Suppressing Non-Semantic Noise in Masked Image Modeling Representations

Martine Hjelkrem-Tan[1], Marius Aasan[1], Rwiddhi Chakraborty[2],
Gabriel Y. Arteaga[1], Changkyu Choi[1], Adín Ramírez Rivera[1]
[1]University of Oslo, [2]UiT The Arctic University of Norway
{matan, mariuaas, gabrieya, changkyc, adinr}@uio.no, rwiddhi.chakraborty@uit.no

Abstract

Masked Image Modeling (MIM) has become a ubiquitous self-supervised vision paradigm. In this work, we show that MIM objectives cause the learned representations to retain non-semantic information, which ultimately hurts performance during inference. We introduce a model-agnostic score for semantic invariance using Principal Component Analysis (PCA) on real and synthetic non-semantic images. Based on this score, we propose a simple method, Semantically Orthogonal Artifact Projection (SOAP), to directly suppress non-semantic information in patch representations, leading to consistent improvements in zero-shot performance across various MIM-based models. SOAP is a post-hoc suppression method, requires zero training, and can be attached to any model as a single linear head. Code available at: https://github.com/dsb-ifi/soap.

1. Introduction

Self-supervised learning (SSL) via Masked Image Modeling (MIM) objectives have become a popular source for strong, generalized vision backbones [1–3, 11, 16, 22, 26, 29, 33, 36, 53]. However, recent works have uncovered key issues with artifacts and noise in the representations in models that rely on MIM-based objectives [16, 26, 27, 29, 36, 53]. Some of these issues can be traced to the objective itself—MIM requires predicting both the semantic content and location of the masked patches [5]. While positional collapse—where the model learns to predict the position of masked tokens instead of content—is a known issue with various suggested mitigation techniques [5, 16, 36], studies directly addressing the extent of this phenomenon are relatively unexplored.

In this paper, we present a novel method to measure the amount of *non-semantic noise* in ViT tokens for state-of-the-art MIM-based models. We characterize non-semantic noise as components that are invariant to the *semantic content* in the input. This can for example be positional encoding, which are necessary for attention mechanisms but seldom useful in inference, or structural artifacts as discussed in Darcet et al. [15]. Our central hypothesis in this work is that *this noise persists in non-semantic images*. As a result, isolating the noise can be approached as quantifying invariant responses between semantic and non-semantic images. This is useful, as suppressing the noise can lead to clearer semantic signals, improving representations for use in downstream tasks. Through our analysis, we discover that strong principal components exhibit high levels of non-semantic noise, and that this feature is *pervasive in MIM-based models* while *nearly non-existent in other, non MIM-based SSL models*. Importantly, this holds *regardless of which positional embedding method is employed* [29] and whether predictions are conducted in latent or input-space, suggesting that this is an *implicit issue in MIM*.

To suppress non-semantic information, we introduce a Semantically Orthogonal Artifact Projection (SOAP) to remove unwanted artifacts that are not useful for inherently semantic tasks, such as instance level classification and salient segmentation—cf. Fig. 1. SOAP is flexible: It is computed directly from data using a Gram-Schmidt based projection, thus requiring no training, and can be attached as an external module to any pretrained SSL backbone.

Our **contributions** include: (i) An in-depth analysis that shows that MIM objectives *uniquely* bias representations toward encoding positional noise rather than semantic information. (ii) A novel *Semantic Invariance Score* to measure the level of semantic invariance in a model's representations, allowing us to diagnose the semantic-positional noise trade-off in SSL representations. (iii) Finally, based on this score, we propose Semantically Orthogonal Artifact Projection (SOAP), a post-hoc denoising strategy that suppresses non-semantic noise, and show improved performance in zero-shot downstream tasks for all MIM-based models.

2. Related Work

Contrastive Learning. Contrastive learning by encouraging representations to be invariant to minor augmentations is a central paradigm for SSL [12, 14, 31, 39, 50]. Modern approaches utilize self-distillation by matching rep-

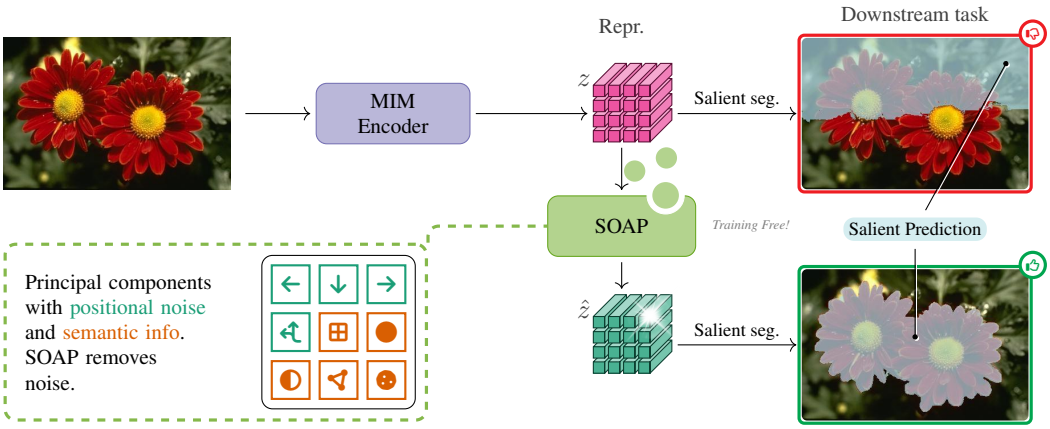


Figure 1. Pipeline overview; a pretrained MIM encoder outputs dense representations z which are used for downstream tasks—we show salient segmentation as an example. By identifying and suppressing principal components encoding positional noise, our SOAP module improves the representations $\hat{z}$ and enhances downstream performance in zero-shot settings.

resentations between a teacher and a student network, with periodic exponential moving average (EMA) updates [10, 11, 13, 20]. These approaches typically rely on strong global cues that match local with global views via aggressive cropping without preserving relative spatial information. However, a stronger reliance on global cues limits performance on dense prediction tasks, as finer-grained semantic relationships are not fully exploited [24, 28, 40, 49].

Masked Image Modeling. A strong alternative to learning invariances through data augmentations, is through choosing reconstruction of masked patches as the pretext task [4, 22, 45]. As opposed to contrastive learning, MIM—popularized by MAE [22]—provides a robust and scalable approach by directly reconstructing the input data via a decoder, thus circumventing representational collapse to trivial solutions. Reconstruction in the representation space, rather than the pixel space, removes the explicit need for a decoder, and has been shown to work in methods like I-JEPA [2]. The MIM objective has been adopted in the self-distillation setup quite successfully—iBOT [53] employs the averaged cross entropy over the student's masked view, and the teacher the non-masked view, and DINOv2 demonstrates additional improvements by separating the MLP projection heads, using adaptive resolutions, and additional regularization [26]. CAPI [16] adopts a pure MIM approach through clustering by generating pseudo-labels in the masked latent space, which removes the need for a delicate balancing act between MIM and instance based contrastive objectives. The recent release of DINOv3 employs modern positional encoding, and a denoised warmup training phase [29]. Across all these approaches, our analysis indicates a consistent level of non-semantic noise not present in purely contrastive approaches.

Positional Noise in MIM Models. A recurring challenge in masked image modeling (MIM) frameworks is the presence of positional noise in the learned embeddings. This has been indirectly addressed through a variety of architectural and training modifications, such as introducing register tokens [15], regularization mechanisms [16, 29], or disentangling positional cues [5, 36, 37, 48]. Despite these efforts, MIM representations typically show weaker out-of-the-box performance compared to frameworks with contrastive objectives, and often require extensive fine-tuning to transfer effectively to downstream tasks. Some works analyze embedding variance to identify noisy or unstable tokens [35], while others propose selective aggregation to prune non-informative dimensions [27]. Venkataramanan et al. [36] introduce RASA, a post-hoc module that suppresses explicit location cues by training to predict patch positions from pretrained embeddings. However, RASA only addresses patch-location noise and does not provide a broader analysis of non-semantic noise in other frameworks. In contrast, our proposed solution requires no training. Additionally, our study directly identifies non-semantic components rather than learning positional cues, probing the nature of semantically invariant noise encoded by state-of-the-art SSL models, and proposing metrics to quantify and interpret these effects.

3. Semantically Orthogonal Artifact Projection

Before introducing our proposal, we provide an exploratory analysis that sheds light on how to decompose the semantic and positional information from the data. Specifically, we use principal component analysis (PCA) to reveal structural patterns that expose positional bias in MIM training (Sec. 3.1). Then, we formalize a linear decomposition of representations into semantic and non-semantic parts, providing a foundation for quantifying the degree to which different components capture meaningful information versus

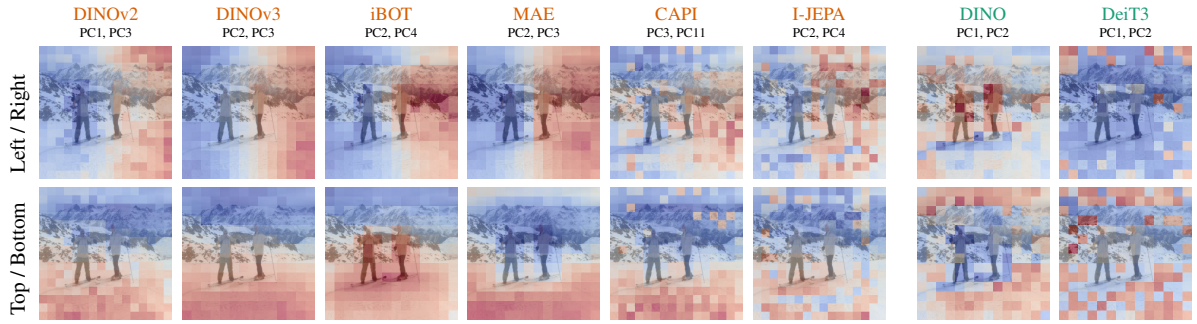


Figure 2. Representations from MIM models exhibit strong positional bias in leading principal components (PC), illustrated here by the response heatmap of selected PCs for one example image. MIM models show clear left/right and top/bottom bias. This behavior is not observed for non-MIM models. In Section 3, we propose a novel method to automatically isolate such positional bias in a post-hoc fashion.

noise (Sec. 3.2). Following, we propose a score to measure semantic invariance by contrasting real and synthetic inputs (Sec. 3.3). Finally, we introduce our Semantically Orthogonal Artifact Projection (SOAP), which uses this score to suppress non-semantic noise, enhancing zero-shot performance in MIM-based models on downstream tasks (Sec. 3.4).

