Learning from Offline Foundation Features with Tensor Augmentations

Emir Konuk[1,2], **Christos Matsoukas** [1,2], **Moein Sorkhei** [1,2] , **Phitchapha Lertsiravaramet** [1,2]
Kevin Smith [1,2]

[1] KTH Royal Institute of Technology, Stockholm, Sweden
[2] Science for Life Laboratory, Stockholm, Sweden
{ekonuk, ksmith}@kth.se

Abstract

We introduce Learning from Offline Foundation Features with Tensor Augmentations (LOFF-TA), an efficient training scheme designed to harness the capabilities of foundation models in limited resource settings where their direct development is not feasible. LOFF-TA involves training a compact classifier on cached feature embeddings from a frozen foundation model, resulting in up to 37× faster training and up to 26× reduced GPU memory usage. Because the embeddings of augmented images would be too numerous to store, yet the augmentation process is essential for training, we propose to apply tensor augmentations to the cached embeddings of the original non-augmented images. LOFF-TA makes it possible to leverage the power of foundation models, regardless of their size, in settings with limited computational capacity. Moreover, LOFF-TA can be used to apply foundation models to high-resolution images without increasing compute. In certain scenarios, we find that training with LOFF-TA yields better results than directly fine-tuning the foundation model.

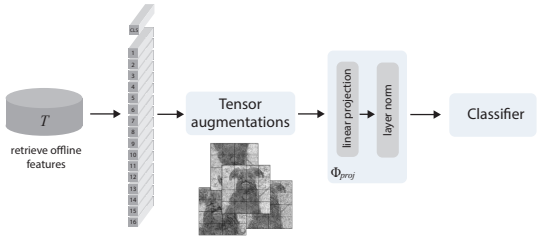
1 Introduction

Large and expensive foundation models, designed to capture general-purpose knowledge, have become a significant focus in machine learning and computer vision research [3]. These models excel in zero- and few-shot learning [4, 23] and adapt to various domains, especially in data-scarce scenarios, through transfer learning. But adapting these models for a specific task is a resource-intensive process [35]. The cost of fine-tuning large foundation models today is already prohibitive to most individuals and organizations. As they continue to grow in size, the rising costs of foundation models risk excluding all but the wealthiest organizations. To mitigate this, parameter-efficient fine-tuning methods have been proposed. These methods incorporate rank-deficient [18] and simple affine modules [31] or learnable prompt parameters injected at the input stage [20]. Their core principle is to limit the number of parameters that need to be trained. We extend this principle to its logical conclusion by introducing no intermediate parameters or prompts, and investigate whether it is possible to completely separate the foundation model from the training process.

In this study, we explore this complete separation of the resource-intensive foundation model from the training process. As seen in Figure 1, training data is passed through the foundation model at a one-time cost, and then cached. The cached feature embeddings are later loaded and used to train a lightweight classifier. By adopting this caching strategy we can train at a significantly faster rate using less memory resources and achieve similar,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3817



	Method	Size	Performance ↑ Relative	Mem., Gb ↓ During training
ViT-G	Frozen + linear	256	0%	7.2
	LOFF-TA	256	+1.9%	13.2
	LOFF-TA + Pooling	512	+4.1%	13.2
	Unfrozen + linear	256	+3.7%	345.2 [1]

Figure 1: *Learning from Offline Foundation Features with Tensor Augmentations (LOFF-TA).* Training data is passed through a foundation model and cached. The cached embeddings are loaded and spatial tensor augmentations are applied in lieu of standard image augmentations. A lightweight classifier is trained on the cached, augmented features. This enables the use of arbitrarily large foundation models and high-resolution images at no additional cost.

or even in some cases, better performance. This framework enables us to leverage the power of *foundation models of any size* in limited resource settings. Moreover, tasks requiring high-resolution imaging, such as medical image diagnosis can benefit from the power of foundation models without increasing computational costs. However, these benefits come at the cost of inference speed, which can be slower as a consequence of our paradigm.

