
Vision Transformers with Self-Distilled Registers

Zipeng Yan^{*1} Yinjie Chen^{*1,2} Chong Zhou³ Bo Dai¹ Andrew F. Luo^{†1}

¹ University of Hong Kong ² Zhejiang University ³ Nanyang Technological University

^{*} Equal contribution – Co-First authors

maxwellcaffrey915@gmail.com, kelvinyzp@gmail.com [†]Corresponding author: aluo@hku.hk

Abstract

Vision Transformers (ViTs) have emerged as the dominant architecture for visual processing tasks, demonstrating excellent scalability with increased training data and model size. However, recent work has identified the emergence of artifact tokens in ViTs that are incongruous with local semantics. These anomalous tokens degrade ViT performance in tasks that require fine-grained localization or structural coherence. An effective mitigation of this issue is the addition of register tokens to ViTs, which implicitly “absorb” the artifact term during training. Given the availability of existing large-scale pre-trained ViTs, in this paper we seek to add register tokens to existing models without retraining the models from scratch, which is infeasible considering their size. Specifically, we propose Post Hoc Registers (**PH-Reg**), an efficient self-distillation method that integrates registers into an existing ViT without requiring additional labeled data and full retraining. PH-Reg initializes both teacher and student networks from the same pre-trained ViT. The teacher remains frozen and unmodified, while the student is augmented with randomly initialized register tokens. By applying test-time augmentation to the teacher’s inputs, we generate denoised dense embeddings free of artifacts, which are then used to optimize only a small subset of unlocked student weights. We show that our approach can effectively reduce the number of artifact tokens, improving the segmentation and depth prediction of the student ViT under zero-shot and linear probing. Our code is publicly available at [this repository](#).

1 Introduction

Vision Transformers (ViTs) are now the dominant architecture in visual modeling, delivering strong performance across classification, detection, and segmentation. Unlike convolutional networks with their built-in locality inductive bias, ViTs process images by spatially splitting them into patches and applying self-attention to enable global feature interactions. This architectural design leads to superior scalability, particularly with contrastive or self-supervised pre-training objectives, and facilitates more flexible representation learning, as it is less constrained by the translation invariance assumptions inherent in CNNs. This flexibility enables remarkable emergent capabilities. Models like CLIP, trained solely on image-text alignment, achieve competitive open-vocabulary segmentation through zero-shot dense queries; while self-supervised approaches learn semantically rich features directly from unlabeled images.

However, the same data-driven attention mechanisms that enable ViT’s representation power can also lead to the emergence of *artifact tokens*. These are outlier features often discordant with local image semantics, meaning they fail to correspond to locally meaningful image structures. The propensity for ViTs to generate such tokens is exacerbated by their lack of strong, built-in spatial priors, which can result in inconsistent dense representations. Ultimately, the presence of these artifact tokens

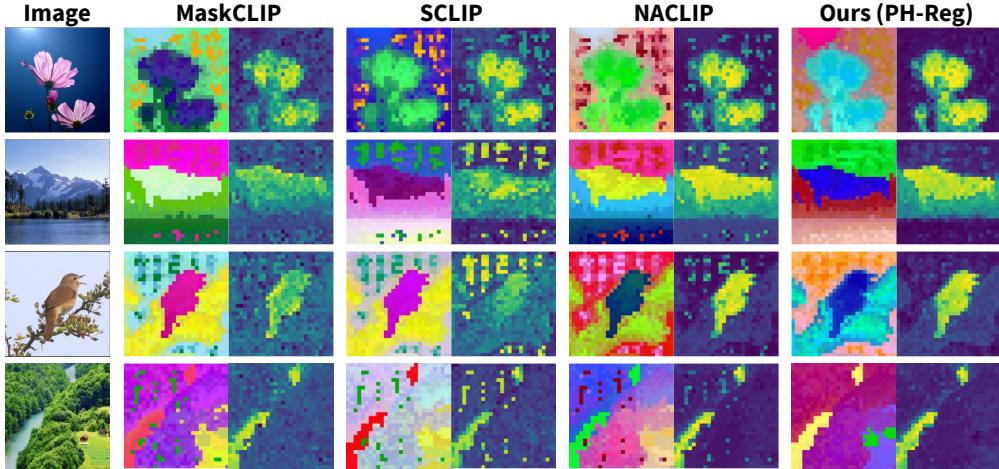


Figure 1: **Effect of PH-Reg on Open-vocabulary Segmentation.** For each image, we compare four methods: MaskCLIP which directly takes the *value* features from the last attention layer; SCLIP which adds correlative self-attention; NAACLIP which further enforces a locality bias; and our PH-Reg method with self-distilled registers. We utilize the same OpenAI CLIP ViT-B/16 weights for all three methods. For each method, we visualize the UMAP of the dense features and a heatmap of one text query. Our method yields noticeably cleaner dense features and high quality localizations, and requires only a small set of additional register parameters compared to the original network.

disrupts fine-grained localization, a critical capability for tasks demanding high spatial precision, such as detailed semantic segmentation or part identification.

Recent work has sought to mitigate artifact tokens via architectural modifications, where *register tokens* are added to the network. These register tokens are randomly initialized, with learnable parameters that participate in the self-attention process similar to the [CLS] token, but are not otherwise used during the output. Although these register tokens are not explicitly supervised during training, they effectively “absorb” the artifact term and learn to attend to global objects. While effective, introducing register tokens constitutes a fundamental architectural modification that requires training from scratch—a time-consuming and computationally demanding process. This significantly limits their applicability, especially given the vast ecosystem of *existing, high-performing pre-trained vision models*.

We present a solution to this issue with **Post Hoc Registers (PH-Reg)**, an efficient self-distillation framework that requires no labeled data or full retraining. We illustrate OpenAI CLIP with PH-Reg in Figure 1. In PH-Reg, both teacher and student networks are initialized from the same pre-trained model weights. And the only extra parameters are the register tokens added to the student network. Our proposed framework freezes the teacher during training. Images provided to the teacher undergo test-time augmentation (e.g. random offsets and horizontal flips). This augmentation strategy effectively denoises the teacher’s dense features without requiring gradient-based updates on the teacher itself, yielding stable dense targets. The denoised dense features are used as a distillation target for the student network, where only a small set of parameters are optimized. This entire process requires only a modest set of unlabeled images, enabling significant enhancements to pre-trained models with minimal computational overhead. Concretely our contributions are as follows: **1.** We propose a test-time augmentation scheme that can effectively denoise dense features in vision transformers. Our denoiser does not require costly neural fields and does not require gradient based optimization. **2.** We elucidate the underlying components in a student model that contribute to learning a clean dense feature map. We show that by finetuning select weights, we are able to achieve clean dense features with minimal additional parameters. **3.** We demonstrate that PH-Reg effectively improves the consistency of dense feature representations in ViTs, leading to quantifiable improvements on downstream tasks that rely on fine-grained spatial understanding (e.g., semantic segmentation or depth prediction). Our method preserves the original utility of dense features without inducing unwanted distribution shift, and functions well with zero-shot language-based dense queries.

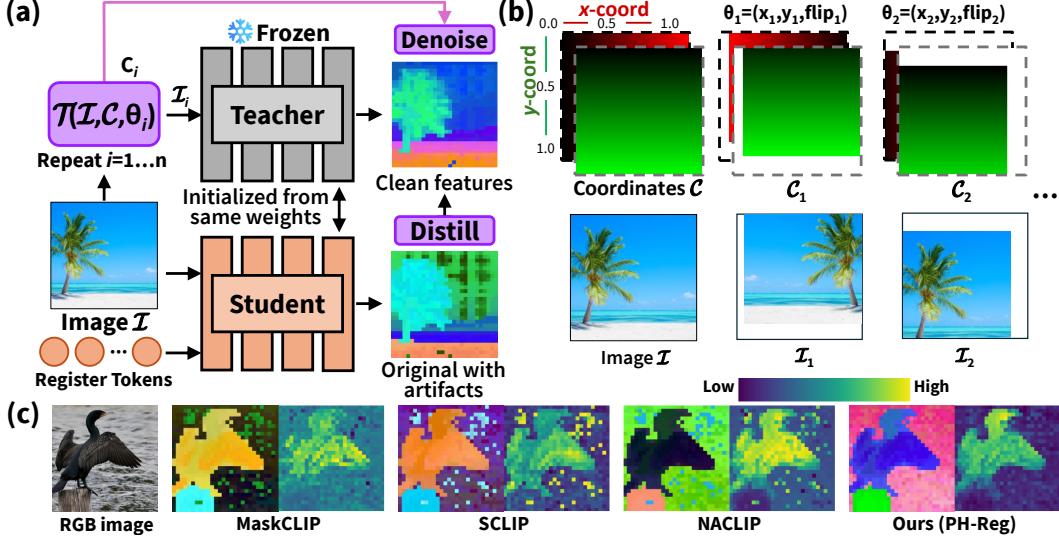


Figure 2: **Learning Framework of PH-Reg.** (a) Our framework begins by creating two networks from the same set of weights. In the teacher, the weights are frozen and unmodified. In the student, the only additional parameters are learnable register tokens. The teacher creates a learning target using denoised representations. (b) An image \mathcal{I} undergoes augmentation by function \mathcal{T} with random augmentation parameters consisting of random offsets and horizontal flips. (c) Given an RGB image, we utilize UMAP to visualize the features, and a heatmap using CLIP text query. Our method can produce significantly cleaner dense representations with minimal additional inference cost.

