Attention, Please!
Revisiting Attentive Probing for Masked Image Modeling

Bill Psomas[2*†] Dionysis Christopoulos[1†] Eirini Baltzi[1] Ioannis Kakogeorgiou[4,7] Tilemachos Aravanis[1]
Nikos Komodakis[3,4,5] Konstantinos Karantzalos[1] Yannis Avrithis[6] Giorgos Tolias[2]

[1]National Technical University of Athens [2]Czech Technical University in Prague
[3]University of Crete [4]Archimedes, Athena RC [5]IACM-FORTH [6]IARAI [7]IIT, NCSR "Demokritos"

Abstract

As fine-tuning (FT) becomes increasingly impractical at scale, probing is emerging as the preferred evaluation protocol for self-supervised learning (SSL). Yet, the standard linear probing (LP) fails to adequately reflect the potential of models trained with Masked Image Modeling (MIM), due to the distributed nature of patch tokens. This motivates the need for attentive probing, an alternative that uses attention to selectively aggregate patch-level features. Despite its growing adoption, attentive probing remains underexplored, with existing methods suffering from excessive parameterization and poor computational efficiency.

In this work, we revisit attentive probing through the lens of the accuracy–efficiency trade-off. We conduct a systematic study of existing methods, analyzing their mechanisms and benchmarking their performance. We introduce efficient probing (EP), a multi-query cross-attention mechanism that eliminates redundant projections, reduces the number of trainable parameters, and achieves up to a 10× speed-up over conventional multi-head attention. Despite its simplicity, EP outperforms LP and prior attentive probing approaches across seven benchmarks, generalizes well beyond MIM to diverse pretraining paradigms, produces interpretable attention maps, and achieves strong gains in low-shot and layer-wise settings. Code available at https://github.com/billpsomas/efficient-probing.

1. Introduction

Self-supervised learning (SSL) [7, 10, 17, 27, 58] has emerged as a powerful paradigm for learning visual representations without labeled data, significantly reducing the reliance on costly human annotations. Recent advances in SSL can be broadly categorized into two families: *joint embedding architectures* (JEA) [3, 7, 10, 17] and *masked image modeling* (MIM) [1, 11, 12, 19, 27, 38, 48, 58]. JEA methods learn global representations by contrasting or clustering different augmentations of the same image, often pooling patch-level features or relying on a dedicated [CLS] token. In contrast, MIM approaches reconstruct masked image regions and thus yield distributed representations with fine-grained spatial information.

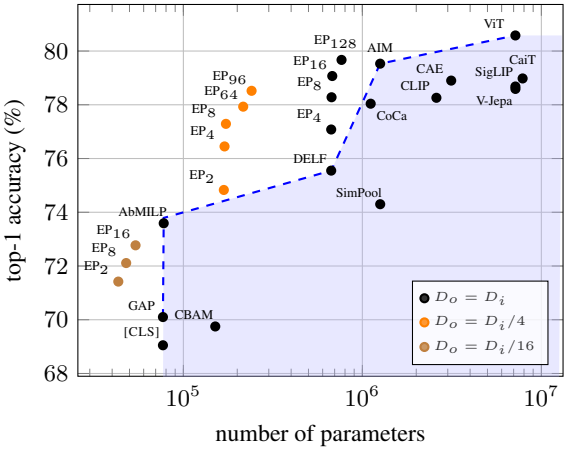

Figure 1. We introduce *efficient probing* (EP), a lightweight attention mechanism improving efficiency without compromising performance. We benchmark *attentive probing* and *attentive pooling* methods to evaluate frozen models pretrained with Masked Image Modeling (MIM). Here: *accuracy* vs. *number of parameters* for different methods on CIFAR-100 [30] using MAE ViT-B [19]. [CLS]: linear probing; GAP: global average pooling of patch tokens; ViT: default transformer block [14]; EP_M: efficient probing with M learnable queries; D_i, D_o: input, output dimensionality.

To evaluate self-supervised pre-training, common protocols include k-NN evaluation, linear probing (LP), and full fine-tuning (FT). While k-NN and LP assess the quality of the learned representations under a frozen backbone, FT measures the utility of pre-training as initialization for downstream tasks. Although FT achieves the highest performance, it is increasingly viewed as unsustainable and prohibitive at scale. As a result, probing is emerging as

*Lead author.
†Equal contribution.

the preferred evaluation strategy [4, 12, 15, 37]. However, MIM models often lack a discriminative global representation [40], making standard LP inappropriate to reflect their true capabilities [19]. Since valuable information is distributed across patch tokens, *attentive probing* [4, 11, 12, 15] has emerged as an alternative, using attention mechanisms to selectively aggregate patch-level features.

Despite its increasing adoption, attentive probing remains underexplored. Existing methods vary significantly in design, are often *over-parameterized* or *inefficient*, and lack a systematic framework for comparison. In this work, we address these limitations by conducting the first comprehensive study of attentive probing, revisiting its design through the lens of the *accuracy–efficiency trade-off*. We introduce a unified framework that encompasses a wide range of attention-based aggregation methods—including those proposed for probing [4, 11, 12, 15] and others from unrelated tasks [11, 36, 41, 43, 53].

Through this framework, we identify key design simplifications that lead to a new method, *efficient probing* (EP): a conceptually simple multiquery cross-attention mechanism that eliminates redundant projections, reduces parameter count and computational cost, and achieves up to $10\times$ speed-up over conventional multi-head attention—while matching or surpassing prior state-of-the-art performance (Figure 1). EP effectively balances accuracy, model size, and efficiency, providing a lightweight yet expressive alternative for probing frozen models.

A central component of both existing methods and EP is the use of multiple independent *attention predictors* (*e.g.*, heads or learnable queries). We establish a connection between the contribution of each predictor to the classification accuracy and its *localization quality*. Our analysis shows that, rather than resorting to shortcut learning, *e.g.*, leveraging background cues like water to classify a fish, EP produces attention maps that effectively focus on foreground objects, enhancing interpretability, robustness, and performance. Notably, each attention predictor in EP specializes in distinct object regions, ensuring complementary feature extraction and a more structured representation.

We validate our findings through extensive experiments on seven classification benchmarks, across four MIM frameworks, using backbones of varying size. Beyond its original motivation for MIM, EP generalizes well to other pre-training paradigms, as shown by results on joint-embedding, hybrid, and vision-language models.

Our contributions are summarized as follows:

1. We conduct the first systematic benchmark and analysis of attentive probing methods, comparing their accuracy, efficiency, and design choices.
2. We derive a new probing mechanism, *efficient probing* (EP), which is on par with the state-of-the-art, while bringing spectacular gains in compute, memory, and parameter efficiency.
3. We uncover a correlation between spatial localization and predictive performance, and show that EP yields interpretable attention maps.

2. Related Work

Evaluation protocols in SSL Self-supervised learning (SSL) has transformed visual representation learning, with evaluation typically performed via (i) *k-NN* on frozen features, (ii) *linear probing* (LP) using a shallow classifier on a frozen encoder, or (iii) *fine-tuning* (FT) the entire model. Although FT achieves the highest accuracy, it is *computationally expensive*.

Two dominant SSL paradigms are joint embedding architectures (JEA) and masked image modeling (MIM). JEA methods (*e.g.*, DINO [7], SimCLR [9]) contrast or cluster augmentations to learn global representations via a [CLS] token or pooled features. In contrast, MIM methods (*e.g.*, MAE [19], SimMIM [51], BEiTv2 [38]) reconstruct masked regions, yielding localized representations distributed across patch tokens.

This global *vs.* local distinction affects evaluation: LP is effective for JEA [7] but underperforms for MIM [19, 40], where discriminative information is not concentrated in a single token. Consequently, FT remains the preferred strategy for MIM [19, 51]. To overcome this, recent work explores *attentive probing* [4, 11, 12, 15], where attention is used to aggregate patch tokens into informative descriptors. While methods like AIM [15], CAE [11], and V-JEPA [4] adopt this idea, no unified evaluation exists. We fill this gap with a comprehensive benchmark and introduce a novel attention mechanism achieving a strong accuracy–efficiency trade-off.

Pooling in vision models Pooling reduces spatial resolution while retaining semantic information. In CNNs, fixed pooling (*e.g.*, global average pooling [18, 33]) is standard; in vision transformers (ViTs) [14], the [CLS] token aggregates features via self-attention.

Recent work proposes attention-based pooling to enhance representation quality. SimPool [41] replaces global average pooling using trainable attention in both CNNs and ViTs. Vision-language models such as CLIP [42], SigLIP [49], and CoCa [53] use attentive pooling or cross-attention to fuse modalities. V-JEPA [4] applies cross-attention pooling for probing pretrained representations. In image retrieval, DELF [36] and DOLG [52] use spatial attention to focus on salient regions. CaiT [45] improves class-token attention, AbMILP [43] uses single-query pooling for multiple-instance learning, and CBAM [50] combines channel and spatial attention to recalibrate features. Although these poolings are originally introduced in diverse

contexts, we repurpose them for probing frozen models, enabling a fair and comprehensive benchmark.

Additional related works are presented in the supplementary material.

3. Method

3.1. Preliminaries

Let $X \in \mathbb{R}^{D_i \times N}$ be the *feature matrix* obtained from a *pre-trained* and *frozen* ViT backbone, where D_i is the number of feature channels and $N = W \times H$ the number of features, one per image patch across the spatial dimensions $W \times H$. Given the *input features* X, the goal is to generate an *output image-level feature* $\mathbf{y} \in \mathbb{R}^{D_o}$ by applying an *attentive pooling* mechanism. The output feature is used to train a C-way linear classifier with $D_o(C+1)$ parameters.

