSt-ViT enables the use of pre-trained ViTs as feature extractors while still achieving consistent and significant improvements across 12 downstream benchmarks—including object discovery, semantic/instance segmentation, and open-vocabulary object detection.

2. Related Work

Artifacts in text-supervised ViT (CLIP-type model [22]). Recent years have witnessed rapid advancements in vision-language contrastive pretraining [15, 22]. Surprisingly, beyond image-level classification, MaskCLIP [36] first find that CLIP model can extract free dense labels (zero-shot semantic segmentation) from the dense alignment in the last layer feature map and the text feature. However, subsequent studies [9, 14, 16, 29, 30, 34] have revealed that while ViT outperforms ResNet in terms of model size and classification performance, it significantly lags behind ResNet in dense alignment tasks. To address the misalignment issue, existing works can be categorized into two approaches: modifying the network architecture and introducing additional alignment training. For the former approach, mainstream methods [9, 14, 16, 29, 31, 34] primarily focus on modifying the final attention layer. For the latter approach, CLIPSelf [30] improves performance on open-vocabulary dense prediction tasks by aligning region-level features with image-level features. Different from previous methods in text-supervised ViTs, which either modify the network structure—potentially affecting its inherent performance—or require additional post-training, our approach addresses this issue directly from the pretraining perspective, which fundamentally avoids the network's lazy behavior.

Artifacts in self-supervised ViT. (DINO-type model [1]). Register [5] found that DINOv2 [20] leads to successful monocular depth estimation and semantic segmentation, but it loses the object detection capability of DINO [1] due to artifacts appearing on the feature map. To address this issue, additional tokens were introduced, designed to store global features and mitigate the impact of these artifacts. During our in-depth analysis of the high-norm phenomenon, we found that high-norm is merely a manifestation of the lazy behavior in later stages. Simply moving the high-norm tokens from the feature map to the register tokens does not fully address the underlying deficiencies in downstream tasks. Therefore, *Vision Transformer requires more than just Registers.*

3. Preliminary

3.1. Network Architecture: Vision Transformer

Vision Transformer [7] (ViT), a unified backbone for various vision tasks, employs a transformer-like [28] architecture over patches of the image. It first splits an image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ into non-overlapping patches and embed these patches as patch embeddings $\mathbf{x}_{\text{emb}} \in \mathbb{R}^{\frac{H}{C}\times\frac{W}{C}\times N}$ (C stands for the downsample ratio and N denotes hidden dimension) using a patch embedding module $\mathcal{P}_{\text{emb}}(\cdot)$, treating each patch as a token. After patchfy, several stacked transformer encoder layers $\mathcal{P}_{\text{enc}}(\cdot)$ are applied on image tokens. Within the transformer layer, tokens are updated by self-attention to capture meaningful local information. To aggregate global features, an additional `CLS` token or global average pooling is performed. The overall feature extraction is computed as:

$$\mathbf{x}_{\text{patch}} = \mathcal{P}_{\text{enc}}(\mathcal{P}_{\text{emb}}(\mathbf{x})), \mathcal{Q}_{\text{CLS}} = \textit{Pooling}(\mathbf{x}_{\text{patch}}), \quad (1)$$

$$\textit{or}$$

$$\mathbf{x}_{\text{patch}}, \mathcal{Q}_{\text{CLS}} = \mathcal{P}_{\text{enc}}(\mathcal{P}_{\text{emb}}(\mathbf{x}), \mathcal{O}_{\text{CLS}}), \quad (2)$$

where *Pooling* denotes global average-pooling in Eq. (1), and $\mathbf{x}_{\text{patch}} \in \mathbb{R}^{\frac{H}{C}\times\frac{W}{C}\times N}$ is final patch feature, $\mathcal{O}_{\text{CLS}}$ in Eq. (2) is a learnable query concatenated with patch embeddings to derive the global `CLS` token $\mathcal{Q}_{\text{CLS}}$.