3.1. PCA of Patch Embeddings

PCA is a natural tool to decompose the representation space into dominant variance directions. This allows us to examine whether leading components reflect semantic content or non-semantic structure, such as positional bias. To this end, we estimate the covariance of the D-dimensional patch embedding space $\mathcal{Z} \subseteq \mathbb{R}^D$ under a given model f over ImageNet [17] using Welford's online algorithm [42], and obtain the principal components by eigendecomposition of the covariance matrix

$$\mathrm{Cov}(\mathcal{Z}) = V \Lambda V^\top \in \mathbb{R}^{D \times D}, \tag{1}$$

with principal component vectors $V = (v_1, \ldots, v_D)$.

For each component $d = 1, ..., D$, we define the *response* of an input z by the inner product $\langle z, v_d \rangle$, which reflects how strongly each patch embedding $z \in \mathcal{Z}$ projects onto direction v_d. By inspecting the responses, we observe that leading principal components in MIM models exhibit strong biases based on patch position. This indicates that a significant share of their variance is devoted to encoding position, *implying that positional bias arises as a direct consequence of MIM objectives*. Fig. 2 shows the responses for an example image for selected principal components that were identified as having a strong positional bias. In contrast, models trained without MIM objectives, such as DINO and DeiT3, do not exhibit this behavior.

To further characterize these components, we evaluate their behavior on a set of images. Let Ω be the input set of images, and $f \colon \Omega \to \mathcal{Z}$ a model that encodes an image $x \in \Omega$ into N patch embedding representations $\boldsymbol{z} = \{z_n\}_{n=1}^N$. The binary activation of component d for x is then defined by thresholding the token responses:

$$A_d(x) = \mathbf{1}[\boldsymbol{z} v_d \geq \eta] \in \{0, 1\}^N. \tag{2}$$

Here η is a scalar threshold; we set $\eta = 0$ in our experiments. The expectation $\bar{A}_d = \mathbb{E}_x[A_d(x)] \in \mathbb{R}^N$ yields an average activation map that highlights spatial patterns.

With these definitions in place, we formulate two diagnostic conditions to identify non-semantic components. First, when the average activation $\bar{A}_d$ exhibits systematic dependence on patch location, we infer that component d predominantly encodes positional cues rather than semantic content. Second, when the per-sample activations $A_d(x)$ exhibit minimal variation across different inputs, component d can be regarded as invariant to image content, which indicates that it captures non-informative signals.

3.2. Patch Embedding as Linear Combinations

To isolate the signals underlying a model's representations, we express each patch embedding z as a mixture of semantic and non-semantic sources. Ideally, we want the model f to capture the relevant semantic information in the input, like global class information and local structures of color, shape, texture, and fine-grained semantics useful for semantic downstream tasks. However, in standard ViTs, f must also encode the positional information directly to each embedding z, since the attention operator is otherwise permutation-invariant.

We formalize this by assuming that each embedding z can be decomposed as a mixture of sources, Φ for semantic information and P for non-semantic noise. Then we express

$$z = \theta_\phi \phi + \underbrace{\theta_\rho \rho + \varepsilon}_{\text{non-semantic}}; \quad \varepsilon \sim \mathbb{P}, \tag{3}$$

where $\theta_\phi, \theta_\rho \in \mathbb{R}_{\geq 0}$ are scalar coefficients for $\phi \in \Phi, \rho \in \mathrm{P}$, and ε represents residual noise from some probability distribution $\mathbb{P}$. The semantic information component $\phi \in \Phi$ encodes both relevant information for global objectives,

such as instance discrimination and classification, and dense objectives like segmentation. The non-semantic information component $\rho \in \mathrm{P}$ encodes local positions and relative geometry among the local patch embeddings $\boldsymbol{z}$, based on their positions in the original image x.[1]

Intuitively, both contrastive and MIM-based training objectives encourage f to encode relevant semantic information. However, in line with the observations of previous work [5], we posit that the inpainting objective in MIM also drives f to encode a substantial amount of non-semantic information ρ, thereby increasing the coefficient θ_ρ at the expense of the semantic coefficient θ_ϕ. This raises the question of how much of a learned representation reflects semantic content, as opposed to non-semantic noise or other non-informative signals. To address this, we introduce a *semantic invariance score* to measure the degree of non-semantic information encoded by each principal component d in a learned patch representation space $\mathcal{Z}$.

3.3. Semantic Invariance

Semantic invariance refers to the property of a component yielding consistent responses even when the semantic content of local representations varies. In other words, a component is semantically invariant if it produces similar activations regardless of whether the input carries meaningful semantic information. We posit that such components are uninformative for downstream understanding, as they do not encode or reflect actual image contents.

Let $\mathcal{X} \subset \Omega$ be the set of semantically informative images, and let $\mathcal{X}^c$ denote a complementary set without semantic information. In practice, $\mathcal{X}$ is instantiated as the ImageNet validation set [17], while $\mathcal{X}^c$ is approximated by a synthetic noise generator. The synthetic images are generated by a mixture of pink noise, modulated white noise, and random low-frequency gradient fields. The details are described in App. C—see the examples in Fig. C.1. For each component d, we compute binary activations A_d using Eq. (2) for samples $x \sim \mathcal{X}$ and $x^c \sim \mathcal{X}^c$. This yields two empirical Bernoulli distributions of activations for the token index $n = 1, \ldots, N$, such that

$$P_{d,n} = \Pr\big(A_{d,n} = 1 \mid x \sim \mathcal{X}\big), \quad \text{and} \tag{4}$$

$$Q_{d,n} = \Pr\big(A_{d,n} = 1 \mid x^c \sim \mathcal{X}^c\big). \tag{5}$$

If $P_{d,n} \approx Q_{d,n}$, then component d behaves similarly for semantic and non-semantic inputs for token index n, indicating semantic invariance. Conversely, large discrepancies between $P_{d,n}$ and $Q_{d,n}$ indicate sensitivity to semantic content at the position with index n, which posits P_d, Q_d as multinomial distributions over all tokens. To quantify this discrepancy, we define a score to measure *semantic invariance (SI)*

$$s_d = \mathrm{SI}(P_d, Q_d), \tag{6a}$$

$$= 2 \cdot \frac{P_d \cdot Q_d + (1 - P_d) \cdot (1 - Q_d)}{\sqrt{P_d^2 + (1 - P_d)^2} + \sqrt{Q_d^2 + (1 - Q_d)^2}}, \tag{6b}$$

$$= 2 \cdot \frac{\langle P_d, Q_d \rangle}{||P_d|| + ||Q_d||}, \tag{6c}$$

which assigns high scores when $P_d \approx Q_d$, and vice versa. However, in the case where $P_{d,n} \approx Q_{d,n} \approx 0.5$ for most token indices n, we cannot really say that the component is strongly semantically invariant, as the model is uncertain. We account for this uncertainty in our definition of the SI-score; it gives lower scores when the model is uncertain. This property is visualized in Fig. 3. Our score is similar in form to the Dice-Sørensen coefficient [9], but we take the inner products and norms over the support set rather than the multivariate dimensions; see App. B.1 for more details. A vectorial formulation of Eq. (6) is given in Eq. (B.1).

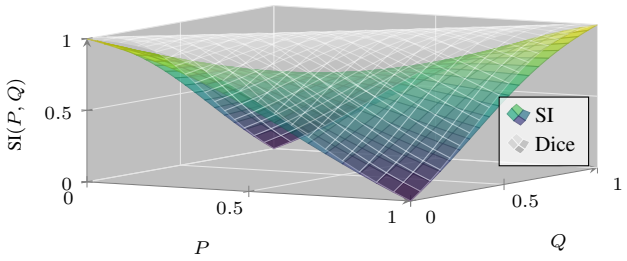


Figure 3. Plot of the semantic invariance (SI) score (viridis). The SI-score increases when $P \approx Q$ and the probabilities are confident (close to 0 or 1). In the uncertain case $P \approx 0.5 \approx Q$, the score is lower to reflect ambiguity of semantic invariance. For comparison we also show the Dice-Sørensen coefficient (gray), which does not have this property, and thus is unable to capture the uncertainty.

3.4. Semantically Orthogonal Artifact Projection (SOAP)

With the SI score in Eq. (6), we introduce Semantically Orthogonal Artifact Projection (SOAP), an off-the-shelf denoising strategy that suppresses non-semantic noise. We hypothesize that suppressing components that are invariant to semantic content in the representation z acts as a denoising step, yielding representations that are closer to the semantic part $\theta_\phi \phi$ in Eq. (3). To achieve this, we operate in the PCA basis, where these components v_d are orthogonal by construction. Each component is assigned a weight w_d, derived from its semantic invariance score s_d and a scaling function t, which determines how strongly it should be suppressed. Our SOAP projector P_ϕ is defined by the Gram–Schmidt process, subtracting the contribution of components identified as non-informative

$$P_\phi = I - VWV^\top; \quad \hat{z} = P_\phi z. \tag{7}$$

[1] We note that positional information is given implicitly in certain models that retain spatial order, like CNNs and MLPs. In this case f does not need to explicitly encode positional information into z. However, we restrict this study to ViTs.

Here, V are the PCA components from Eq. (1), I is the identity matrix, and $W = \text{diag}(w_1, \ldots, w_D)$ is a diagonal weight matrix.

Since directly weighting by the SI-scores leads to strong suppression across all components (see Fig. 3) we introduce a scaling function t so only the most invariant ones are suppressed. We propose filtering using a Fermi window [7, 8]—commonly used in MRI imaging—using the rank of the scores r. This corresponds to a smooth regularization of the scores [21] using a sigmoid gating approach [25] with explicit control over truncation and smoothness. The scaling function is defined by

$$w_d = t(s_d, r) = s_d \times \frac{\sigma((\mu - r)/\tau)}{\sigma(\mu/\tau)}; \quad r = \text{rank}(s_d). \quad (8)$$

Coupling statistical variance with semantic relevance, the hyperparameters μ and τ provide flexible control of the cut-off and smoothness of suppression, providing a simple way to adapt denoising strength to the spectral structure of the embeddings. Fig. E.1 shows the effect of the scaling function on the semantic invariance scores.

4. Experiments

For our study, we consider MAE [22], I-JEPA [2] and CAPI [16] for models with MIM objectives, DINO [11] for global contrastive objective, and iBOT [53], DINOv2 [26], and DINOv3 [29] for models with both. For completeness we also evaluate the supervised model DeiT3 [32]. Tab. 1 provides an overview of the models.

We estimate the principal components for the representation space of each model over ImageNet train [17] as described in Sec. 3.1, and calculate the SI-scores using ImageNet validation paired with 50000 synthetic images as described in App. C. SOAP is directly computed from the principal components and corresponding SI scores. We show strong improvements in zero-shot salient segmentation using the widely adopted TokenCut method [41] when applying SOAP prior to inference. Further, we show improvements in kNN semantic segmentation and kNN classification using patch embeddings. Additional details on our evaluation methods and ablations are given in App. F.