We call this approach "Learning from Offline Foundation Features with Tensor Augmentations", or LOFF-TA. It is a simple approach, but has not yet been explored as it is complementary to existing adaptation methods. LOFF-TA can be trivially combined with existing adaptation methods [18, 31, 20] by caching features from the adapted foundation models themselves to achieve even better performance. We conduct a series of comprehensive experiments using LOFF-TA on eleven well-known image classification benchmark datasets to demonstrate its benefits and limitations. Our key findings and contributions are outlined as follows:

- We propose, LOFF-TA which decouples the training process from the resource-intensive foundation model – a classifier is trained on cached features from foundation models instead of images.
- Since integrating image augmentations within LOFF-TA presents a challenge due to the tremendous storage cost of caching the embeddings of augmented images, we propose to apply spatial tensor augmentations to the cached embeddings of original images when training the compact classifier. We show that they perform nearly as well as standard image augmentations.
- We show that, using LOFF-TA, it is possible to achieve similar performance to a fine-tuned foundation model at a fraction of the computational cost – training speed is accelerated up to 37×, and GPU memory usage is reduced up to 26×. Additionally, LOFF-TA allows flexibility in choosing any input image size or foundation model according to need and available resources.
- Surprisingly, in some cases, LOFF-TA outperforms a fine-tuned foundation model.

Despite the simplicity of our approach, our findings indicate that there are many potential benefits, in terms of both performance and economics, to training from cached foundation features. The source code used in this work can be found at https://github.com/emirkonuk/loffta.

2 Related work

Foundation models [3] such as LLaMA [38] and GPT [4] have revolutionized natural language processing through large-scale training, requiring huge datasets and substantial computational resources. In a rapidly advancing competitive landscape, these models are

[1] It was not possible to train ViT-G on a single GPU with batch size of 64. Instead we report the memory footprint across 8 NVIDIA Quadro RTX 8000 using distributed training.

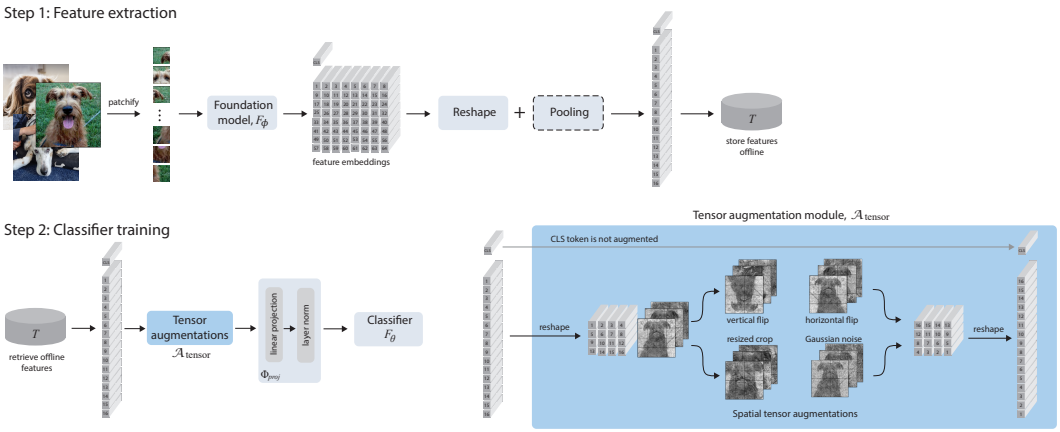


Figure 2: *Overview of LOFF-TA.* **Step 1:** We leverage a foundation model to process the training data and store the extracted features offline. **Step 2:** The cached tensors are loaded, tensor augmentations are applied, then the augmented tensors are passed through projection and normalization layers and used to train a lightweight classifier. The tensor augmentations include spatial-based transforms, such as flips and crops, along with additive Gaussian noise. An optional pooling step (dashed operation) reduces the spatial dimension of the stored features, allowing for training with high-resolution images at no additional cost.

scaling up dramatically to achieve unprecedented capabilities. In computer vision, this trend is mirrored with models like DINOv2 [35], CLIP [37], OpenCLIP [19], BLIP [30], SAM [23] and SEEM [47].

The standard approach for adapting a foundation model for a specific task involves fine-tuning its parameters or integrating additional trainable layers or classifiers [3, 13]. However, as the sizes of foundation models expand, these computational demands will escalate, posing significant challenges. To overcome these challenges, researchers have developed several strategies. These include fine-tuning only a subset of a model's parameters [46] and using techniques like gradient check-pointing for resource optimization [7]. Adaptation methods like [31, 17, 6, 18] introduce a limited number of tunable modules within the foundation model and incorporate learnable 'prompts' in the stem [20]. The result from these approaches is an efficiently fine-tuned, slightly bigger foundation model. Prior to deep learning, machine learning typically involved a two-stage process: first extract relevant features from data, then train a model on these features [2]. LOFF-TA, with its caching of features, echoes this latter approach [15]. As such, it complements the adaptation methods [31, 17, 6, 18, 20] and can be used in conjunction with them to improve overall performance, as we show in Section 5.5.

