Which Direction to Choose? An Analysis on the Representation Power of Self-Supervised ViTs in Downstream Tasks

Yannis Kaltampanidis[1][0009-0008-9624-7783],
Alexandros Doumanoglou[1][0000-0002-4337-1720], and
Dimitrios Zarpalas[1][0000-0002-9649-9306]

Information Technologies Institute, Centre for Research and Technology Hellas, 1st Km Charilaou - Thermi Road, Thessaloniki, Greece
{ykalt, aldoum, zarpalas}@iti.gr
https://www.iti.gr

Abstract. Self-Supervised Learning (SSL) for Vision Transformers (ViTs) has recently demonstrated considerable potential as a pre-training strategy for a variety of computer vision tasks, including image classification and segmentation, both in standard and few-shot downstream contexts. Two pre-training objectives dominate the landscape of SSL techniques: Contrastive Learning and Masked Image Modeling. Features (or tokens) extracted from the final transformer attention block –specifically, the keys, queries, and values– as well as features obtained after the final block's feed-forward layer, have become a common foundation for addressing downstream tasks. However, in many existing approaches, these pre-trained ViT features are further processed through additional transformation layers, often involving lightweight heads or combined with distillation, to achieve superior task performance. Although such methods can improve task outcomes, to the best of our knowledge, a comprehensive analysis of the intrinsic representation capabilities of unaltered ViT features has yet to be conducted. This study aims to bridge this gap by systematically evaluating the use of these unmodified features across image classification and segmentation tasks, in both standard and few-shot contexts. The classification and segmentation rules that we use are either hyperplane based (as in logistic regression) or cosine-similarity based, both of which rely on the presence of interpretable directions in the ViT's latent space. Based on the previous rules and without the use of additional feature transformations, we conduct an analysis across token types, tasks, and pre-trained ViT models. This study provides insights into the optimal choice for token type and decision rule based on the task, context, and the pre-training objective, while reporting detailed findings on two widely-used datasets.

Keywords: ViT · SSL · DiNO · MAE · Directions · Hyperplane · Cosine Similarity

1 Introduction

Vision transformers [14,37,18,23,21], have shown exceptional performance in addressing complex computer vision and multi-modal tasks [8,40,32,39]. However, their effectiveness is highly dependent on the size of the training dataset, requiring an extensive amount of data to generalize effectively and avoid overfitting. Training these models from scratch is resource-intensive, both in terms of computational power and processing time. Given that related tasks, such as classification and segmentation, often share foundational knowledge, training separate models for each task from scratch is inefficient. Therefore, it has been proposed to train a large model once, using substantial data and resources to capture general knowledge, and then specialize or distill this model for specific downstream tasks by leveraging the knowledge acquired during the initial training phase.

Self-supervision, based on Masked Image Modeling (MIM) [19,2,7] or Contrastive Learning (CL) [6,11,7], has been proposed as a way for ViTs to capture this general knowledge from large datasets without the need for explicit labels. However, to achieve top performance, in most approaches [43,42,17,20] the pretrained ViT features undergo further transformations before the final prediction, in order to align the feature representations with the solution of the downstream task. Moreover, different methods utilize various feature types –such as query-key-value pairs from the last attention block, or the output tokens of the final feed forward layer– and employ diverse decision rules, being either hyperplane-based or direction similarity-based. Even though these approaches have demonstrated their effectiveness in solving downstream tasks, yet to our knowledge, a comprehensive evaluation of the intrinsic representation capabilities of unaltered self-supervised ViT features is missing from the literature.

In this work, we present a comprehensive analysis of the representational power of unaltered features from two self-supervised ViTs, pre-trained on a large dataset [35] using the previously mentioned self-supervision objectives [19,6]. To the best of our knowledge, this is the first study to examine all of the following aspects simultaneously: a) two ViTs pre-trained with different **self-supervised objectives** b) the five possible **token types** from the last transformer layer –keys, queries, values, and features before and after the final feed-forward block– c) two downstream **tasks**: image classification and segmentation, across both standard and 1-way-k-shot **contexts** and d) two commonly used prediction methods (or, as otherwise mentioned, **decision rules**), based on either hyperplane separation (linear probing) or cosine similarity.

We find that the hyperplane decision rule is more effective in semantic separability across most experiments, indicating that the cosine similarity between the tokens of these pretrained models is a suboptimal semantic proximity metric. Furthermore, our experiments indicate that the optimal token type depends heavily on the pre-training objective, task, context and decision rule –with some previously overlooked tokens proving to be the most effective. Beyond practical guidelines, our work challenges existing intuitions about ViT token interpretations and underscores the need for a deeper understanding of the role of each computational block within ViT layers.

2 Related Work

Self-Supervised Pre-Training Self-supervised pre-training [44] stands out as the leading method towards developing vision, vision-language, and various multi-modal foundation models [5, 39, 48]. The core strategies in this field involve CL, MIM, or an integration of both. On the one hand, CL methods [7, 41, 10, 11] utilize image augmentation techniques to generate views with similar or dissimilar semantic content, which, in turn, are considered for feature alignment. On the other hand, representation learning in MIM methods [2, 30, 40] is driven by masking patches and then reconstructing pixels or features.

Within ViTs, MIM approaches, largely represented by Masked Autoencoders (MAEs) [19, 45], typically **require** supervised **fine-tuning** to achieve competitive performance on downstream tasks [19, 2, 45, 50, 53, 27]. These models tend to exhibit narrow self-attention receptive fields [49] and capture texture-based features, making them best suited for dense prediction tasks such as object detection [29]. They also tend to exhibit great scaling with an increasing number of parameters which can be attributed to the high attention-map variance between transformer heads, meaning that a larger portion of the network can being utilized during fine-tuning [29].

ViTs trained with a CL framework, such as DiNO [6], generate semantic-level feature representations [1], allowing them to serve as universal feature extractors **without further fine-tuning** [38]. Similar to other contrastive learning methods, the self-attention maps of a ViT pre-trained with a DiNO objective, have a broad receptive field, effectively capturing global patterns, but CL also faces the challenge of *collapse into homogeneity* [29], leading to similar self-attention maps for all heads. This limitation has motivated the development of hybrid SSL techniques that combine MIM and CL learning objectives to address their respective limitations [28, 29, 24, 31].

Transfer-Learning Self-Supervised ViTs on Downstream Tasks In dense prediction tasks, the *patch* tokens of the final encoder layer are commonly used as regional embeddings [39, 47, 19], while the corresponding *class* token ([CLS]) remains the standard representation for image classification [14, 6, 50]. The ability of DiNO to induce discriminative saliency maps in the self-attention mechanism of ViTs [6] has inspired the extraction of features directly from the self-attention blocks. Beyond the vanilla approach that uses the class token for image classification tasks, various techniques have been explored that leverage the *key* tokens in the self-attention block of a frozen DiNO backbone (a ViT pre-trained with the DiNO objective), to tackle unsupervised segmentation and localization tasks [36, 43, 42], often employing a cosine similarity-based signal. Alternative methods that utilize a similar backbone seek to distill its knowledge in both standard [17] and few-shot [20] contexts through lightweight heads, using the backbone as a means to detect semantic similarities within the data.

Unlike CL which is able to build strong frozen backbones, MIM pre-training is best capitalized with task-specific fine-tuning. In [50], a MAE is pre-trained on a face dataset and subsequently fine-tuned on a dataset with facial expressions for facial affective analysis. In the medical domain, where annotated data are more scarce, self-pre-training [53] has been proposed as the paradigm of pre-training a MAE directly on the data of the downstream task. Subsequently, the learned encoder can be combined with a trainable linear head or a convolutional decoder to demonstrate superior performance compared to supervised baselines or baselines pre-trained on out-of-domain data. Beyond masking pixels, the MIM objective can be utilized to train lightweight student models that learn to reconstruct masked features from a larger state-of-the-art teacher, providing efficient solutions to solve the downstream segmentation task [46].