4. Analysis and Hypothesis

We first propose a probing metric applicable to different pretraining approaches in Sec 4.1, along with the Point-in-Box benchmark, to provide a quantitative analysis of various pretraining strategies and their evolution during training. Based on the patch score, we first investigate from a spatial perspective which specific patches in a well-trained ViT yield high scores, and from a temporal perspective how the distribution of patch scores evolves over the course of training in Sec. 4.2. From the conclusions of Sec. 4.2, we hypothesize that both the sparsity of annotations and ViT's lazy behavior are underlying causes of the artifacts observed in the patch scores. Therefore, in Sec. 4.3 and Sec. 4.4, we validate our findings by systematically isolating these two factors. Notably, we present our investigation using fully supervised [26] experiments on ImageNet [6], the most controlled and extensively studied setting. Nevertheless, we have observed the same trends across other datasets [2, 23] and training settings [1, 12]. Due to space constraints, we present several intriguing conclusions, including the relationship between high-norm tokens [5] and artifacts, as well as why DINO-v1 [1] is an exception to the high-norm phenomenon, in the appendix. We strongly encourage readers to review it.

4.1. New Metric: Patch Score and Point-in-Box

Patch Score. To enable a unified exploration across different pretraining settings, we propose a new metric: Patch Score. Specifically, we employ the `CLS` features to compute the dot product similarity for each patch as follows:

$$\mathcal{S}_{\text{p}} = \mathbf{x}_{\text{patch}} \cdot \mathcal{Q}_{\text{CLS}}, \quad (3)$$

where patches with higher scores indicate stronger alignment with image-level information (the `CLS` token).

Point-in-Box benchmark. Building on the patch score, we assess artifacts by determining whether the highest scoring regions correspond to foreground objects. We use images from the ImageNet [6] validation set that feature a single object annotation to avoid ambiguity. We define the Point-in-Box score as the proportion of instances in which the highest patch score falls within the foreground bounding box.

4.2. Artifacts in Patch Score

Q: We wonder, where dose `CLS` token "look" at?

Experiment Setting: Given a ViT-B/16 [7] fully supervised on ImageNet [6] provided by *Torchvision* [19], we mask the corresponding high-score or low-score patches on the original input RGB image and then re-evaluate. We report both the classification performance and the difference (Δ).

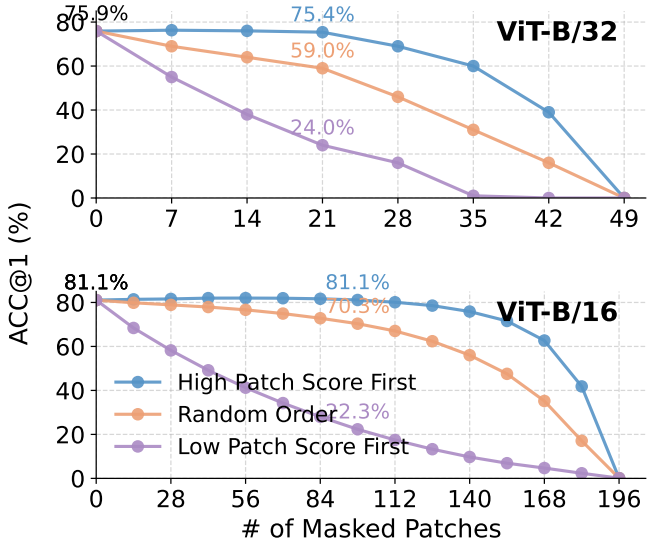

Figure 2. **Masked Image classification** on ImageNet. Although masking high-score patches removes more than half of the image, it surprisingly does not harm performance, whereas masking low-score patches does. This suggests that high-score patches primarily belong to the background and do not contribute to classification.

Experiment Results: As shown in Fig. 2, masking high-score patches not only avoids performance degradation but even improves it (+1.2% for ViT-B), with minimal loss even when over half the patches are masked. In contrast, masking low-score patches causes a greater performance drop, with the largest difference of 60% Acc observed at 70% masking.

Experiment Conclusion: Based on the experiment results:

- Patches with higher scores are more often located in semantically irrelevant background regions, while foreground patches tend to have lower scores.
- Commonly used datasets for pretraining ViTs are object-centered, with *a large portion of image patches being redundant or irrelevant.* As a result, masking half of the image patches does not affect classification.

Q: We wonder, when does this phenomenon begin?

Experiment Setting: We train ViT-B/16 [7] and ResNet-50 [10] on ImageNet using the same batch size, and report both classification accuracy and Point-in-Box score over training iterations.