Augmentations. Developing novel image augmentation strategies mostly rely on visual insights to design augmentations [45]. The underlying principle is the manifold hypothesis [5] and effective augmentations should not move samples too far from the image manifold. While previous work on feature augmentations relied on interpolating features of different images [42] and adding noise to these interpolations [11] to ensure the augmented features stay close to the manifold, we find that feature augmentations need not be so limited. In this work, we show that it is surprisingly beneficial to apply spatial augmentations to features, in a manner similar to image augmentations.

3 Methods

In this work, we propose to pass training data through the foundation model, cache its features, and use them to train a lightweight classifier. We term this *Learning from Offline Foundation Features* (LOFF). A key challenge is LOFF's inability to incorporate image augmentations. To overcome this, we apply spatial and noise tensor augmentations directly

https://doi.org/10.52202/079017-3817

to the pre-stored foundation features, resulting in LOFF-TA. Finally, to allow the use of high-resolution images, we pool the foundation features.

3.1 LOFF

Our approach embraces a straightforward yet powerful idea: to separate feature extraction using a powerful foundation model from the training process using a lightweight classifier, as depicted in 2. We use a foundation model to process training data upfront, then extract and store the output features (Step 1). These serve as rich representations of the data which can be retrieved at a later time for training one (or more) classifiers, as depicted in Step 2.

In detail, given a dataset D, we generate and store a new dataset T by applying a foundation model F_ϕ to all images x from D. Each sample t in T is a $k \times d$ dimensional tensor where k is the number of tokens for each input image x and d is the embedding dimensionality.

$$T \coloneqq \{t \in \mathbb{R}^{k \times d}, \mid t = F_\phi(x)\}. \tag{1}$$

Importantly, *this phase only needs to be performed once*. The extracted features, T, along with the corresponding labels for each image, are cached and used as a replacement for D. The offline foundation features T are retrieved to train a lightweight classification model F_θ. Finally, we employ a standard cross-entropy loss, denoted as $\mathcal{L}$:

$$\arg\min_\theta \mathcal{L}(y, F_\theta(\Phi_{proj}(t))) \tag{2}$$

where, y is the true label, $t \in T$ the stored features and Φ_{proj} is a projection module. The role of the projection module Φ_{proj} is to match the cached features to the dimensions expected by the classification model. Φ_{proj} consists of a learnable linear projection layer followed by a Layer Normalization (LN) operation [1], which we found to improve performance.

High-resolution images Many foundation models are trained to handle images larger than 224×224, such as [23, 35], but GPU memory limitations on conventional hardware make it impossible for them to be fine-tuned on high-resolution images. LOFF can mitigate this issue by pooling the features before storing them as shown in Fig. 2 (Step 1). The reduced size of the pooled features allow them to be stored and used to train a classifier efficiently. We investigate both average pooling and max pooling operations [27] and assess their impact on classification performance and computational cost.

3.2 LOFF-TA

Given that image augmentations are crucial for effective model training, the inability to apply them in the LOFF framework poses a significant challenge. The obstacle lies in the impracticality of storing tensors from augmented images during Step 1, which would result in prohibitive storage demands. To address this drawback, we introduce tensor augmentations $\mathcal{A}_{\text{tensor}}(t)$ on the features $t \in T$. A tensor augmentation module is applied dynamically online, before a batch of features is fed into the classifier, as depicted in 2. This method, named *Learning from Offline Foundation Features with Tensor Augmentations* (LOFF-TA), allows augmentations to be used for overcoming the aforementioned obstacle with image augmentations.

In a standard setting, the objective for image classification is given by

$$\arg\min_\theta \mathcal{L}(y, F_\theta(\mathcal{A}_{\text{img}}(x))) \tag{3}$$

where F_θ is the model to be trained on each sample (x, y) from the dataset $\mathcal{D}$ and $\mathcal{A}_{\text{img}}$ indicates stochastic image augmentations. LOFF-TA changes the optimization task to become

$$\arg\min_\theta \mathcal{L}(y, F_\theta(\Phi_{proj}(\mathcal{A}_{\text{tensor}}(t)))) \tag{4}$$

where $\mathcal{A}_{\text{tensor}}$ denotes our tensor augmentation operator. The stored features $t \in T$ and projection module Φ_{proj} remain the same as in LOFF.