Relation to the Present Work In contrast to other studies that shed light on self-supervised ViTs from varying perspectives [11, 29, 49, 33, 26], our research adopts a latent space probing approach, regularly explored in mechanistic interpretability [51, 15, 16, 13, 34]. To our knowledge, this study is the first to rigorously evaluate the effectiveness of tokens derived from a frozen MAE to solve downstream tasks. This is even without taking into account the extensive breadth of this study on variation in token types, decision rules, and downstream tasks and contexts. Instead, previous work tends to prefer DiNO features for segmentation tasks with works considering frozen MAE features being almost non-existent, possibly due to the known fact that MIM works better when fine-tuned. Yet, a quantitative evaluation of the effectiveness of MAE's features compared to DINO's is currently missing, and our work addresses this gap with a detailed analysis. Our findings suggest that for semantic segmentation, while the downstream performance of MAE's features is inferior to DINO's, in some aspects the gap between them is not as large as one might initially believe.

Regarding DiNO, methods such as [17, 36, 43, 1, 20, 42] address the unsupervised segmentation task using token feature transformations derived from a frozen backbone. In our work, we differentiate and take a step back to meticulously assess the effectiveness of DiNO's **vanilla** tokens (without any transformations or extra processing) on downstream tasks using annotations, revealing to some extent the best starting point of those previous approaches. Furthermore, many previous approaches [36, 17, 43, 20, 42] have applied the cosine similarity rule to the tokens of a frozen DINO backbone, utilizing it as an implicit supervisory signal for semantic similarities. However to our knowledge, a rigorous assessment of its potential is missing from the literature and our work aims to address this, by being the first to assess the effectiveness of DiNO's features with the cosine rule on semantic tasks with ground-truth labels. Our work is also unique in providing a thorough study over the representation power of different token types, being either the attention layer's queries, keys, values, or tokens from either side of the final feed forward transformer block, expanding on the shallow analysis of [6]. In principle, our findings are aligned with previous work that prefers to use the attention layer's key tokens for semantic segmentation [36, 43, 1, 42] but also highlights a detailed comparison with the alternative tokens. Finally, image classification based on the two SSL approaches is also less explored in the literature [6, 20], and our work provides a detailed analysis, in terms similar to the segmentation task.

3 Approach

As briefly stated in the preface, our work aspires to address the following questions that innately arise when employing pre-trained ViTs in downstream tasks:

- Which self-supervised **pre-training objective** (MIM, CL, implemented by MAE and DiNO respectively) produces frozen backbones, which are more aligned to each **downstream task** (classification, segmentation)?
- Which ViT **token types** (queries $\boldsymbol{q}$, keys $\boldsymbol{k}$, values $\boldsymbol{v}$ from the final ViT's self-attention block or tokens $\boldsymbol{x}_1, \boldsymbol{x}_2$ from either side of the transformer feed forward block) provide semantically meaningful representations?
- Which **decision rule** (hyperplane based, cosine similarity based) should be utilized to separate the feature space into semantic regions?

Additionally, we also consider two downstream contexts: standard (where a plethora of labeled examples are available for learning a decision rule) and few-shot (where only a limited number of samples are available for the same purpose). In the following subsections, we aim to clarify these research questions by conducting experiments with combinatorial variability across pre-trained models, tasks, contexts, decision rules, and token types.

3.1 Self-supervised Pre-Training Objectives

This study concentrates on two well-known SSL ViT architectures: MAE [19] and DiNO [6]. MAE is part of the group of pre-training techniques focused on masked image modeling, whereas DiNO aligns with self-distillation and contrastive learning approaches. For the sake of computational efficiency, we opted for the smallest pre-trained ViT models accessible to the public (DiNO: ViT-S/8 21M parameters, MAE: ViT-B/16 86M parameters).

3.2 Downstream Tasks

We investigate the semantic representation power of ViT tokens in two exemplary downstream tasks: image classification and semantic segmentation. In the context of image classification, we develop a subset of ImageNet [35] resembling ImageNet-Tiny [22], constructed by randomly selecting 550 samples for each of ImageNet-Tiny's 200 classes. For image segmentation, we utilize the Broden dataset [3], which consolidates multiple datasets that are densely annotated [9, 12, 4, 25, 52]. Broden encompasses 1197 concepts distributed across approximately $63K$ images within 5 distinct concept categories (object, part, material, texture, color). In this research, we have excluded the color category to focus on the remaining categories which are deemed to hold greater semantic significance. Fig. 1 demonstrates the extensive annotations present in Broden, which incorporate low-level concept categories, such as material and texture, alongside high-level concepts, such as object and scene.

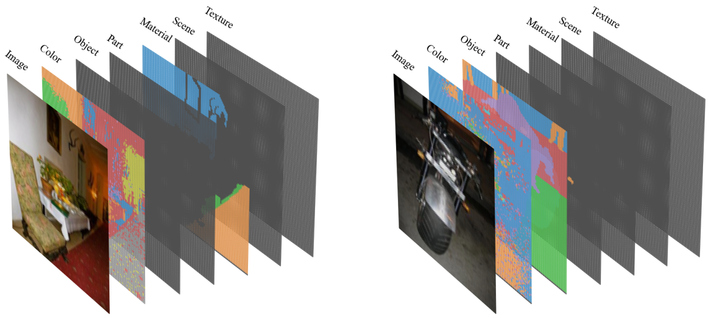


Fig. 1: Broden samples. Each image in the dataset is associated with multiple segmentation maps, covering six primary categories (color, object, part, material, scene, texture). For instance, the image in the left has a *color* and a *material* category-mapping whereas the image on the right a *color* and an *object* segmentation map.

We address **both** tasks through a unified binary classification framework, taking inspiration from [51]. Using independent binary classifiers offers a straightforward yet effective learning scheme suited for Broden's multilabel annotation structure. For image classification, we use the [CLS] token as a global feature representation of the entire image, whereas for semantic segmentation, we leverage the corresponding patch tokens to represent individual regions. Consequently, each object –whether the entire image for classification or an image-region for segmentation– is represented by a single feature vector, which serves as input to a set of binary classifiers. In other words, beyond the typical image classification task, the segmentation task is tackled by treating it as a patch classification problem.

3.3 Token Types

In our analysis we account for various token types derived from the final transformer layer to address the downstream tasks. We consider the query $\boldsymbol{q}$, key $\boldsymbol{k}$, and value $\boldsymbol{v}$ tokens of the self-attention block (Fig. 2 top), the output of the self-attention block, denoted as $\boldsymbol{x}_1$ and the output tokens of the feed forward block (MLP), referred to as $\boldsymbol{x}_2$ (Fig. 2 bottom).

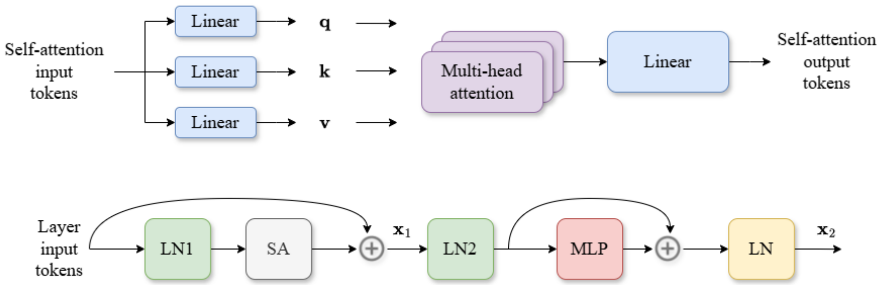


Fig. 2: (Top) Multi-head attention schematic diagram. $\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v} \in \mathbb{R}^D$ depict the *queries, keys, values* tokens respectively, with D representing the ViT's embedding dimension. (Bottom) Schematic diagram of the transformer's final layer, where LN denotes layer normalization and SA represents the multi-head self-attention mechanism. Note that the final normalization layer (LN) is applied exclusively at the last transformer layer. We denote $\boldsymbol{x}_1 \in \mathbb{R}^D$ the transformer tokens prior to the MLP and the second layer normalization layer, while $\boldsymbol{x}_2 \in \mathbb{R}^D$ the output-tokens after the MLP (layer output).

3.4 Classifier Decision Rules

We examine the semantic separability of ViT tokens using two different decision rules: hyperplane-based and cosine similarity-based. As illustrated in Fig. 3, each classification rule is associated with a distinct decision boundary, dissecting the feature space into two disjoint subspaces.

Specifically, the hyperplane rule is comprised of a normal vector $\boldsymbol{w}$ and a bias term b, defining the orientation and position of the hyperplane respectively. A feature vector $\boldsymbol{z}$ is classified positively if $\boldsymbol{w}^T\boldsymbol{z} - b \geq 0$. In contrast, the cosine decision rule defines a convex cone via a conical axis vector $\boldsymbol{\alpha}$ and an angular threshold θ, such that $\boldsymbol{z}$ is positively classified if $\arccos(\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_2} \cdot \frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|_2}) \leq \theta$, with $\cdot$ denoting the dot product.