3.2. Attentive Pooling

We consider M *attention predictors*, to be discussed in subsection 3.3. For each predictor $j \in \{1, \dots, M\}$, let $\mathbf{a}_j \in \mathbb{R}^N$ be the ℓ_1-normalized *attention vector* it generates. Each vector, reshaped to $W \times H$, is an attention map indicating the locations on which the predictor focuses. Let $V \in \mathbb{R}^{D_o \times N}$ be the *value features*, commonly obtained by a linear transformation $V = W_V X$, where $W_V \in \mathbb{R}^{D_o \times D_i}$ is a learnable *projection matrix*.

Let the output feature $\mathbf{y}$, value features V and projection matrix W_V be partitioned into M subvectors / submatrices according to

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_M \end{bmatrix}, V = \begin{bmatrix} V_1 \\ \vdots \\ V_M \end{bmatrix}, W_V = \begin{bmatrix} W_{V_1} \\ \vdots \\ W_{V_M} \end{bmatrix}, \tag{1}$$

with $\mathbf{y}_j \in \mathbb{R}^{d_o}$, $V_j \in \mathbb{R}^{d_o \times N}$, $W_{V_j} \in \mathbb{R}^{d_o \times D_i}$ and $d_o = \frac{D_o}{M}$.

The attentive pooling operation is then given by

$$\mathbf{y}_j = V_j \mathbf{a}_j = W_{V_j} X \mathbf{a}_j. \tag{2}$$

Each attention predictor is responsible for the weighted pooling of N features into a d_o-dimensional subspace of the final representation space. In the following, we explore existing and novel ways for designing these attention predictors. We focus on the number of additional parameters to be learnt on top of the frozen backbone and the computational complexity of the pooling operation.

3.3. Attention Predictors

Multi-Head Cross-Attention (MHCA) A standard approach is to perform multi-head cross-attention between the input features and an *input vector* $\mathbf{u} \in \mathbb{R}^{D_i}$, where each head corresponds to a separate attention predictor. The *query feature* $\mathbf{q} \in \mathbb{R}^{D_a}$ and *key features* $K \in \mathbb{R}^{D_a \times N}$ are obtained by linear transformations $\mathbf{q} = W_Q \mathbf{u}$, $K = W_K X$ with projection matrices $W_Q, W_K \in \mathbb{R}^{D_a \times D_i}$ (Figure 2).

Let the query feature $\mathbf{q}$ and projection matrix W_Q be partitioned into M subvectors / submatrices according to

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_M \end{bmatrix}, W_Q = \begin{bmatrix} W_{Q_1} \\ \vdots \\ W_{Q_M} \end{bmatrix}, \tag{3}$$

with $\mathbf{q}_j = W_{Q_j} \mathbf{u} \in \mathbb{R}^{d_a}$, $W_{Q_j} \in \mathbb{R}^{d_a \times D_i}$ and $d_a = \frac{D_a}{M}$. Similarly, let the key features K and projection matrix W_K be partitioned according to

$$K = \begin{bmatrix} K_1 \\ \vdots \\ K_M \end{bmatrix}, W_K = \begin{bmatrix} W_{K_1} \\ \vdots \\ W_{K_M} \end{bmatrix}, \tag{4}$$

with $K_j = W_{K_j} X \in \mathbb{R}^{d_a \times N}$ and $W_{K_j} \in \mathbb{R}^{d_a \times D_i}$.

The attention vector for head j is then given by

$$\mathbf{a}_j = \mathrm{softmax}(\hat{\mathbf{a}}_j) \tag{5}$$

with

$$\hat{\mathbf{a}}_j = K_j^\top \mathbf{q}_j = (W_{K_j} X)^\top (W_{Q_j} \mathbf{u}). \tag{6}$$

That is, the input features X and input vector $\mathbf{u}$ are projected to d_a-dimensional subspaces where attention subvectors are computed via dot product followed by softmax normalization over patches. This attention predictor requires $D_a(2D_i + 1)$ parameters and has complexity $\mathcal{O}(ND_aD_i)$. As discussed in subsection 3.4, there are several existing methods that fit within this generic framework with little or no differences.

MHCA with a learnable query If we consider input vector $\mathbf{u}$ to be learnable, then there is no need for the projection matrix W_Q in (6). Instead, we can set the query feature $\mathbf{q}$ to be learnable, thus absorbing W_Q and $\mathbf{u}$:

$$\hat{\mathbf{a}}_j = (W_{K_j} X)^\top \mathbf{q}_j = X^\top W_{K_j}^\top \mathbf{q}_j \tag{7}$$

where the query feature $\mathbf{q}_j \in \mathbb{R}^{d_a}$ is learnable.

We observe that $W_{K_j}^\top$ maps $\mathbf{q}_j$ to the D_i-dimensional space of input features to compute the attention vector. Thus, standard MHCA ensures that each query subvector is interacting with the full representation space of the input features, despite being defined in a smaller dimensional space. Using a learnable query feature directly simplifies the architecture, reduces the amount of computations and the number of parameters to $D_a(D_i + 1)$.

We simplify the architecture by removing the key transformation. By letting W_K be fixed to the identity matrix, (7) becomes

$$\hat{\mathbf{a}}_j = X_j^\top \mathbf{q}_j \tag{8}$$

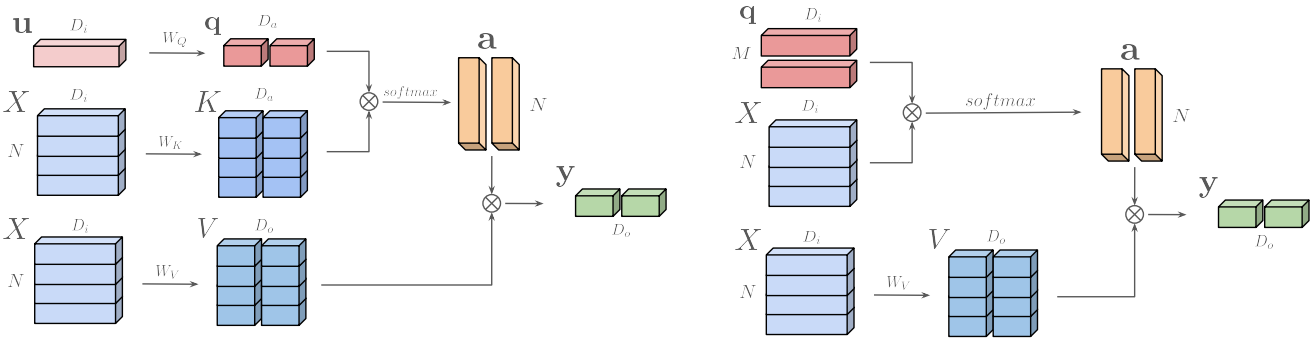

Figure 2. *Comparison of multi-head cross attention (MHCA, left)* vs. *our transformation-free cross-attention (right)*, *i.e.* the mechanism of EP. MHCA uses a learnable input vector $\mathbf{u}$ projected into query space and interacts with key features K in two separate subspaces, each corresponding to an attention predictor. Attention predictor outputs $\mathbf{a}_j$ are used to aggregate value features V into sub-vectors $\mathbf{y}_j$, forming the final output $\mathbf{y}$. In contrast, EP employs two *learnable queries* $\mathbf{q}_j$, one per attention predictor, to compute attention with input features directly in the full representation space. Attention predictor outputs $\mathbf{a}_j$ are used as in MHCA to perform the aggregation.

where the feature matrix X is partitioned into M submatrices according to

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_M \end{bmatrix}, \tag{9}$$

with $X_j \in \mathbb{R}^{d_i \times N}$ and $d_i = \frac{D_i}{M}$. We thus observe that the query feature only interacts with a d_i-dimensional subspace of the input features. This is a *limitation* that we experimentally verify to perform poorly. In the following, we suggest a new way to design attention predictors that have less parameters, require less compute and overcome this limitation, making them perform better.

Transformation-free Cross-Attention with multiple learnable queries Instead of using the key submatrices W_{K_j} to project the query subvectors $\mathbf{q}_j$ to the D_i-dimensional input feature space, we propose to learn M query features $\mathbf{u}_j \in \mathbb{R}^{D_i}$ in that space directly (Figure 2). Thus, $\mathbf{u}_j$ absorbs W_{K_j} and $\mathbf{q}_j$ and attention prediction becomes

$$\hat{\mathbf{a}}_j = X^\top \mathbf{u}_j \tag{10}$$

for $j \in \{1, \dots, M\}$. As a result, there are no projection matrices and there are no parameters other than the learnable query features $\mathbf{u}_j$.

This choice reduces the number of additional parameters to be learned and saves from one more matrix-vector multiplication. In particular, it requires $D_i M$ parameters for the attention compared to $D_a(D_i+1)$ for (6), while the number of operations drops to ND_iM compared to $ND_a(D_i+1)$. Typically, M is one to two orders of magnitude smaller than D_i and D_a, which are commonly equal to each other, making the proposed approach more efficient in terms of both parameters and operations.

There is a connection between EP and *slot attention* [34], where slots are also multiple vectors in the input feature space. To derive EP from slot attention, one needs to perform only a single iteration; remove LayerNorm, GRU and MLP; make slot vectors learnable rather than initialized at random; and concatenate the output features into a global representation of appropriate dimension. Thus, EP can be seen as a lightweight counterpart of slot attention, where the absence of interactions is compensated by the query features being learned.

3.4. Existing variants

We analyze existing methods as instances of the presented framework, and examine common variants, considering their relationship to the framework despite slight deviations.

AbMILP Attention-based Multiple Instance Learning Pooling [43] is the simplest variant. It fixes W_V to identity and is equivalent to MHCA with a learnable query feature framework in (7), with a single head ($M = 1$). It can also be seen as a special case of our proposed method in (10) with one learnable query feature, *i.e.* $M = 1$. AbMILP requires only D_i parameters and computes attention with a single matrix-vector multiplication, but its performance is limited by the single head/query.