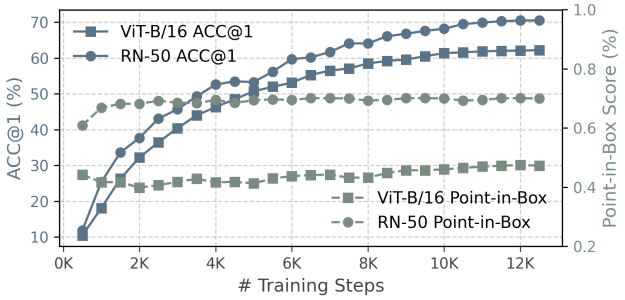

Figure 3. **Image classification and Point-in-Box score** on ImageNet-1k [6]. The left axis presents classification accuracy over training step, while the right axis displays the Point-in-Box score. ViT's Point-in-Box score is significantly lower than ResNet's from the outset, and it shows no improvement as training progresses.

Experiment Results: As shown in Fig. 3, while classification performance improves over training, the Point-in-Box score for ViT remains nearly unchanged—rising only slightly from 0.42 at 10% ACC@1 to 0.44 at 60% ACC@1. ViT model exhibit significantly lower scores than ResNet.

Experiment Conclusion: This confirms that artifacts in patch scores emerge during begining of training. It seems that ViT has been "looking" at the background regions from the very beginning. Given that irrelevant background regions occupy more than 50% of the image, does this imply that **the most convenient—and laziest—learning strategy for the model is to first diffuse the small foreground semantics into the majority background patches and then aggregate these background patch feature through the CLS token?** In the next two sections, we separately eliminate the effects of annotation sparsity (Sec. 4.3) and the model's ability to diffuse semantics (Sec. 4.4) to verify our hypothesis.

4.3. The Sparsity of Annotations

Validation Experiment Setting: To validate our hypothesis of annotation sparsity, we reduce annotation sparsity by enlarging the kernel of $\mathcal{P}_{\text{emb}}(\cdot)$—thereby lowering the proportion of background tokens even without ground truth. We train ViT-base with a notably large 28×28 kernel on ImageNet compared with default ViT-B/16.

Validation Experiment Results: As shown in Fig. 4, this adjustment leads to a modest improvement in the Point-in-Box score, increasing from 0.44 to 0.52. To better visualize the improvement, we additionally provide patch score visualizations, showing that the highest-scoring patches shift from the background to the foreground. However, the improvement is unstable and may degrade classification performance, dropping from 62% to 55%. This also explains why larger models suffer more [5], as ViT-B uses a 16×16 kernel, whereas ViT-L typically employs a 14×14 kernel.

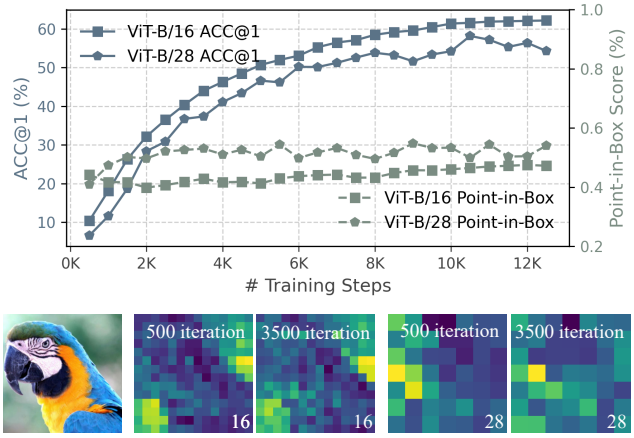

Figure 4. **Image classification and Point-in-Box score** on ImageNet-1k [6]. To better better understand the improvement, we visulize the patch score at two training step. By increasing the kernel size to boost the proportion of foreground patches, the artifacts in patch scores are alleviated.

4.4. Lazy Behavior from ViT's Global Dependencies

Validation Experiment Setting: To validate our hypothesis of model's ability, we progressively reduce the model's capacity for global representation by replacing the original attention mechanism with a window-based one [18] at each layer. We train ViT-Small [7] on ImageNet [6] for 50 epochs and report classification performance alongside the Point-in-Box score, as well as the layers and window sizes.

Window Index	Window Size	**IN1K**	**Point-in-Box**
None	None	72.3	50.1
1,5,9,11	4	71.7	52.1
All	4	63.9	59.8

Table 2. **Window attention experiment:** By reducing the model's long-range capability, the artifacts are alleviated.