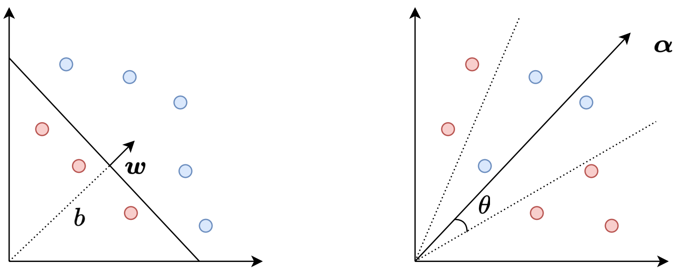


Fig. 3: Classifier decision rules. (Left) Hyperplane classifier $(\boldsymbol{w}, b)$. (Right) Cosine similarity classifier $(\boldsymbol{\alpha}, \theta)$. Each classifier dissects the feature space into two disjoints subspaces. Positively classified samples are depicted in blue, while negatively classified samples are illustrated in red.

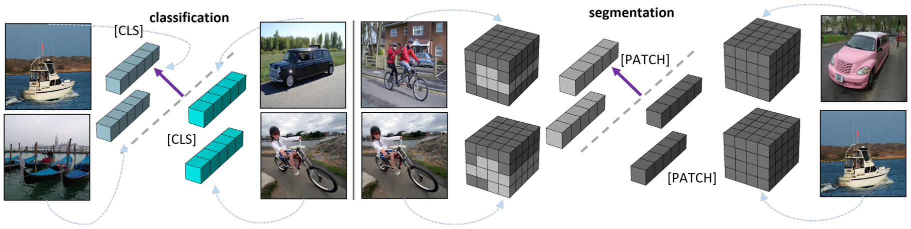


Fig. 4: Hyperplane decision rule: The class token represents the global image content, while individual image regions are represented by their corresponding patch tokens. A hyperplane is learned for each image class or semantic concept to distinguish positive samples from negative ones.

Concept Templates: Both decision rules are associated with a class-specific (in image classification) or concept-specific (in semantic segmentation) *directional* vector, a *threshold* and a *projection function*, which altogether may be utilized to classify a feature vector for the downstream task. We use the term *concept template* to encompass these attributes and also refrain from making explicit distinction regarding the label type of each downstream task (*image class* vs *patch concept*) as we treat both tasks within a common framework of similar principles. In the rest of the paper we will mostly refer to the downstream task's labels as *concept labels*, when in fact for image classification these labels correspond to image classes.

Formally, given the dimensionality of the embedding space D, a feature vector $\boldsymbol{z} \in \mathbb{R}^D$ and a concept $c \in \mathbb{N}$, the concept template is a triplet $\tau_c \coloneqq (\boldsymbol{d}, t, f)$, where $\boldsymbol{d} \in \mathbb{R}^D$ is the directional vector, $t \in \mathbb{R}$ is the threshold and $f(\boldsymbol{z}; \boldsymbol{d}) : \mathbb{R}^D \to \mathbb{R}$ is the projection function.

The concept template τ_c detects the existence of concept c in the feature vector $\boldsymbol{z}$ (positive classification) if:

$$f(\boldsymbol{z}; \boldsymbol{d}) \geq t \tag{1}$$

In the case of a hyperplane decision rule: $\boldsymbol{d} \coloneqq \boldsymbol{w}, t \coloneqq b$ and $f(\boldsymbol{z}; \boldsymbol{w}) \coloneqq \boldsymbol{w}^T \boldsymbol{z}$, while for a cosine decision rule: $\boldsymbol{d} \coloneqq \boldsymbol{\alpha}$, $t \coloneqq \cos(\theta)$ and $f(\boldsymbol{z}; \boldsymbol{\alpha}) \coloneqq \frac{1}{\|\boldsymbol{\alpha}\|_2 \ \|\boldsymbol{z}\|_2} \boldsymbol{\alpha}^T \boldsymbol{z}$. Based on the underlying decision rule, we distinguish two cases of concept templates: **hyperplane-templates** and **cosine-templates**.

3.5 Analysis Framework

This section provides details on how we learn the concept templates. In this and the rest of the sections, we often use the terms *concept template* and *classifier* interchangeably, preferring the former to emphasize its geometric interpretation and the latter to focus on its functional application.

Hyperplane Templates: To compute hyperplane templates, for each concept, we learn a hyperplane classifier $(\boldsymbol{w}, b)$ with the process illustrated in Fig. 4.

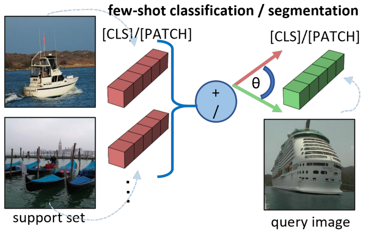


Fig. 5: Cosine similarity decision rule: In the few-shot context, the concept template's direction is derived by averaging intra-class token representations extracted from the support set. Specifically, a token originating from the query image set is classified positively if the cosine similarity between the token and the template's direction exceeds a threshold θ.

Given a training feature dataset $D_f : \{(\boldsymbol{z}_i, c_i), i = 1, \ldots, N\}$, where $\boldsymbol{z}_i \in \mathbb{R}^D$ represents the feature vector of an object (image/image-region) and $c_i \in \mathbb{N}$ represents its ground-truth label, we construct a positive sample pool for each concept c, denoted as $D_c^+ = \{\boldsymbol{z}_i \mid (\boldsymbol{z}_i, c_i) \in D_f,\ c_i = c\}$, and a corresponding negative sample pool, $D_c^- = \{\boldsymbol{z}_i \mid (\boldsymbol{z}_i, c_i) \in D_f,\ c_i \neq c\}$, where $|D_c^-| \gg |D_c^+|$. In semantic segmentation, when forming the negative sample pool for a concept c_i, we only consider concepts within the same primary category as c_i. To manage the significant class imbalance between the two sample pools, we initially limit the size ratio of $D_c^- : D_c^+$ to be no more than $20 : 1$ by random subsampling. During template learning, we conduct five rounds of hard negative mining, following [51]. In each of these rounds, the hyperplane template is fitted to the mined dataset over 3 epochs, ensuring a positive-to-negative sample ratio of $1 : 2$. The evaluation of **each** learned template is performed on a reserved test-set (approximately $10K$ image samples for ImageNet and $18K$ image samples for Broden from its validation split) via a set of **balanced** binary classification metrics.

Cosine Templates: Since the cosine decision rule is frequently utilized in unsupervised settings [36, 17, 43, 42, 20] including few-shot contexts, we **explicitly** consider learning cosine-templates in a few-shot regime by constructing support-query image sets for template learning and evaluation. The directional vector $\boldsymbol{\alpha}$ and similarity threshold $t \doteq \cos(\theta)$ of cosine templates, are computed in a non-parametric 1-way-k-shot setting. For a concept $c \in \mathbb{N}$, we construct a *support image set* $S_{c,k}$ by randomly sampling k training images that contain c. $S_{c,k}$ is further processed to construct the respective positive and negative *support feature pools* $D_{c,k}^+, D_{c,k}^-$; Notice that for image classification $D_{c,k}^- = \emptyset$, as every image in the support set is mapped to a single feature vector ([CLS] token). The cosine template's directional vector $\boldsymbol{\alpha}$ is then computed by averaging the positive support features:

$$\boldsymbol{\alpha} = \frac{1}{|D^+_{c,k}|} \sum_{\boldsymbol{z} \in D^+_{c,k}} \boldsymbol{z} \tag{2}$$

while the angular threshold θ is computed by maximizing the F1-score of the classifier τ_c on the *support feature set*:

$$\theta = \underset{\hat{\theta}}{\operatorname{argmax}}(F1(\hat{\theta}, D_{c,k}; \boldsymbol{\alpha}, f)) \tag{3}$$

where $F1$ is the F1 score of a classifier $\tau_c = (\boldsymbol{\alpha}, t, f)$ computed on the support feature set $D_{c,k} = D^+_{c,k} \cup D^-_{c,k}$, given the directional vector $\boldsymbol{\alpha}$ and the cosine-similarity projection function as f. Due to the fact that we use an empty $D^-_{c,k}$ for image classification, in Eq. (3) we consider the smallest possible angle θ that maximizes F1 score. The overall process is illustrated in Fig. 5. Furthermore, we vary $k \in \{1, 5, 10, 50, 100, 500\}$, leveraging different proportions of the available data. Finally, the templates are evaluated on a balanced randomly sampled query test set of 50 positive and 50 negative images using the same set of balanced binary classification metrics as in the hyperplane templates. Due to the stochastic nature of this 1-way-k-shot setting, we average and present the results from $N = 10$ independent trials reporting mean scores and their standard deviation.