AIM AIM [15] is an instance of MHCA with a learnable query feature. It deviates from the generic framework by applying batch normalization on the input features. It does follow (2) and (7) with M heads and $D_a = D_i = D_o$, but replaces X by $\mathrm{BN}(X)$. Batch normalization introduces additional parameters and a slight computational overhead compared to the default variant of the framework.

DELF We establish an useful relation with DELF [36], although it was not introduced in the context of attentive pooling for SSL. DELF feeds each of the N input features to a two-layer MLP whose output is a scalar attention value

in $[0, 1]$. It can be viewed as an instance of the MHCA with a learnable query feature and $M = 1$ with the following two modifications. A non-linearity is introduced in equation (7) by $\hat{\mathbf{a}} = \mathrm{ReLU}(W_K X)^\top \mathbf{q}$, where subscript j is skipped due to $M = 1$, and softmax in (5) is replaced by element-wise softplus, $\mathbf{a} = \mathrm{softplus}(\hat{\mathbf{a}})$. In the context of DELF, the query feature $\mathbf{q}$ can be seen as the parameter of a 1×1 convolutional layer. DELF does not introduce any additional parameters and has the same complexity as the default variant of the framework.

SimPool SimPool [41] can be seen as an instance of MHCA with a single head ($M = 1$) but it has a data-dependent query vector, W_V fixed to identity, and layer normalization on the input features. Specifically, the query feature is obtained as the average of the input features, $\mathbf{q} = \frac{1}{N} X^\top \mathbf{1}$, and X is replaced by $\mathrm{LN}(X)$. Compared to the default variant of the framework, SimPool saves D_i parameters and has the same complexity.

V-JEPA The first part of V-JEPA [4] is identical to the MHCA framework but applies layer normalization on the input features, like SimPool. Its second part is an MLP with GeLU [21] activation and residual connections, making the overall process equivalent to a transformer block.

Additional methods considered in experiments like CLIP [42] and CoCa [53] are further variants of the MHCA framework with slight differences compared to the variants presented above. For brevity, we present these in the supplementary material.

4. Experiments

4.1. Experimental setup

Datasets We evaluate attentive probing across diverse image classification benchmarks, including ImageNet-1k [13], CIFAR-100 [30], Places365 [57], CUB-200 [47], FGVC Aircraft [35], Stanford Cars [29], and Food-101 [6]. Full dataset details are provided in the supplementary material.

SSL methods We conduct attentive probing experiments using frozen self-supervised learning methods, including MAE [19], SimMIM [51], BEiTv2 [38], and CAPI [12]. We evaluate MAE using ViT-S, ViT-B, and ViT-L, SimMIM and BEiTv2 using ViT-B, and CAPI using ViT-L.

Evaluation protocols Attentive probing is performed for 90 epochs. We evaluate top-1 classification accuracy on the validation set of each dataset. Additionally, we compute the number of parameters and measure the FLOPs for each method to assess computational efficiency and scalability.

4.2. Competitors

We compare attentive probing against a diverse set of methods, covering different paradigms. First, we evaluate attentive poolings originally designed for probing, including AIM [15], CAE [11], CAPI [12], and V-JEPA [4]. Second, we include attentive poolings originally proposed in other contexts but applicable to probing, such as AbMILP [43], SimPool [41], CLIP [42], SigLIP [49, 54], CoCa [53], CaiT [45], and DELF [36]. Additionally, we include feature re-weighting methods like CBAM [50], applying global average pooling to obtain the global descriptor.

As baselines, we include [CLS], which corresponds to standard linear probing using the classification token, and GAP, which serves as the baseline attentive probing approach with uniform attention over the patch tokens. To establish a reference, we also evaluate a ViT [14] block as a probing mechanism, applying global average pooling to extract the global representation. All methods operate on the same input features—namely, the patch tokens extracted from the frozen backbone—ensuring a fair and consistent comparison. Unless otherwise stated, $D_o = D_i = D_a$.

4.3. Benchmark

Accuracy *vs.* parameters In Figure 3, we compare efficient probing (EP) with baseline and competitor methods using MAE ViT-B, SimMIM ViT-B, BEiTv2 ViT-B, and CAPI ViT-L on Imagenet-1k, and MAE ViT-B on Food-101 and Cars-196. We plot top-1 accuracy against the number of trainable parameters, including both attentive pooling and classifier parameters, and overlay the Pareto frontier to highlight optimal trade-offs. The two primary baselines, [CLS] and GAP, are the most parameter-efficient, as they introduce no overhead beyond the classifier parameters, but yield noticeably lower accuracy. In contrast, methods like V-JEPA, CaiT, SigLIP, and the reference ViT block employ significantly more parameters, though within the attentive probing setting, their increased complexity provides mostly marginal accuracy improvements. Among the existing attentive probing or pooling methods, SimPool provides moderate accuracy but is not particularly parameter-efficient, while CAE and CLIP achieve stronger performance at the cost of higher parameter counts. AbMILP, DELF, AIM, and CoCa lie on the Pareto frontier, striking the optimal balance.

EP consistently achieves the best accuracy–parameter trade-off, positioning itself on the left or upper-left region of the Pareto frontier across self-supervised methods and datasets. A key factor is its flexibility in controlling the number of queries M and the output dimensionality D_o (because of W_V), allowing adaptation to different parameter constraints. Notably, on ImageNet-1k with MAE ViT-B, EP_{64} (64 queries) achieves a state-of-the-art top-1 accuracy of 75.6% with less than 1.4M parameters. EP_{48} with 48 queries and $D_o = D_i/8$ achieves 70.3% top-1 accuracy,

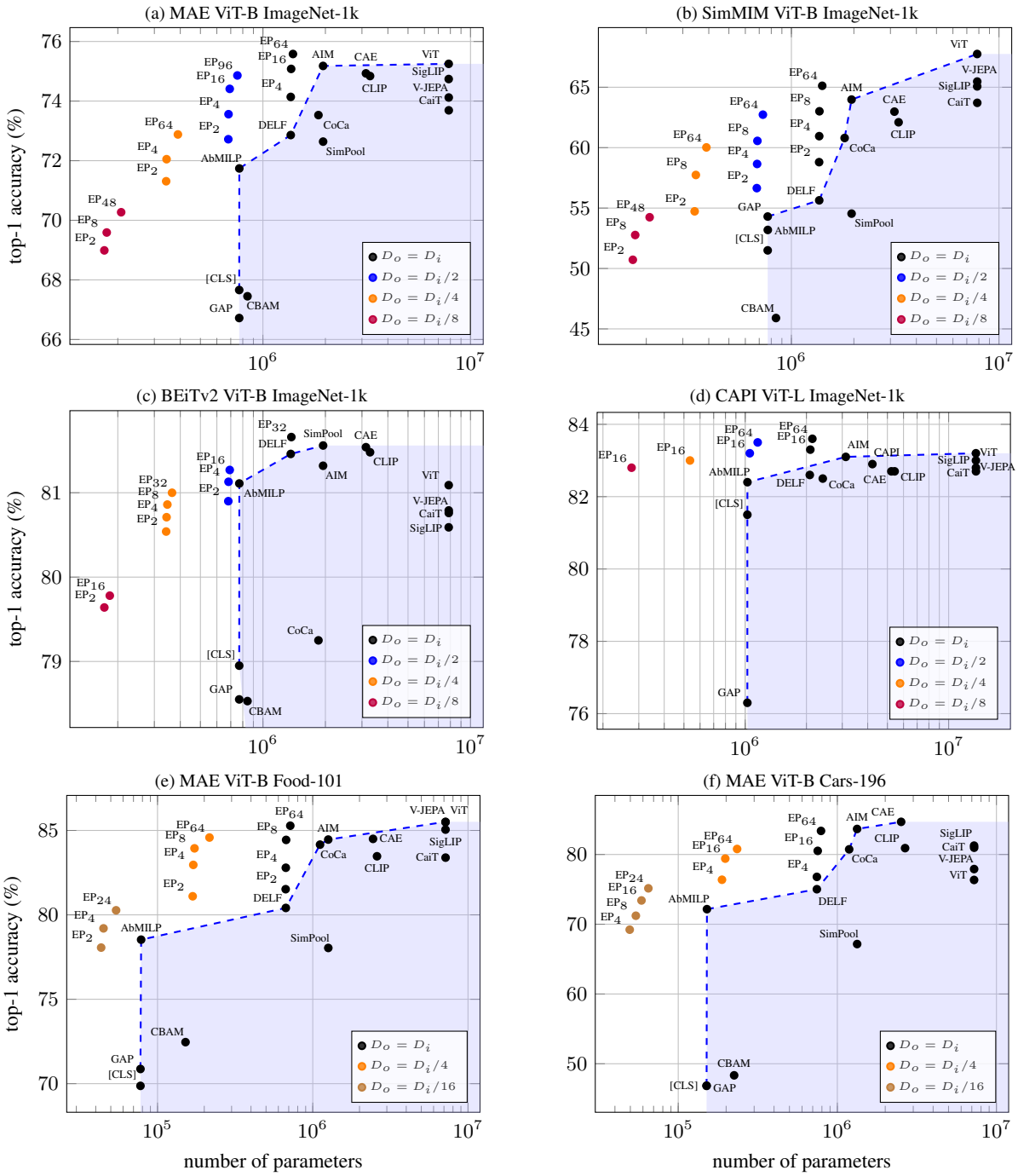

Figure 3. *Top-1 classification accuracy* vs. *number of parameters* for various self-supervised learning methods (a, b, c, d) across different datasets (c, d). We evaluate both dedicated probing mechanisms (*e.g.*, V-JEPA) and repurposed attentive pooling methods (*e.g.*, CLIP). EP variants are marked with different colors for different output dimensionalities D_o. EP_M: efficient probing with M learnable queries. [CLS]: linear probing using the classification token; GAP: global average pooling over patch tokens; ViT: default transformer block.

while having a little more than 200k parameters, *i.e.* almost 4× less than linear probing ([CLS]). On ImageNet-1k with SimMIM ViT-B, the trend is almost consistent, where EP_{64} achieves competitive accuracy, while with BEiTv2 ViT-B and with CAPI ViT-L, EP_{32} and EP_{64} achieve a state-of-the-art top-1 accuracy of 81.7% and 83.6% respectively. On Food-101 and Cars-196, we observe that reducing D_o even to $D_i/16$ does not significantly hurt performance.