Table 1: *Main results.* We train models on features extracted by DINOv2 [35] ViT-B and ViT-G models. We report the results using LOFF (no augmentations) and LOFF-TA (with tensor augmentations). We consider features extracted from 256×256 and 512×512 images (using pooling as described in 3.1). *Frozen + linear* and *Unfrozen + linear* are points of comparison consisting of a frozen/unfrozen foundation model with a linear layer trained on images directly, with image augmentations.

	Method	Size	**APTOS**, $\kappa \uparrow$ n = 3,662	**AID**, Acc. $\uparrow$ n = 10,000	**DDSM**, AUC $\uparrow$ n = 10,239	**ISIC**, Rec. $\uparrow$ n = 25,333	**NABirds**, Acc. $\uparrow$ n = 48,562	**TP**, Im/sec $\uparrow$ Train (Infer.)	**Mem.**,Gb $\downarrow$ Training
ViT-B	Frozen + linear	256	88.6 ± 0.3.	90.9 ± 0.1	90.3 ± 0.2	51.7 ± 1.0	86.0 ± 0.1	153 (313)	**1.8**
	LOFF	256	89.6 ± 0.2	91.9 ± 0.3	94.2 ± 1.2	70.8 ± 2.1	83.0 ± 0.1	**228** (236)	13.2
	LOFF-TA		90.4 ± 0.6	92.3 ± 0.7	94.4 ± 0.1	72.8 ± 1.7	83.5 ± 0.3	227 (236)	13.2
	LOFF + Pool	512	89.4 ± 1.5.	93.2 ± 0.6	95.3 ± 0.5	74.3 ± 1.5	86.2 ± 0.3	**228** (61)	13.2
	LOFF-TA + Pool		**90.5 ± 1.0**	**93.7 ± 0.3**	**95.5 ± 0.1**	**77.4 ± 0.0**	**86.8 ± 0.4**	227 (61)	13.2
	Unfrozen + linear	256	90.5 ± 0.9	93.7 ± 0.8	93.3 ± 0.9	76.8 ± 0.7	85.8 ± 0.1	77 (313)	28.2
ViT-G	Frozen + linear	256	88.2 ± 0.3	92.8 ± 0.2	90.8 ± 0.6	66.4 ± 1.1	89.8 ± 0.2	14 (28)	**7.2**
	LOFF	256	88.6 ± 1.5	93.3 ± 0.5	94.8 ± 1.6	73.1 ± 0.5	87.4 ± 0.2	**222** (27)	13.2
	LOFF-TA		89.9 ± 0.4	94.0 ± 0.2	95.3 ± 0.1	76.0 ± 0.7	88.5 ± 0.2	218 (27)	13.2
	LOFF + Pool	512	90.3 ± 0.6	94.1 ± 0.2	95.4 ± 0.4	74.0 ± 1.6	88.8 ± 0.1	**222** (7)	13.2
	LOFF-TA + Pool		**91.8 ± 0.3**	**94.6 ± 0.2**	**96.3 ± 0.6**	**79.9 ± 0.2**	**90.1 ± 0.2**	218 (7)	13.2
	Unfrozen + linear	256	89.6 ± 0.6	96.2 ± 0.1	96.7 ± 0.2	87.3 ± 1.3	90.2 ± 0.1	6 (28)	345.2 [1]

3.3 Tensor augmentations

Spatial relationships in image data are crucial for understanding the content of the image. Directly applying augmentations haphazardly to the unstructured output tokens from the foundation model may lead to undesirable results. A key aspect of our approach involves the utilization of spatial tensor augmentations during the training phase, as depicted in 2. These augmentations are chosen to consider the spatial relationships in the data, similar to how image augmentations operate. We apply spatial augmentations, denoted as $\mathcal{A}_{\text{tensor}}$, to foundation features $t \in T$ after a reshaping operation. These augmentations are conceptually analogous to image augmentations, treating the foundation features as if they were low-resolution, hyper-spectral images. We select a set of *spatial augmentations* suited for feature-level transformations, chosen to enhance the training process while maintaining the integrity of spatial relationships. We *flip* by mirroring the tensor on its height or width axis, *resize* it by upsampling or downsampling its spatial dimensions using linear interpolation. We *shear* the tensor using nearest neighbor interpolation and *translate* it by shifting along its spatial dimensions. We *rotate* the tensor in its spatial dimensions around its spatial center using nearest neighbor interpolation. In addition to spatial augmentations, we apply additive *Gaussian noise* with zero mean to the feature tensor, similar to [11]. Although we considered channel augmentations, analogous to contrast or color augmentations in images, we did not find them to be beneficial. We also note that, like image augmentations, not all tensor augmentations types are appropriate in every setting[2].