4 Experimental Results

The subsequent subsections detail the outcomes of our comprehensive experimental evaluation, structured by downstream task and decision rule. In our analysis, the term *token performance* is used to denote the efficacy of concept templates that incorporate a particular token. It is important to highlight that for image classification tasks, the mentioned tokens refer to the [CLS] tokens, while for image segmentation, they pertain to patch tokens. Lastly, we underline that all the binary performance metrics presented in this work are **balanced**.

4.1 Task: Classification. Rule: Hyperplane

TLDR: We observe a substantial disparity in the classification performance of the hyperplane template between the pre-trained MAE and DiNO models. While MAE tokens resemble the performance of random classifiers, DiNO demonstrates exceptional classification capacity. Specifically, DiNO's $\boldsymbol{x}_2$ token is particularly well-suited for classification tasks via linear probing, while MAE should not be considered in this context.

Details: Fig. 6 (Left) compares MAE and DiNO tokens in terms of accuracy. Most of MAE tokens approximately score an accuracy of 0.5, which is equivalent to a random classifier. This may be attributed to the fact that the [CLS] token is not participating in the MAE's loss function. In contrast, DiNO attains its maximum accuracy with $\boldsymbol{x}_2$ (0.946). A detailed analysis of DiNO's token performance is presented in Fig. 6 (Right). We observe near-perfect precision for

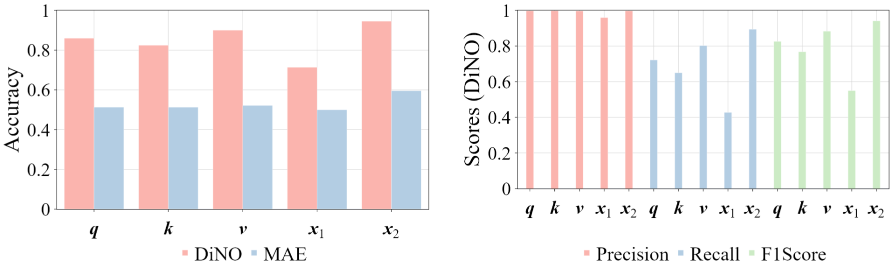


Fig. 6: Hyperplane-template classification: (Left) Accuracy between DiNO and MAE tokens. (Right) Precision, recall and F1 metrics for DiNO tokens.

$\boldsymbol{q}$, $\boldsymbol{k}$, $\boldsymbol{v}$, and $\boldsymbol{x}_2$ (> 0.99), while $\boldsymbol{x}_1$ achieves a precision of 0.96. This enables the construction of a hyperplane with minimal false positives (FP) across all tokens. Furthermore, $\boldsymbol{x}_2$ exhibits the highest recall (0.89), followed by $\boldsymbol{v}$ (0.80), $\boldsymbol{q}$ (0.72) and $\boldsymbol{k}$ (0.65). These results indicate that $\boldsymbol{x}_2$ provides the optimal linear separability of semantic concepts.

Notably, $\boldsymbol{x}_1$ demonstrates the lowest performance across all evaluated metrics. To better understand this phenomenon, we also assess the performance of $\boldsymbol{x}_1$ after layer normalization, which we denote as $\boldsymbol{x}_n$. Table 1 presents the impact of the normalization layer on DiNO's $\boldsymbol{x}_1$ hyperplane classification metrics. Layer normalization positively affects the semantic linear separability of the feature space. However, a more detailed analysis of the effects of layer normalization is beyond the scope of this work.

Table 1: Layer normalization effects on DiNO's $\boldsymbol{x}_1$ performance metrics.

DiNO	**Accuracy**	**Precision**	**Recall**	**F1**
$\boldsymbol{x}_1$	0.714	0.959	0.427	0.550
$\boldsymbol{x}_n$	0.940	**0.997**	0.884	0.935
$\boldsymbol{x}_2$	**0.946**	**0.997**	**0.894**	**0.941**

4.2 Task: Classification. Rule: Cosine

TLDR: Similar to hyperplane-based classification, DINO outperforms MAE under the cosine similarity decision rule. Notably, DINO's $\boldsymbol{x}_1$ token achieves the highest accuracy and F1 scores. Furthermore, MAE shows substantial improvement with cosine templates compared to the hyperplane decision rule, with its $\boldsymbol{k}$ token yielding the highest accuracy and F1 score in this context. Finally, increasing the support set size beyond 50 samples results in diminishing gains in average accuracy and F1 scores for both models.

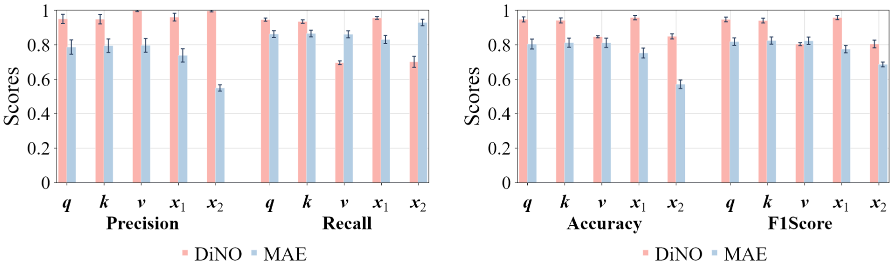


Fig. 7: Cosine-template classification with $k = 500$ support samples per concept. (Left) Precision and recall comparison between MAE and DiNO tokens. (Right) Accuracy and F1 score comparison. The error bars denote the standard deviation across 10 independent trials.

Details: Fig. 7 shows the classification metrics for DiNO and MAE tokens using the cosine decision rule, averaged over 10 independent trials with $k = 500$ support images per concept. DiNO's $\boldsymbol{x}_1$ emerges as the optimal token, achieving the highest accuracy (0.958 ± 0.01) and F1 score (0.958 ± 0.01), while $\boldsymbol{q}$ and $\boldsymbol{k}$ perform similarly. Although all DiNO tokens demonstrate high precision, $\boldsymbol{v}$ and $\boldsymbol{x}_2$ exhibit the lowest recall in this setting. For MAE, $\boldsymbol{k}$ achieves the highest accuracy (0.812 ± 0.03) and F1 score (0.824 ± 0.02), while $\boldsymbol{q}$ and $\boldsymbol{v}$ demonstrate similar performance. Notably, $\boldsymbol{x}_2$ exhibits the highest recall (0.929 ± 0.02), making it particularly well-suited for critical risk detection applications where minimizing false negatives (FN) is essential.

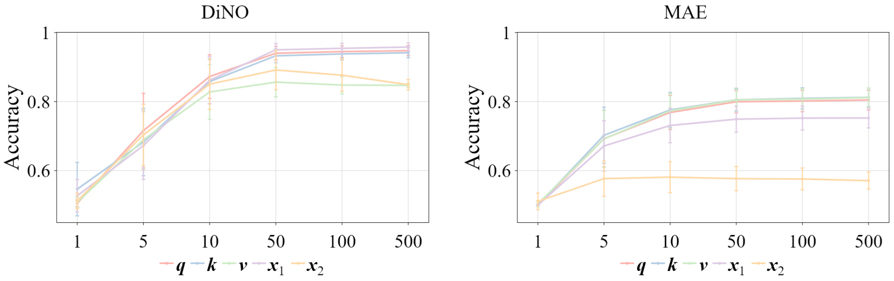


Fig. 8: Cosine-template classification accuracy for $k \in \{1, 5, 10, 50, 100, 500\}$ support samples per concept, for DiNO (Left) and MAE (Right). Error bars denote standard deviation across 10 independent trials.

Fig. 8 illustrates the impact of k (number of support samples used to compute cosine-templates) on model accuracy. Notably, performance gains diminish significantly beyond 50 samples. However, increasing the number of support samples leads to a more representative support set, thereby reducing the standard deviation across trials.