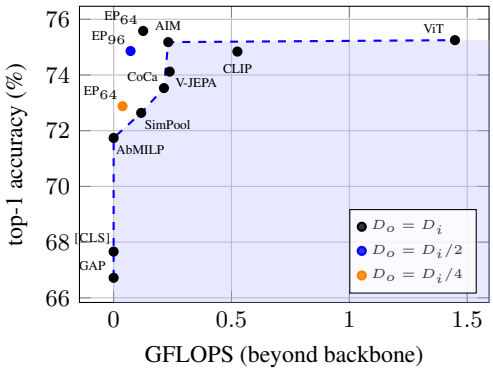

Figure 4. *Top-1 classification accuracy* vs. *GFLOPS* for MAE ViT-B with different probings on ImageNet-1k.

Accuracy *vs*. computational cost In Figure 4, we compare different probing or pooling methods in terms of top-1 accuracy against computational cost, measured in GFLOPS. Baseline approaches [CLS] and GAP are the most efficient but suffer from lower accuracy. Methods using self-attention, such as ViT and CLIP, exhibit higher GFLOPs due to additional attention computations. EP requires only 0.12 GFLOPs for probing, compared to 1.44 GFLOPs for ViT—comparable to the vanilla MHCA—achieving a more than 10× reduction in compute. Our EP again lies on the left or upper-left of the Pareto frontier, demonstrating superior accuracy-to-compute efficiency. By controlling the classifier's output dimensionality D_o, we achieve competitive performance at significantly reduced computational cost. Notably, reducing D_o (*e.g.*, $D_o = D_i/2$) yields efficiency gains without major accuracy loss, establishing the scalability of EP to different computational budgets.

Comparison of pre-training methods Table 1 compares various pre-training methods—masked image modeling, joint embedding, hybrid, and vision-language—under multiple evaluation protocols on ImageNet-1k. Fine-tuning (FT) achieves the highest accuracy but is compute-intensive and unsustainable at scale, making it increasingly rare in recent evaluations (×). k-NN performs poorly, highlighting the limited discriminative power of raw features. Linear probing (LP) and efficient probing (EP) offer efficient alternatives. EP consistently outperforms LP while remaining lightweight. It also reveals trends that reverse those seen in LP and k-NN—for instance, MAE surpasses BYOL and CAPI outperforms CLIP—suggesting that MIM methods are stronger than often assumed. While originally motivated by MIM, EP proves broadly applicable and narrows the performance gap to FT across all paradigms.

4.4. Classification *vs*. localization

We investigate whether the quality of attention maps in terms of localization contributes positively to classification accuracy. Specifically, we evaluate each attention predictor by comparing its localization quality against its influence on overall classification performance. To assess localization quality, we measure: (i) the sum of attention values allocated to image patches within the ground-truth bounding box [13], and (ii) the entropy of the attention distribution. We average these metrics over the validation set. To quantify each predictor's importance for classification, we replace its learned attention distribution with a uniform distribution and report the resulting accuracy drop.

Table 1. *Comparison of pre-training methods in terms of different evaluation protocols* on ImageNet-1k; (ranking).

Method	Pub.	Arch.	Pretrain	k-NN	LP	EP	FT
MAE [19]	CVPR22	ViT-B	IN-1k	46.1 (10)	67.7 (10)	75.6 (9)	83.6
BEiTv2 [38]	arXiv22	ViT-B	IN-1k	74.8 (8)	79.0 (6)	81.7 (6)	85.0
SimMIM [51]	CVPR22	ViT-B	IN-1k	15.1 (11)	51.5 (11)	65.1 (11)	83.8
CAPI [12]	TMLR25	ViT-L	IN-1k	76.7 (6)	81.5 (5)	83.6 (4)	×
BYOL [17]	NeurIPS20	RN-50	IN-1k	64.8 (9)	74.3 (9)	75.1 (10)	77.7
DINO [7]	ICCV21	ViT-B	IN-1k	76.1 (7)	78.2 (8)	78.7 (8)	82.8
iBOT [58]	ICLR22	ViT-B	IN-1k	77.0 (5)	78.7 (7)	79.2 (7)	84.0
DINOv2 [37]	TMLR24	ViT-B	LVD	81.8 (3)	83.2 (3)	84.0 (3)	×
CLIP [42]	PMLR21	ViT-L	WIT	77.2 (4)	82.3 (4)	83.4 (5)	×
SigLIP [54]	ICCV23	ViT-L	WebLI	83.7 (2)	85.9 (2)	86.1 (2)	×
SigLIP2 [46]	arXiv25	ViT-L	WebLI	84.4 (1)	86.9 (1)	87.1 (1)	×

Figure 5 reveals a strong correlation between the attention predictors' localization quality or entropy and their impact on classification accuracy: better localization quality and lower entropy consistently result in higher influence on accuracy. This trend holds across various attentive methods, including EP and its variants with reduced output dimensionality D_o. As previously observed (Figure 3), lower D_o values generally degrade classification accuracy. Interestingly, Figure 5 (rightmost plot) further suggests that this decline is not solely due to reduced representational capacity, but also due to diminished attention quality, as indicated by increased entropy in the attention distributions.

4.5. Visualizations

We visualize the attention maps of EP_8. As shown in Figure 6, different queries consistently focus on distinct object parts, such as the head, wings, or body, revealing a complementary decomposition of the object. This suggests that EP distributes attention effectively across informative regions, resulting in richer and more structured representations.

4.6. Ablations

Impact of W_K in MHCA Our analysis in subsection 3.3 posits that while a single learnable query $\mathbf{q}$ can effectively absorb the key transformation in single-head attention, the same does not hold in multi-head. To empirically validate this, we probe MAE ViT-B with four variants: single-head *vs*. multi-head, each with and without W_K. Specifically, we evaluate single-head AbMILP and AIM with 12 heads (AIM_{12}). In single-head attention, removing W_K has minimal impact on performance (71.8% → 71.7%), while in multi-head the drop is noticeable (75.1% → 72.9%).

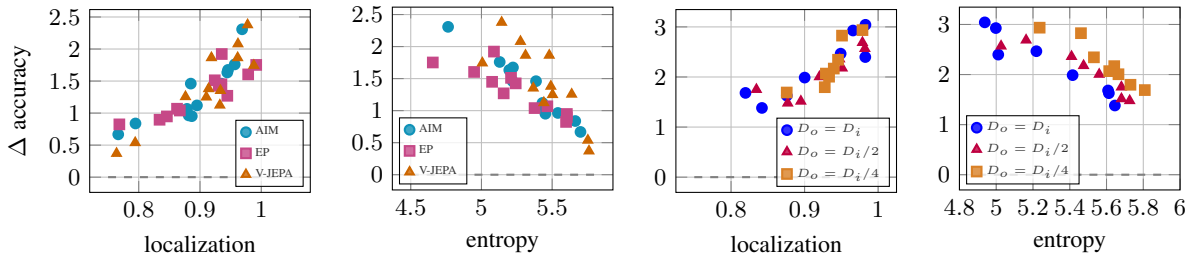

Figure 5. *Classification accuracy* vs. *attention quality* on ImageNet-1k. Each point corresponds to an attention predictor (head or query). We measure the classification accuracy drop (Δ accuracy) when replacing predictor's attention with a uniform distribution, and relate this to localization quality (1st and 3rd plots) and entropy (2nd and 4th plots). The first two plots compare different attentive probing methods, while the latter two vary output dimensionality D_o of EP.

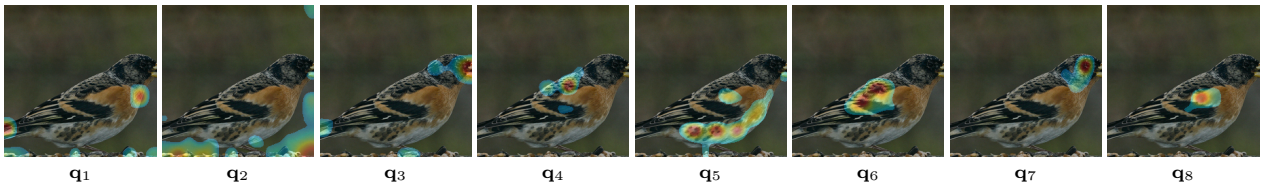

Figure 6. *Attention maps of efficient probing* with 8 queries (EP_8). Each query $\mathbf{q}_i$ learns to focus on distinct and complementary regions, capturing diverse spatial and semantic information. MAE ViT-B pre-trained on ImageNet-1k, probed with EP.

Impact of W_V We ablate the effect of the projection matrix W_V, which operates on patch tokens (2), by adding or removing it across pooling methods. Introducing W_V to GAP results in a top-1 accuracy improvement from 66.7% to 68.0%. Conversely, removing W_V from EP_{12} degrades performance from 75.1% to 72.1%. A similar accuracy drop is observed for other methods, such as AIM (75.1% → 72.0%) and CAE (74.9% → 72.2%), confirming that W_V is a critical component in maintaining performance.

Impact of attention predictors, D_a, and D_o We analyze the effect of increasing the number M of heads (in MHCA-based method AIM) and the number M of queries (in EP) on probing performance. Figure 7 shows that both lead to accuracy improvements. For AIM, increasing the number of heads incurs no additional cost in terms of parameters, but its effectiveness heavily depends on the presence of W_K. In contrast, EP achieves similar or better performance by leveraging additional queries, while removing W_K. AIM introduces an additional attention dimensionality D_a, since its query is learnable and interacts with W_K. Lowering D_a reduces the parameters but leads to a greater accuracy drop (green points), indicating that the learned query formulation benefits from a large attention space. We also evaluate the impact of reducing the output dimensionality D_o (blue points). On EP, we observe that lowering D_o to $D_i/2$ reduces parameters while maintaining competitive performance. Interestingly, this strategy also generalizes well to AIM, demonstrating that extracting lower-dimensional features can achieve comparable accuracy with reduced computational cost. In all cases, EP consistently matches or outperforms AIM, while remaining more parameter efficient.