4.1. Analyzing Information Content in SSL Tokens

We examine the nature of the non-semantic information revealed by the SI-score for each model. To do so, we aggregate the activation maps for each component over patch embeddings from ImageNet validation and generated synthetic images. Fig. 4 visualizes the averaged activations of the principal components with the highest SI-scores, where we selected a few models for demonstration. Extended results for all models in our study is included in Fig. B.2. Across all MIM-based models, we observe strong positional bias in the form of left/right and top/bottom alignment (first two columns). In contrast, DINO and DeiT3 do not exhibit the same structured positional patterns. This empirically supports our claim that MIM objectives amplify the encoding of positional information in patch tokens. Notably, this phenomenon is present in both MIM-based models with standard additive positional embeddings (DINOv2, iBOT, CAPI, IJEPA) and those that inject positional information via RoPE in the attention mechanism (DINOv3). We also note that the two components with the highest semantic invariance are the exact same components identified as encoding positional noise in Fig. 2 for all MIM models, demonstrating that the SI-score indeed captures positional noise.

Next, we take a closer look at the SI-scores for the principal components of all the models—Fig. 5 shows the scores in descending order. We observe that MIM-models exhibit much higher maximum SI-scores than models trained without MIM. In particular, at least two components have an SI-score higher than 0.75 for all MIM-models. Meanwhile, models trained without MIM score below this threshold for all components, as demonstrated by the red-green gradient.

To highlight the discrepancy between models trained with and without MIM objectives, we look at the maximum

Table 1. Overview of models in our study. We select a representative group of models by including models with different architectures, objectives, MIM modes, and positional encoding in our experimental setup.

Model	Arch.	Objective	MIM mode	Pos. enc.
DINOv2	ViT-B/14	MIM + CL	Latent	Add.
DINOv3	ViT-B/16	MIM + CL	Latent	RoPE
iBOT	ViT-B/16	MIM + CL	Latent	Add.
MAE	ViT-B/16	MIM	Pixel	Add.
CAPI	ViT-L/14	MIM	Latent	Add.
I-JEPA	ViT-H/14	MIM	Latent	Add.
DINO	ViT-B/16	CL	NA	Add.
DeiT3	ViT-B/16	Supervised	NA	Add.

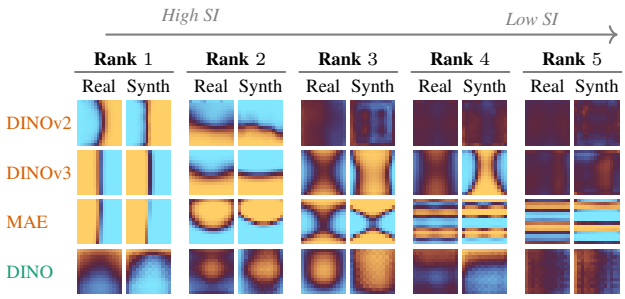
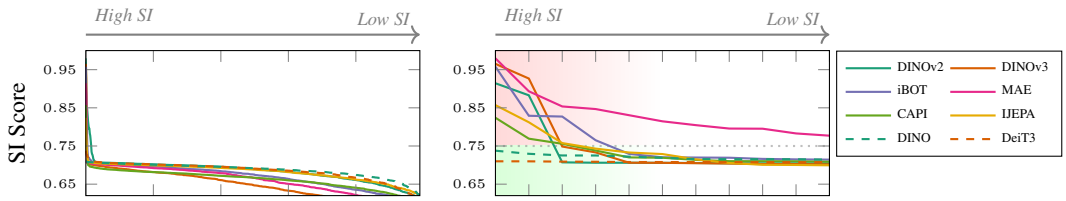


Figure 4. Semantic Invariance (SI) scores reveal positional bias in top-ranked principal components. **Real** and **Synth** columns show activations for real and synthetic images, respectively. The top two components encode left/right and top/bottom biases (see Fig. 2), which diminish in lower-ranked components. MIM models exhibit clear semantic invariance, whereas the non-MIM model (DINO) does not. See Fig. B.2 for additional models and examples.



(a) All principal components ranked by SI. (b) Top 10 principal components.

Figure 5. Semantic invariance (SI) score in descending order. All scores are shown in the left plot, while the right focuses on the top 10 semantically invariant scores. Note that all MIM-models have a max-score ≥ 0.75, while all non MIM models have a lower score.

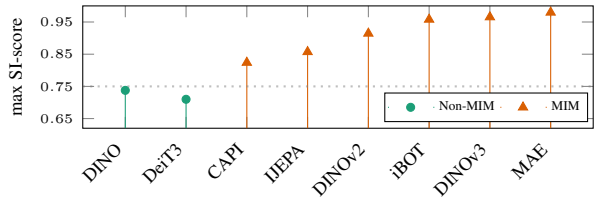


Figure 6. Models with MIM objectives exhibit higher max semantic invariance (SI) than models with other objectives.

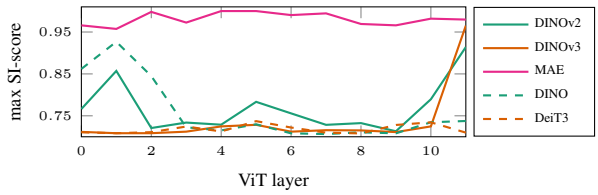


Figure 7. Maximum semantic invariance score for MIM models (solid lines) and non-MIM models (dashed lines) vs. model depth. Critically, MIM models show high SI-scores in the last layers. This can be explained by the MIM objective encouraging positional information in the patch embeddings of deeper layers.

SI-score for each model in Fig. 6. This reflects the level of non-semantic noise encoded by each model, and shows a prominent gap between non-MIM and MIM-based models.

We also probe semantic invariance over model depth. Fig. 7 shows the maximum SI-score per layer for a selection of the models. We observe that semantic invariance starts out high in the early layers, which is expected for models with additive positional embeddings, although DeiT3 is a somewhat surprising exception to this. Specific to the MIM-based models is that the SI-score increases again for the last layers. This can be explained as the model saturates more positional information into the embeddings in preparation for solving the MIM task. For MAE however, the score remains high across all layers.

To summarize, we confirm that the most semantically invariant components revealed by the SI score encode positional information, we show that MIM-based models consistently exhibit higher SI-scores, and we observe that the non-semantic noise either increases in the last transformer layer or is high throughout the MIM-based models.

4.2. Cleaning with SOAP for Zero-Shot Performance

We use SOAP to correct for semantically invariant components in local embeddings, and find that this improves performance in zero-shot downstream tasks for all MIM models. Note that since non-semantic noise can be identified through linear methods, evaluation using learnable heads is unsuitable as they can adapt to unintentionally mask the issue. See App. G for further details.

Based on the observations from Sec. 4.1, we let μ be the number of components with an SI-score above a threshold of 0.75, and use a sharper cut-off $\tau = 0.05$ to reduce suppression of the remaining components. This makes the suppression more sparse and retains all but the most semantically invariant components. In the case that no components have an SI-score higher than 0.75, we set all weights $w_d = 0$; this is equivalent to no suppression.

Salient segmentation. We select TokenCut [41] for zero-shot evaluation of relative saliency information present in local embeddings, and follow their evaluation on three datasets—ECSSD [46], DUTS [38], and DUTOMRON [47]—reporting intersection over union (IoU), accuracy, and the F-measure[2].

Tab. 2 shows that correcting the patch embeddings, in addition to using salient principal components as guides to foreground selection, can significantly boost performance.

We observe that DINO performs better out-of-the-box compared to the models with MIM objectives. We believe this is because DINO is trained with a global objective only. Positional information, and to a degree local semantic information, is not as useful for the salient segmentation task which relies on global (class specific) correlations. However, after suppressing semantically invariant components, most MIM models perform on par or better than DINO. Some notable exceptions are I-JEPA and DINOv3, which perform poorly in general for this task.

[2]F-measure is a standard metric used in saliency detection, defined by $F_\beta = \frac{(1+\beta^2)\text{Precision}\times\text{Recall}}{\beta^2\text{Precision}+\text{Recall}}$. Prediction and Recall are defined by the binary prediction mask and the ground truth. We report the max value of 255 uniformly distributed binarization thresholds, denoted maxF_β. IoU, Accuracy, and maxF_β are all computed following Wang et al. [41].

Table 2. Zero-shot salient segmentation with TokenCut. We evaluate on ECSSD [46], DUTS [38], and DUT-OMRON [47]. Correcting the embeddings with SOAP improves results for all MIM-based models.

		ECSSD			DUTS			DUT-OMRON		
Pretrain	Model	max F_β	IoU	Acc.	max F_β	IoU	Acc.	max F_β	IoU	Acc.
Original embeddings										
DINO	ViT-B16	82.580	74.325	90.929	83.932	75.769	89.705	59.289	52.851	83.019
DINOv2	ViT-B16	71.319	63.937	83.147	69.701	63.064	78.891	45.923	39.643	75.067
DINOv3	ViT-B16	36.975	29.122	52.953	39.874	31.302	52.264	19.656	15.623	46.258
iBOT	ViT-B16	62.873	56.248	78.785	66.353	59.657	77.987	33.731	29.602	67.883
CAPI	ViT-L14	72.456	66.083	84.334	68.148	61.913	78.634	49.150	42.423	77.762
MAE	ViT-B16	79.952	71.067	89.410	79.229	70.227	86.078	54.758	45.630	78.552
I-JEPA	ViT-H14	37.670	27.989	68.898	33.008	24.890	63.592	33.345	24.340	71.343
Corrected embeddings										
DINOv2	ViT-B16	80.633	72.559	88.687	84.387	76.785	89.440	50.610	43.472	71.762
DINOv3	ViT-B16	42.633	33.742	61.975	47.624	39.057	63.033	23.329	17.485	51.390
iBOT	ViT-B16	66.557	60.167	78.340	72.192	65.618	80.595	36.330	31.991	63.552
CAPI	ViT-L14	**85.219**	**78.084**	**92.600**	**85.710**	**78.439**	**91.906**	59.872	**51.315**	80.291
MAE	ViT-B16	82.094	72.118	91.444	82.877	72.293	90.107	**59.931**	48.297	**82.974**
I-JEPA	ViT-H14	40.239	31.162	71.406	33.371	25.922	65.001	35.472	27.038	76.841

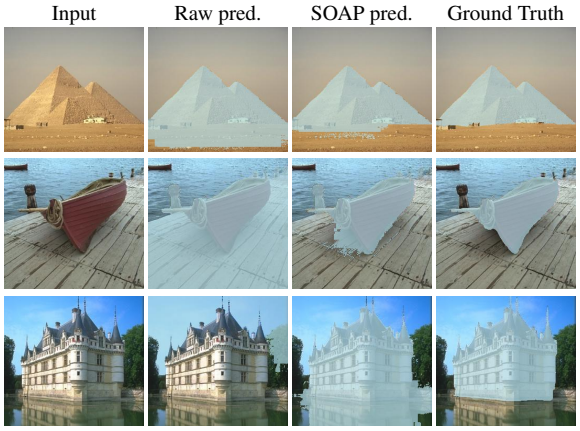


Figure 8. Examples of salient segmentation using TokenCut [41] with frozen CAPI [16] on the raw embeddings (Raw pred.) and after correcting with SOAP (SOAP pred.). Suppressing positional noise with SOAP improves the zero-shot saliency maps. In the last row, TokenCut is unable to generate a salient prediction from the raw embeddings, yet it succeeds when provided with SOAP embeddings.

kNN segmentation. We evaluate zero-shot segmentation on ADE20k [51, 52] by performing per-patch k-nearest neighbors (kNN) and upsampling the predictions to full image resolution using nearest neighbor interpolation. Tab. 3 shows that correcting the patch embeddings boosts performance for all models. Additional details and results for PascalVOC [18] are given in App. F.2.

kNN classification. To probe the degree to which instance information is encoded in the patch embeddings, we perform classification by weighted aggregation of patch-level kNN predictions. We use `cls`-attention weighted aggregation, except for the models trained without an instance objective (CAPI, MAE, I-JEPA), where we aggregate weighted by patch-prediction entropy instead. More details

Table 3. kNN segmentation on ADE20k [52] reporting mean IoU and pixel accuracy (Acc). Correcting with SOAP improves results.