4.3 Task: Segmentation. Rule: Hyperplane

TLDR: Both MAE and DINO demonstrate strong and comparable hyperplane-template accuracy, yet inferior to the scores for image classification. Between the two pre-trained models, DINO achieves a higher overall F1 score. Notably, $\boldsymbol{k}$ is the optimal token in terms of overall accuracy and F1 score for both models. However, while $\boldsymbol{k}$ consistently yields the highest F1 score across all concept categories in DINO, MAE shows a slight advantage for $\boldsymbol{x}_2$ over $\boldsymbol{k}$ when considering textures, objects, or scenes.

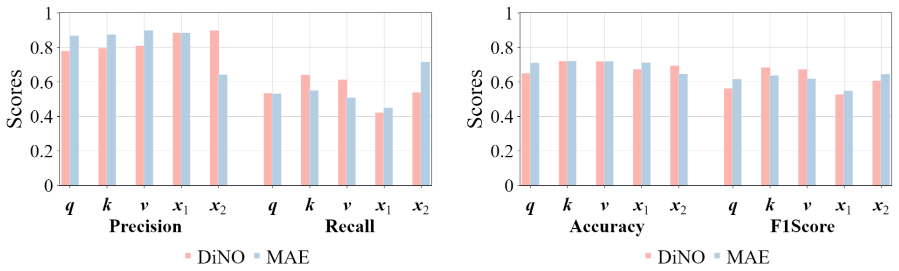


Fig. 9: Hyperplane-template segmentation: (Left) Precision and recall comparison between MAE and DiNO tokens. (Right) Accuracy and F1 score comparison between MAE and DiNO tokens.

Details: Fig. 9 presents the overall hyperplane-template segmentation performance of DiNO and MAE tokens. Among DiNO tokens, $\boldsymbol{k}$ achieves the highest accuracy (0.721) and F1 score (0.684), while $\boldsymbol{v}$ attains similar accuracy (-0.001) but a slightly lower F1 score (-0.01). DiNO's $\boldsymbol{x}_2$ exhibits the highest precision (0.899) making it particularly well-suited for quality assurance applications where minimizing false positives (FP) is essential. MAE's $\boldsymbol{k}$ achieves the highest accuracy (0.721), while $\boldsymbol{x}_2$ attains the highest F1 score (0.645). Comparing the two, $\boldsymbol{k}$ appears to be the optimal choice, with a significantly higher accuracy (+0.07) and only a slight reduction in F1 score (-0.01). On the other hand, $\boldsymbol{v}$ demonstrates the highest precision (0.899), while $\boldsymbol{x}_2$ excels in recall (0.716). Notably, $\boldsymbol{x}_2$ shows a substantial precision drop compared to $\boldsymbol{x}_1$ (-0.24), coupled with a significant recall increase (+0.27). This suggests that critical semantic information may be lost in $\boldsymbol{x}_2$, likely in favor of low-level textural patterns, as $\boldsymbol{x}_2$ tokens are processed through a decoder for masked patch reconstruction.

Fig. 10 presents the F1 scores of hyperplane templates, grouped by semantic category. DiNO's $\boldsymbol{k}$ token consistently outperforms others regardless of the semantic category. Among all concept categories, DiNO performs better in part

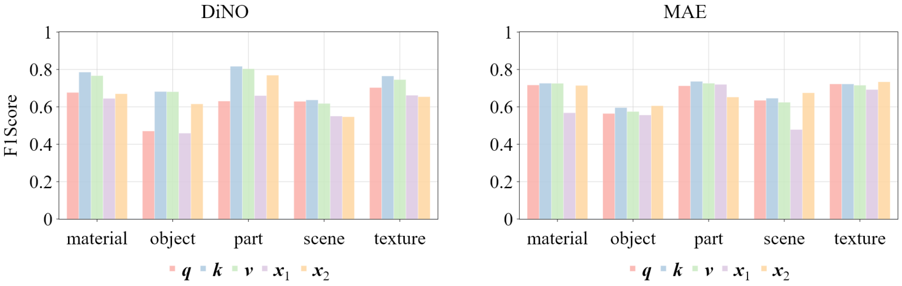


Fig. 10: F1 score for DiNO (Left) and MAE (Right) templates, grouped by label category. The scores for each concept template, are grouped and averaged according to their Broden primary semantic category (material, object, part, scene, texture).

(0.817), material (0.786), and texture (0.765) but is less effective in object (0.682) and scene (0.637) categories. This pattern suggests that DiNO's $\boldsymbol{k}$ token excels at segmenting fine-grained semantic concepts, aligning with prior findings [1]. For MAE, the $\boldsymbol{k}$ token achieves the highest F1 scores in part (0.734) and material (0.727), whereas $\boldsymbol{x}_2$ leads in object (0.606), scene (0.676), and texture (0.734). Notably, the most significant disparity occurs in the part category, where $\boldsymbol{k}$ significantly outperforms $\boldsymbol{x}_2$ (+0.08). Interestingly DiNO achieves higher F1 score compared to MAE, in all categories except for scene (-0.13).

Fig. 11 further examines precision and recall across concept categories. In terms of precision, DiNO's $\boldsymbol{x}_2$ token achieves the highest overall score (Fig. 9), a trend that persists across most categories, except for texture, where $\boldsymbol{x}_1$ exhibits superior precision (+0.09). For MAE, $\boldsymbol{v}$ achieves the highest precision in object and part categories, while $\boldsymbol{x}_1$ is the most precise in material, scene, and texture. Notably, MAE's $\boldsymbol{x}_1$ consistently outperforms $\boldsymbol{x}_2$ in precision across all Broden categories. When analyzing recall, DiNO's $\boldsymbol{k}$ token demonstrates the best performance in material, object, and part categories, whereas $\boldsymbol{q}$ and $\boldsymbol{v}$ emerge as the top-performing tokens for scene and texture, respectively. Regarding recall for MAE, $\boldsymbol{x}_2$ consistently performs best across all categories.

Cross-model comparisons reveal that MAE's $\boldsymbol{v}$ or $\boldsymbol{x}_1$ tokens achieve higher precision than DiNO in part, scene, and texture categories, while DiNO tokens exhibit superior precision in material and object categories, reinforcing its strength in segmenting individual structures.

4.4 Task: Segmentation. Rule: Cosine

TLDR: For both MAE and DiNO, the utilization of the cosine-decision rule is evidently inferior to hyperplane-templates, as their overall accuracy across all concepts is not significantly superior to a random-classifier (≈ 0.6). However, both models can achieve notable accuracy and F1 scores for textural concepts.

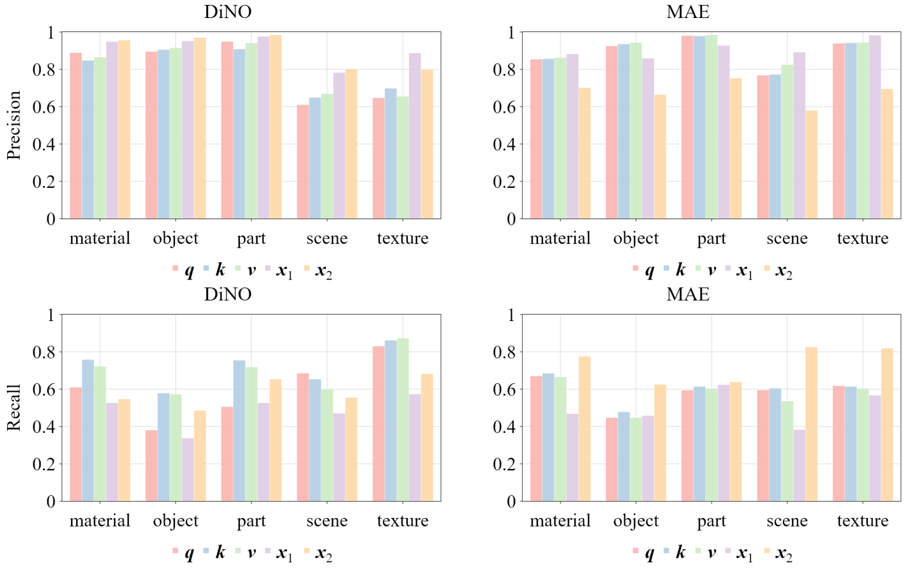


Fig. 11: Precision (Top) and recall (Bottom) for DiNO (Left) and MAE (Right) tokens, grouped by label category.