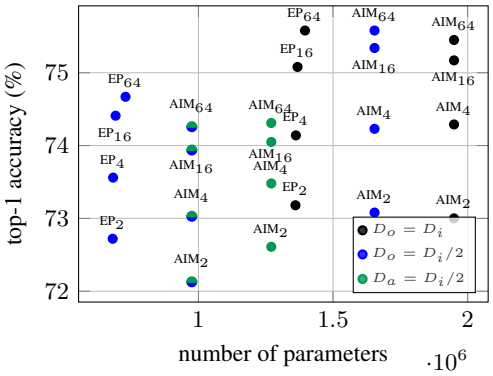

Figure 7. *Effect of varying the number of heads/queries M, D_a and output dimension D_o* on probing accuracy. Black: standard setting ($D_o = D_i$); blue: reduced classifier dimension ($D_o = D_i/2$); green: reduced attention dimension ($D_a = D_i/2$); half-blue-half-green: simultaneous reduction of both D_o and D_a.

5. Conclusion

We revisit evaluation protocols for pre-training methods and introduce EP, a scalable alternative to the increasingly unsustainable fine-tuning. EP is a lightweight attentive probing method that eliminates redundant projections and leverages multiple learnable queries for efficient and expressive feature aggregation. It produces interpretable attention maps with strong localization, generalizes well across models and pretraining paradigms, and consistently outperforms linear probing, achieving up to +13.6% on ImageNet-1k.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023. 1

[2] Maximilian Augustin, Syed Shakib Sarwar, Mostafa Elhoushi, Sai Qian Zhang, Yuecheng Li, and Barbara De Salvo. Petah: Parameter efficient task adaptation for hybrid transformers in a resource-limited context, 2024. 12

[3] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 1

[4] Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024. 2, 5, 12

[5] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Dublin, Ireland, 2022. Association for Computational Linguistics. 12

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 5, 13

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 2, 7

[8] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 12

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 2

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 1

[11] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *Int. J. Comput. Vision*, 132(1):208–223, 2023. 1, 2, 5, 12

[12] Timothée Darcet, Federico Baldassarre, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Cluster and predict latents patches for improved masked image modeling. *arXiv preprint arXiv:2502.08769*, 2025. 1, 2, 5, 7

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5, 7, 13

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2, 5

[15] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models, 2024. 2, 4, 5, 12

[16] Cheng Fu, Hanxian Huang, Xinyun Chen, Yuandong Tian, and Jishen Zhao. Learn-to-share: A hardware-friendly transfer learning framework exploiting computation and parameter sharing. In *Proceedings of the 38th International Conference on Machine Learning*, pages 3469–3479. PMLR, 2021. 12

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 1, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15979–15988. IEEE, 2022. 1, 2, 5, 7

[20] Xin He, Yushi Chen, Lingbo Huang, Danfeng Hong, and Qian Du. Foundation model-based multimodal remote sensing data classification. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–17, 2024. 12

[21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 5

[22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. 12

[23] Leiyi Hu, Hongfeng Yu, Wanxuan Lu, Dongshuo Yin, Xian Sun, and Kun Fu. Airs: Adapter in remote sensing for parameter-efficient transfer learning. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–18, 2024. 12

[24] Chia-Chien Hung, Lukas Lange, and Jannik Strötgen. Tada: Efficient task-agnostic domain adaptation for transformers. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023. 12

[25] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 15

[26] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 12

[27] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *Computer Vision – ECCV 2022*, pages 300–318. Springer Nature Switzerland, 2022. 1

[28] Samar Khanna, Medhanie Irgau, David B. Lobell, and Stefano Ermon. Explora: Parameter-efficient extended pretraining to adapt vision transformers under domain shifts, 2025. 12

[29] Jonathan Krause, Michael Stark, Jia Deng, and Fei-Fei Li. 3d object representations for fine-grained categorization. *ICCVW*, 2013. 5, 13

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 5, 13

[31] Minglei Li, Peng Ye, Yongqi Huang, Lin Zhang, Tao Chen, Tong He, Jiayuan Fan, and Wanli Ouyang. Adapter-x: A novel general parameter-efficient fine-tuning framework for vision. *CoRR*, abs/2406.03051, 2024. 12

[32] Tao Li, Zhengbao He, Yujun Li, Yasheng Wang, Lifeng Shang, and Xiaolin Huang. Flat-lora: Low-rank adaptation over a flat loss landscape, 2025. 12

[33] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013. 2

[34] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020. 4

[35] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 5, 13

[36] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017. 2, 4, 5

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. 2, 7

[38] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *CoRR*, abs/2208.06366, 2022. 1, 2, 5, 7

[39] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 12

[40] Marcin Przewieźlikowski, Randall Balestriero, Wojciech Jasiński, Marek Śmieja, and Bartosz Zieliński. Beyond [cls]: Exploring the true potential of masked image modeling representations. *arXiv preprint arXiv:2412.03215*, 2024. 2

[41] Bill Psomas, Ioannis Kakogeorgiou, Konstantinos Karantzalos, and Yannis Avrithis. Keep it simpool: Who said supervised transformers suffer from attention deficit? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5350–5360, 2023. 2, 5

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 5, 7, 12

[43] Dawid Rymarczyk, Adriana Borowa, Jacek Tabor, and Bartosz Zieliński. Kernel self-attention in deep multiple instance learning, 2021. 2, 4, 5, 12

[44] Romain Thoreau, Valerio Marsocci, and Dawa Derksen. Parameter-efficient adaptation of geospatial foundation models through embedding deflection. *arXiv preprint arXiv:2503.09493*, 2025. 12

[45] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 32–42. IEEE, 2021. 2, 5, 12

[46] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 7

[47] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5, 13

[48] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14668–14678, 2022. 1

[49] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 2, 5, 12

[50] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. In *ECCV (7)*, pages 3–19. Springer, 2018. 2, 5

[51] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: a simple framework for masked image modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition,*

CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pages 9643–9653. IEEE, 2022. 2, 5, 7

[52] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11772–11781, 2021. 2

[53] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. 2, 5, 12

[54] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023. 5, 7, 12

[55] Bingchen Zhao, Haoqin Tu, Chen Wei, Jieru Mei, and Cihang Xie. Tuning layernorm in attention: Towards efficient multi-modal llm finetuning, 2023. 12

[56] Houqiang Zhong, Shaocheng Shen, Ke Cai, Zhenglong Wu, Jiangchao Yao, Yuan Cheng, Xuefei Li, Xiaoyun Zhang, Li Song, and Qiang Hu. Serial low-rank adaptation of vision transformer, 2025. 12

[57] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, 2014. 5, 13

[58] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 1, 7

[59] Yitao Zhu, Zhenrong Shen, Zihao Zhao, Sheng Wang, Xin Wang, Xiangyu Zhao, Dinggang Shen, and Qian Wang. Melo: Low-rank adaptation is better than fine-tuning for medical image diagnosis. In *IEEE International Symposium on Biomedical Imaging, ISBI 2024, Athens, Greece, May 27-30, 2024*, pages 1–5. IEEE, 2024. 12

A. Additional related works

A.1. Parameter-efficient fine-tuning

Parameter-efficient fine-tuning (PEFT) adapts large pre-trained models to downstream tasks without updating all model parameters. PEFT techniques broadly include *additive*, *selective*, and *low-rank adaptation* methods.

Additive methods introduce small, task-specific modules into the frozen backbone, leaving the pre-trained weights untouched. These modules often reside within the transformer blocks and are trained to specialize the model for a new domain. Notable examples include AdapterFusion [39], LeTS [16], and TADA [24] in natural language processing (NLP), VPT [26], AdaptFormer [8], and Adapter-X [31] in computer vision (CV), and FMA [20], AiRs [23], and DEFLECT [44] in remote sensing (RS).

Selective methods fine-tune only specific subsets of parameters, typically chosen based on their functional role or estimated importance. Examples include BitFit [5], which adjusts only the bias terms, and norm-tuning approaches [55] which update only the normalization layers. These techniques avoid introducing new components, making them lightweight, though sometimes at the expense of performance.

Low-rank adaptation methods like LoRA [22] in NLP assume that parameter updates lie in a low-dimensional subspace. They inject trainable low-rank matrices into existing layers, yielding strong performance with minimal parameter growth. In the vision domain, LoRa and its variants have been effectively adapted to vision transformers (ViTs), often rethinking where and how low-rank modules are inserted to align with the spatial and hierarchical nature of visual representations. Notable examples include structure-aware methods like Serial LoRA [56] and Flat-LoRA [32], layer-wise extensions such as AdaptFormer [8], and task specific designs like PETAH [2] and MeLo [59] which adapt LoRa to mobile inference and medical imaging, respectively. Continued pretraining approaches such as ExPLoRA [28] further extend low-rank adaptation to domain-shifted self-supervised settings.

EP naturally fits the additive PEFT family. It introduces a compact learnable query set interacting with frozen tokens via multi-head attention. Unlike typical prompt-based methods, it avoids backbone modifications and focuses training on minimal parameters. Thus, EP efficiently combines additive PEFT simplicity with task-specific attentive pooling.

B. Additional methods

B.1. Existing variants

We present here additional attentive probing or pooling methods evaluated in our experiments but not detailed in the main paper. These approaches represent variations of the MHCA framework, highlighting only their key deviations from the default design.