4 Experimental setup

To evaluate the effectiveness of LOFF-TA we benchmark over eleven datasets from various domains using different foundation models, model capacities and image resolutions.

4.1 Models and implementation details

Foundation models. We employ two foundation model families: DINOv2 [35] and CLIP [37] (implemented by OpenCLIP [19]) as the basis for our investigations. For the majority of our experiments, we utilize the ViT-B and ViT-G architectures.

Classifiers. LOFF and LOFF-TA train a lightweight classifier on the features from a foundation model. While, in principle, any classifier can be used in this role, our experiments use DeiT-S [39]. The classifier is initialized using ImageNet [10] pre-trained weights, as we found empirically this gave a significant improvement over random initialization [14]. In some

[2]We omit vertical flips for the SUN397 dataset [44].

https://doi.org/10.52202/079017-3817

Table 2: *Expanded results on seven standard datasets.* We compare LOFF (no augmentations) and LOFF-TA (with tensor augmentations) against baselines *Frozen + linear*, *Unfrozen + linear* and *Frozen + DeiT-S* consisting of a frozen/unfrozen foundation model with a linear layer/DeiT-S classifier trained on images directly (with image augmentations). Results are reported for features extracted from OpenCLIP [19] and DINOv2 [35] using 256×256 images.

		Method	Oxford-III Pet n = 7,349	Flowers102 n = 8,189	Caltech-101 n = 8,677	StanfordCars n = 16,185	StanfordDogs n = 20,580	SUN397 n = 39,700	NABirds n = 48,562	TP / Mem. Im/s ↑ / Gb ↓
DINOv2	ViT-B	Frozen + linear	95.7 ± 0.1	99.7 ± 0.1	96.7 ± 0.4	87.9 ± 0.1	87.8 ± 0.1	75.4 ± 1.2	86.0 ± 0.1	153 / 1.8
		LOFF	94.5 ± 0.4	99.2 ± 0.1	96.2 ± 0.7	87.7 ± 0.6	84.6 ± 0.3	76.5 ± 0.1	83.0 ± 0.1	228 / 13.2
		LOFF-TA	95.2 ± 0.1	99.5 ± 0.1	97.0 ± 0.5	88.9 ± 0.4	85.3 ± 0.2	76.8 ± 0.1	83.5 ± 0.3	227 / 13.2
		Unfrozen + linear	94.8 ± 0.2	99.0 ± 0.1	97.1 ± 0.3	93.7 ± 0.1	86.4 ± 0.4	76.2 ± 0.2	85.8 ± 0.1	77 / 28.2
		Frozen + DeiT-S	94.8 ± 1.0	99.6 ± 0.0	97.0 ± 0.3	91.6 ± 0.2	87.4 ± 0.4	76.8 ± 0.1	85.4 ± 0.1	94 / 13.8
DINOv2	ViT-G	Frozen + linear	96.2 ± 0.1	99.7 ± 0.0	96.1 ± 0.4	90.2 ± 0.2	89.8 ± 0.1	78.2 ± 0.1	89.8 ± 0.2	14 / 7.2
		LOFF	95.5 ± 0.2	99.7 ± 0.1	96.0 ± 0.5	91.8 ± 0.0	89.0 ± 0.3	78.2 ± 0.2	87.4 ± 0.2	222 / 13.2
		LOFF-TA	95.8 ± 0.4	99.7 ± 0.1	96.8 ± 0.3	92.8 ± 0.1	89.2 ± 0.1	79.2 ± 0.3	88.5 ± 0.2	218 / 13.2
		Unfrozen + linear	96.0 ± 0.2	99.7 ± 0.1	97.5 ± 0.1	94.5 ± 0.6	90.1 ± 0.2	79.6 ± 0.6	90.2 ± 0.1	6 / 345.2 [1]
		Frozen + DeiT-S	96.1 ± 0.1	99.7 ± 0.0	96.7 ± 0.2	93.3 ± 0.3	89.3 ± 0.1	78.7 ± 0.3	88.9 ± 0.2	13 / 18.2
OpenCLIP	ViT-B	Frozen + linear	91.7 ± 0.3	90.4 ± 0.6	96.0 ± 0.3	94.1 ± 0.3	76.8 ± 0.1	77.7 ± 0.0	61.0 ± 0.3	206 / 1.8
		LOFF	91.4 ± 0.3	89.2 ± 0.1	94.6 ± 0.8	93.3 ± 0.2	72.7 ± 1.2	77.5 ± 0.1	70.7 ± 0.1	228 / 13.2
		LOFF-TA	91.8 ± 0.2	94.1 ± 0.7	95.4 ± 0.1	93.4 ± 0.1	74.1 ± 0.9	77.7 ± 0.2	71.2 ± 0.2	227 / 13.2
		Unfrozen + linear	92.5 ± 0.2	96.4 ± 0.2	96.4 ± 0.5	94.2 ± 0.2	80.1 ± 0.7	75.7 ± 0.1	79.1 ± 0.1	101 / 12.9
		Frozen + DeiT-S	92.9 ± 0.2	95.3 ± 0.3	96.0 ± 0.1	94.4 ± 0.4	79.5 ± 0.4	78.1 ± 0.3	72.1 ± 0.2	124 / 14.5