Validation Experiment Results: As shown in Tab. 2, We observe that as the number of global attention layers in the network decreases, the points-in-box score increase significantly. Please note that the second row represents the default window attention setting [18], which results in only a slight improvement in the Point-in-Box score (+2%). The highest score is achieved when all attention layers are replaced with window-based attention. However, this comes at the cost of reduced classification performance. After an in-depth analysis, we propose a simple, straightforward, and unified hypothesis to explain the peculiar mechanisms in ViT pre-training:

Algorithm 1 LaSt-ViT Pytorch-type pseudo-code

```
h_dim = hidden_dim
M = number of patch we want to aggregate
class ViT(nn.Module):

  def gauss_k(self, kernel_size, sigma):
    # generate gaussian_kernel_1d
    x = torch.arange(-kernel_size // 2 + 1,
     kernel_size // 2 + 1)
    kernel = torch.exp(-0.5 * (x / sigma) ** 2)
    kernel = kernel / torch.max(kernel)
    return kernel

  def __call__(self, image):
    # Patchify, Add position embeddings::
    x = conv(image) + pe
    # Encoder
    x = TransformerEncoder(...)(x)
    x_fft = torch.fft(x, dim=-1)
    x_fft = x_fft * gauss_k(h_dim, h_dim ** 0.5)
    x_ifft = torch.ifft(x, dim=-1).real
    diff = torch.abs(x - x_ifft) / torch.abs(x)
    _, indices = torch.topk(diff, k=M, dim=1)
    sel_p = torch.gather(x, 1, indices)
    return torch.mean(sel_p, dim=1)
```

Notes: Changes to existing code highlighted via violet background.

> ***Takeaway***: *The sparsity of annotations, coupled with ViT's global dependencies, allows its lazy behavior to diffuse class-related image semantics to all redundant, semantically irrelevant background patches.*

5. Method

5.1. Motivation

Based on these observations and hypotheses, we propose a simple and general solution to prevent the lazy behavior in ViT: *Stop using the learnable `CLS` token to arbitrarily aggregate global information within self-attention.* Instead, foreground semantics should be locked. *When aggregating the CLS token, we use the Fourier transform to discard redundant tokens, ensuring that pooling is performed selectively on meaningful patch features.*

We want to highlight that: During our investigation, we also identify alternative methods to mitigate lazy behavior, such as max pooling. Once we understand how the problem arises, we can devise a wide range of solutions—after all, understanding the issue is already half the battle. In the following sections, we present the approach that proved to be the most robust with the best downstream performance, and compare it with other simple alternatives.

5.2. LaSt-ViT

In LaSt-ViT, during the final pooling stage, we first transform the patch features into the frequency domain and apply a high-pass filter to remove low-frequency components. The filtered features are then converted back to the original domain, where we measure their distance from the original features. Patch features with small distances—indicating

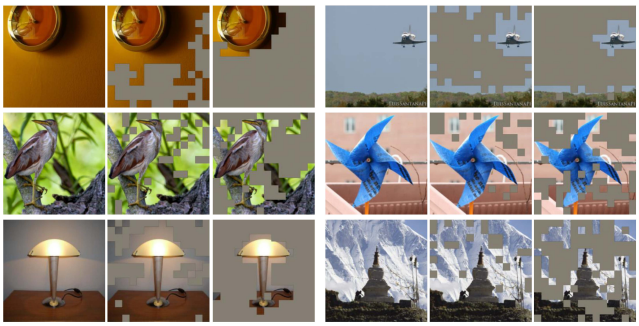

Figure 5. Selection results on ImageNet *validation* images: For each triple, we display the redundant tokens identified by LaSt-ViT, with varying numbers of selected tokens.

minimal information loss—are selected and aggregated into the CLS token. Specifically, the distance computation can be formulated as:

$$\mathbf{x}_{\text{FFT}} = \mathit{FFT}(\mathbf{x}_{\text{emb}}), \tag{6}$$

$$\mathbf{x}_{\text{FFT}} = \mathbf{x}_{\text{FFT}} * \mathit{Gaussian\_kernel}, \tag{7}$$

$$\mathbf{ix}_{\text{emb}} = \mathit{IFFT}(\mathbf{x}_{\text{FFT}}), \tag{8}$$

$$\mathcal{D} = \frac{\mathbf{x}_{\text{emb}}}{\mathit{abs}(\mathbf{ix}_{\text{emb}} - \mathbf{x}_{\text{emb}})}, \tag{9}$$

where $\mathit{FFT}(\cdot)$ denotes Fourier transform and $\mathit{IFFT}(\cdot)$ denotes inverse Fourier transform, $\mathit{abs}(\cdot)$ represents absolute value, *Gaussian_kernel* denotes a high-pass filter with gaussian kernel. After the distance score $\mathcal{D}$ is obtained, LaSt-ViT performs average pooling only on the patches with the lowest M distance scores, while discarding the others. Notably, LaSt-ViT introduces only minimal modifications to ViT. As a result, our method can be easily integrated into many ViT variants [1, 20, 25, 26, 32]. The detailed Pytorch-type pseudo-code is provided in Alg. 1.