CLIP CLIP [42] differs from MHCA by employing self-attention rather than cross-attention. Specifically, CLIP prepends a global average pooled (GAP) token to the layer-normalized input features, treating this token as a global representation. All tokens, including the GAP token, are augmented with learnable positional encodings and processed through a single self-attention block (which includes a query projection matrix W_Q). The global representation is extracted from the output corresponding to the GAP token. Additionally, CLIP includes a linear projection matrix W_{proj} after attention aggregation. These modifications enable interactions across all tokens but increase parameter count and computational complexity.

CAiT CAiT [45] adapts MHCA-with-learnable-query formulation by concatenating the learnable query token with the input features and applying self-attention rather than cross-attention. It retains the query projection matrix W_Q and includes a linear projection matrix W_{proj} after attention aggregation, followed by an MLP block with GELU activations, residual connections, and LayerScale parameters. The global representation is obtained from the updated query token after these operations, thereby increasing complexity and parameter count relative to the default MHCA variant.

SigLIP SigLIP [49, 54] remains close to the MHCA-with-learnable-query formulation but retains the query projection matrix W_Q. After the attention aggregation, SigLIP incorporates an output projection W_{proj}, followed by a transformer-style MLP block with GELU activation and residual connections. Optional layer normalization can also be applied before the MLP. These changes add further parameters and computational overhead compared to the baseline MHCA design.

CAE CAE [11] follows the MHCA-with-learnable-query template closely but retains the query projection matrix W_Q and applies separate layer normalization to both input features and the query token prior to attention. After attention aggregation, it employs an additional output projection matrix W_{proj}. These modifications introduce additional parameters and computational complexity.

CoCa CoCa [53] is aligned with the MHCA-with-learnable-query framework but retains the query projection matrix W_Q and layer-normalizes the query token before computing attention. Attention and value aggregation both occur in a reduced-dimensional space, with dimension $D_a = D_o < D_i$. A final linear projection matrix W_{proj} is then applied to restore the feature dimension to the original backbone dimension D_i. These choices introduce a controlled amount of additional complexity and parameters.

Figure 8 presents a visual comparison of three selected attentive probing or pooling techniques: AbMILP [43], AIM [15], and V-JEPA [4]. AbMILP (top-left) serves as a lightweight method, employing a single-head learnable query without additional linear projection matrices, thus requiring only D_i parameters. AIM (top-right) extends this by adopting multi-head attention, operating within multiple subspaces. This approach introduces linear

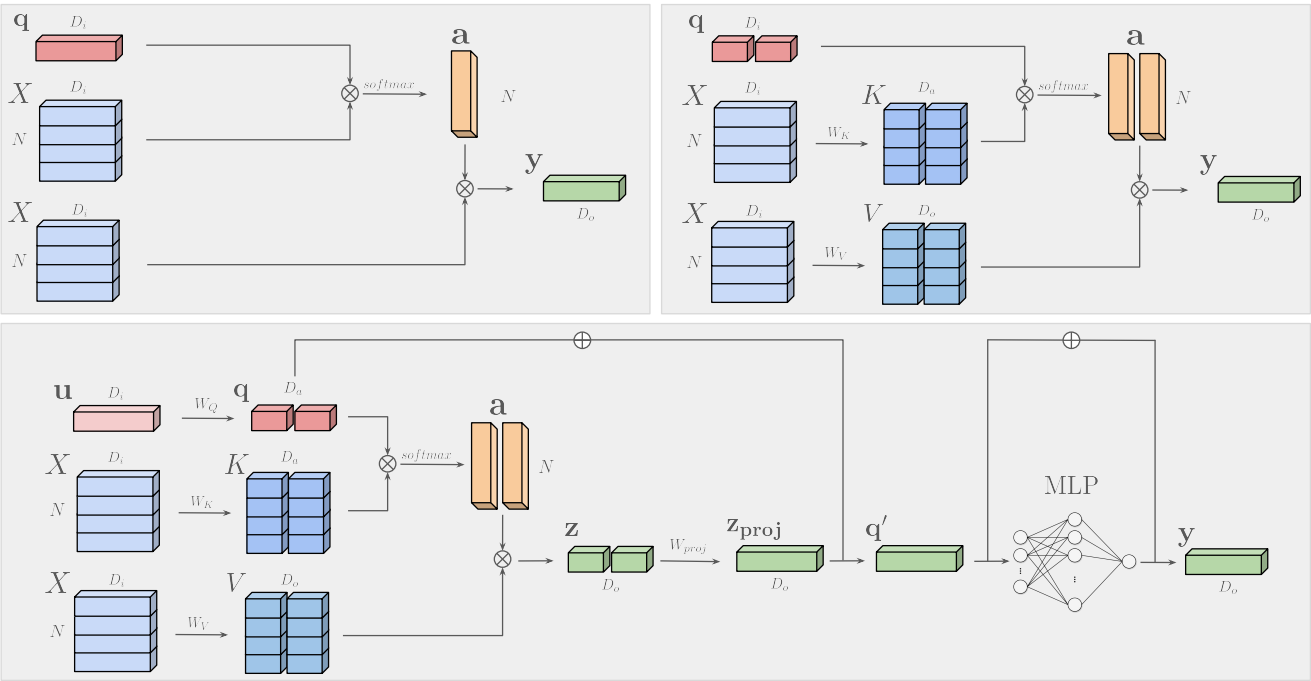

Figure 8. *Visual comparison of three attentive pooling methods*. AbMILP (top-left) employs a single-head, learnable query without linear projections, minimizing complexity. AIM (top-right) extends the approach by introducing multi-head attention, operating in multiple subspaces, and applies linear projections to keys and values. V-JEPA (bottom) offers a more comprehensive architecture by integrating multi-head attention with extensive linear projections and an additional MLP block with a residual connection, increasing representational capacity.

projection matrices for keys and values, increasing the number of parameters, yet allowing more expressive query-key interactions. V-JEPA (bottom) represents a significantly more complex and computationally intensive architecture. Beyond multi-head attention and multiple linear projections for queries, keys, and values, it integrates an additional projection step, followed by a multi-layer perceptron (MLP) featuring GeLU activation and residual connections.

C. More experiments

C.1. Setup

Datasets We evaluate attentive probing across diverse image classification benchmarks. As a large-scale dataset, ImageNet-1k [13] serves as the primary benchmark, containing 1.28M images across 1,000 categories. CIFAR-100 [30] provides a smaller yet challenging 100-class task with 60K images. To assess scene understanding, we use Places365 [57], comprising 1.8M images spanning 365 scene types. For fine-grained classification, we evaluate on CUB-200 [47] (11,788 images, 200 bird species), FGVC Aircraft [35] (10K images, 100 aircraft models), Stanford Cars [29] (16K images, 196 car types), and Food-101 [6] (101K images, 101 food categories).

C.2. Benchmark

We extend our benchmark presented on the main paper to include additional backbone sizes and datasets. Figure 9 presents the trade-off between top-1 accuracy and the number of parameters for various pooling and probing methods integrated into MAE with different backbone sizes (ViT-S and ViT-L). The evaluation spans multiple datasets, including FGVC-Aircraft, CUB200, Places365, and SUN397.

As shown in Figure 9a and Figure 9a, our proposed EP method consistently outperforms standard linear probing across MAE ViT-S and ViT-L backbones respectively. Notably, on MAE ViT-L, EP_{16} with $D_o = D_o/2$ achieves an accuracy boost of 79.1% surpassing linear probing by 3.1% while maintaining the same number of trainable parameters. Furthermore, EP_{128} reaches 79.4%, outperforming SigLIP, while reducing the number of trainable parameters by over 11M.

GAP and [CLS], the two primary baselines, exhibit high parameter efficiency but low classification accuracy. In contrast, multi-head attention methods such as SigLIP (54.39%), ViT (54.84%), and V-JEPA (52.44%) achieve higher accuracy, albeit at the cost of increasing the number of trainable parameters by up to 100×. More specifically, in Figure 9e we can observe that EP has the best trade-off between accuracy and parameters, achieving top-1 classification accuracy of 53.7% with just 1M extra trainable parameters (EP_{64}). In Figure 9c, our EP_{24} variant, for the FGVC-Aircraft dataset, achieves a remarkable accuracy boost of 61.2% (+19.5%), while maintaining lower parameter count than linear probing (41.7%). Similarly, in Figure 9d for the CUB200 dataset, our EP_{64} with $D_o = D_o/4$ variant achieves comparable accuracy (75.9%) with computationally costly poolings such as SigLIP (77.8%) with around 7M trainable parameters less.