Details: Fig. 12 presents the overall segmentation metrics for DiNO and MAE tokens under the cosine decision rule, averaged over 10 trials with $k = 500$ support images per concept. In both models, $\boldsymbol{q}$ tokens achieve the highest accuracy (DiNO: 0.574, MAE: 0.622) and F1 scores (DiNO: 0.419, MAE: 0.464). While MAE outperforms DiNO, both models perform significantly worse compared to the hyperplane decision rule, highlighting the limitations of the cosine decision rule in this setting.

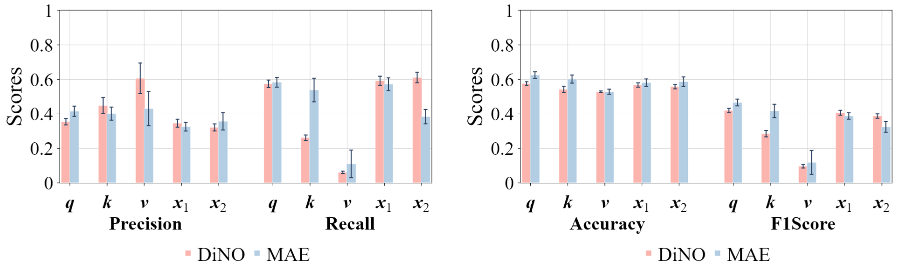


Fig. 12: Cosine-template segmentation with $k = 500$ support samples per concept. (Left) Precision and recall comparison between MAE and DiNO tokens. (Right) Accuracy and F1 score comparison. The error bars denote the standard deviation across 10 independent trials.

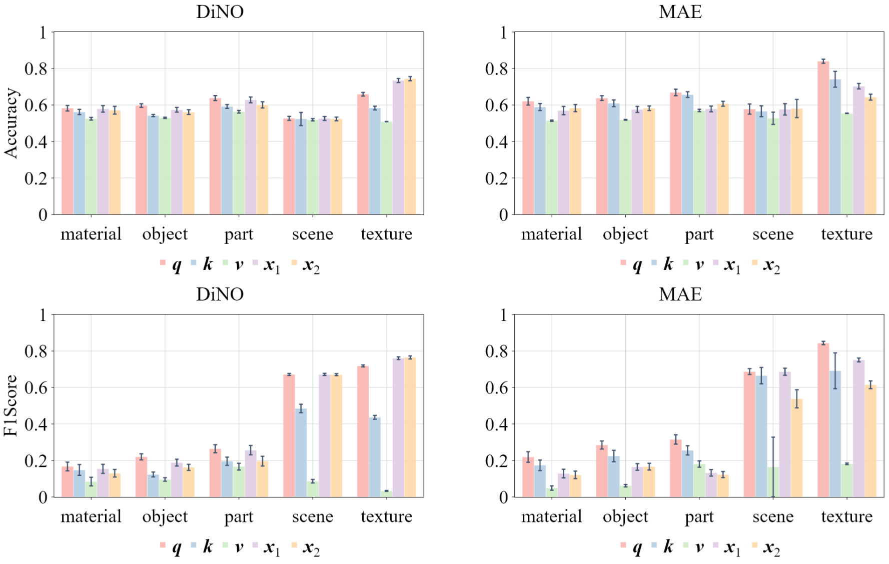


Fig. 13: Cosine-template segmentation with $k = 500$ support samples per concept. (Top) Accuracy for DiNO (Left) and MAE (Right) tokens, grouped by label category. (Bottom) F1 score for DiNO (Left) and MAE (Right) tokens, grouped by label category. The error bars denote the standard deviation across 10 independent trials.

Figure 13 shows the accuracy and F1 scores for cosine templates, grouped by semantic category. Notably, both models perform well on textural concepts, and partially well (low accuracy, but higher F1 score) on scenes. MAE's $\boldsymbol{q}$ token achieves an average accuracy of 0.840 and an F1 score of 0.844, while DiNO's $\boldsymbol{x}_2$ token reaches an average accuracy of 0.744 and an F1 score of 0.765. Fig. 14 illustrates the impact of k (number of support samples used to compute cosine-templates) on model accuracy. Similar to cosine-template classification, performance gains diminish significantly beyond 50 samples.

4.5 Qualitative Results

In the following subsection, we qualitatively examine the segmentation capabilities of learned concept templates on **unseen** image samples. Based on our previous analysis, we use the $\boldsymbol{k}$ tokens for hyperplane templates and the $\boldsymbol{q}$ tokens for cosine templates for both DiNO and MAE.

In Fig. 15, we present image samples organized by their primary category (material, object, part, scene, texture). Within each category, we select five representative concepts, and examine one image sample per concept. The selected concepts are chosen to ensure a balanced representation of DiNO hyperplane template performance, incorporating both the highest and lowest F1 scores. To

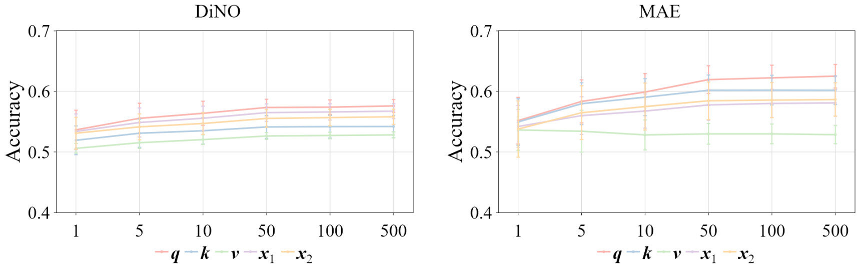


Fig. 14: Cosine-template segmentation accuracy for $k \in \{1, 5, 10, 50, 100, 500\}$ support samples per concept, for DiNO (Left), MAE (Right). Error bars denote standard deviation across 10 independent trials.

improve visualization clarity, the representative image is selected from the test set based on the largest area coverage of the corresponding concept. Additionally, for each image sample, we provide its ground truth segmentation mask (GT Mask) alongside the predicted masks generated by hyperplane-based (DiNO-C, MAE-C) and cosine decision rule-based (DiNO-C, MAE-C) template models.

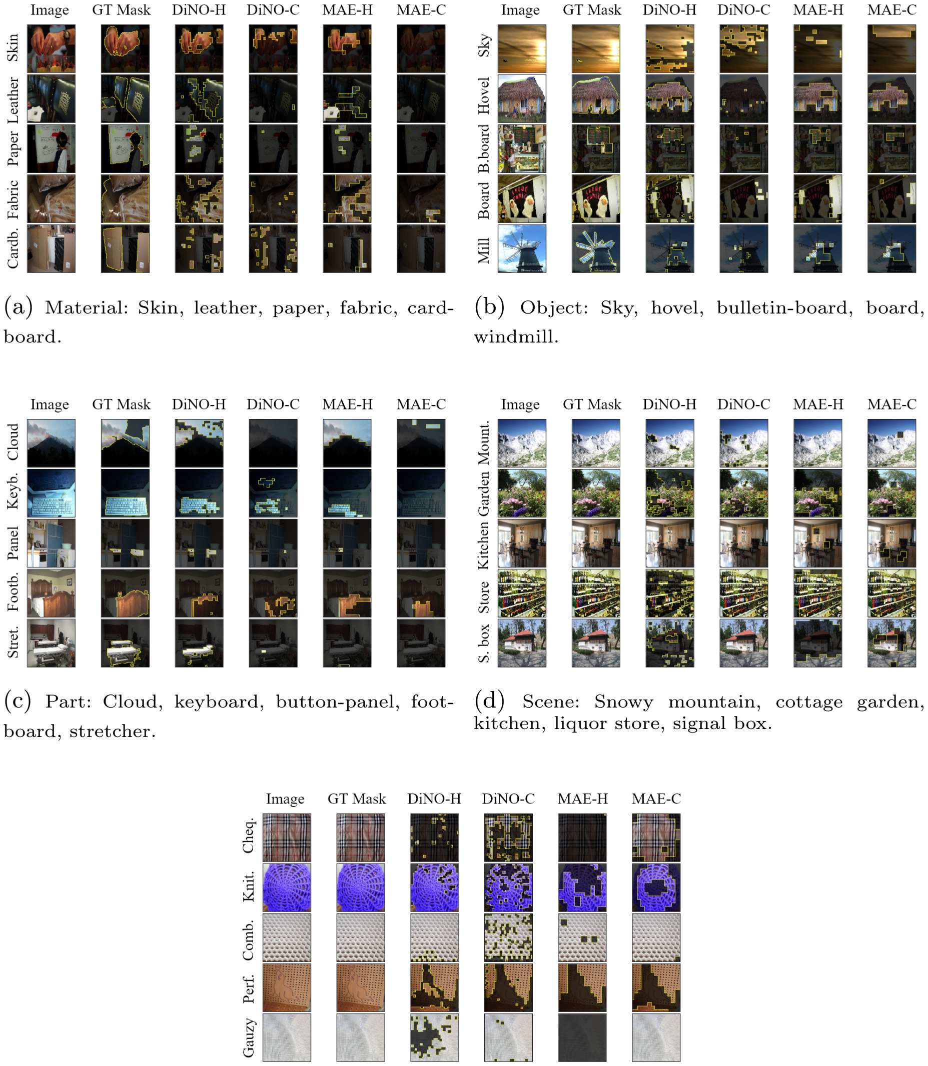
In Fig. 16, we present segmentation visualizations for the concept labels with the highest F1 scores. Specifically, for each model (DiNO, MAE) and decision rule (hyperplane, cosine), we identify the top-five concept labels based on their F1 scores. For each selected concept, we showcase segmentation masks for five image samples where the template achieves the highest intersection over union (IoU).