2 Related Work

Transformers in Visual Learning. Building upon the success of self-attention in language modeling, architectures that leverage transformer based token-mixing have been proposed for visual generation [1, 2, 3] and recognition tasks [4, 5], cumulating in the ViT architecture which relies on very few locality biases [6]. In the years since, many improvements and variants have been proposed [7, 8, 9, 10, 11, 12]. The improvements have largely focused on data [13] and compute efficiency [14, 15, 16, 17, 18, 19, 20, 21, 22]. In general, vision transformers tokenize an image into a set of patches, where each patch is first processed using an MLP or convolution block [23, 24, 25, 26], the patches are further processed with self-attention which enables global token interactions beyond those in convolutional networks. As self-attention is permutation invariant, positional information is typically injected using learnable positional embeddings or relative positions [27, 28, 29, 16, 30, 31, 32]. Positional embeddings have been suggested to play a role in the emergence of dense ViT artifacts, as networks with positional embeddings removed have smooth feature maps [33].

Representation Learning with Vision Transformers. The lack of restrictive local inductive biases in Vision Transformers enables strong scaling behavior across a diverse set of tasks. Beyond traditional supervised learning on categorical datasets such as ImageNet, methods have been proposed to learn on large scale datasets by leveraging language contrastive objectives [34, 35], or self-supervised image-level objectives [36, 37, 38] and patch-level objectives [39, 40, 41, 42, 43, 44, 45, 46]. While the training objectives are very different, these methods enable strong zero-shot and linear-probe performance across a diverse set of tasks, suggesting that these methods effectively learn the *underlying statistics* of visual input. While these methods lack explicit dense supervision, the dense features from these models have been shown to have strong zero-shot emergent behavior with language-based segmentation [47], object part correspondence [48], and structural understanding [49].

Artifacts in Vision Transformer Dense Features. Recent work on DINOv2 [46] has found that Vision Transformers can have artifacts in their dense features. It has been proposed that artifacts can be mitigated with ‘‘register’’ tokens [50, 51, 52, 53]. These register tokens are effectively randomly initialized embeddings that are analogous to the [CLS] token. While registers participate in the self-attention process, they are discarded during output. This approach requires a model to be trained from scratch. The nature and the mechanisms that cause the emergence of artifact tokens are unclear, and there exists conflicting results on what information (global or no information) these artifact tokens contain [50, 54]. Recent work has further investigated the mechanism of register

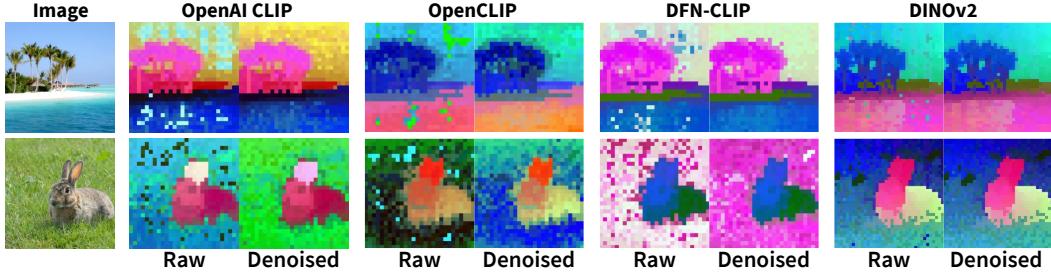


Figure 3: **Denoising Teacher Representations with Augmentations.** For each model, we visualize the UMAP of dense features before and after applying test-time augmentation. The results show that our proposed method produces noticeably cleaner dense feature representations without requiring gradient-based learning. Please zoom in for details.

tokens [55]. Our own results have found that artifact tokens are not necessarily high-norm, and can be low-norm as well. Unlike the observation by [33], we find that positional embeddings alone cannot account fully for the artifacts. Regardless of “why” artifact tokens emerge, removing these artifacts is an active area of research, with proposals based on registers [50], magnitude smoothness priors [54], and the foundational work on leveraging neural fields to denoise ViTs with a static artifact component [33, 56, 57]. A concurrent line of work has sought to remove artifacts for open-vocabulary segmentation with training-free attention modifications [47, 58, 59, 60, 61, 62]. Our framework can be applied to existing pretrained networks, introduces minimal additional parameters, can be applied to tasks beyond open-vocabulary segmentation, and makes no assumptions on the magnitude or static nature of the artifacts.

3 Methods

In this section, we will describe the PH-Reg framework, which we illustrate in Figure 2. This framework enables *existing pretrained ViTs* to benefit from register tokens, yielding significantly cleaner dense representations. During training, PH-Reg requires only unlabeled images for the self-distillation process. In section 3.1, we will first describe the denoising process of teacher network outputs. Unlike prior work that rely on a neural field/hash-grid, this method denoises dense features without the use of expensive gradient-based learning. In section 3.2, we will describe how we initialize and modify the student architecture. This approach only introduces a small set of additional parameters to the network. Finally in section 3.3 we will describe our distillation process.

3.1 Efficient Denoising of Teacher Representations

Our denoising process starts from the observation that artifact tokens are not static relative to image content. Put another way, if an image is shifted by a certain amount (with the gaps padded with whitespace), the artifacts do not shift by the same amount. As shown in Figure 2, given an RGB image $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$, we randomly sample n random augmentation parameters $(\theta_1, \theta_2, \dots, \theta_n)$, where each θ_i defines a horizontal/vertical offset (x_i, y_i) and boolean $\text{flip}_i \in \{0, 1\}$ defined horizontally. For each image, we also compute the image space coordinate grid $\mathcal{C} = (\text{x-coords}, \text{y-coords})$. Where (x, y) are respectively in range $[0, 1]$: x defines the left-right axis, and y defines the top-down axis. The coordinates \mathcal{C} help us keep track of the original location of an image region after augmentation. In practice, as we are working with a ViT model with patch size $k \times k$, where the tokenization process yields a $(\frac{H}{k}, \frac{W}{k})$ grid of image tokens, we define our parameters using offsets that are integer multiples of k to facilitate efficient indexing. Together, the image \mathcal{I} , the coordinates \mathcal{C} , and augmentation parameter θ_i are provided to transform function to yield an augmented image \mathcal{I}_i and new coordinates \mathcal{C}_i : $\mathcal{T}(\mathcal{I}, \mathcal{C}, \theta_i) \Rightarrow (\mathcal{I}_i, \mathcal{C}_i)$.

Algorithm 1 Denoising Process

Input: Image $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$;
 Image space coordinates $\mathcal{C} \in [0, 1] \times [0, 1]$;
 Augmentation parameters $\theta_1, \theta_2, \dots, \theta_n$;
 Augmentation function \mathcal{T} ;
 ViT teacher model f_{teacher} ;
 1. Zero init clean feature tensor Q
 2. Zero init count tensor K
 3. **For** i in $\{1, \dots, n\}$:
 4. $\theta_i = (x_i, y_i, \text{flip}_i)$
 5. $(\mathcal{I}_i, \mathcal{C}_i) = \mathcal{T}(\mathcal{I}, \mathcal{C}, \theta_i)$
 6. Dense feature $F_i = f_{\text{teacher}}(\mathcal{I}_i)$
 7. $(F_i^{\text{valid}}, \mathcal{C}_i^{\text{valid}}) = \mathcal{T}^{-1}(F_i, \mathcal{C}_i, \theta_i)$
 8. $Q[\mathcal{C}_i^{\text{valid}}] = Q[\mathcal{C}_i^{\text{valid}}] + F_i^{\text{valid}}$
 9. $K[\mathcal{C}_i^{\text{valid}}] = K[\mathcal{C}_i^{\text{valid}}] + 1$
 10. **return** Q/K

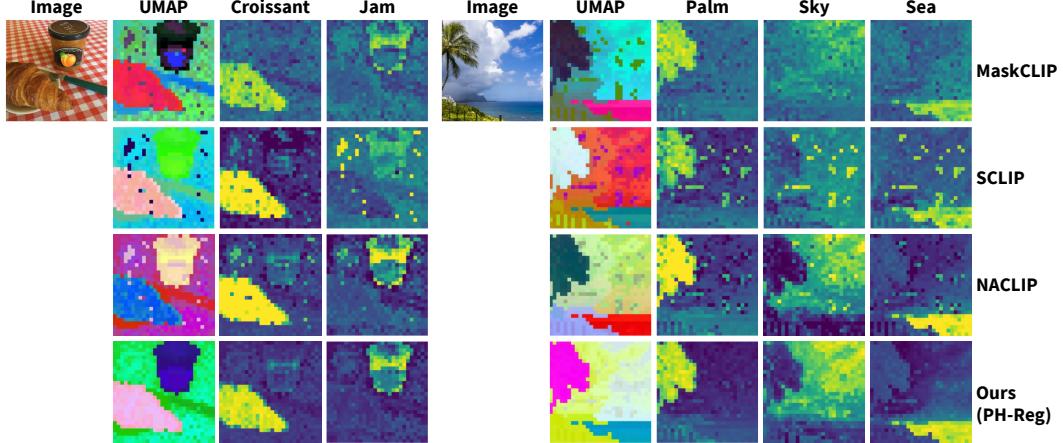


Figure 4: **Visualization of Open-vocabulary Semantic Segmentation.** We compare against MaskCLIP, SCLIP, NAACLIP, and find that our method yields clean feature maps free of artifacts.