5.3. Transfer to Downstream Tasks

In this section, we provide further details and explain how each downstream task is conducted.

Unsupervised Object Discovery. Since LazyStrike guides the CLS token to focus on foreground objects, as shown in Fig. 5, we can achieve unsupervised object localization using patch scores. *This expansion is independent of the training method—typically a privilege of self-supervised approaches like DINO in earlier works—**for the first time allowing any training objective to accomplish this.*** We construct the mask by applying a threshold defined as the mean score plus one standard deviation. Patches with scores above this threshold are classified as foreground.

Zero-shot Open-Vocabulary Tasks. Since LazyStrike ensures that the CLS feature aggregates information from the correct patch features, and the CLS feature itself is directly supervised by the learning signal, this effectively leads to an indirect alignment between patch features and the supervision signal. For text-supervised ViTs, we can obtain zero-shot semantic segmentation results by computing the similarity between patch features and arbitrary text features, thereby enabling applications across various open-vocabulary tasks.

Method	High Norm	Points-in-Box
ResNet [10]	✗	68.4
ViT [7]	✓	42.7
ViT (+LazyStrike)	✗	55.1 (+12.4)
DINO-ResNet [1]	✗	71.1
DINO-v1 [1]	✗	44.5
DINO-v1 (+LazyStrike)	✗	69.7 (+25.2)
CLIP-ResNet [22]	✗	53.9
CLIP [22]	✓	39.8
CLIP (+LazyStrike)	✗	50.1 (+10.3)

Table 3. Evaluation of the LazyStrike in Points-in-Box score.

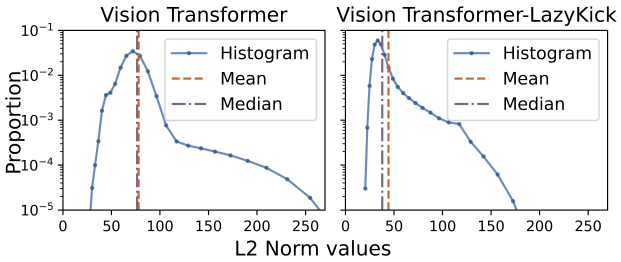

Figure 6. Evaluation of the LaSt-ViT in feature norm. Specifically, the elimination of artifacts also removes the high-norm phenomena [5], highlighting our deeper perspective on addressing artifacts.

6. Experiment

6.1. Experiment Settings

We first verify the elimination of artifacts in patch score (Sec. 6.2) and validate our proposed method on three training methods: fully supervised (Sec. 6.3), text-supervised (Sec. 6.4), and self-supervised (Sec. 6.5), and examine multiple downstream tasks for ViT under different supervision, including object discovery [1, 24], zero-shot semantic segmentation [22, 25, 32], open-vocabulary object detection [13], instance segmentation [30] and coarse segmentation [1].

Implementation Details. Without additional specification, we set M to half the number of image tokens (for example, with an input of 224, ViT-16 produces 196 tokens, hence $M = 98$ in this case) in Alg. 1, which aligns with our finding that over half of the patches in an image are redundant.

6.2. Artifact Elimination

Elimination of artifacts in feature norm and patch score. Tab. 3 presents the results under different training methods, demonstrating that LazyStrike not only eliminates the