		Original embeddings		Corrected embeddings	
Pretrain	Model	IoU	Acc	IoU	Acc
DINOv2	ViT-B16	40.25	74.60	40.81 ↑0.56	74.72 ↑0.12
DINOv3	ViT-B16	43.85	77.94	44.53 ↑0.68	78.09 ↑0.15
iBOT	ViT-B16	27.73	70.86	28.51 ↑0.79	71.29 ↑0.43
CAPI	ViT-L14	31.38	71.63	31.64 ↑0.26	71.78 ↑0.16
MAE	ViT-B16	11.88	58.00	13.74 ↑1.86	59.54 ↑1.54
I-JEPA	ViT-H14	20.95	60.27	21.26 ↑0.31	60.29 ↑0.01
DINO	ViT-B16	21.21	66.48	21.21 0.00	66.48 0.00

are given in App. F. We compare top-1 and top-5 accuracies on ImageNet [17]. The results in Tab. 4 show improvements after correcting for invariant components, although the benefit is in general less than for dense tasks. This is expected, as classification is not as reliant on local semantics and is therefore less affected by positional noise.

4.3. Ablations

SI-score sensitivity to dataset choice. To probe the SI-score sensitivity to dataset choice, we ablate over Caltech256 [19], COCO-Stuff164k [23], CUB200 [43], PASCAL VOC [18], and ImageNet [17] using DINOv2 as the backbone. We calculate the cosine distance between the score vectors over the $D = 768$ components. We observe that the distances are low (between 0.0025 and 0.0032), which means that the scores for each component remain consistent despite using different datasets with different distributions and of various sizes to instantiate semantically informative input. This indicates that the SI-score is not sensitive to dataset choice. We show the SI-score distance for each pair of datasets in App. D.

Use of scaling function. We ablate the necessity of filtering the SI-scores with the scaling function from Eq. (8). The results for kNN classification in Tab. 5 show that ap-

Table 4. Weighted kNN classification on ImageNet [17] by aggregating patch predictions. We compare original vs. SOAP-corrected embeddings across backbones.

		Original embeddings		Corrected embeddings	
Pretrain	Model	Acc@1	Acc@5	Acc@1	Acc@5
DINOv2	ViT-B16	82.32	96.29	82.60 ↑0.28	96.30 ↑0.01
DINOv3	ViT-B16	81.47	95.57	81.48 ↑0.01	95.63 ↑0.05
iBOT	ViT-B16	71.45	90.03	71.72 ↑0.27	90.10 ↑0.08
CAPI†	ViT-L14	70.81	91.09	71.25 ↑0.43	91.33 ↑0.24
MAE†	ViT-B16	59.98	81.89	62.83 ↑2.85	84.38 ↑2.49
I-JEPA†	ViT-H14	75.38	91.55	75.63 ↑0.24	91.64 ↑0.09
DINO	ViT-B16	66.08	86.13	66.08 0.00	86.13 0.00

†Aggregation weighted by entropy for models with no class token objective; otherwise weighted by class attention.

Table 5. Ablation on the scaling function in Eq. (8), evaluated on kNN classification of average pooled patch embeddings on ImageNet [17].

		SOAP without scaling		SOAP with scaling	
Pretrain	Model	kNN Acc@1	kNN Acc@5	kNN Acc@1	kNN Acc@5
DINOv2	ViT-B16	77.068	91.588	77.102	91.635
DINOv3	ViT-B16	76.354	91.306	76.588	91.612
iBOT	ViT-B16	59.194	79.610	59.498	79.918
CAPI	ViT-L14	55.720	76.950	56.444	77.742
MAE	ViT-B16	47.596	69.108	47.758	69.442
I-JEPA	ViT-H14	71.422	86.112	71.390	86.168

Table 6. Comparing SOAP with RASA on Franca ViT-B/14 for zero-shot salient segmentation on ECSSD [46] and kNN classification on ImageNet [17].

	ECSSD (Sal. Seg)			IN1k (kNN cls.)	
Method	max F_β	IoU	Acc.	Acc@1	Acc@5
Franca	71.615	64.899	83.982	64.920	85.872
Franca + RASA	68.220	68.220	85.935	64.890	85.886
Franca + SOAP	**84.176**	**76.985**	**91.514**	**65.084**	**86.006**

plying SOAP without scaling the SI-scores can reduce performance. Evaluating on salient segmentation shows the same result; see Tab. E.1. We posit that this is because too many components are suppressed when the scores are not scaled in the projection, resulting in detrimental information loss. While the SI-score allows us to identify the most semantically invariant components, the raw scores remain too high for the remaining components, unnecessarily dampening their contribution.

Comparison with RASA. We compare SOAP with RASA by correcting the patch embeddings from Franca using pretrained RASA weights from Venkataramanan et al.'s [36] work. The results in Tab. 6 show that SOAP yields higher performance. We argue that the significant improvement stems from SOAP correcting non-semantic noise directly, while RASA has to learn the positional cues to remove. We also note that RASA requires training 9 linear layers of dimension $D \times 2$ ($D = 768$ for ViT-B), while SOAP is **simpler**, requires **no training**, and can be **attached to any model** as a single $D \times D$ linear head.

5. Discussion

Our analysis indicates a tendency for MIM to "cheat" to perform the matching of masked tokens by dedicating capacity to patch location cues. Furthermore, Fig. 4 shows that this happens in both latent and reconstructive MIM. The results are not totally unexpected; the reconstruction objective requires positional information to perform the task to some extent. However, we note that latent MIM models exhibit this property with or without additive positional encoding, such as for DINOv3. The models learn to dedicate capacity to non-semantic information. This makes sense if we consider how much this actually helps the MIM objective; by being able to correctly select the patch position in the image, the search is reduced by a ratio of $H_z \times W_z$. For a ViT-B/14 model, this results in a reduction of $\times 256$, significantly improving the loss of the model. While MIM improves performance on dense objectives, it does so at a *cost*. The importance and severity of this positional noise is corroborated by several works [16, 37, 48]. Our method shows that *the problem is pervasive for MIM models, and does not meaningfully occur in non-MIM models*.

Limitations and Further Work. We restrict evaluation to raw patch embeddings; both kNN and TokenCut operate directly on the representations without additional projections or heads. This is a conscious choice, as further transformations would confound the effect of SOAP with that of the evaluation model itself, which is beyond the scope of the current work. An avenue for future work is to study how SOAP interacts with more elaborate evaluation protocols. Lastly, the linear assumption as a mixture of sources (Sec. 3.2) is a simplification, but given our empirical results, this assumption seems to at least partially hold.

6. Conclusion

We demonstrate that masked image modeling (MIM) objectives bias vision transformers toward encoding positional noise, corroborating our hypothesis that such noise persists even in inputs devoid of semantic content. This suggests that MIM learning signals are solved by short-cutting the intended objective of learning better local representations, and while this helps solve the pretext task, it reduces zero-shot generalization and semantic fidelity. To diagnose and address this issue, we introduce a Semantic Invariance Score and the lightweight, post-hoc denoising method SOAP, which consistently suppresses positional noise and improves downstream performance. Our findings highlight a fundamental trade-off in MIM and offer practical tools for building more semantically robust self-supervised models.

Acknowledgments

This work was funded by RCN (the Research Council of Norway) through Visual Intelligence, Centre for Research-based Innovation (309439), and in part by the RCN–NRF (National Research Foundation of Korea) joint project AURoRA (359216, RS-2025-03522980). We acknowledge Sigma2 (Project NN8104K) for access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through Sigma2, Norway.

References

[1] Benedikt Alkin, Lukas Miklautz, Sepp Hochreiter, and Johannes Brandstetter. MIM-Refiner: A contrastive learning boost from intermediate pre-trained representations. In *Inter. Conf. Learn. Represent. (ICLR)*, 2025. 1

[2] 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. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023. 2, 5, 1

[3] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *Inter. Conf. Mach. Learn. (ICML)*, 2022. 1

[4] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. In *Inter. Conf. Learn. Represent. (ICLR)*, 2022. 2

[5] Amir Bar, Florian Bordes, Assaf Shocher, Mido Assran, Pascal Vincent, Nicolas Ballas, Trevor Darrell, Amir Globerson, and Yann Lecun. Stochastic positional embeddings improve masked image modeling. In *Inter. Conf. Mach. Learn. (ICML)*, 2024. 1, 2, 4

[6] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Trans. Mach. Learn. Res.*, 2024. 6

[7] Matt A. Bernstein, Sean B. Fain, and Stephen J. Riederer. Effect of windowing and zero-filled reconstruction of mri data on spatial resolution and acquisition strategy. *Journal of Magnetic Resonance Imaging*, 14(3):270–280, 2001. Introduces/uses a radial *Fermi* window in k-space; discusses SNR and truncation artifact trade-offs. 5

[8] Elisabeth C. Caparelli and Dardo Tomasi. K-space spatial low-pass filters can increase signal loss artifacts in echo-planar imaging. *Biomedical Signal Processing and Control*, 3(1):107–114, 2008. Explicitly notes Fermi filters (as used on GE systems) and Hamming filters for fMRI k-space smoothing. 5

[9] Aaron Carass, Snehashis Roy, Adrian Gherman, Jacob C Reinhold, Andrew Jesson, Tal Arbel, Oskar Maier, Heinz Handels, Mohsen Ghafoorian, Bram Platel, et al. Evaluating white matter lesion segmentations with refined sørensen-dice analysis. *Scientific reports*, 10(1):8242, 2020. 4