A teacher model f_{teacher} is based on a frozen set of original network weights, without any additional parameters. Given an augmented image \mathcal{I}_i , this network outputs feature F_i . We restore the features to the original location within an image using the inverse transform function $\mathcal{T}^{-1}(F_i, \mathcal{C}_i)$. The restored features are additively accumulated across different augmentation parameters, while keeping track of the number of occurrences for each location. At the end, the dimension-wise sample mean is computed for the accumulated features. We present our full denoising process in Algorithm 1. The patch-wise expected value of this representation is the same as the optimal value when optimizing a discrete grid of representations to minimize mean squared error (as used in DVT [33] and traditional neural field based methods). However, as we do not require gradients, this denoising process can be done in less than $200ms$, roughly two magnitudes faster than neural field based denoising in DVT. The comparison between raw and denoised dense feature visualizations is shown in Figure 3.

3.2 Design of the Student Network

Our objective is to preserve maximal computational efficiency of the student model, while leveraging the knowledge of the pre-trained weights. For this purpose we introduce m number of register tokens, providing a minimally invasive enhancement to the base architecture. After the addition of register tokens, a total of $m + 1 + \frac{H}{k} \times \frac{W}{k}$ tokens participate in the self-attention process. Unlike prior work that trains registers from scratch [50], this approach updates only these registers and selectively unfreezes specific components during distillation, preserving the majority of the ViT's pretrained weights. Through ablation studies, we identify optimal unfreezing strategies, such as adjusting convolution layers, positional embeddings, or the last transformer block.

3.3 Learning and Optimization of the Student

We employ a multi-objective distillation strategy, combining cosine similarity and mean squared error losses to ensure both directional and magnitude alignment between teacher representations F_{teacher} and student representations F_{student} . Our final loss is: $\text{Loss}_{\text{total}} = 1 - \text{cossim}(\text{target}, \text{predicted}) + \text{MSE}(\text{target}, \text{predicted})$.

4 Experiments

In this section, we comprehensively evaluate the performance of PH-Reg on a diverse set of dense tasks, first using a zero-shot setup for open-vocabulary segmentation in section 4.1, followed by linear probe based segmentation and depth tasks in section 4.2. Finally we perform ablation studies to explore design decisions and investigate the nature of artifacts across different models in section 4.3. All implementation details are provided in the appendix.

4.1 Open-vocabulary Semantic Segmentation Evaluation

Datasets. In this section, we follow prior works [59, 62, 61] to evaluate our approach on six semantic segmentation datasets, with their names abbreviated (in parentheses) to conserve table space: PASCAL

Table 1: **Open-vocabulary Semantic Segmentation Quantitative Evaluation Results on 8 Benchmarks.** While the first 3 benchmarks (VOC21, PC60 and Obejct) include a background class, the remaining benchmarks do not. We report the mean Intersection over Union (mIoU, %) metric, where higher values indicate better performance, for our method and all baseline models. The best result for each dataset is highlighted in **bolded**. Additional results are provided in the supplementary material.

Method	VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	Avg.
CLIP [34]	18.60	7.84	6.50	49.05	11.17	7.19	6.65	3.16	13.77
MaskCLIP [47]	49.27	25.46	26.94	66.56	28.62	18.80	28.33	13.70	32.21
SCLIP [59]	59.62	31.74	33.52	81.53	34.46	22.65	32.34	16.45	40.08
ClearCLIP [61]	59.76	32.56	32.77	84.56	35.91	23.89	30.04	16.65	39.52
NACLIP [62]	58.88	32.20	33.15	79.70	35.16	23.30	35.48	17.42	39.41
MaskCLIP + DVT [47, 33]	44.29	25.08	20.89	65.88	29.50	17.10	30.89	14.06	30.96
NACLIP + DVT [62, 33]	60.25	32.73	32.89	80.26	35.91	23.41	36.31	17.54	39.91
Ours (PH-Reg)	63.01	34.52	35.27	83.05	37.88	24.66	37.17	19.22	41.85

VOC 2012 (VOC21) [63], PASCAL Context (PC 60) [64], COCO-Object (Object) [65], COCO-Stuff (Stuff) [66], Cityscape (City) [67], ADE20K-150 (ADE) [68]. In addition to these standard benchmarks, we also evaluate on two commonly used variants, PASCAL VOC 2012 (VOC20) and PASCAL Context (PC 59), in which the background class is excluded from the evaluation. For all experiments, we utilize the same evaluation harness for all methods, and apply a sliding window inference strategy for non-square images. We also resize input images such that the shorter side is fixed to specific resolutions, accommodating the varying original image sizes across datasets.

Baselines. We compare our method against several relevant approaches in open-vocabulary semantic segmentation, including MaskCLIP [47], SCLIP [59], ClearCLIP [61], and NACLIP [62]. We also include vanilla CLIP as a baseline in our comparison, as it can be adapted for semantic segmentation. Unless otherwise specified, all visual encoders use the widely adopted pretrained ViT backbone with the same OpenAI CLIP ViT-B/16 weights to ensure a fair comparison. We also include denoised versions of MaskCLIP and NACLIP produced by DVT, as DVT represents a closely related method to our approach. Unless otherwise noted, for CLIP we adopt the most basic MaskCLIP framework (direct v output without any attention modifications) as our student model. Notably, we re-implemented all baselines using the same prompt templates as in [59, 62]. All reported results are obtained without any post-processing refinement.

Quantitative Results. Table 1 summarizes the quantitative comparison results of various open-vocabulary semantic segmentation models. We observe that PH-Reg CLIP consistently outperforms all compared methods on 7 out of 8 evaluated benchmarks, with particularly strong results in VOC21 (63.01%) and COCO Object (35.27%). Moreover, PH-Reg CLIP surpasses the denoised versions of MaskCLIP and NACLIP, where DVT fails to yield significant performance gains. We believe this is caused by the residual estimator in DVT, which assumes stationary artifacts – an assumption that does not hold consistently for training-free open vocabulary segmentation methods based CLIP. We note that ClearCLIP slightly outperforms our method on the VOC20 dataset. This may be attributed to its use of correlative self-attention, *i.e.* q - q attention, which incorporates feature localization cues. This explanation is plausible, as SCLIP, which also employs q - q attention, similarly outperforms NACLIP on the VOC20 dataset.

Qualitative Results. Figure 4 presents a qualitative comparison between our PH-Reg CLIP and three baseline models: MaskCLIP, SCLIP, and NACLIP. We visualize the UMAP of the dense features produced by each model, as well as the corresponding heatmaps generated from different text queries. Our qualitative observations are as follows: **1.** Artifact tokens are frequently observed in the UMAP visualizations of MaskCLIP, SCLIP, and NACLIP. While some artifact tokens are reduced in the heatmap of MaskCLIP, they remain prevalent in the heatmaps of both SCLIP and NACLIP. **2.** The presence of artifact tokens hinders the models’ ability to maintain fine-grained spatial alignment with the text queries, leading to suboptimal localization. **3.** In contrast, PH-Reg CLIP consistently produces cleaner UMAPs and more fine-grained, semantically aligned heatmaps, which correspond well to meaningful local image structures. These visualizations demonstrate that our method effectively preserves the consistency of dense feature representations and enhances semantic alignment between visual and textual modalities.

Table 2: **Linear Probe Based Evaluation Results on Segmentation and Depth.** PH-Reg improves pretrained ViT backbones across various dense prediction tasks. For semantic segmentation, we report the mean Intersection over Union (mIoU, %) metric and mean accuracy (mAcc, %). For monocular depth estimation, we report Root Mean Squared Error (RMSE), absolute relative error (Abs Rel), and accuracy under threshold δ_1 . The best result for each dataset is highlighted in **bold**.