experiments, we compare against a frozen/unfrozen foundation model as a benchmark, with either a linear layer or DeiT-S on top as a classifier.

Implementation details. In our experiments, we utilize the AdamW optimizer [32], and a batch size of 64. We incorporate a learning rate warm-up strategy and manually decrease the learning rate by a factor of 0.1 when the validation performance plateaus. For lightweight classifier in LOFF and LOFF-TA, we implement modifications to the DeiT-S architecture [39]. We remove the patchifier from the model's stem and introduce a linear projection layer followed by a normalization layer as detailed in 3.

4.2 Datasets

Our evaluation spans eleven image classification datasets, covering a diverse spectrum of object categories. We begin with datasets with high-resolution images, as this setting highlights new capabilities made possible by LOFF-TA. We include APTOS2019 [21] for diabetic retinopathy detection, DDSM [29] for identifying masses in mammography, ISIC [40, 8, 9] for skin lesion classification, AID [43] for aerial image classification, and NABirds [41] for fine-grained bird species classification. The resolution of these datasets varies, but we resize them to 512 × 512. We extend our evaluation to a number of standard 256 × 256 resolution benchmark datasets: Flowers102 [34], NABirds [41], StanfordCars [26], StanfordDogs [22], Oxford-III Pet [36], Caltech-101 [12], and SUN397 [44]. For each dataset, we report metrics appropriate to its specific evaluation criteria. We adhere to official train/validation/test splits when available, or follow [24] in their absence. Standard practice of image normalization is maintained, using the mean and variance from the training sets.

5 Results

In this section we show that LOFF-TA achieves competitive, sometimes superior, results compared to the baselines while significantly reducing memory usage and training time.

5.1 Effectiveness of LOFF-TA

We begin our analysis with Table 1 where we focus on the cost-benefit trade-off of using LOFF-TA, considering DINOv2 as the foundation model. Our evaluation spans five datasets where higher resolution images are known to improve performance. We measure each approach in terms of performance, throughput (TP), and memory (Mem.) footprint. The study includes a comparison with two key configurations: a baseline approach, where a linear classifier is appended to a frozen foundation model (*frozen + linear*), and an upper-bound where the entire foundation model is fine-tuned in a typical fashion for transfer

Table 3: *Ablations*. We systematically remove components of LOFF-TA to investigate the impact of each contribution. Results are reported with DINOv2 [35] as the foundation model, adding trivial augment [33] as an augmentation strategy.

Foundation CLS	Layer norm	Gaussian noise	Spatial aug.	Trivial augment [33]	**Oxford-III Pet**	**Caltech-101**
✗	✓	✓	✓	✗	94.1	95.6
✓	✗	✓	✓	✗	94.9	94.8
✓	✓	✗	✗	✗	94.9	95.4
✓	✓	✗	✓	✗	95.1	96.2
✓	✓	✓	✓	✗	95.2	96.4
✓	✓	✗	✗	✓	**95.4**	**96.8**

learning (*unfrozen + linear*). Both employ standard image augmentations. In contrast, LOFF operates without augmentations, while LOFF-TA introduces tensor augmentations. We also explore the effects of pooling on foundational features, increasing image resolution from 256 to 512, to understand how these adjustments influence the performance and efficiency.