Model	Backbone	COCO-Obj.	ADE20K	City.	VOC20	Context59	COCO-Stf.
CLIP [22]	ViT-B/16	8.8	3.1	6.5	49.0	11.2	7.2
CLIP (+LazyStrike)	ViT-B/16	13.3 (+4.5)	8.3 (+5.2)	12.1 (+5.6)	75.0 (+26.0)	15.2 (+4.0)	11.8 (+4.6)
MetaCLIP [32]	ViT-B/16	4.8	2.9	5.8	39.6	9.3	6.2
MetaCLIP (+LazyStrike)	ViT-B/16	14.1 (+9.3)	7.9 (+5.0)	11.1 (+5.3)	72.8 (+33.2)	15.5 (+6.2)	12.0 (+5.8)
EVACLIP [25]	ViT-B/16	15.0	6.7	12.2	56.5	14.1	9.7
EVACLIP (+LazyStrike)	ViT-B/16	**26.2** (+11.2)	**14.8** (+8.1)	**24.5** (+12.3)	**79.6** (+23.1)	**24.7** (+10.6)	**18.3** (+8.6)
CLIP [22]	ViT-L/14	3.0	1.6	2.7	17.1	5.1	3.2
CLIP (+LazyStrike)	ViT-L/14	15.0 (+12.0)	8.4 (+6.8)	12.3 (+9.6)	72.4 (+55.3)	15.1 (+10.0)	11.9 (+8.7)
MetaCLIP [32]	ViT-L/14	5.0	3.3	6.2	25.7	8.9	6.1
MetaCLIP (+LazyStrike)	ViT-L/14	13.9 (+8.9)	9.2 (+5.9)	13.9 (+7.7)	75.6 (+49.9)	16.0 (+7.1)	12.5 (+6.4)
EVACLIP [25]	ViT-L/14	15.7	8.4	13.8	53.8	16.6	10.1
EVACLIP (+LazyStrike)	ViT-L/14	**24.0** (+8.3)	**11.3** (+2.9)	**17.7** (+3.9)	**76.4** (+22.6)	**21.7** (+5.1)	**14.8** (+4.7)

Table 4. Evaluation results (mIoU, %) on six **semantic segmentation benchmarks**. Our results are marked in gray. LazyStrike consistently improves semantic segmentation results under text supervision across different type of CLIP [22] and model sizes, demonstrating that, after understanding the essence of the problem, a simple approach can uniformly address issues across different models.

Method	Backbone	COCO Detection			LVIS Segmentation			
		$AP50^{box}$	$AP50^{box}_{base}$	$AP50^{box}_{novel}$	AP^{mask}	AP^{mask}_{freq}	AP^{mask}_{comm}	AP^{mask}_{novel}
ConvNet based								
F-VLM [13]	RN50	39.6	/	28.0	24.2	26.9	24.0	18.6
F-VLM [13]	RN50x64	/	/	/	34.9	/	/	32.8
ViT based								
F-ViT [30]	ViT-B/16	34.9	41.0	17.5	15.4	20.6	12.3	11.5
F-ViT (+LazyStrike)	ViT-B/16	45.7 (+10.8)	50.1 (+11.1)	33.3 (+15.8)	21.7 (+6.3)	25.2 (+4.6)	18.0 (+5.7)	22.8 (+11.3)
F-ViT [30]	ViT-L/14	46.0	53.6	24.7	28.7	31.5	27.9	24.2
F-ViT (+LazyStrike)	ViT-L/14	**53.2** (+7.2)	**68.2** (+14.6)	**39.1** (+14.4)	**34.3** (+5.4)	**35.1** (+3.6)	**34.4** (+6.6)	**32.1** (+6.6)

Table 5. Evaluation results on **open-vocabulary benchmark**. Our results are marked in gray. LazyStrike consistently enhances performance on open-vocabulary dense tasks, by demonstrating that frozen ViT can achieve comparable performance with ConvNet [10].

Model	Train	mIoU
ViT-B/16	*Supervised*	22.3
ViT-B/16 (+LazyStrike)	*Supervised*	32.8 (+10.5)
ViT-S/16	*Supervised*	29.5
ViT-S/16 (+LazyStrike)	*Supervised*	41.9 (+12.4)
ViT-S/16	*DINO*	47.7
ViT-S/16 (+LazyStrike)	*DINO*	55.1 (+7.4)

Table 6. **Coarse segmentation** via patch score. We follow [33] to conduct coarse segmentation on VOC12. With LazyStrike, ViT under label-supervision also appears emergence of segmentation.

high-norm phenomenon but also enhances Point-in-Box score. With LazyStrike applied, ViT's Point-in-Box score approaches that of ResNet [10]. Fig. 6 provides a detailed analysis of feature norms under fully supervised training [26], revealing that LazyStrike reduces the maximum feature values, thereby mitigating the high-norm phenomenon.