Method	VOC21		ADE		NYUd		
	mIoU(\uparrow)	mAcc(\uparrow)	mIoU(\uparrow)	mAcc(\uparrow)	RMSE(\downarrow)	Abs Rel(\downarrow)	$\delta_1(\uparrow)$
CLIP [34]	73.88	83.37	35.78	47.3	0.6843	0.2115	64.93
CLIP + DVT [34, 33]	74.74	84.33	36.39	48.14	0.6800	0.2089	65.07
NACLIP [62]	74.01	83.16	37.06	48.33	0.6852	0.2082	64.52
NACLIP + DVT [62, 33]	74.47	82.98	36.91	48.56	0.6845	0.2122	65.11
MaskCLIP [47]	70.28	79.06	34.43	44.74	0.6645	0.2030	67.71
MaskCLIP + DVT [47, 33]	71.38	80.49	34.43	44.86	0.6792	0.2091	64.96
Ours (PH-Reg)	75.32	84.96	38.07	49.58	0.6746	0.1995	68.17
OpenCLIP [70]	71.31	80.64	37.68	49.8	0.6853	0.2113	64.86
OpenCLIP + DVT [70, 33]	72.58	83.42	38.30	50.91	0.6811	0.2159	64.73
OpenCLIP + Ours	73.25	83.99	39.32	51.24	0.6784	0.2019	65.32
DFN-CLIP [71]	71.98	82.07	36.81	47.83	0.6860	0.2118	64.50
DFN-CLIP + DVT [71, 33]	73.09	83.52	37.73	49.39	0.6852	0.2092	64.65
DFN-CLIP + Ours	72.97	82.48	39.15	50.61	0.6768	0.2052	65.26
DINOv2 [46]	84.13	92.00	47.82	60.50	0.4566	0.1391	82.92
DINOv2 + DVT [46, 33]	85.43	93.37	48.86	61.61	0.4329	0.1289	85.23
DINOv2 + Ours	84.85	92.46	48.66	61.57	0.4306	0.1216	86.35

4.2 Linear Probe Based Segmentation & Depth Evaluation

Datasets & Baselines. In this section, we evaluate our approach in two semantic segmentation datasets: PASCAL VOC 2012 (VOC21) [63] and ADE20K-150 (ADE) [68] and one monocular depth estimation dataset: NYUv2-Depth dataset (NYUd) [69]. Since DVT [33] targets a similar objective with our approach, we include both the vanilla models and their denoised versions produced by DVT as comparison baselines. We adopt the same linear probe experimental setup as in [33], and train a linear layer integrated into the backbones, as a decode head to predict pixel-wise segmentation or depth logits from patch tokens. Table 2 summarizes the main experiment results.

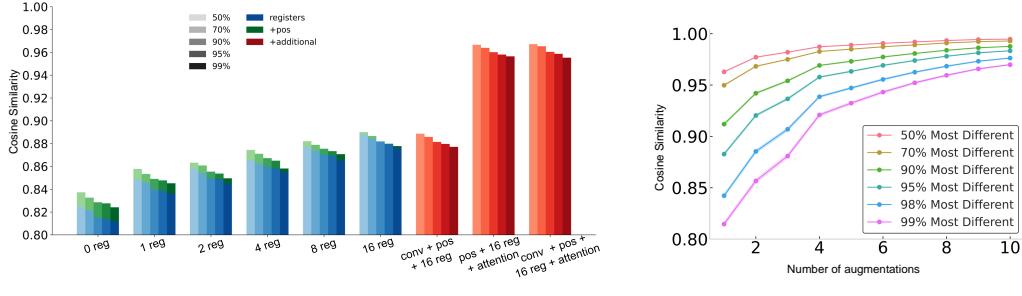
Semantic Segmentation Results. As shown in Table 2, We observe significant and consistent improvements, outperforming at least 4 out of 6 denoised ViT backbones across the evaluated datasets. While DVT consistently enhances the performance of DINOv2 and vanilla CLIP, it provides only limited improvements for other ViT backbones derived from other CLIP models. In contrast, our approach yields substantial performance boosts across these backbones, especially a notable +5.04% mIoU on VOC21 and +3.64% mIoU on ADE20k. These results demonstrate that our method can be robustly adopted to enhance the performance of diverse ViT backbones in semantic segmentation.

Notably, DVT relies on neural fields and requires gradient-based optimization, making the iterative denoising process applied to each image individually highly time-consuming. Our method leverages test-time augmentation for denoising, enabling the generation of cleaner dense feature representations without incurring excessive computational overhead. However, our results also show that the residual estimator as introduced in DVT may be beneficial to some model types (DINOv2, DFN-CLIP) more so than others. These results highlight that PH-Reg achieves superior performance in suppressing artifact tokens through a more robust and efficient design.

Depth Estimation Results. Following prior work [46, 33] we adopt AdaBins [72] for monocular depth evaluation. As shown in Table 2, our method consistently improves the performance of pretrained ViT backbones whereas the DVT assumption of stationary artifacts mostly hold true for DINOv2. Additionally, DVT achieves performance gains using an additional transformer block with 0.08x the parameters of the base models [33], our method achieves superior results with only a negligible increase in parameter count introduced by the register tokens. These results demonstrate the efficiency of our approach, yielding noticeable performance gains with minimal model overhead.

4.3 Ablation Studies and Investigation of Artifacts

In this section, we conduct ablation studies on OpenAI CLIP ViT-B/16 to investigate various architectural and training components, focusing on both model performance and training feasibility.



(a) Registers' behavior. This plot illustrates adding registers improve PH-Reg teacher performance. In the blue settings, only registers are unfrozen. The green settings represent the improvements when positional embeddings are unlocked additionally. The red settings represent performance of unlocking more layers.

(b) Augmentations improves cosine similarity. The plot illustrates how increasing the number of augmentations improves the alignment of the model’s predictions with the target of 200 augmentations’ features, as measured by cosine similarity.

Figure 5: Ablation on number of registers and augmentations

The Number of Register Tokens. We evaluate the influence of the number of register tokens on the cosine similarity between the student model’s outputs and the target values using the COCO Caption dataset [73]. Specifically, we distill the student model with 0, 1, 2, 4, 8, or 16 register tokens.

As illustrated in Figure 5a, the cosine similarity increases as the number of register tokens grows, indicating improved alignment with the target representations. However, the performance gain becomes marginal when increasing the number of registers from 4 to 8 and from 8 to 16. Based on this observation, we use 16 register tokens in all subsequent experiments.

Distillation Architectural Settings. We further evaluate the impact of architectural configurations during distillation by analyzing cosine similarity on the COCO Caption dataset. In this setting, we vary the number of register tokens from 0 to 16 while allowing the positional embeddings to be updated during training. As shown in Figure 5a, the improvement in the cosine similarity from unlocking the position embedding becomes less pronounced as the number of register tokens increases. Nonetheless, unlocking positional embeddings continues to provide a positive effect on alignment. This result suggests that in contrast to DVT, the positional embedding itself is unlikely to fully explain the artifact tokens.

Next, we fix the number of register tokens to 16 and evaluate the effect of unlocking additional layers, including the convolutional patch embedding layer and the later attention layers. For all experiments, we report 50th, 70th, 90th, 95th, and 99th percentiles (of cosine similarity to capture the distribution of the most dissimilar features.) Our analysis reveals that incorporating even a single register leads to substantial improvements. In particular, the 99th percentile of feature cosine similarity in the 1-register configuration exceeds the 50th percentile (median) of the raw case without registers. This indicates that registers significantly enhance the quality of feature representations across the distribution, not only in extreme cases.

As suggested by [54, 33], attention layers close to the output also play an important role, which we confirm in our experiments. As shown in Figure 5a, unlocking the last attention layer significantly increases the cosine similarity between the student model’s outputs and the target values. While unlocking the convolutional patch embedding layer alone slightly reduces cosine similarity value, the overall value improves when both the convolutional patch embedding and later attention layers are unlocked, compared to the baseline with only unlocked the position embeddings and later attention layers with 16 registers. Therefore, we unlock the positional embeddings, the convolutional patch embedding layer, and the final attention layer during distillation.

The Number of Augmentations in the Denoising Process. We evaluate our approach using cosine similarity on the COCO Caption dataset to investigate the effect of the number of augmentations. The

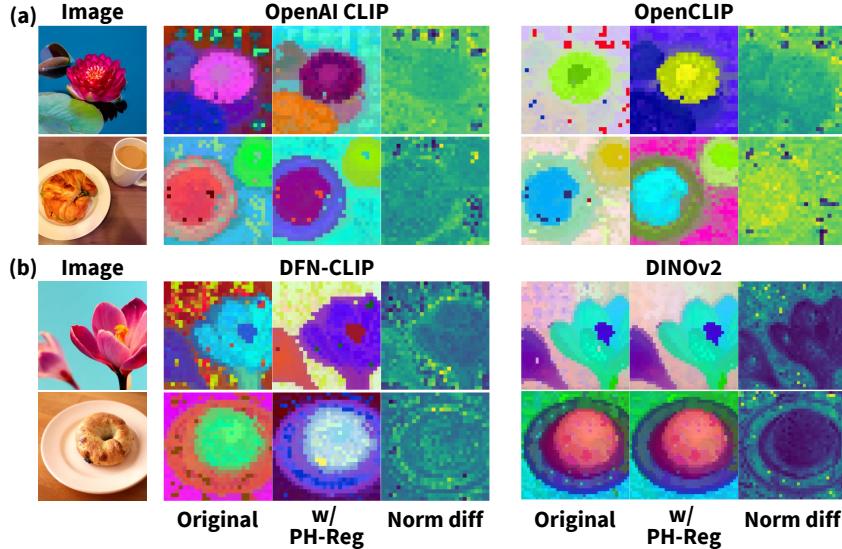


Figure 6: **Comparison of Original and PH-Reg Features and Norms.** While prior work has noted artifact tokens in DINOv2 as having higher norm than other tokens, we observe this is not the case for all models. Some models have artifact tokens with lower magnitude.