4.6 Summary

Our post-hoc concept direction analysis provides insights into the representation power of pretrained DiNO and MAE models, offering guidelines for practical applications while raising questions for future work. A key observation is that the hyperplane classification rule consistently delivers better semantic separability than the cosine counterpart in both classification and segmentation downstream tasks. While MAE's [CLS] tokens seem to be an exception to this finding, we demonstrated that cosine distance between tokens is a suboptimal intra-class similarity metric.

Additionally, we showed that depending on the downstream task, context, and pretraining objective, different ViT tokens –some of which had not been extensively explored in the literature– yield better semantic separability. This challenges current intuitions regarding the interpretation of query, key, and value tokens within transformer architectures and highlights the importance of understanding the role of each block within a transformer layer.

Furthermore, when utilizing pretrained DiNO and MAE models in downstream tasks, the following observations should be mentioned: For image classification, DiNO's $\boldsymbol{x}_2$ token combined with the hyperplane classification rule results in optimal classification results. Respectively, MAE's tokens should not



(a) Material: Skin, leather, paper, fabric, cardboard.

(b) Object: Sky, hovel, bulletin-board, board, windmill.

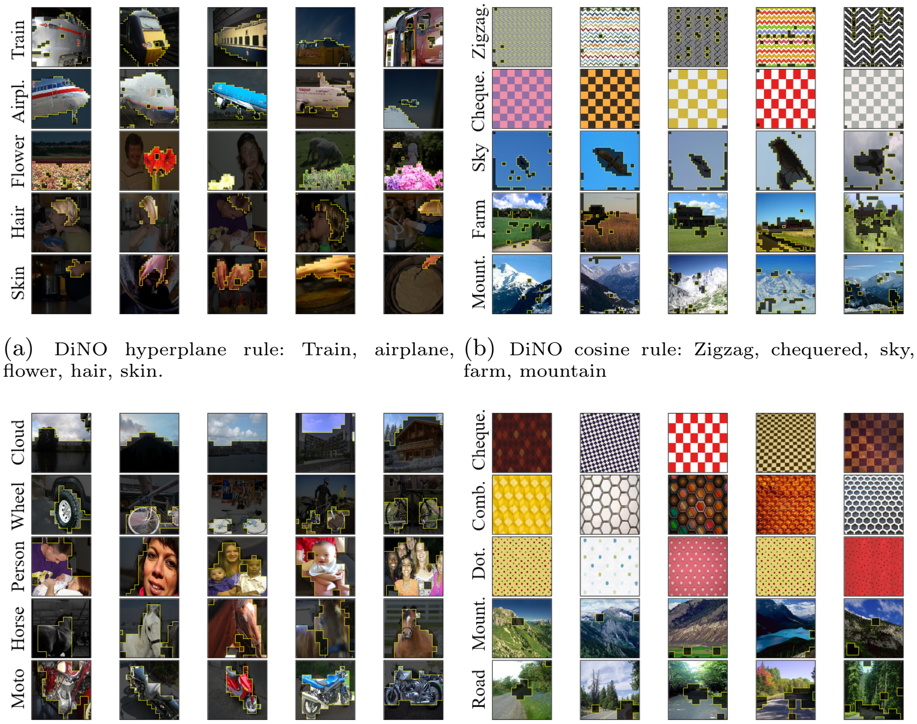
(c) Part: Cloud, keyboard, button-panel, footboard, stretcher.

(d) Scene: Snowy mountain, cottage garden, kitchen, liquor store, signal box.

(e) Texture: Chequered, knitted, honeycombed, perforated, gauzy.

Fig. 15: Segmentation visualizations for DiNO and MAE templates using cosine (DiNO-C, MAE-C) and hyperplane (DiNO-H, MAE-H) decision rules. Each figure showcases five unseen images from a specific concept category (material, object, part, scene, texture)

be considered in this context as they produce random image classifiers. When labels are sparse and a few-shot context is required, DiNO's $\boldsymbol{x}_1$ is better aligned with the cosine classification rule compared to other token types. We also observe



(a) DiNO hyperplane rule: Train, airplane, flower, hair, skin.

(b) DiNO cosine rule: Zigzag, chequered, sky, farm, mountain

(c) MAE hyperplane rule: Cloud, wheel, person, horse, motorcycle.

(d) MAE cosine rule: Chequered, honeycombed, dotted, mountain, forest road

Fig. 16: Segmentation visualizations for DiNO and MAE utilizing cosine and hyperplane decision rules. Each figure presents segmentation masks of unseen samples, produced by a particular model (DiNO, MAE) and decision rule (cosine, hyperplane). We showcase five samples per concept, highlighting the top five concepts with the highest F1 scores.

that a support set size of 50 samples represents the point at which performance gains begin to significantly diminish.

For semantic segmentation tasks, the models achieve their highest scores when leveraging their respective $\boldsymbol{k}$ tokens and the hyperplane decision rule. While DiNO outperforms MAE, the latter's strong performance in this context highlights that masked image modeling could serve as an important pretext (sub)task in the development of foundational vision transformers. Furthermore, DiNO's $\boldsymbol{k}$ tokens achieve the highest performance across all object categories, a trend that's not evident in MAE. Finally, in a few-shot context, both models' overall performance across all concept categories is inadequate. However, the $\boldsymbol{q}$ token for both DiNO and MAE provides excellent separability for textural concepts.

5 Limitations

While our study provides a thorough analysis of self-supervised ViT properties across various pre-training objectives, token types, decision rules, downstream tasks, and contexts, it has certain limitations. A primary constraint was computational resources, which restricted our evaluation solely to ViT tokens extracted from the final transformer layer. Additionally, we treat image segmentation as a non-overlapping patch-level classification rather than pixel-level classification. Since ViT-based segmentation methods using frozen backbones [17] perform spatial interpolation of the feature maps to restore the spatial dimensionality of the input space prior to classification, our approach does not significantly deviate from this norm. Finally, regarding classification via the cosine decision rule, we did not account for feature-space centering prior to the computation of cosine-similarity between features. While it would be interesting to investigate its effects, we will consider it in future works.

6 Conclusion

Our work conducted an in-depth post-hoc concept direction analysis to evaluate the representational power of pretrained DiNO and MAE token types in classification and segmentation downstream tasks. We examined their performance in both standard and few-shot learning contexts, utilizing hyperplane and cosine-similarity decision rules. Our findings show that the cosine decision rule –often used in unsupervised learning approaches– consistently results in inferior semantic separability compared to its hyperplane counterpart. We also demonstrate that the optimal token type selection is highly dependent on these factors, while confirming that masked modeling effectively constructs competent backbones for image segmentation tasks.

Future research toward the development of foundational vision architectures should focus on deepening our understanding and interpretation of ViT tokens (arising from the unintuitive and possibly unexpected efficiency of key and query tokens, disproving the hypothesis that value tokens possess superiority), as well as assessing the efficacy of transformer layers, particularly under self-supervised pretraining objectives. Additionally, in unsupervised learning applications –where the cosine distance between ViT tokens is commonly used as an intra-class similarity metric– exploring semantic proximity metrics beyond cosine similarity could enhance downstream task performance. Alternatively, a possible future research direction could be to work towards pre-training methods that will enforce interpretable concept alignment through the cosine rule, offering imminent enhancement of many existing unsupervised works that rely on a self-supervised backbone.