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2020. 2

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2021. 1, 2, 5

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Inter. Conf. Mach. Learn. (ICML)*, 2020. 1

[13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021. 2

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

[15] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *Inter. Conf. Learn. Represent. (ICLR)*, 2024. 1, 2

[16] Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latents patches for improved masked image modeling. *Trans. Mach. Learn. Res.*, 2025. 1, 2, 5, 7, 8, 9

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2009. 3, 4, 5, 7, 8, 6

[18] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2012 (voc2012) results, 2012. 7, 4, 5

[19] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, 2022. 7

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2020. 2

[21] Per Christian Hansen. *Discrete Inverse Problems: Insight and Algorithms*. SIAM, 2010. 5

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 16000–16009, 2022. 1, 2, 5

[23] 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 *European Conf. Comput. Vis. (ECCV)*, 2014. 7

[24] Yanbin Liu and Stephen Gould. Unsupervised dense prediction using differentiable normalized cuts. In *European Conf. Comput. Vis. (ECCV)*, pages 19–36. Springer, 2024. 2

[25] Huy Nguyen, Nhat Ho, and Alessandro Rinaldo. Sigmoid gating is more sample efficient than softmax gating in mix-

ture of experts. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, pages 118357–118388, 2024. 5

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024. 1, 2, 5

[27] Marcin Przewięzlikowski, Randall Balestriero, Wojciech Jasiński, Marek Śmieja, and Bartosz Zieliński. Beyond [cls]: Exploring the true potential of masked image modeling representations. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2025. 1, 2

[28] Gilles Puy, Yuwei Li, Hang Xu, Andrea Scamarcio, Xin Xu, Ishan Misra, Yujun Shen, Cordelia Schmid, and Camille Couprie. Three pillars improving vision foundation model distillation for lidar. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 14460–14470, 2024. 2

[29] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 1, 2, 5

[30] E P Simoncelli and B A Olshausen. Natural image statistics and neural representation. *Annu. Rev. Neurosci.*, 24:1193–1216, 2001. 2

[31] Yonglong Tian, Dilip Krishnan Chen, and Phillip Isola. What makes for good views for contrastive learning? In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2020. 1

[32] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *European Conf. Comput. Vis. (ECCV)*, 2022. 5

[33] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint*, 2025. 1

[34] Grant Van Horn, Oisin Mac Aodha, Yang Song, Ying Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist Species Classification and Detection Dataset. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 8769–8778, 2018. 5, 6

[35] Ani Vanyan, Alvard Barseghyan, Hakob Tamazyan, Vahan Huroyan, Hrant Khachatrian, and Martin Danelljan. Analyzing local representations of self-supervised vision transformers. *arXiv preprint*, 2024. 2

[36] Shashanka Venkataramanan, Valentinos Pariza, Mohammadreza Salehi, Lukas Knobel, Spyros Gidaris, Elias Ramzi, Andrei Bursuc, and Yuki M. Asano. Franca: Nested matryoshka clustering for scalable visual representation learning. *arXiv preprint*, 2025. 1, 2, 8

[37] Haoqi Wang, Tong Zhang, and Mathieu Salzmann. SINDER: Repairing the singular defects of DINOv2. In *European Conf. Comput. Vis. (ECCV)*, 2024. 2, 8

[38] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 6, 7, 8

[39] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Inter. Conf. Mach. Learn. (ICML)*, 2020. 1

[40] Yangtao Wang, Enze Xie, Xingyi Song, Peize Sun, Ping Luo, Wenhai Wang, Zhe Li, Tong Xu, and Changsheng Deng. Self-supervised transformers for unsupervised object discovery using normalized cut. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 14595–14604, 2022. 2

[41] Yangtao Wang, Jie Xu, Enze Xie, Ding Liang, Ping Wang, Tong Lu, and Ping Luo. TokenCut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12): 15790–15801, 2023. 5, 6, 7, 3, 4, 8, 9

[42] Barry Payne Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3): 419–420, 1962. 3

[43] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, California Institute of Technology, 2010. 7

[44] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 5

[45] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianwei Bao, Zhuliang Yao, Qiang Dai, Han Hu, and Guoqiang Gao. Simmim: A simple framework for masked image modeling. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 9643–9653, 2022. 2

[46] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013. 6, 7, 8, 3, 4

[47] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2013. 6, 7, 9

[48] 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 Conf. Comput. Vis. (ECCV)*, 2024. 2, 8

[49] Yike Yuan, Kaicheng Cao, Yiheng He, Xiaojie Zhang, Jiaying Xu, Yichen Zhang, Ziwei Liu, and Yizhou Yu. DenseDINO: boosting dense self-supervised learning with token-based point-level consistency. In *Inter. Joint Conf. Artif. Intell. (IJCAI)*, pages 1765–1773, 2023. 2

[50] Junbo Zhang and Kaisheng Ma. Rethinking the augmentation module in contrastive learning: Learning hierarchical

augmentation invariance with expanded views. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022. 1

[51] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017. 7, 4, 5, 6

[52] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *Inter. J. Comput. Vis.*, 2019. 7, 4, 5, 6

[53] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *Inter. Conf. Learn. Represent. (ICLR)*, 2022. 1, 2, 5

Suppressing Non-Semantic Noise in Masked Image Modeling Representations

Supplementary Material

A. Self-Supervised Objectives

Contrastive Learning. In this work, we focus on the contrastive self-distillation objective as presented in Caron et al. [11]. Given an image x, two random augmentations are applied yielding the views u, v. The views are sent through a teacher-student framework giving $p = f_\theta(v) \in \mathbb{R}^d$ and $q = f_{\hat{\theta}}(u) \in \mathbb{R}^d$, where θ and $\hat{\theta}$ denote the teacher and student weights, respectively. The loss minimizes the cross-entropy

$$\mathcal{L}_{\text{CLS}} = -q^\top \log p. \tag{A.1}$$

Importantly, this loss is applied on the global representations, given by the `CLS` token in ViTs. The teacher and student share the same architecture, comprising a backbone and a projection head for the global representation. The teacher is updated as an exponential moving average of the student.

Masked Image Modeling. The core idea of the MIM objective is to reconstruct masked parts of an image when given visible parts as context. While the masking strategy varies between the MIM-based methods, the general setup can be summarized as follows. The input x is obscured by a random mask m and passed through an encoder to give a context $z = f(m(x))$ with which to make a prediction $\hat{s} = g(z)$ about the unmasked image x. The prediction can be made in the pixel space or in the latent space where the target s is given by passing x through the encoder. The loss is

$$\mathcal{L}_{\text{MIM}} = \ell(\hat{s}, s) \tag{A.2}$$

where ℓ is the mean squared error [22], euclidean distance [2] or negative cross entropy [16, 26, 29, 53]. Notably, several frameworks [26, 29, 53] employ a combination of contrastive loss over the global `CLS`-tokens in a multi-view setup with a local MIM-based loss over masked patch tokens.

B. Details on Activation and Distributions

In this section, we elaborate on the details of responses, activations, and distributions. As explained in the main text, we compute responses for all images in the dataset for each principal component. We binarize the activations, and compute the token-wise empirical distributions as individual Bernoulli distributions for each token in the image, $P_{d,n}$ from Section 3.3. As an ensemble, this can be taken as $P_d \sim \text{Multinomial}(2, N)$, forming a full distribution over the image.

Figure B.1. The relation of soft responses of DINOv2 to PC_1 in Figure 2 (left), compared with the multinomial distribution of activations P_1 from Figure 4 and Section 3.3 (right). The responses are binarized, and the probability distribution P_1 is computed as a multinomial distribution over all tokens in the image.

Divergence measures would be a natural choice to compare P_d, Q_d, however, we find that pure divergence measures such as Jensen-Shannon yield suboptimal scores in our testing. If $P_{d,n} = Q_{d,n} = 0.5$, a proper divergence such as the Jensen-Shannon requires that the distributions are equal, e.g., $D_{\text{JS}}(P_{d,n}, Q_{d,n}) = 1$. However, in this case we are unsure if individual activations actually agree or not. In other words, a proper divergence yields high scores when activation maps are highly uncertain.

In contrast, our proposed SI-score given in Eq. (6) yields $\text{SI}(P_{d,n}, Q_{d,n}) = \sqrt{0.5}$ for the same example, reflecting the inherent uncertainty in agreement in the two responses, and correctly identifies similar responses between real data and the non-semantic synthetic data.

Figure B.1 illustrates the relation between responses and empirical estimates of the distribution. To expand on Fig. 4 in the main article, Fig. B.2 shows the top 10 principal components for each model in our study, sorted by descending SI-scores.

B.1. Form of the SI-score

When we calculate the SI-scores in Section 3.3, we do so over the multinomial distributions P_d, Q_d. To clarify, we consider the inner products and norms as being taken over the *support set*, not the multivariate dimensions.

Let $P_d, Q_d \in \mathbb{R}^N$ and let $1 \in \mathbb{R}^N$. Define the augmented vectors

$$\tilde{P}_d = \big[P_d, 1 - P_d\big], \qquad \tilde{Q}_d = \big[Q_d, 1 - Q_d\big] \in \mathbb{R}^{2N}.$$

Then we calculate the scores by averaging over the multi-

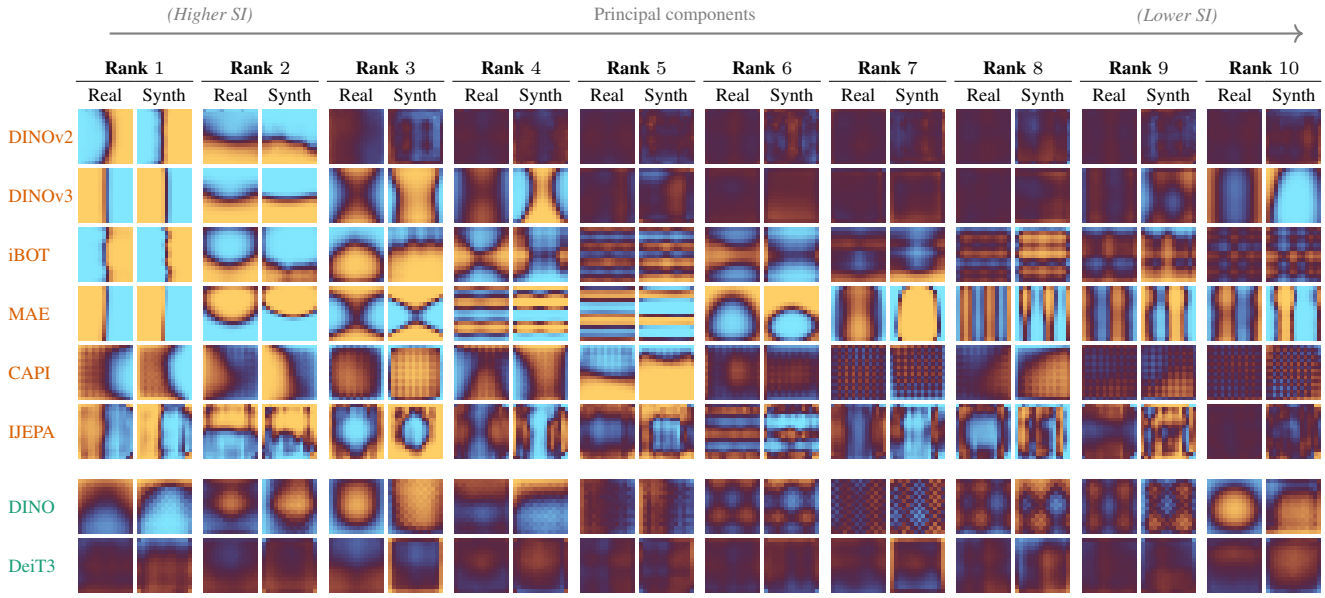


Figure B.2. Distributions for the top 10 principal components, ranked by semantic invariance (SI). **Real** columns correspond to P_d and **Synth** correspond with Q_d from Section 3.3, and each column compares the activations for real images from ImageNet, and generated synthetic images. MIM models are shown in the top rows; non-MIM models in the bottom rows. MIM models exhibit higher semantic invariance than the non-MIM models, as seen by the clear positional bias in the activations.