Table 3: **Distillation Approaches.** This table reports the contributions of different components in our distillation framework. *Vanilla* refers to the student MaskCLIP, *Denoising Only* refers to MaskCLIP with 10x averaging, the other rows refer to using NACLIP as teacher and MaskCLIP as student.

Approach	VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	Avg.
Vanilla	49.27	25.46	26.94	66.56	28.62	18.80	28.33	13.70	32.21
Denoising only (10x aug)	51.41	28.13	29.00	69.58	31.03	20.25	31.82	15.20	34.55
Distill, no reg, no denoise	61.16	33.51	34.51	81.51	36.70	23.96	35.74	18.34	40.68
Distill, with reg, no denoise	61.27	33.52	34.39	81.52	36.74	23.92	35.55	18.38	40.66
Distill, no reg, with denoise	62.48	34.28	35.00	82.27	37.62	24.46	36.83	18.92	41.48
Full Pipeline	63.01	34.52	35.27	83.05	37.88	24.66	37.17	19.22	41.85

student model is distilled with 1 to 10 augmentations, where one of them is always an unmodified image. As shown in Figure 5b, a high convergence threshold is observed at the *99th* percentile where even the most dissimilar cases exhibit cosine similarity values above 0.95, indicating a substantial reduction in feature space outliers. As a result, we employ 10 augmentations to generate high-quality features efficiently, thereby reducing computational overhead.

Distillation Approaches. We evaluate the contribution of different components in our distillation approach by conducting open-vocabulary semantic segmentation task on 8 benchmarks. In this setting, the student model is distilled by sequentially removing each component from our approach. These components include the denoising process with 10 augmentations, 16 additional register tokens, and self-distillation. As shown in Table 3, each component contributes to improving the student model’s performance. By comparing our full pipeline, where student model is distilled with denoising process and registers, we find that approximately half of the improvement comes from registers, and the other half results from denoising process applied to the teacher model.

The Ratio of Shifting in the Denoising Process. To investigate the impact of different shifting ratios in our denoising process, we conduct open-vocabulary semantic segmentation task on 8 benchmarks. We gradually increase the shifting ratio from 0% to 30%, while fixing the number of augmentation at 10. As shown in Table 4, a shifting ratio of 10% demonstrates the best performance across 4 datasets, while a shifting ratio of 15% achieves the best performance across 3 datasets and provides optimal average performance across all 8 datasets. Therefore, we adopt a shifting ratio of 15% in denoising process when applied to the teacher model.

Table 4: **Effect of Shifting Ratio.** This table reports the impact of different shifting ratios used in the denoising process, as defined in 3.1. Throughout this experiment, we use OpenAI CLIP ViT-B/16 as the backbone for the model and apply 10 augmentations in the denoising process.

Ratio	VOC21	PC60	Object	VOC20	PC59	Stuff	City	ADE	Avg.
0%	58.88	32.20	33.15	79.70	35.16	23.30	35.48	17.42	39.41
10%	60.49	32.91	33.73	80.47	35.94	23.86	36.60	17.94	40.24
15%	60.47	32.89	33.61	80.36	35.94	23.89	36.87	18.10	40.27
20%	60.39	32.82	33.46	79.93	35.85	23.86	36.98	18.07	40.17
25%	60.26	32.75	33.26	79.75	35.75	23.78	37.02	18.00	40.07
30%	60.14	32.76	33.25	79.39	35.77	23.77	37.07	18.10	40.03

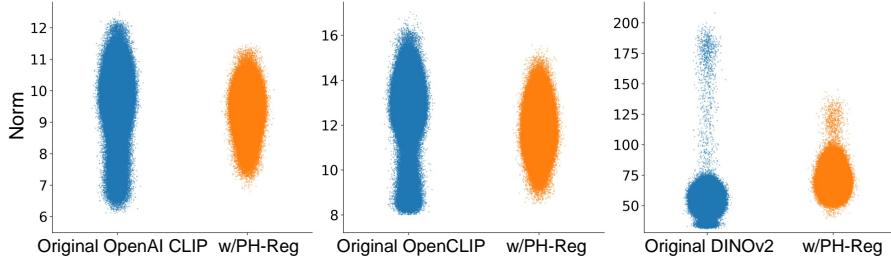


Figure 7: **Patch Norms.** This figure illustrate norms of patch tokens of different backbones. Our method effectively reduces the variance of token norms and reduces the outliers, regardless if the artifacts are lower/higher norm.

4.4 Investigation of Artifacts and Registers

While prior work has noted that artifact tokens are high magnitude in DINOv2 [50], in Figure 6 we find that this is not always the case. In OpenAI CLIP and OpenCLIP, the artifacts are generally lower norm than their surrounding patches. In contrast, in DFN-CLIP and DINOv2, the artifacts are higher norm. This illustrates that there may be elements of the training dynamic at play, as the artifact norms can differ even when the training objective is very similar. In Figure 7 we visualize the norms of the original network and those with registers added, we find that our method effectively reduces the variance of the patch norms.

5 Discussion

Limitations and Future Work In this work we proposed PH-Reg, a method to reduce the artifact tokens in existing pre-trained vision transformers. We show that PH-Reg can eliminate artifact tokens in ViTs effectively and generate clean dense feature maps, enhancing the performance in downstream dense prediction tasks. This approach relies on test time augmentation to denoise dense feature presentations in the teacher model. While our method generally outperforms DVT on CLIP based models, we sometimes underperform when using the DINOv2 backbone. We believe this is due to the static artifact estimator present in DVT. The assumption of static artifacts holds true for some models (DINOv2), but not for others (CLIP). A potential avenue for additional investigation is how to dynamically determine the artifacts without strong stationary assumptions.

Conclusion We introduce a novel post-training method PH-Reg, for learning clean dense feature representations in ViTs through an efficient self-distillation framework that does not require additional labeled data. Our approach leverages test-time augmentation to denoise the teacher model, and guide the student model to optimize the dense feature representations. This enables us to eliminate artifact tokens effectively by integrating learnable registers into existing pretrained models, without the need for training from scratch. We demonstrate that the distilled ViTs generate fine-grained dense feature maps, enhancing the consistency of feature representations in ViTs. We further show that cleaner dense feature maps in ViTs leads to quantifiable improvements on dense prediction tasks. Finally, we illustrate that the distilled ViTs can accurately capture meaningful semantic structures in images, as shown by heatmaps generated from CLIP text queries. We validate our conclusions with extensive evaluations across multiple dense prediction benchmarks.

References

- [1] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018.
- [2] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.
- [3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.
- [4] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in neural information processing systems*, 32, 2019.
- [5] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019.
- [6] 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.
- [7] Tobias Christian Nauen, Sebastian Palacio, Federico Raue, and Andreas Dengel. Which transformer to favor: A comparative analysis of efficiency in vision transformers. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6955–6966. IEEE, 2025.
- [8] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12894–12904, 2021.
- [9] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272, 2021.
- [10] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. *arXiv preprint arXiv:2102.08602*, 2021.
- [11] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, pages 10183–10192. PMLR, 2021.
- [12] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021.
- [13] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021.
- [14] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International conference on machine learning*, pages 9438–9447. PMLR, 2020.
- [15] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.
- [16] 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.

[17] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

[18] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

[19] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wanhai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[20] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10809–10818, 2022.

[21] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. In *European conference on computer vision*, pages 328–345. Springer, 2022.

[22] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[23] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021.

[24] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021.

[25] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 589–598, 2021.

[26] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[27] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.

[28] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[29] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

[30] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model. *arXiv preprint arXiv:2402.12376*, 2024.

[31] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024.

[32] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11144–11154, 2023.

[33] Jiawei Yang, Katie Z Luo, Jiefeng Li, Congyue Deng, Leonidas Guibas, Dilip Krishnan, Kilian Q Weinberger, Yonglong Tian, and Yue Wang. Denoising vision transformers. In *European Conference on Computer Vision*, pages 453–469. Springer, 2024.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[35] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=Ee277P3AYC>.

[36] 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.

[37] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[38] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *International Conference on Learning Representations (ICLR)*, 2022.

[39] Trieu H. Trinh, Minh-Thang Luong, and Quoc V. Le. Selfie: Self-supervised pretraining for image embedding, 2019. URL <https://arxiv.org/abs/1906.02940>.

[40] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/chen20s.html>.

[41] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[43] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

[44] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=p-BhZSz59o4>.

[45] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

[46] Maxime Oquab, Timothée Darcret, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[47] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022.

[48] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022.

[49] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12490–12500, June 2024.

[50] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[51] Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020.

[52] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[53] Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Ganelsman. Vision transformers don't need trained registers. *arXiv preprint arXiv:2506.08010*, 2025.