Our observations reveal several trends. Firstly, all LOFF and LOFF-TA variants (with an exception of NABirds) surpass the baseline in performance, with only a slight increase in memory usage but significantly faster training. Secondly, LOFF-TA consistently outperforms LOFF, confirming the importance of tensor augmentations. Moreover, upgrading to a larger foundation model (from ViT-B to ViT-G) doesn't alter memory or training speed. Pooling and working with higher resolution further improves performance, again without affecting memory or throughput. Remarkably, in many cases (6 out of 10), LOFF-TA with pooling exceeds the performance of the intended upper-bound model (*unfrozen + linear*). Finally, it's worth noting the training speed and memory efficiency of LOFF and LOFF-TA compared to fine-tuning the foundation model (*unfrozen + linear*); our approach offers a 37× acceleration during training, and a remarkable savings in memory as well – in fact, ViT-G is too large to fine-tune on a conventional GPU with a batch size of 64.

5.2 Further evidence

In Table 2, we continue our analysis in a similar fashion to Table 1, but in this case focusing on seven standard visual object recognition datasets. Our focus remains on assessing performance, throughput (TP), and memory usage (Mem.), but with image resolution of 256. The analysis compares LOFF, LOFF-TA, *frozen + linear*, and *unfrozen + linear*, and introduces *frozen + DeiT-S* as a benchmark for image vs. tensor augmentations (see Section 5.3).

The findings in Table 2 reinforce the patterns observed in Table 1. LOFF-TA maintains its superiority over LOFF, highlighting the efficacy of tensor augmentations. It's important to note that in this set of experiments, the best performing LOFF-TA + Pooling configuration using high-resolution images is not considered due to the datasets' inherent resolution limit of 256. Despite this, LOFF-TA generally matches or surpasses the baseline *frozen + linear* in performance, while delivering a notable increase in throughput. Although *frozen + linear* claims the smallest memory footprint, LOFF and LOFF-TA again show significant savings compared to fine-tuning a foundation model (*unfrozen + linear*). Comparing foundation models of similar capacity, we notice that DINOv2 outperforms OpenCLIP in five of the seven datasets using LOFF-TA. Finally, we once again see that swapping the foundation model (between DINOv2 ViT-B, OpenCLIP ViT-B, and DINOv2 ViT-G) results in no appreciable change in throughput or memory consumption for LOFF-TA, only differences in performance.

5.3 Image vs. tensor augmentations

Looking at Table 2 again, we investigate the effect of image augmentations versus tensor augmentations. LOFF-TA and *frozen + DeiT-S* share the same architecture (a cascaded model that consists of a frozen foundation model with an appended DeiT-S), and in both cases the foundation is frozen – the only difference is LOFF-TA uses tensor augmentations while *frozen + DeiT-S* uses image augmentations. We also report results (Table 5 in the Appendix) when we unfreeze the foundation model in the cascaded setting for both OpenCLIP and DINOv2 for completeness. In Table 2, we observe that image augmentations outperform

 https://doi.org/10.52202/079017-3817

tensor augmentations. As one might expect, tensor augmentations such as tensor rotation or cropping can not replicate the exact effects of image rotation or cropping since foundation models are not linear operators. Yet, the performance impact is surprisingly less than anticipated, which, considering LOFF-TA's significant computational savings, is noteworthy.

5.4 Ablation study

In this section, we try to understand the impact of each contribution to LOFF-TA. Utilizing DINOv2 [35], we conduct a study where components are systematically removed and then tested on the Oxford-III Pet and Caltech-101 datasets. Results are presented in Table 3.

CLS token. A perhaps obvious, but critical, insight from our study is the important role played by the foundation model's CLS token in contributing to classifier performance. When this token, representing global information, is integrated into the classifier training, it proves to be a key contributor to performance gains. However, the choice of how to integrate it was not trivial. We opted to integrate the offline CLS token from the foundation model with the learned CLS token of the classifier model by summation. Other possible ways to incorporate it could be to initialize the classifier's CLS to the offline foundation CLS, or concatenate them – although our non-exhaustive experiments testing these approaches were inferior.