6.3. Fully-Supervised Comparison

Emergence of Coarse Segmentation. Following [1], we evaluate emerging properties, a phenomenon only appears in self-supervised training before, on the validation set of VOC12. As shown in Tab. 6, our method consistently improves emerging properties across different model sizes and training methods. Notably, our approach achieves performance close to DINO in the supervised setting (41.9% vs. 47.7%), *demonstrating that LazyStrike prompts emerging properties and those are not exclusive to self-supervised.*

Emergence of PCA. As shown in Fig. 7, we compute the PCA of the patch features from LaSt-ViT and visualize the first three components for the foreground. FocusLock refines the previously entangled PCA features, *effectively distinguishing and highlighting the salient foreground.*

6.4. Weakly-Supervised Comparison

Zero-shot Semantic Segmentation benchmarks. Tab. 4 illustrates our proposed method against several baseline models on six semantic segmentation benchmarks. The improvements achieved by integrating our modifications into these models are highlighted in blue. Our method consistently outperforms the baseline models across all evaluated benchmarks, demonstrating significant gains. For instance, when applied to the CLIP [22] model with ViT-B/16 architecture, our method achieves a substantial increase in mIoU on the

Method	FPS	VOC07	VOC12	COCO
SS [27]	-	18.8	20.9	16.0
EdgeBoxes [37]	-	31.1	31.6	28.8
DINO-seg [1]	29.4	45.8	46.2	42.1
LOST [24]	29.4	61.9	64.0	50.7
DINO (+LazyStrike)	**55.9**	**64.4**	**67.6**	**51.6**

Table 7. **Object discovery CorLoc**. All models adopt ViT-S. Previous best-performing methods relied on eigenvector computations, whereas LazyStrike avoids such heavy computational demands.

Figure 7. **Visualization of PCA components.** We compute the PCA of the patch features and visualize the first 3 components for the foreground object. With LazyStrike, ViT under label-supervision also shows the emergence of PCA, which helps distinguish foreground from background and separate object parts, enhancing feature representation.

Pascal (from 11.2% to 15.2%), Cityscapes (from 6.5% to 12.1%), and VOC (from 49.0% to 75.0%). When scaled up to the larger ViT-L architecture, our method continues to deliver remarkable results. For the CLIP model, the mIoU on VOC jumps from 17.1% to an impressive 72.4%, and on Cityscapes, it increases from 2.7% to 12.3%. In summary, integrating our method into the baseline models results in significant improvements across all benchmarks, demonstrating its robustness and effectiveness across various CLIP models and models of different sizes.

Open-vocabulary Object Detection and Segmentation benchmarks. As shown in Tab. 5, We choose F-VLM [13] and F-ViT [30] as baselines. Both methods use a **frozen CLIP [25]** as the backbone for object detection and instance segmentation. After obtaining the region of interest, they weigh the semantic scores of the corresponding area to determine the object class scores. The only difference is that F-VLM uses a ConvNet-based backbone, while F-ViT employs a ViT-based backbone. For OV-COCO, LaSt-ViT achieves a gain of 15.8% and 14.4% over the baseline on the novel category for ViT-B and ViT-L, respectively. For OV-LVIS, it also improves the baseline by 11.3% and 6.6% over the rare category for ViT-B and ViT-L.

6.5. Self-Supervised Comparison

Unsupervised Object Discovery. We adopt DINO-seg [1] and LOST [24] as baselines for comparison, both utilizing ViT-S [7] as the backbone for object discovery tasks. The comparisons are illustrated in Tab. 7. LaSt-ViT exhibits significant performance improvements. Specifically, our model achieves the highest CorLoc scores across all datasets, surpassing both DINO-seg and LOST models. Notably, our model attains a CorLoc score of 64.4% on VOC 2007, 67.6% on VOC 2012, and 51.6% on COCO, representing improvements of 2.7%, 3.6%, and 0.9% points, respectively, over the best-performing LOST model. Moreover, our method demonstrates a remarkable throughput of 55.9 images per second. This indicates that our model achieves superior object discovery performance and operates more efficiently, making it highly suitable for practical applications.

Dataset →	ImageNet-1k		COCO	
(a). Full-sup.	*Acc#1*	*Acc#5*	*Det.*	*Seg.*
DeiT	81.0	95.3	47.6	42.4
+DynamicViT	81.4	95.4	41.2	37.1
+Ours	81.7	95.4	47.4	42.4
Dataset →	ImageNet-1k		COCO	
(b). Self-sup.	*KNN*	*Linear*	*Det.*	*Seg.*
DINO	74.5	77.0	50.3	44.9
+DynamicViT	66.4	68.2	42.7	39.9
+Ours	74.7	76.4	50.6	45.3

Table 8. Lazy behavior in ViTs does not harm classification accuracy but limits their use as general feature extractors. Our method, LazyStrike, addresses this while preserving versatile features for tasks like detection and segmentation. In contrast, token pruning improves classification efficiency but sacrifices generality for different tasks and adaptability for different pretraining methods, making it ineffective for general feature extractor.