Acknowledgement This work has been supported by the EC funded Horizon Europe Framework Programme: CAVAA Grant Agreement 101071178.

References

1. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. ArXiv **abs/2112.05814** (2021)
2. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers (2022), https://arxiv.org/abs/2106.08254
3. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3319–3327 (2017)
4. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM Transactions on Graphics (TOG) **33**, 1 – 12 (2014)
5. Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., Arx, S.v., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramèr, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., Liang, P.: On the opportunities and risks of foundation models (arXiv:2108.07258) (Jul 2022). https://doi.org/10.48550/arXiv.2108.07258, http://arxiv.org/abs/2108.07258, arXiv:2108.07258 [cs]
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9650–9660 (October 2021)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
8. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A.J., Padlewski, P., Salz, D.M., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A.V., Bradbury, J., Kuo, W., Seyedhosseini, M., Jia, C., Ayan, B.K., Riquelme, C., Steiner, A., Angelova, A., Zhai, X., Houlsby, N., Soricut, R.: Pali: A jointly-scaled multilingual language-image model. ArXiv **abs/2209.06794** (2022)
9. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.L.: Detect what you can: Detecting and representing objects using holistic models and body parts. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 1979–1986 (2014)
10. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning (arXiv:2003.04297) (Mar 2020).

https://doi.org/10.48550/arXiv.2003.04297, http://arxiv.org/abs/2003.04297, arXiv:2003.04297 [cs]

11. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9620–9629 (2021)
12. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 3606–3613 (2013)
13. Cunningham, H., Ewart, A., Riggs, L., Huben, R., Sharkey, L.: Sparse autoencoders find highly interpretable features in language models. arXiv preprint arXiv:2309.08600 (2023)
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv **abs/2010.11929** (2020)
15. Doumanoglou, A., Asteriadis, S., Zarpalas, D.: Unsupervised interpretable basis extraction for concept–based visual explanations. IEEE Transactions on Artificial Intelligence (2023)
16. Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., Serre, T.: Craft: Concept recursive activation factorization for explainability. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2711–2721 (2023)
17. Hamilton, M., Zhang, Z., Hariharan, B., Snavely, N., Freeman, W.T.: Unsupervised semantic segmentation by distilling feature correspondences. ArXiv **abs/2203.08414** (2022)
18. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al.: A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence **45**(1), 87–110 (2022)
19. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16000–16009 (June 2022)
20. Kang, D., Koniusz, P., Cho, M., Murray, N.: Distilling self-supervised vision transformers for weakly-supervised few-shot classification & segmentation. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 19627–19638 (2023)
21. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM computing surveys (CSUR) **54**(10s), 1–41 (2022)
22. Le, Y., Yang, X.S.: Tiny imagenet visual recognition challenge (2015)
23. Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., He, Z.: A survey of visual transformers. IEEE Transactions on Neural Networks and Learning Systems (2023)
24. Mishra, S.K., Robinson, J., Chang, H., Jacobs, D., Sarna, A., Maschinot, A., Krishnan, D.: A simple, efficient and scalable contrastive masked autoencoder for learning visual representations. ArXiv **abs/2210.16870** (2022)
25. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.L.: The role of context for object detection and semantic segmentation in the wild. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 891–898 (2014)
26. Naseer, M.M., Ranasinghe, K., Khan, S.H., Hayat, M., Shahbaz Khan, F., Yang, M.H.: Intriguing properties of vision transformers. Advances in Neural Information Processing Systems **34**, 23296–23308 (2021)

27. Nguyen, H.H., Yamagishi, J., Echizen, I.: Exploring self-supervised vision transformers for deepfake detection: A comparative analysis. ArXiv **abs/2405.00355** (2024)
28. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.Q., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y.B., Li, S.W., Misra, I., Rabbat, M.G., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. ArXiv **abs/2304.07193** (2023)
29. Park, N., Kim, W., Heo, B., Kim, T., Yun, S.: What do self-supervised vision transformers learn? ArXiv **abs/2305.00729** (2023)
30. Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F.: Beit v2: Masked image modeling with vector-quantized visual tokenizers (arXiv:2208.06366) (Oct 2022). https://doi.org/10.48550/arXiv.2208.06366, http://arxiv.org/abs/2208.06366, arXiv:2208.06366 [cs]
31. Qian, Y., Wang, Y., Lin, J.: Enhancing the linear probing performance of masked auto-encoders. In: International Conference on Pattern Recognition. pp. 289–301. Springer (2022)
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021), https://api.semanticscholar.org/CorpusID:231591445
33. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? Advances in neural information processing systems **34**, 12116–12128 (2021)
34. Rao, S., Mahajan, S., Böhle, M., Schiele, B.: Discover-then-name: Task-agnostic concept bottlenecks via automated concept discovery. In: European Conference on Computer Vision. pp. 444–461. Springer (2024)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**, 211 – 252 (2014)
36. Siméoni, O., Puy, G., Vo, H.V., Roburin, S., Gidaris, S., Bursuc, A., P'erez, P., Marlet, R., Ponce, J.: Localizing objects with self-supervised transformers and no labels. ArXiv **abs/2109.14279** (2021)
37. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., J'egou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning (2020)
38. Vanyan, A., Barseghyan, A., Tamazyan, H., Huroyan, V., Khachatrian, H., Danelljan, M.: Analyzing local representations of self-supervised vision transformers. ArXiv **abs/2401.00463** (2023)
39. Wang, P., Wang, S., Lin, J., Bai, S., Zhou, X., Zhou, J., Wang, X., Zhou, C.: One-peace: Exploring one general representation model toward unlimited modalities. ArXiv **abs/2305.11172** (2023)
40. Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O.K., Singhal, S., Som, S., Wei, F.: Image as a foreign language: Beit pretraining for all vision and vision-language tasks. ArXiv **abs/2208.10442** (2022)
41. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). p. 3023–3032. IEEE,

Nashville, TN, USA (Jun 2021). https://doi.org/10.1109/CVPR46437.2021.00304, https://ieeexplore.ieee.org/document/9578497/

42. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3124–3134 (2023)
43. Wang, Y., Shen, X., Hu, S.X., Yuan, Y., Crowley, J.L., Vaufreydaz, D.: Self-supervised transformers for unsupervised object discovery using normalized cut. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 14523–14533 (2022)
44. Wu, H., Gao, Y., Zhang, Y., Lin, S., Xie, Y., Sun, X., Li, K.: Self-supervised models are good teaching assistants for vision transformers. In: Proceedings of the 39th International Conference on Machine Learning. p. 24031–24042. PMLR (Jun 2022), https://proceedings.mlr.press/v162/wu22c.html
45. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: a simple framework for masked image modeling. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9643–9653 (2021)
46. Xiong, Y., Varadarajan, B., Wu, L., Xiang, X., Xiao, F., Zhu, C., Dai, X., Wang, D., Sun, F., Iandola, F., et al.: Efficientsam: Leveraged masked image pretraining for efficient segment anything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16111–16121 (2024)
47. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10371–10381 (2024)
48. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: Contrastive captioners are image-text foundation models. Trans. Mach. Learn. Res. **2022** (2022)
49. Yue, X., Bai, L., Wei, M., Pang, J., Liu, X., Zhou, L., Ouyang, W.: Understanding masked autoencoders from a local contrastive perspective. ArXiv **abs/2310.01994** (2023)
50. Zhang, W., Ma, B., Qiu, F., qiong Ding, Y.: Multi-modal facial affective analysis based on masked autoencoder. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 5793–5802 (2023)
51. Zhou, B., Sun, Y., Bau, D., Torralba, A.: Interpretable basis decomposition for visual explanation. In: European Conference on Computer Vision (2018)
52. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5122–5130 (2017)
53. Zhou, L., Liu, H., Bae, J., He, J., Samaras, D., Prasanna, P.: Self pretraining with masked autoencoders for medical image classification and segmentation (arXiv:2203.05573) (Apr 2023). https://doi.org/10.48550/arXiv.2203.05573, http://arxiv.org/abs/2203.05573, arXiv:2203.05573 [eess]