[54] Haoqi Wang, Tong Zhang, and Mathieu Salzmann. Sinder: Repairing the singular defects of dinov2. In *European Conference on Computer Vision*, pages 20–35. Springer, 2024.

[55] Alexander Lappe and Martin A. Giese. Register and cls tokens yield a decoupling of local and global features in large vits, 2025. URL <https://arxiv.org/abs/2505.05892>.

[56] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023.

[57] Andrew F Luo, Jacob Yeung, Rushikesh Zawar, Shaurya Dewan, Margaret M Henderson, Leila Wehbe, and Michael J Tarr. Brain mapping with dense features: Grounding cortical semantic selectivity in natural images with vision transformers. *arXiv preprint arXiv:2410.05266*, 2024.

[58] Yi Li, Hualiang Wang, Yiqun Duan, Jiheng Zhang, and Xiaomeng Li. A closer look at the explainability of contrastive language-image pre-training. *Pattern Recognition*, 162:111409, 2025. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2025.111409>. URL <https://www.sciencedirect.com/science/article/pii/S003132032500069X>.

[59] Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference, 2024. URL <https://arxiv.org/abs/2312.01597>.

[60] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers, 2023. URL <https://arxiv.org/abs/2312.00878>.

[61] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Clearclip: Decomposing clip representations for dense vision-language inference. In *European Conference on Computer Vision*, pages 143–160. Springer, 2024.

[62] Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025.

[63] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[64] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014.

[65] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

- [66] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.
- [67] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [68] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.
- [69] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012.
- [70] 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, July 2021.
- [71] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.
- [72] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4008–4017, 2021. doi: 10.1109/CVPR46437.2021.00400.
- [73] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

A Technical Appendices and Supplementary Material

Sections

1. Implementation Details for Self-Distillation (section **B**)
2. Implementation Details for Quantitative Evaluation (section **C**)
3. Additional Efficiency Analysis of PH-Reg Compared to DVT (section **D**)
4. Additional Qualitative Examples for Segmentation (section **E**)
5. Additional Qualitative Heatmaps for PH-Reg Zero-Shot (section **F**)
6. Proof of Test Time Augmentation (section **G**)

B Implementation Details for Self-Distillation

In this section, we provide a detailed overview of how we implement self-distillation in PH-Reg. Our self-distillation framework consists of one teacher model and one student model. While both the teacher and student model are initialized from the same weights, the teacher is frozen, while additional register parameters are added to the student network.

Our codes and weights are available in <https://github.com/Oraiser0/PH-Reg.git>.

B.1 Model Architectures

Teacher Model Architecture. For CLIP based models, since we focus on zero-shot open-vocabulary segmentation, we utilize the NACLIP modification to the final layer. This modification does not introduce any additional weights to the teacher network, and is training-free. Our empirical analysis in Section B.4 shows that NACLIP’s neighborhood attention mechanism improves feature consistency. For DINOv2, we directly use the final output layer, without any modification to the teacher network.

Student Model Architecture. Based on the results from the ablation studies we integrate 16 register tokens into the student model. For the CLIP based student, to ensure representational alignment we directly take the v head from the output layer (the MaskCLIP output). For DINO based students, we do not apply such modifications. We bicubically upsample the positional embedding so it matches the input image. Unless otherwise specified, this modification is applied consistently, while all other layers remain unchanged.

B.2 Model Implementation

Table S.1: **Model Implementation Libraries and Weights.** We compare models trained using different datasets and objectives.

Model	Library	Weight
CLIP	clip (OpenAI)	ViT-B-16
OpenCLIP	open_clip	hf-hub:laion/CLIP-ViT-B-16-laion2B-s34B-b88K
DFN-CLIP	open_clip	hf-hub:apple/DFN2B-CLIP-ViT-B-16
DINOv2	transformers (Hugging Face)	facebook/dinov2-base

We provide the model weights and the corresponding implementation libraries in Table S.1.

B.3 Optimization

In the distillation process, the shorter side of each input image is resized using bicubic interpolation to 448 for CLIP-based models and 518 for DINOv2. The resized image is then randomly cropped into a square of size (448, 448) or (518, 518), respectively. For each input image, we generate $N = 10$ augmentations using random shifts and horizontal flips. Assuming an image length of 1, we uniformly sample the shift for both the horizontal and vertical axes from $[-0.15, 0.15]$. While the horizontal flip is sampled with probability 0.5. To ensure each patch is covered, we do not apply any augmentation to the first image of the 10. All shifted images are concatenated and fed into the teacher model, while the original (unshifted) images are used as input to the student model. The target feature is computed as the average of these 10 augmentations. To accommodate the resized input images for both the teacher and student models, we consistently resize the positional embeddings using bicubic interpolation. During training, the weights of the teacher model are frozen. In the student model, we allow updates to registers, the positional embeddings, the convolutional patch embedding layer, and the final transformer layer containing the self-attention mechanism.

The distillation framework is implemented in PyTorch, with distributed training managed via PyTorch Accelerate. Training is conducted on 4 NVIDIA Ada 6000 GPUs, with mixed-precision optimization to balance computational efficiency and numerical stability. Detailed training configurations are provided in Table S.2 and Table S.3.

Table S.2: Configs for CLIP-based models.

Config	Value
optimizer	AdamW
initial learning rate	3e-4
final learning rate	1e-5
weight decay	1e-2
optimizer momentum (β_1, β_2)	(0.9, 0.999)
learning rate scheduler	Exponential Scheduler
batch size	16
training epochs	100
augmentation	RandomSquareCrop

Table S.3: Configs for DINOv2.

Config	Value
optimizer	AdamW
initial learning rate	1e-4
final learning rate	5e-6
weight decay	1e-2
optimizer momentum (β_1, β_2)	(0.9, 0.999)
learning rate scheduler	Exponential Scheduler
batch size	8
training epochs	100
augmentation	RandomSquareCrop

B.4 Pearson Analysis of Open-vocabulary Segmentation

Table S.4: **Open-vocabulary Semantic Segmentation Quantitative Comparison on 7 datasets.** We report the Pearson correlation coefficient for the zero-shot query against the one-hot ground truth labels. The results are averaged within each image, then averaged across images. Compared to mIoU, pearson does not require knowledge of all of the categories present an image (via softmax). The value ranges from -1 to 1, where 1 = perfect positive correlation, -1 = perfect negative correlation, and 0 = no linear correlation. The best result for each dataset is highlighted in **bolded**.

Method	VOC21	PC60	VOC20	PC59	Stuff	City	ADE20k	Avg.
SCLIP	-0.005	0.349	0.409	0.443	0.323	0.291	0.308	0.303
ClearCLIP	0.012	0.428	0.489	0.543	0.393	0.336	0.418	0.374
NACLIP	0.011	0.422	0.470	0.543	0.392	0.363	0.425	0.375
NACLIP+DVT	0.003	0.438	0.487	0.551	0.395	0.367	0.427	0.381
Ours (PH-Reg)	0.013	0.468	0.494	0.590	0.424	0.381	0.461	0.404

In this section we present additional evaluation results on open-vocabulary semantic segmentation via the pearson metric. Results are illustrated in Table S.4. Overall, PH-Reg CLIP significantly outperforms the baseline models on 7 datasets. Even in the absence of prior category knowledge, PH-Reg CLIP achieves an average performance of 0.404, representing a clear improvement over the second-best method, DVT enhanced NACLIP, with an average performance of 0.381. These results highlight that our approach improves the consistency of dense feature representations by reducing artifact tokens, thereby offering a robust and generalizable enhancement over existing methods.

We further observe that both ClearCLIP and NACLIP achieve competitive results; however, NACLIP significantly outperforms ClearCLIP on ADE20K and Cityscapes. The former requires the model to handle a large number of categories, while the latter demands fine-grained localization of small objects. Based on this observation, we choose NACLIP as our primary teacher model, leveraging its neighbor attention mechanism to enhance the student model’s performance on these challenging tasks.

Table S.5: **Dataset Specific Details for Open-vocabulary Semantic Segmentation.** We list the per-dataset resolution, crop size, and stride used for each dataset. We maintain the same settings for all methods within a given dataset.

Dataset	VOC21	PC 60	Object	VOC20	PC 59	Stuff	City	ADE
Resize resolution	448	448	336	336	448	448	560	448
Crop size	336	336	336	336	336	336	224	336
Stride	112	112	112	112	112	112	112	112

Table S.6: **Training Time Efficiency Analysis.** We report the training time of our PH-Reg method compared to DVT. For fairness, we restrict all stages of our model to a single GPU and evaluate the same model (DINOv2 ViT-B).

Method	Stage 1 Extraction	Stage 1 Distillation	Stage 2 Training	Total
DVT	2998 min	18340 min	570 min	21908 min (365.1 h)
PH-Reg	-	-	-	9000 min (150 h)

C Implementation Details for Quantitative Evaluation

In this section, we provide detailed implementation information for our quantitative evaluation experiments. In section C.1, we present the evaluation details for open-vocabulary semantic segmentation (OVSS). In section C.2, we describe the evaluation details for linear probe based semantic segmentation and monocular depth estimation.