Method	IN1K [6]	VOC [8]	COCO [8]
Attention-Pool	55.8	10.7	3.3
Max-Pool	53.1	71.9	12.2
w/ LazyStrike			
M = 1	53.5	72.7	13.5
M = 49	55.8	75.8	**18.5**
M = 98	**56.2**	**75.9**	18.0
M = 196 (Full)	55.3	13.5	4.8

Table 9. **Ablation study** on text-supervised ViT [12]. We report ImageNet classification and downstream semantic segmentation results, where LazyStrike significantly addresses the artifact issue and even leads to a improvement in classification.

6.6. Ablation study

Other method to alleviate artifacts. In Tab. 9, we compare Maxpool, a method that can also drastically reduce redundancy to mitigate artifacts. While it reduces the artifact phenomenon and improves ViT's semantic segmentation performance, it leads to a loss of important feature details, resulting in diminished performance in classification and

Method	IN1K [6]	VOC07 [8]	VOC12 [8]
Attention-Pool	59.1	14.1	28.7
Mean-Pool	64.3	15.3	29.6
w/ LazyStrike			
M = 1	64.6	30.4	35.6
M = 7	64.8	32.1	37.6
M = 49 (Full)	**64.9**	15.8	30.3

Table 10. **Ablations stduy** on label-supervised ViT [26]. We report ImageNet classification performance and downstream object location results, where LazyStrike significantly addresses the artifact.

other downstream tasks.

Number of cutted tokens. In Tab. 9, we examine the impact of the number of dropped tokens by training OpenCLIP [12] ViT-B/16 with different M. Performance improves significantly with LazyStrike, peaking when half of the tokens are selected. Tab. 10 shows further ablation studies on label-supervised ViT-B/32, with pretraining on ImageNet-1k and classification performance and CorLoc results.

7. Conclusion

In this work, we first introduce a unified probing metric to uncover the root cause of artifacts in the Vision Transformer. We reveal that ViT lazily adopts semantically irrelevant local patches as shortcuts to encode global image semantics. Based on these findings, we propose drop half of the image patches to prevent ViT's lazy behavior, achieving strong performance across 12 benchmarks. Our work provides a new baseline for future research on ViT.

References

[1] 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 International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3, 6, 7, 8

[2] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 3

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 1

[5] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 1, 2, 3, 4, 6

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 3, 4, 5, 8, 9

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 3, 4, 5, 6, 8

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 8, 9

[9] Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. *arXiv preprint arXiv:2404.08181*, 2024. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4, 6, 7

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *CVPR*, 2017. 1

[12] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 2, 3, 8, 9

[13] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. *arXiv preprint arXiv:2209.15639*, 2022. 6, 7, 8

[14] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. *arXiv preprint arXiv:2408.04883*, 2024. 3

[15] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 3

[16] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks, 2023. 3

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 5

[19] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision, 2016. 4

[20] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal,

Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1, 3, 6

[21] Bill Psomas, Ioannis Kakogeorgiou, Konstantinos Karantzalos, and Yannis Avrithis. Keep it simpool: Who said supervised transformers suffer from attention deficit? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5350–5360, 2023. 1

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 6, 7

[23] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 3

[24] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021. 1, 2, 6, 8

[25] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 6, 7, 8

[26] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2, 3, 6, 7, 9

[27] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171, 2013. 8

[28] 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. 3

[29] Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. In *European Conference on Computer Vision*, pages 315–332. Springer, 2025. 2, 3

[30] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. Clipself: Vision transformer distills itself for open-vocabulary dense prediction. *arXiv preprint arXiv:2310.01403*, 2023. 1, 3, 6, 7, 8

[31] Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks. *arXiv preprint arXiv:2312.12359*, 2023. 3

[32] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023. 6, 7

[33] Yaodong Yu, Tianzhe Chu, Shengbang Tong, Ziyang Wu, Druv Pai, Sam Buchanan, and Yi Ma. Emergence of segmentation with minimalistic white-box transformers. In *Conference on Parsimony and Learning*, pages 72–93. PMLR, 2024. 7

[34] Dengke Zhang, Fagui Liu, and Quan Tang. Corrclip: Reconstructing correlations in clip with off-the-shelf foundation models for open-vocabulary semantic segmentation. *arXiv preprint arXiv:2411.10086*, 2024. 3

[35] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 1, 2

[36] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 2, 3

[37] C Lawrence Zitnick and Piotr Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*. Springer, 2014. 8