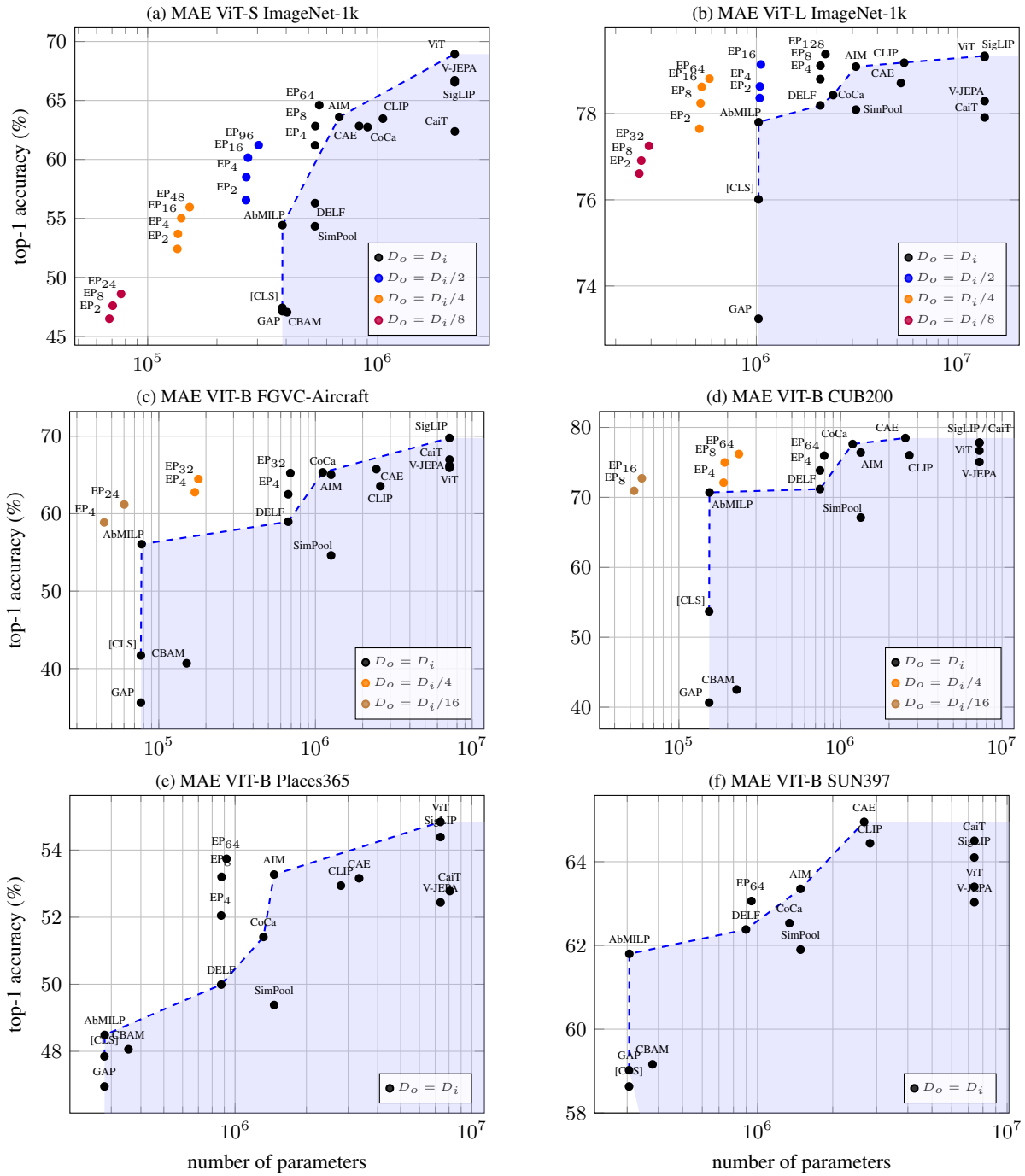

Figure 9. *Top-1 classification accuracy* vs. *number of parameters* for MAE framework, with backbones of varying size (a, b) and across various datasets (c, d, e, f). We evaluate attentive probing mechanisms originally designed for probing, alongside attentive pooling methods cast as probing. EP variants are marked with different colors for different output dimensionalities D_o. EP_M: efficient probing with M learnable queries. [CLS]: linear probing using the class token; GAP: global average pooling over patch tokens; ViT: default transformer block. EP consistently achieves better performance than linear probing *with up to 4.5× fewer parameters*.

Layer-wise probing Table 2 presents a layer-wise comparison between standard linear probing (LP) and efficient probing (EP) using patch token representations from intermediate layers of a pre-trained and frozen MAE with ViT-B. While LP exhibits a clear degradation in performance as we move toward earlier layers (dropping from 67.7% at layer 12 to just 45.8% at layer 6),

Table 2. *Top-1 accuracy on ImageNet-1k for MAE/ViT-B* using patch token representations from intermediate ViT layers. We report results for standard linear probing (LP) and efficient probing (EP), along with the accuracy gains achieved by EP over LP at each layer.

Layer	LP	EP	gains
12	**67.7**	75.6	+7.9
10	66.2	**75.9**	+9.7
9	64.5	75.4	+10.9
6	45.8	69.6	**+23.8**

EP demonstrates remarkable robustness. It maintains high accuracy even from lower layers, with performance stabilizing beyond layer 9. Notably, EP yields a significant relative improvement of +23.8% at layer 6 over LP, underscoring its ability to extract and utilize meaningful representations from less semantically enriched stages of the encoder. These results highlight the effectiveness of EP in unlocking information from earlier layers that standard LP fails to exploit.

Low-shot probing Table 3 evaluates the performance of linear probing (LP), efficient probing (EP), and fine-tuning (FT) under limited supervision, using only 5% and 10% of the ImageNet-1k training set, stratified by class. While LP struggles in this low-shot regime, EP substantially bridges the gap toward FT. Specifically, EP closes 74.8% and 71.5% of the LP→FT performance gap for the 5% and 10% subsets, respectively. These improvements are particularly impressive given that EP remains significantly more parameter-efficient than FT, with complexity comparable to LP (as further illustrated in Figure 9). These findings highlight the strong data efficiency of EP, making it a compelling alternative to fine-tuning when data or computational resources are limited.

Table 3. *Top-1 accuracy on ImageNet-1k with limited training data for MAE/ViT-B*. We report results for linear probing (LP), efficient probing (EP), and fine-tuning (FT) on 5% and 10% subsets of ImageNet-1k. The last column shows the percentage of the LP→FT performance gap closed by EP. For reference, the gap closed by EP on the full training set (100%) is 49.7%.

Subset	LP	EP	FT	% gap
5%	49.6	60.9	64.7	74.8%
10%	55.9	65.2	68.9	71.5%

C.3. Visualizations

To better understand the behavior of different attentive probing or pooling methods, we present qualitative visualizations of attention maps across various configurations.

Figure 10 shows the attention maps obtained from four single-head attention probing methods (CBAM, AbMILP, DELF, and SimPool) using an ImageNet-1K pretrained MAE ViT-B model. Among them, CBAM exhibits poor localization, often failing to focus on the target object, which is consistent with its low classification accuracy across datasets. In contrast, AbMILP, DELF, and SimPool produce more precise and meaningful attention, highlighting relevant object regions while suppressing background noise. Due to their single-head nature, these methods are compelled to concentrate all semantic information into a single attention vector, which encourages a global view of the input image rather than fine-grained discrimination.

Figure 11 compares attention maps from multi-head probing methods. Rather than visualizing just the average attention across heads,which can obscure useful per-head behavior, we show the minimum, maximum, and standard deviation across attention heads. The first column contains maps from the [CLS] token of the pretrained MAE ViT-B model. The remaining columns display maps from CAE, CaiT, CLIP, CoCa, ViT, V-JEPA, SigLIP, and AIM, alongside EP using 16 learnable queries (EP_{16}). Notably, EP produces high-quality attention maps that rival the best-performing methods in both clarity and relevance, while retaining computational efficiency.

Figure 12 presents the attention maps corresponding to each individual query in EP. We observe that each query $\mathbf{q}_i$ attends to distinct, complementary regions of the object (*e.g.*, head, torso, boundaries), illustrating how EP distributes attention cooperatively across salient features without redundancy. This diversity among queries reveals the model's capacity to decompose complex objects into meaningful sub-parts.

Figure 13 explores the effect of varying the number of queries in EP, visualizing configurations with 1, 2, 4, and more queries. When only a single query (EP_1) is used, the attention map tends to capture a coarse, global representation of the object. As the number of queries increases, the attention becomes more fine-grained and spatially distributed, with each query specializing in distinct object regions. This highlights the flexibility of EP in controlling the granularity of attention: fewer queries encourage holistic coverage, while more queries promote detailed, part-based localization.

C.4. Implementation details

We evaluate four masked image modeling (MAE, BEiTv2, SimMIM, CAPI), two joint-embedding (BYOL, DINO), two hybrid (iBOT, DINOv2), and three vision-language pre-training methods (CLIP, SigLIP, SigLIP2). All models use the ViT-B architecture unless it is not available; for example, BYOL is implemented with ResNet-50, and CAPI uses ViT-L. To ensure a fair comparison of different probing and pooling methods, we use the LARS optimizer and perform a learning rate search in the range [0.1, 5.0] with a step size of 0.1 for each model. For large-scale datasets such as ImageNet-1k and Places365, we fix the learning rate to 0.1 due to the computational cost of an exhaustive search. All models are trained for 90 epochs with 10 warmup epochs, ensuring consistent training schedules even though many models converge earlier. The effective batch size is set to 4096 for all datasets except FGVC Aircraft, where it is reduced to 512 due to the dataset's smaller size. All models use standard image pre-processing, including RandomResizedCrop, horizontal flipping, and normalization. For vision-language models, we adopt their official pre-processing pipelines (*e.g.*, OpenCLIP [25] transforms for CLIP and SigLIP) to ensure consistency with pre-training distributions. All experiments are conducted on a cluster of 8 NVIDIA A100 GPUs, each equipped with 40 GB of VRAM.