variate dimensions, yielding

$$s_d = \mathrm{SI}(P_d, Q_d) \tag{B.1a}$$

$$= \frac{1}{N} 1^\top \left(2 \frac{\langle \tilde{P}_d, \tilde{Q}_d \rangle}{\|\tilde{P}_d\|_2 + \|\tilde{Q}_d\|_2} \right), \tag{B.1b}$$

$$= \frac{2}{N} \sum_n \frac{P_{d,n} Q_{d,n} + (1 - P_{d,n})(1 - Q_{d,n})}{\sqrt{P_{d,n}^2 + (1 - P_{d,n})^2} + \sqrt{Q_{d,n}^2 + (1 - Q_{d,n})^2}}, \tag{B.1c}$$

where $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_2$ denotes the Euclidean inner product and norm (over the support set), respectively.

C. Generating synthetic images

Let $\Omega = \{1, \ldots, H\} \times \{1, \ldots, W\}$ and $X \in \mathbb{R}^{C \times H \times W}$. For each image, draw mixture weights

$$\mathbf{w} = (w_1, w_2, w_3) \sim \mathrm{Dir}(\alpha_1, \alpha_2, \alpha_3). \tag{C.1}$$

We generate three components independently:

1. **Pink Noise**: X_{pink} is zero-mean with isotropic power spectrum

$$\mathbb{E}\big[\, |\mathcal{F}\{X_{\text{pink}}\}(\xi)|^2 \,\big] \propto \|\xi\|^{-\beta}, \qquad \xi \in \mathbb{Z}^2 \setminus \{0\},$$

with $\beta \approx 2$, following the power-law slope typical of natural images [30].

2. **Modulated White Noise**: $X_{\text{white}} = M \odot W$, where $W \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and the nonnegative modulation $M = g(P)$ is a smooth field obtained from a pink process P (as above) and a bounded mapping g that sets the local standard deviation. This yields a heteroscedastic Gaussian field with variance $\sigma^2(x) = M(x)^2$ and long-range variance correlations.

3. **Gradient Field**: X_{grad} is a random low-degree polynomial—or equivalently, a very low-pass random field—concentrating energy near $\xi = 0$.

Synthesized images are then given by the convex mixture

$$X = w_1 X_{\text{white}} + w_2 X_{\text{pink}} + w_3 X_{\text{grad}}. \tag{C.2}$$

Assuming zero mean and independence between components, the expected power spectrum of X is

$$S_X(\xi) = \mathbb{E}\big[\, |\mathcal{F}\{X\}(\xi)|^2 \,\big] = \sum_{i=1}^{3} \mathbb{E}[w_i^2]\, S_{X_i}(\xi), \tag{C.3}$$

i.e., a convex combination of the components' spectra.

Simoncelli and Olshausen [30] show that natural images exhibit approximate scale invariance with a $1/\|\xi\|^\beta$ law in the power spectrum (amplitude $\sim 1/\|\xi\|^{\beta/2}$), together with large-scale illumination/contrast fluctuations. In the construction above, X_{pink} directly imposes the $1/\|\xi\|^\beta$ decay, giving second-order statistics aligned with natural image ensembles. Meanwhile, X_{grad} injects additional low-frequency near-DC energy, modeling global trends and illumination. Finally, X_{white} introduces spatially varying contrast via a pink variance field, capturing long-range correlations of local variance found in natural scenes.

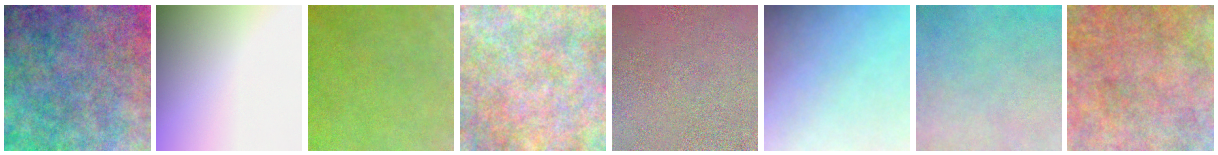


Figure C.1. Examples of generated synthetic non-semantic images.

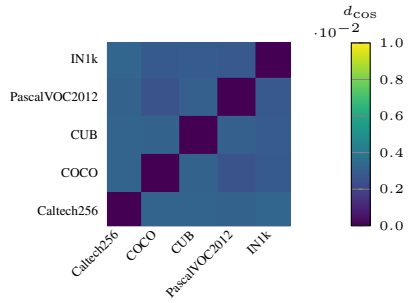


Figure D.1. Cosine distance between SI-scores for DINOv2 using various datasets for semantically informative images. The low values in $(.0025, .0032)$ indicate that the SI-score is consistent across datasets.

Thus $S_X(\xi)$ inherits a natural-image-like spectrum: a power-law falloff dominated by X_{pink}, boosted near $\xi = 0$ by X_{grad}, and with heteroscedasticity from X_{mw}. We illustrate examples of synthetic images in Fig. C.1.

During testing, we also experimented with standard probability distributions such as Gaussian or Uniform noise, in addition to simple mono-colored and gradient images, which provides similar responses to our proposed synthetic data. However, we considered these less appropriate given the mismatch in frequency response, which differs significantly from natural images. Hence, we designed our synthesis to better match key properties of natural images without additional semantic content.

D. SI-score sensitivity to dataset choice

Expanding on Sec. 4.3, we show the SI-score distance for each pair of datasets in Fig. D.1.

E. Effect of the Scaling Function

The filtering of SI scores acts as a smooth low-pass filter over the PCA spectrum. We selected the Fermi window as a smooth approximation of hard truncation, due to its precedence as a regularizer in image reconstruction. Figure E.1 illustrates the effect of the scaling function on the SI scores.

We ablate the effect of removing the scaling function in SOAP in Section 4.3, showing the downstream performance of kNN classification on the aggregated patch embeddings in Table 5. We provide additional results in Table E.1 by showing the effect for salient segmentation. We

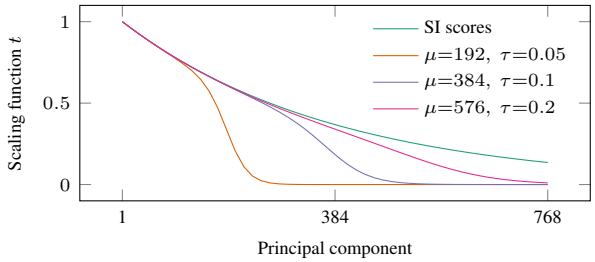


Figure E.1. Plot showing the effect of the scaling function, filtering with the Fermi window in Eq. (8), on semantic invariance (SI) scores with various choices of μ, τ. The μ parameter controls how many components are filtered, while τ controls the smoothness.

Table E.1. Ablation on the scaling function in Eq. (8), evaluated on salient segmentation for ECSSD [46]. Removing the scaling function from SOAP generally leads to reduced performance.

	SOAP without scaling			SOAP with scaling		
Pretrain	max F_β	IoU	Acc.	max F_β	IoU	Acc.
DINOv2	81.615	73.533	89.472	80.633	72.559	88.687
DINOv3	39.867	32.426	59.745	42.633	33.742	61.975
iBOT	64.702	58.401	76.419	66.557	60.167	78.340
CAPI	85.555	78.446	93.460	85.219	78.084	92.600
MAE	73.264	62.877	85.655	82.094	72.118	91.444
I-JEPA	32.646	24.327	68.927	40.239	31.162	71.406

see a boost in performance by 1–9% points for all models, except DINOv2 and CAPI, which have reduced performance by $< 1\%$. Since the majority of the SI-scores are > 0.6 for all models, projecting the representations with SOAP without scaling the SI-scores results in suppressing a majority of the principal components, reducing performance when useful information is encoded in any but the most semantic components. We chose to include the ablation with kNN in the main paper, as the salient segmentation task requires less fine-grained information about the patch contents, making kNN more informative as an ablation task.

F. Evaluation Details

Throughout the paper, we provide several evaluations and experiments. In this section, we exposition some of the details for each evaluation method.

F.1. TokenCut

We use the official TokenCut [41] implementation with their graph cut segmentation algorithm for the patch embeddings and the bilateral solver for edge-aware post-processing to

Table F.1. Ablation over the TokenCut parameter τ_{TC} on ECSSD [46] for DINOv2 and CAPI.

	DINOv2 ViT-B/16			CAPI ViT-L/14		
τ_{TC}	max F_β	IoU	Acc.	max F_β	IoU	Acc.
0.2	79.037	70.033	87.720	64.927	53.906	78.027
0.3	79.803	71.751	87.953	72.456	66.083	84.334
0.4	79.177	71.708	87.655	70.604	64.627	84.139

refine the segmentations up to original image size. TokenCut is a natural extension of spectral clustering and graph cuts to token representations, where patches are either classified as belonging to foreground or background by taking the inner product of the Fiedler vector of a filtered Gramian matrix over the tokens. Contrary to Wang et al. [41], we find that using the final output features yields better results for all models except MAE. Our reported results are thus for the out features for all models in our study, except MAE, for which we use the key features. We set $\tau_{\text{TC}} = 0.3$ for all models; Table F.1 shows this setting yield better results for DINOv2 and CAPI. Otherwise, we follow original implementation.

Selecting foreground partition using salient principal components. We also observe that some principal components have a strong center bias in the activations. This can be partially explained by object center bias in the training data. Comparing with the activation maps of synthetic input data in Fig. B.2 shows that in several cases the center bias is not present. This indicates that these components are responding to relative saliency or instance level correlations for each of the local patches, rather than an encoded positional center bias in the model.