C.1 Implementation Details of Open-vocabulary Semantic Segmentation.

We follow SCLIP and NAACLIP in the setup for the open-vocabulary semantic segmentation evaluation. For fairness, we utilize the same parameters for all models. We resize input images such that the shorter side is scaled to a specific resolution, while maintaining the original aspect ratio for the longer side. Additionally, we set fixed crop sizes and strides during evaluation. All evaluation parameters are summarized in Table S.5, while all other settings follow their default configurations.

C.2 Implementation Details of Linear Probe Based Evalution.

Our linear probe evaluation follows prior works (Vision Transformers Need Registers, Denoising Vision Transformers), where a linear layer is trained as a decoding head to predict pixel-wise segmentation or depth logits.

Semantic Segmentation. We extract the final output features from the frozen backbone and, if applicable, pass them through the denoiser (for the DVT baseline). A single learnable linear layer is then trained to predict the segmentation logits. For CLIP-based models, both training and testing images are resized to (448, 448), while for DINOv2, the images are resized to (518, 518).

Monocular Depth Estimation. Similar to semantic segmentation, we extract features from the backbone, and pass them through the denoiser if applicable. Following the method in DVT and DINOv2, we then append the [CLS] token to each patch token to enrich the feature representations for all methods. A linear layer is trained using SigLoss and gradient loss (scaled by a factor of 0.5) to predict depth values into 256 uniformly distributed bins. We adopt DVT’s learning rate of 5e-3 for all experiments.

D Additional Efficiency Analysis of PH-Reg Compared to DVT

In this section, we provide a detailed efficiency analysis of PH-Reg compared to DVT, focusing on three aspects: training time, space usage, and inference cost.

Training Time. When evaluating training time, we utilize the official code provided in the DVT repository without any modification. The original DVT code specifies a single GPU for feature

Table S.7: **Inference Efficiency Analysis.** We report the inference cost results for all models, evaluating efficiency based on model size and GFLOPs.

Method	GFLOPs	Params (M)
MaskCLIP	62.89	86.19
NACLIP	64.76	86.19
NACLIP w/ denoising (10x)	647.6	86.19
NACLIP + DVT	70.32	94.07
CLIP + PH-Reg (Ours)	64.16	86.66

extraction and learning – for fairness we also limit all stages of our own model to a single GPU here. For DVT and PH-Reg, we evaluate the same model (DINOv2 ViT-B). Results are illustrated in Table S.6. Our method has the advantage of not utilizing gradient-based neural field learning as done in DVT. Therefore, our method trains the student model in a single stage, saving over 58.9% of the time compared to DVT.

Space Usage. A further advantage is the space utilization during training. DVT requires saving 1.4 terabytes of intermediate neural fields in the form of serialized Instant-NGP files. Our method computes all distillation targets on the fly, and requires no additional space.

Inference Cost. When evaluating testing cost for all models, we cast parameters to *fp32* dtype, and use eager attention implementation for all models. For our method, we include the positional embeddings adapted for 448 resolution. For MaskCLIP and NACLIP, current official implementations have the same number of parameters as the original CLIP, although a small reduction can be achieved in MaskCLIP by discarding the last q, k head. For DVT, we evaluate their stage 2 model, corresponding to the transformer block denoiser coupled to a original vision transformer. Results are illustrated in Table S.7. Our method utilizes approximately 10% fewer FLOPs and 10% fewer parameters during inference compared to DVT.

E Additional Qualitative Examples

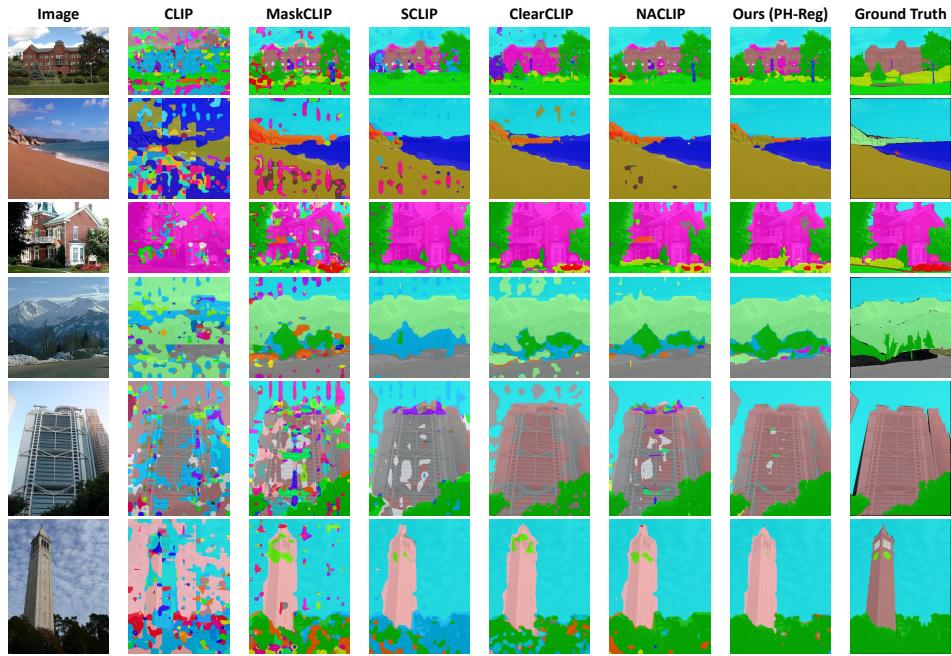


Figure S.1: Open-vocabulary semantic segmantation qualitative comparision between different baseline models on ADE20K.

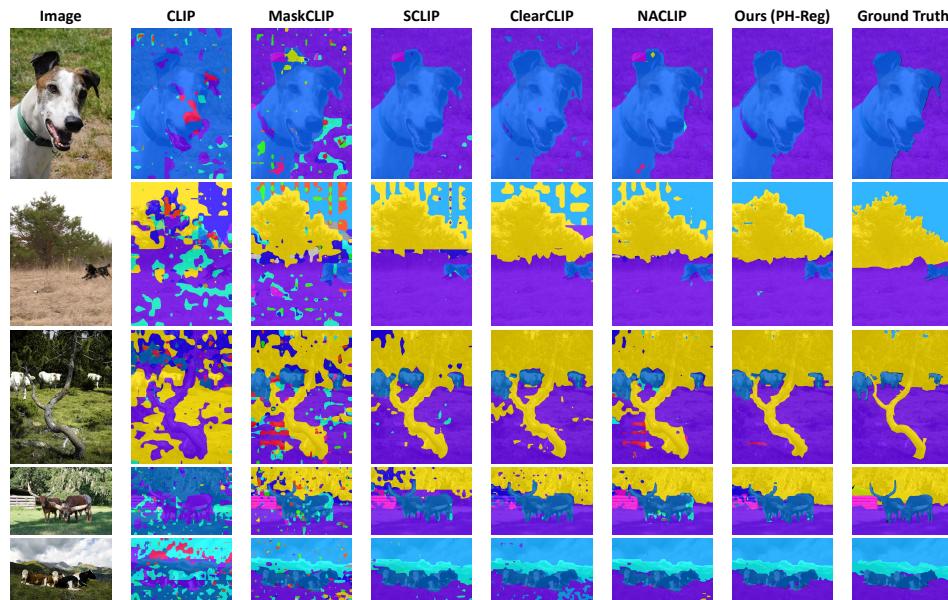


Figure S.2: Open-vocabulary semantic segmantation qualitative comparision between different baseline models on Pascal Context59.

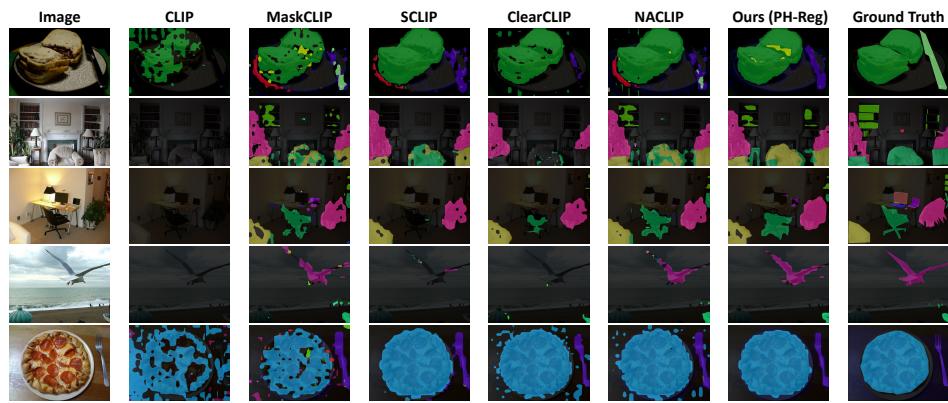


Figure S.3: Open-vocabulary semantic segmentation qualitative comparision between different baseline models on COCO Obejct.