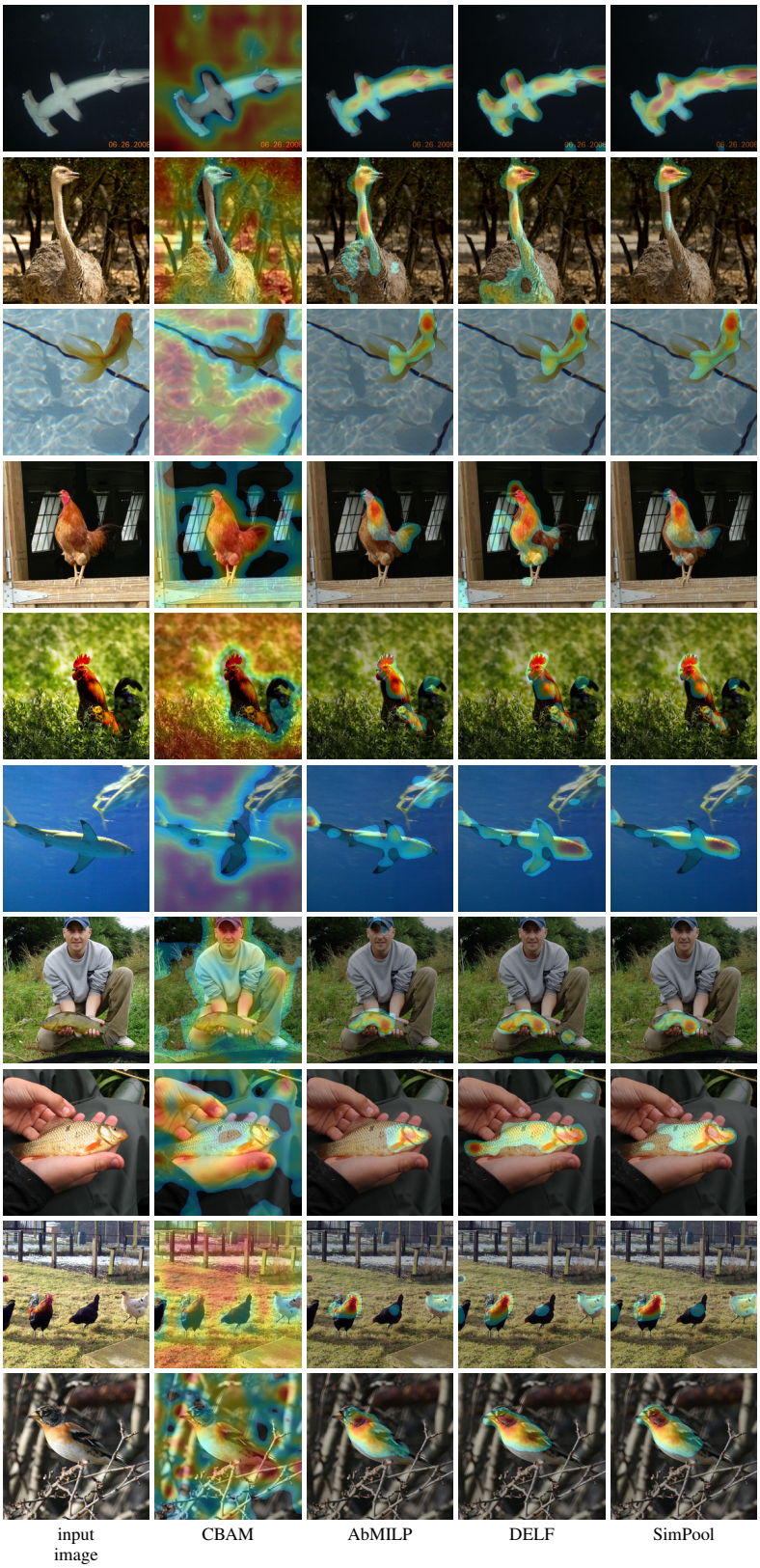

Figure 10. *Attention maps of single-head attention pooling methods*. MAE ViT-B pre-trained on ImageNet-1k. Images: ImageNet-1k validation set.

Figure 11. *Attention maps of multi-head attention pooling methods* for different attention predictor aggregators: mean, standard deviation (std), minimum (min), and maximum (max). MAE ViT-B pre-trained on ImageNet-1K. Images: ImageNet-1k validation set. EP_{16}: efficient probing (EP) with 16 queries.

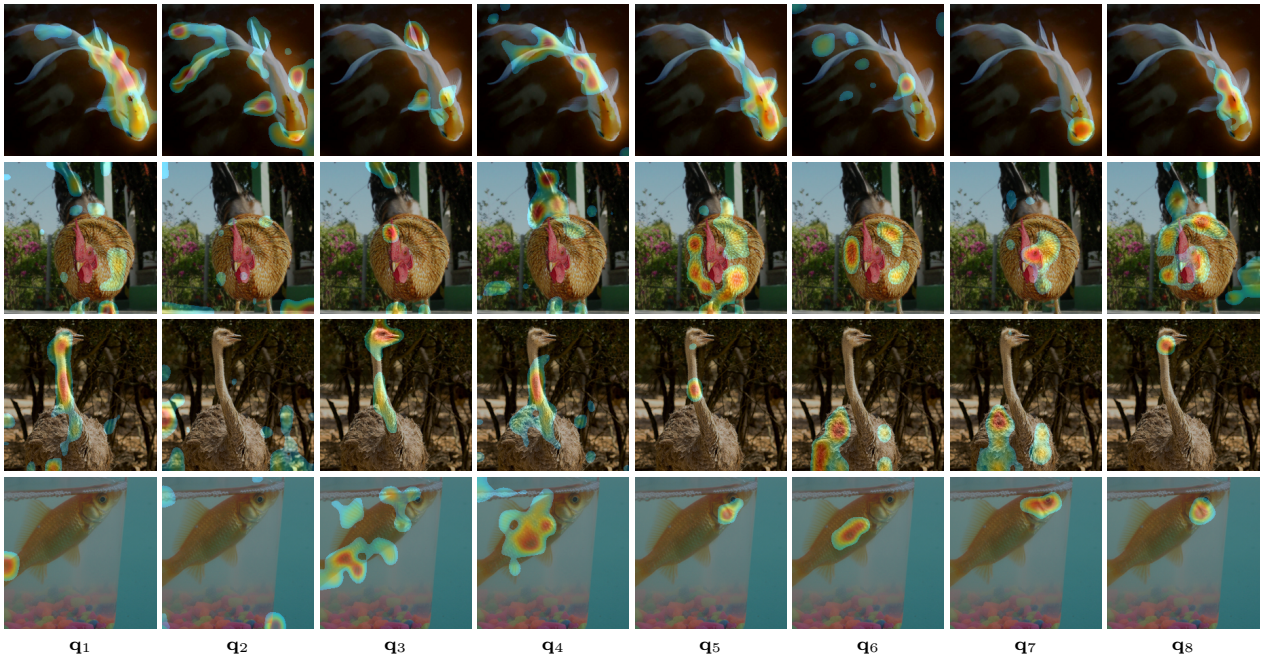

Figure 12. *Attention maps of efficient probing* (EP) with 8 queries. Each query $\mathbf{q}_i$ learns to focus on distinct and complementary regions, capturing diverse spatial and semantic information. MAE ViT-B pre-trained on ImageNet-1K, probed with EP. Images: ImageNet-1k validation set.

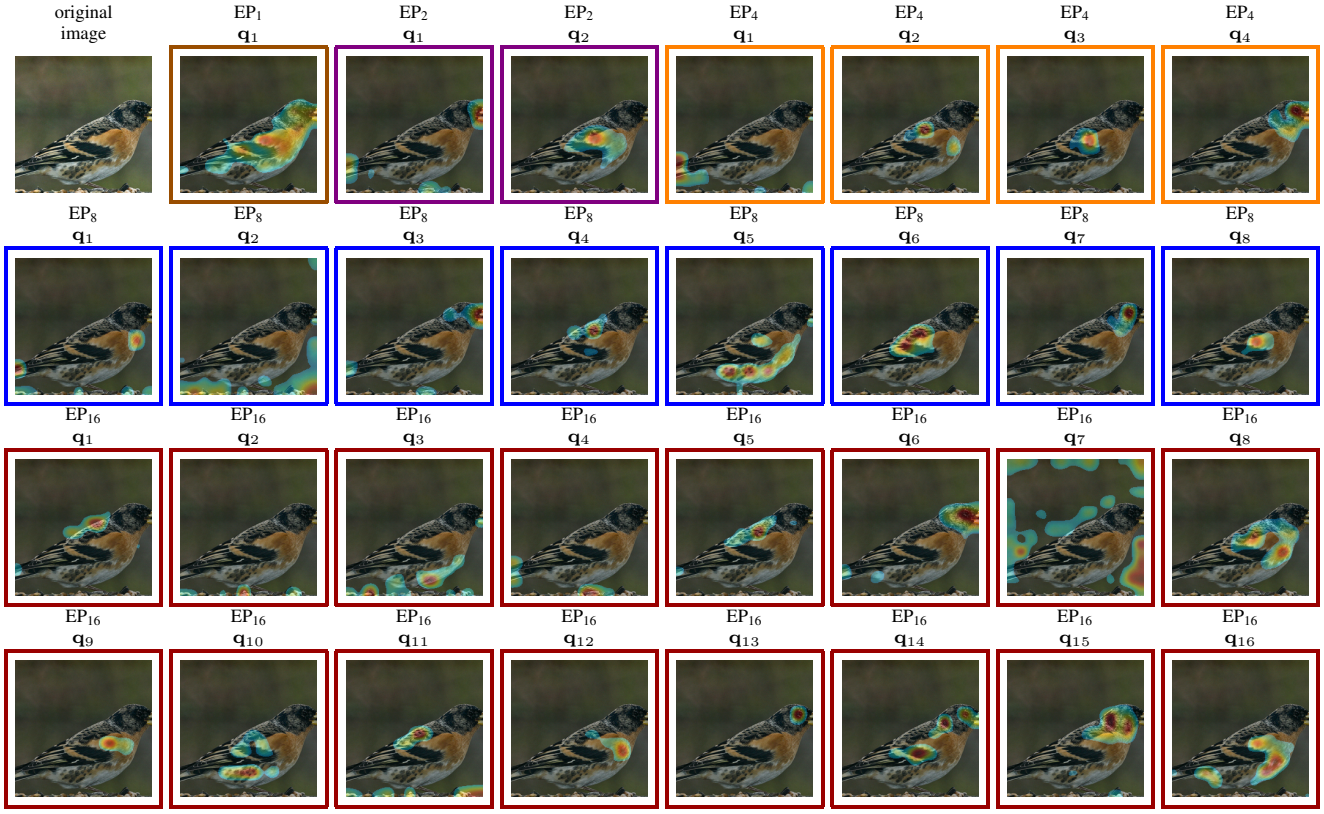

Figure 13. *Attention maps of efficient probing* (EP) variants grouped by the number of queries (1, 2, etc.). MAE ViT-B pre-trained on ImageNet-1K, probed with EP. Images: ImageNet-1k validation set.