Given a principal component v_d with salient activations, we can effectively determine which patches are likely to be part of the foreground or class level object. We find that we can improve zero-shot salient segmentation with TokenCut by selecting the foreground partition based on the patch with the highest response $\langle v_d, z \rangle$. In contrast, TokenCut selects the patch with maximum absolute value in its feature vector. The results in Table F.2 show out-of-the-box improvement across the board, where we use the strongest salient principal component to guide foreground selection for each model. We show results for all MIM models in our study, except I-JEPA which did not have a good salient principal component. We observed low performance on salient segmentation for all our experiments with I-JEPA, which suggests that the patch embeddings are not informative for this task.

F.2. kNN segmentation

We evaluate segmentation quality without any learnable parameters using a patch-level knearest neighbor approach. We first extract patch features from all training images using the frozen backbone and build a feature bank of ℓ_2-normalized patch embeddings paired with their ground truth labels, obtained by downsampling the segmentation masks to the patch grid via nearest-neighbor interpolation. At evaluation time, each query patch is classified by retrieving its $k = 30$ nearest neighbors from the feature bank using cosine similarity, with temperature-scaled ($\tau = 0.07$) weighted voting over the neighbor labels. This approach directly probes the spatial quality of frozen patch representations without introducing any learnable parameters, making it a good diagnostic for raw feature quality.

Table F.2. Using salient principal components to select foreground partition in TokenCut. Results are shows for corrected embeddings after cleaning with SOAP.

		Max. abs. val.			Max. Sal. PC reponse		
Model	Arch.	max F_β	IoU	Acc.	max F_β	IoU	Acc.
DINOv2	ViT-B/14	71.461	64.129	83.292	80.633	72.559	88.687
DINOv3	ViT-B/16	33.360	25.111	46.521	42.633	33.742	61.975
iBOT	ViT-B/16	64.432	57.978	80.093	66.557	60.167	78.340
CAPI	ViT-L/14	74.506	68.045	86.3234	85.219	78.084	92.600
MAE	ViT-B/16	80.525	70.705	90.506	82.094	72.118	91.444
Franca	ViT-B/14	77.660	70.830	87.577	84.176	76.985	91.514

In Tab. F.3 we provide additional results for PascalVOC [18], expanding the results for ADE20k [51, 52] that were reported in Tab. 3 in the main paper.

F.3. kNN classification

In this section we provide details on our kNN classification evaluation protocol. In the main article we report kNN classification by attention- and entropy-weighted aggregation of patch predictions; the exact details are in Sec. F.3.1. Furthermore, we report kNN classification on the averaged patch embeddings in Sec. F.3.2 for completeness.

F.3.1. kNN classification by weighted aggregation of patch predictions

In Tab. 4 in Sec. 4.2, we evaluate frozen patch features on ImageNet [17] using a patch-level knearest neighbor classification protocol with $k = 20$ and temperature $\tau = 0.07$. We extract patch tokens from the last layer of each frozen backbone and reduce their dimensionality to 256 using PCA. Features are ℓ_2-normalized and stored in a feature bank sharded across 2 GPUs in float16 precision. Each patch in a query image retrieves its k nearest neighbors from the training feature bank via cosine similarity, producing per-patch class probability distributions through temperature-scaled softmax weighting. The per-patch probabilities are then aggregated into a single image-level prediction using one of two weighting schemes:

1. ***CLS attention weighting***, which uses the attention weights from the `[CLS]` token in the last self-attention layer of the backbone to weight each patch's contribution, thus leveraging the model's own learned notion of patch importance.

Table F.3. kNN segmentation comparison of original vs. SOAP-corrected embeddings across backbones, reporting mean IoU and pixel accuracy (Acc) for the ADE20 [51, 52] and PascalVOC [18] benchmarks.

		ADE20k				PascalVOC			
		Original emb.		Corrected emb.		Original emb.		Corrected emb.	
Pretrain	Model	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
DINOv2	ViT-B16	40.25	74.60	40.81 ↑0.56	74.72 ↑0.12	73.34	93.74	74.07 ↑0.73	93.90 ↑0.16
DINOv3	ViT-B16	43.85	77.94	44.53 ↑0.68	78.09 ↑0.15	78.46	95.41	79.37 ↑0.91	95.57 ↑0.16
iBOT	ViT-B16	27.73	70.86	28.51 ↑0.79	71.29 ↑0.43	61.10	91.25	62.32 ↑1.22	91.53 ↑0.28
CAPI	ViT-L14	31.38	71.63	31.64 ↑0.26	71.78 ↑0.16	60.40	91.40	60.84 ↑0.44	91.51 ↑0.11
MAE	ViT-B16	11.88	58.00	13.74 ↑1.86	59.54 ↑1.54	27.99	82.62	31.31 ↑3.32	83.51 ↑0.89
I-JEPA	ViT-H14	20.95	60.27	21.26 ↑0.31	60.29 ↑0.01	57.77	89.22	58.49 ↑0.72	89.32 ↑0.10
DINO	ViT-B16	21.21	66.48	21.21 0.00	66.48 0.00	47.32	88.01	47.32 0.00	88.01 0.00

2. ***Entropy weighting***, which assigns higher weight to patches whose kNN probability distributions have lower entropy, favoring patches that yield more confident predictions. Specifically, each patch i produces a kNN class probability distribution $\mathbf{p}_i \in \mathbb{R}^C$, for which we measure the patch confidence via the Shannon entropy

$$H_i = -\sum_{c=1}^{C} p_{i,c} \log p_{i,c}.$$

We assign aggregation weights by

$$w_i = \frac{\exp(-H_i/t)}{\sum_j \exp(-H_j/t)},$$

so uncertain patches get lower weights and confident patches are favored. Here, $t > 0$ is a softmax temperature parameter.

We aggregate by CLS attention weighting for models trained with an instance discrimination objective (DINO, DINOv2, DINOv3, iBOT), as these methods explicitly train the `[CLS]` token to capture global image semantics through their contrastive or self-distillation losses, yielding meaningful attention distributions over patches. For models trained exclusively with a masked image modeling objective (I-JEPA, MAE, CAPI), the `[CLS]` token is either absent or not trained to aggregate global information, so its attention weights are not informative for patch weighting. We therefore use entropy-based aggregation for these models, which is agnostic to the pretraining objective and instead relies on the confidence of the per-patch kNN predictions themselves.

In Tab. F.4 we provide additional results for iNat2018 [34], expanding the results for ImageNet [17] that were reported in Tab. 4 in the main paper. We use $t = 0.5$ for ImageNet and $t = 0.25$ for iNAT2018 for the entropy aggregation—selected based on validation performance for CAPI.

F.3.2. kNN classification on averaged patch embeddings.

We additionally perform kNN classification on ImageNet [17] by average pooling the patch embeddings, and matching the validation embeddings to the k nearest embeddings from the training set. We follow the kNN evaluation script by Caron et al. [11], and set $k = 20$ number of neighbors and 0.07 temperature for the voting coefficient. We compare the top-1 and top-5 accuracies of the average of the patch embeddings, and the corrected patch embeddings. The results in Table F.5 show only modest improvements after correcting for invariant components. This is the expected result, as instance tasks are not as reliant on local semantics, and the benefit of SOAP may be dampened by the uniform aggregation, as spatial information about the locality of patches reduces due to averaging them out.

G. Linear evaluation protocols

Our intention with restricting evaluation to zero-shot protocols (TokenCut, kNN) is to measure the *intrinsic fidelity* of the representations, while *limiting confounding factors*. As SOAP uses linear PCA, a learnable head can adapt to suppress positional noise when this information is unhelpful for the task. Linear evaluation protocols are thus unsuitable for diagnosing positional noise, as they can adapt, unintentionally masking the issue. Indeed, linear probing, attentive probing, and linear segmentation yield similar results with SOAP; see Tab. G.1, Tab. G.2, and Tab. G.3. The difference in performance with and without SOAP is very low—this level of variation in performance is expected when probing with learnable heads, and the difference is thus too small to attribute any change in performance to SOAP. We describe each of these evaluation protocols in detail below.

In contrast, kNN directly probes representation geometry without learned transformations, making it a faithful diagnostic for raw feature quality [11, 44]. This matters in practice: using SSL representations out-of-the-box is common, and practitioners unaware of positional noise may encounter false positives from patch embeddings at similar relative locations.

G.1. Linear evaluation.

We perform linear evaluation on the averaged patch embeddings of the last output layer of each model on Ima-

Table F.4. Weighted kNN classification by aggregating patch prediction on iNat2018 [34] and ImageNet [17], expanding results from Tab. 4. We compare original vs. SOAP-corrected embeddings across backbones.

		ImageNet				iNat			
		Original emb.		Corrected emb.		Original emb.		Corrected emb.	
Pretrain	Model	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
DINOv2	ViT-B16	82.32	96.29	82.60 ↑0.28	96.30 ↑0.01	58.80	82.99	60.36 ↑1.56	83.75 ↑0.76
DINOv3	ViT-B16	81.47	95.57	81.48 ↑0.01	95.63 ↑0.05	58.25	79.52	58.76 ↑0.51	80.12 ↑0.60
iBOT	ViT-B16	71.45	90.03	71.72 ↑0.27	90.10 ↑0.08	27.64	49.06	27.70 ↑0.06	49.73 ↑0.66
CAPI†	ViT-L14	70.81	91.09	71.25 ↑0.43	91.33 ↑0.24	26.88	51.18	27.64 ↑0.76	52.25 ↑1.07
MAE†	ViT-B16	59.98	81.89	62.83 ↑2.85	84.38 ↑2.49	16.43	31.36	19.12 ↑2.70	36.61 ↑5.25
I-JEPA†	ViT-H14	75.38	91.55	75.63 ↑0.24	91.64 ↑0.09	17.29	35.34	18.01 ↑0.73	36.06 ↑0.72
DINO	ViT-B16	66.08	86.13	66.08 0.00	86.13 0.00	24.64	44.60	24.64 0.00	44.60 0.00

†Aggregation weighted by entropy for models with no class token objective; otherwise weighted by class attention.

Table F.5. kNN classification of average pooled patch embeddings on ImageNet [17]. We compare the original embeddings with the SOAP-corrected embeddings for each backbone, and report top-1 and top-5 accuracies.