F Additional Qualitative Heatmaps for PH-Reg Zero-Shot

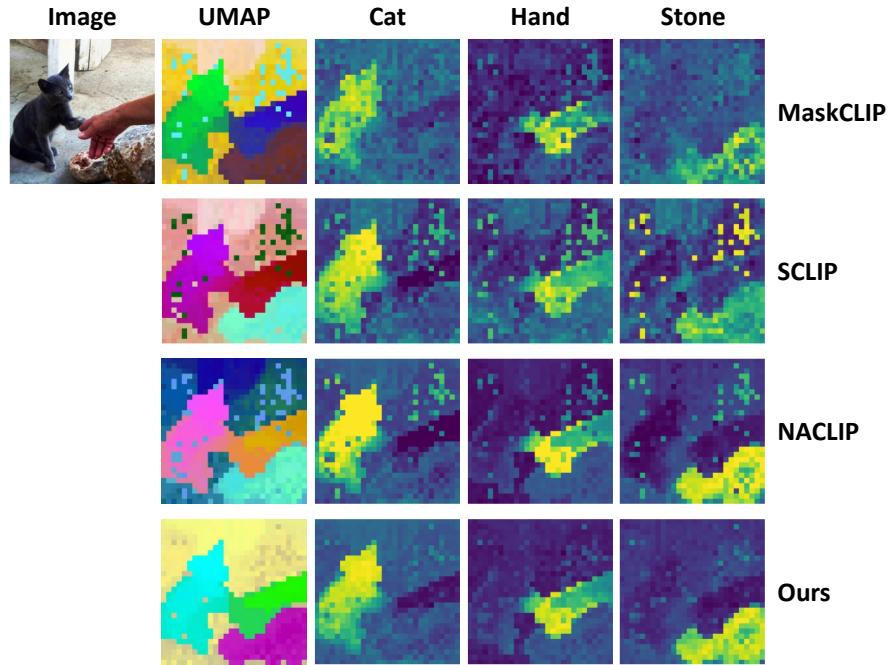


Figure S.4: **Zero-shot Heatmap Results.** Our results have fewer artifacts than other methods.

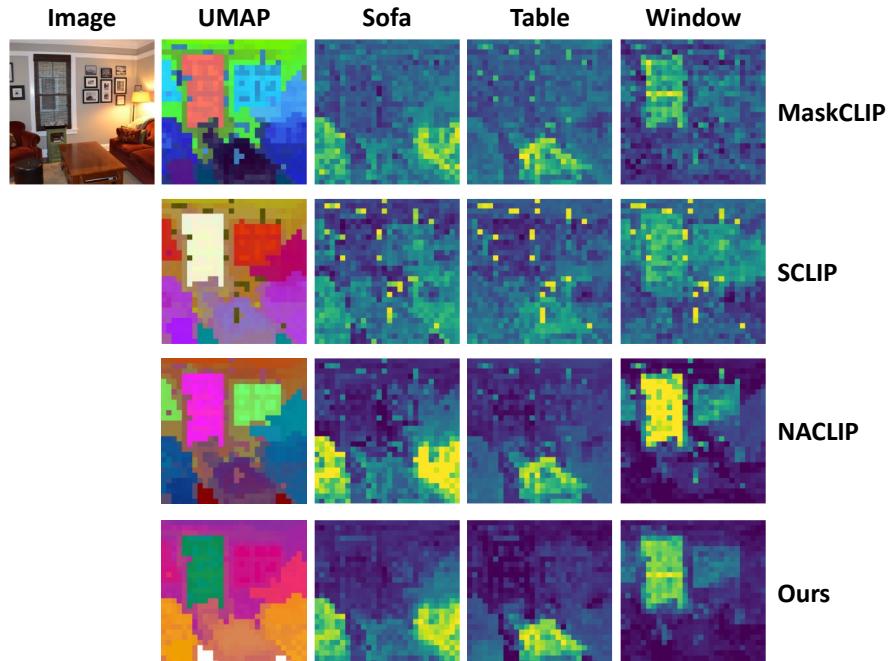


Figure S.5: **Zero-shot Heatmap Results.** Our results have fewer artifacts than other methods.

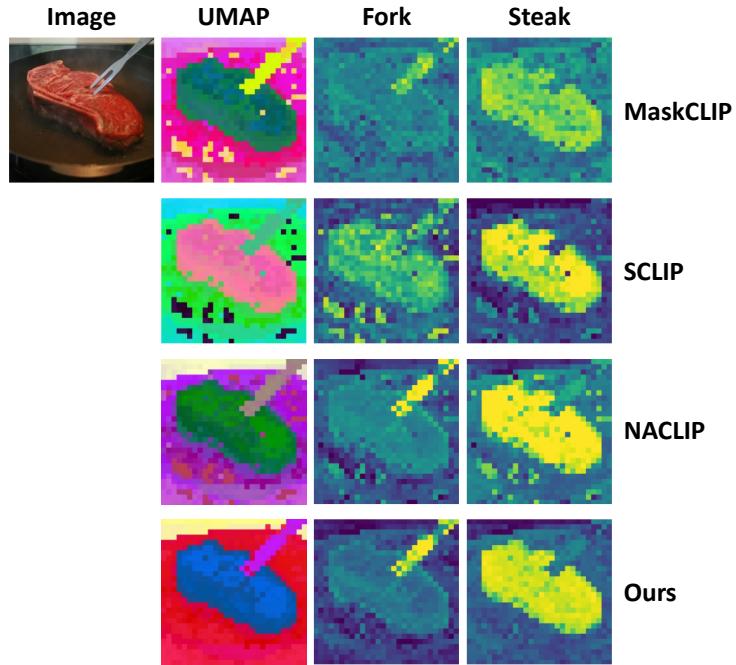


Figure S.6: **Zero-shot Heatmap Results.** Our results have fewer artifacts than other methods.

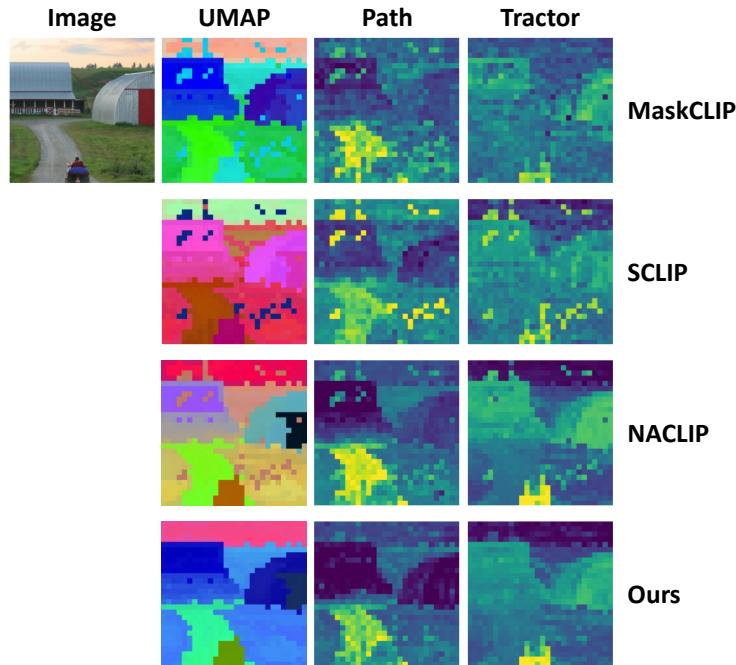


Figure S.7: **Zero-shot Heatmap Results.** Our results have fewer artifacts than other methods.

G Optimal Feature Aggregation

Let $f_1, \dots, f_n \in \mathbb{R}^d$ be feature vectors of a single patch from n different transformations of an input image \mathcal{I} . We seek the optimal aggregated feature f^* that minimizes the total squared error:

$$f^* = \arg \min_f \sum_{i=1}^n \|f_i - f\|_2^2 \quad (1)$$

Expanding the objective gives us:

$$\arg \min_f \sum_{i=1}^n (f_i^\top f_i - 2f_i^\top f + f^\top f) \quad (2)$$

Dropping constant terms that do not change the optimum:

$$= nf^\top f - 2 \left(\sum_{i=1}^n f_i^\top \right) f \quad (3)$$

Dividing and multiplying the right side by n :

$$= nf^\top f - 2n \left(\sum_{i=1}^n \frac{1}{n} f_i^\top \right) f \quad (4)$$

Dividing the equation by n as whole shows us that we need to minimize:

$$\|f - \frac{1}{n} \sum_{i=1}^n f_i\|_2^2 \quad (5)$$

So it can be derived that the mean of the feature vectors is the minimizer under MSE loss:

$$f^* = \frac{1}{n} \sum_{i=1}^n f_i \quad (6)$$

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We justify each claim with extensive experiments.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We have a limitations section and discuss potential future work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: This paper does not have theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide extensive details on our ablation studies. Additional details can be found in the supplemental.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code will be included in the supplemental.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: A high-level outline is included in the main paper. Implementation details will be included in the supplemental.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to cost of running experiments, we only perform one run.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: Please see the supplemental for computational details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: The research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: This is discussed in the supplemental.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper has no risk related to safeguards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets that we use are public, and are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Code will be included in the supplemental.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not collect any human data.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We do not collect any human data.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.