		Original embeddings		Corrected embeddings	
Pretrain	Model	Acc@1	Acc@5	Acc@1	Acc@5
DINOv2	ViT-B16	77.064	91.624	77.100 ↑0.036	91.636 ↑0.012
DINOv3	ViT-B16	76.542	91.530	76.588 ↑0.046	91.612 ↑0.082
iBOT	ViT-B16	59.170	79.612	59.498 ↑0.328	79.918 ↑0.306
CAPI	ViT-L14	56.250	77.490	56.444 ↑0.194	77.742 ↑0.252
MAE	ViT-B16	47.488	69.168	47.758 ↑0.27	69.442 ↑0.274
I-JEPA	ViT-H14	71.382	86.144	71.390 ↑0.008	86.168 ↑0.024
DINO	ViT-B16	55.216	75.740	55.216 0.000	75.740 0.000

geNet [17]. We follow the standard protocol of training a single linear layer for classification on top of the frozen features for 100 epochs. We use a standard stochastic gradient descent optimizer (SGD) with a base learning rate of 0.001, momentum 0.9, cosine learning rate decay, and a batch size of 256 with 4 GPUs (effective batch size 1024). Following protocol, the learning rate is scaled by

$$\text{lr} = \frac{\text{base lr} \times \text{batch size} \times \text{num GPUs}}{256}.$$

The results in Tab. G.1 show minor changes in performance when the embeddings are corrected with SOAP, reflecting that suppressing positional noise has little effect when the classification head is learnable.

G.2. Attentive probing.

The attentive probing protocol replaces the global average pooling of patch tokens with a learnable attention mechanism that computes a weighted aggregation over the patch tokens before classification [6]. Specifically, a lightweight two-layer MLP computes per-patch attention logits, which are normalized via softmax to produce attention weights over the spatial positions. The attended feature is then passed to a linear classifier. This allows the probe to selectively focus on the most informative patches, which is particularly beneficial for methods where discriminative information is distributed across patch tokens rather than concentrated in a single global representation. We otherwise follow the linear evaluation protocol, training for 100 epochs with SGD, a base learning rate of 0.0025, momentum 0.9, cosine learning rate decay, and a batch size of 256 with 4 GPUs (effective batch size 1024). The results in Tab. G.2 show minor changes in performance when the embeddings are corrected with SOAP, reflecting that suppressing positional noise has little effect when the classification head is learnable.

G.3. Linear Segmentation.

We evaluate segmentation quality by training a linear segmentation head on top of frozen patch features on ADE20k [51, 52]. The segmentation head consists of a single 1×1 convolution applied to the spatial patch feature map, followed by bilinear upsampling to the original image resolution. We train with cross-entropy loss and SGD with momentum 0.9, polynomial learning rate decay with power 0.9, and a batch size of 32 per GPU across 4 GPUs (effective batch size 128) for 80 epochs. The learning rate is selected from $\{0.08, 0.04, 0.008\}$ based on validation mIoU for the baseline results of the original embeddings for each model. The crop size is set to be divisible by the patch size of the backbone: 512 for patch size 16 and 518 for patch size 14. Training images are augmented with random scaling (ratio 0.5–2.0×), random cropping, and random horizontal flipping; validation images are resized to the crop size. We report mean intersection over union (IoU) and per pixel accuracy in Tab. G.3, once again showing that learnable evaluation heads confound the effect of suppressing positional noise with SOAP.

H. Additional salient segmentation examples

We show additional examples of salient segmentation using TokenCut [41] in Figs. H.1 and H.2, for the DUTS [38] and DUTOMRON [47] datasets, respectively. The images displayed are from the first samples in the datasets, and were not cherry picked except to show different outcomes of using SOAP in the case of DUTOMRON. We show both the

Table G.1. Linear evaluation protocol.

		Original embeddings		Corrected embeddings	
Pretrain	Model	Acc@1	Acc@5	Acc@1	Acc@5
DINOv2	ViT-B16	81.13	96.01	81.15 ↑0.02	96.01 0.00
DINOv3	ViT-B16	76.77	93.89	76.76 ↓-0.01	93.90 ↑ 0.01
iBOT	ViT-B16	72.89	91.37	72.93 ↑0.04	91.35 ↓ -0.02
CAPI	ViT-L14	63.09	85.41	63.17 ↑0.08	85.46 ↑ 0.05
MAE	ViT-B16	50.60	74.83	50.63 ↑0.03	74.87 ↑ 0.04
I-JEPA	ViT-H14	74.99	90.65	74.99 0.00	90.69 ↑ 0.04
DINO	ViT-B16	66.36	86.43	66.39 ↑0.03	86.41 ↓ -0.02

Table G.2. Attentive probing.

		Original embeddings		Corrected embeddings	
Pretrain	Model	Acc@1	Acc@5	Acc@1	Acc@5
DINOv2	ViT-B16	84.97	97.20	85.00 ↑0.03	97.22 ↑ 0.02
DINOv3	ViT-B16	83.40	96.56	83.46 ↑0.06	96.61 ↑ 0.05
iBOT	ViT-B16	79.05	94.34	79.04 ↓-0.01	94.31 ↓ -0.03
CAPI	ViT-L14	81.75	95.84	81.72 ↓-0.03	95.93 ↑ 0.09
MAE	ViT-B16	67.80	87.44	67.90 ↑0.10	87.47 ↑ 0.03
I-JEPA	ViT-H14	77.66	92.84	77.60 ↓-0.06	92.77 ↓ -0.07
DINO	ViT-B16	72.56	90.58	72.50 ↓-0.06	90.58 0.00

Table G.3. Linear segmentation.

		Original embeddings		Corrected embeddings	
Pretrain	Model	IoU	Acc	IoU	Acc
DINOv2	ViT-B16	47.54	80.21	47.53 ↓-0.01	80.20 ↓ -0.01
DINOv3	ViT-B16	49.35	82.31	49.28 ↓-0.07	82.25 ↓ -0.06
iBOT	ViT-B16	35.61	75.97	35.61 ↑0.00	76.12 ↑ 0.15
CAPI	ViT-L14	41.96	78.73	41.98 ↑0.02	78.55 ↓ -0.18
MAE	ViT-B16	20.43	65.10	20.52 ↑0.09	65.48 ↑ 0.38
I-JEPA	ViT-H14	27.43	68.71	27.53 ↑0.10	68.77 ↑ 0.06
DINO	ViT-B16	28.95	71.28	28.88 ↓ -0.07	71.19 ↓ -0.09

coarse per-patch prediction maps, and the refined maps after using the bilateral solver for edge aware post-processing; see App. F.1 for more TokenCut details.

Figure H.1. Examples of salient segmentation from DUTS [38] using TokenCut [41] with frozen CAPI [16] on the raw embeddings (Raw pred.) and after correcting with SOAP (SOAP pred.). We show the predictions per patch and after refining the segmentation maps with the bilateral solver (BS). Suppressing positional noise with SOAP either matches or improves the zero-shot saliency maps. These examples are from the first samples in the dataset.

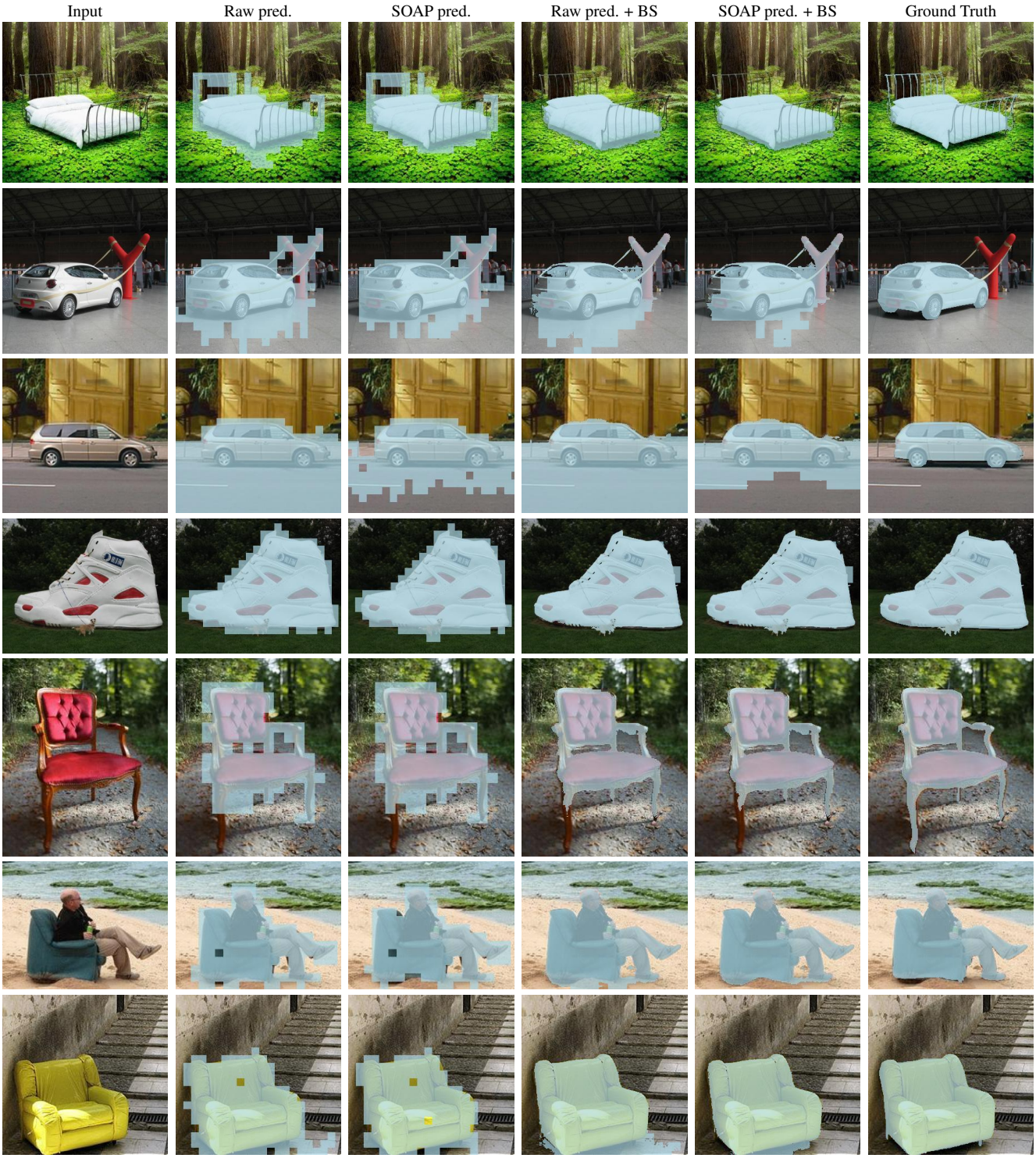


Figure H.2. Examples of salient segmentation from DUTOMRON [47] using TokenCut [41] with frozen CAPI [16] on the raw embeddings (Raw pred.) and after correcting with SOAP (SOAP pred.). Suppressing positional noise with SOAP either matches or improves the zero-shot saliency maps. We show the predictions per patch and after refining the segmentation maps with the bilateral solver (BS). These examples are from the first samples in the dataset, and were not cherry picked except to show different outcomes of using SOAP.