Layer norm. We applied a Layer Norm [1] operation in the projection layer at the stem of the classifier. Similar to the CLS token ablation, we find that adding a layer norm operation before passing the foundation features to the classifier boosts performance. By centering input features, we believe the normalization plays a pivotal role similar to that in Dual Patchnorm [28], aligning and standardizing the input.

Pooling. Table 1 shows that pooling larger image features enhances performance without extra computational costs, enabling the use of large models for high-resolution tasks. In Appendix Table 6, we compare max and average pooling, finding both perform similarly, with max pooling slightly outperforming in larger models, possibly due to better noise reduction. Thus, max pooling is recommended for larger models. While this work focuses on pooling, alternative dimensionality reduction methods like strided convolutions or bi-linear interpolation could also be effective.

Augmentations. Since the cached features contain long-range spatial information that may potentially be harmed by our tensor augmentations, it is important to assess if these augmentations have a meaningful contribution to the performance of LOFF-TA. Comparing Row 3 and Row 4 of Table 3, we see that adding spatial augmentations results in a significant boost in performance. A smaller boost is observed when Gaussian noise is added in Row 5. Replacing our augmentations with Trivial augment [33] provides a further boost to performance, although this setup was not used in our other experiments.

5.5 LOFF-TA vs. foundation adaptation methods

LOFF-TA enables practitioners to use large foundation models *without any modifications*. A class of methods exists that modify foundation models for new tasks, so-called adaptation methods. Adaptation methods can be used in conjunction with LOFF-TA, but we also provide a comparison between them in Table 4. The results show that standalone LOFF-TA achieves competitive performance with VPT [20], Adaptformer [6] and SSF [31]. When LOFF-TA is combined with these methods, we observe noticeable improvements across the board, demonstrating that computationally efficient SOTA performance can be achieved by adapting foundation models and then applying LOFF-TA.

6 Discussion

Performance-efficiency trade-off. The aim of this paper is to consider the use of foundation models in a resource-limited setting. To do so, we propose to work with cached foundational features. The benefits of doing so are that training speed is significantly accelerated (up to 37×), and GPU memory usage is significantly reduced (up to 26×). Of course, these benefits

Table 4: *LOFF-TA can be used alongside foundation adaptation methods*. Below we report results using ViT-B. Standalone LOFF-TA performs comparable with other foundation adaptation methods and can easily be combined with them to further enhance performance.

Method	**APTOS**, $\kappa \uparrow$ n = 3,662	**AID**, Acc. ↑ n = 10,000	**DDSM**, AUC ↑ n = 10,239	**ISIC**, Rec. ↑ n = 25,333	**NABirds**, Acc. ↑ n = 48,562
LOFF-TA	90.4 ± 0.6	92.3 ± 0.7	94.4 ± 0.1	72.8 ± 1.7	83.5 ± 0.3
VPT [20]	89.6 ± 0.1	93.0 ± 0.1	91.4 ± 0.3	75.2 ± 1.1	85.8 ± 0.2
VPT + LOFF-TA	90.8 ± 0.4	93.1 ± 0.3	92.4 ± 0.3	79.7 ± 0.9	83.7 ± 0.1
SSF [31]	90.2 ± 0.1	92.1 ± 0.2	96.7 ± 0.6	76.4 ± 0.9	88.2 ± 0.0
SSF + LOFF-TA	91.1 ± 0.7	93.1 ± 0.0	97.2 ± 0.3	81.6 ± 1.5	85.6 ± 0.1
AdaptFormer [6]	89.6 ± 0.6	94.3 ± 0.1	91.8 ± 0.8	82.6 ± 1.0	87.1 ± 0.3
AdaptFormer + LOFF-TA	90.0 ± 0.3	94.3 ± 0.2	93.2 ± 0.5	83.5 ± 0.3	85.3 ± 0.2

come with trade-offs. In most cases, LOFF-TA will perform slightly worse than directly fine-tuning the foundation model, although we did observe some cases where LOFF-TA was superior. Also, the computational cost savings at training time do not translate to inference.

An important benefit of LOFF-TA is that it affords flexibility, allowing practitioners to choose the most suitable foundation model for their task from a range of sizes and pretraining methods (refer to Table 2), including the capacity to handle high-resolution images (see Table 1). The caching and pooling of features makes these different choices come with equivalent computational costs during training, determined by the compact classifier.

Societal impact: As foundation models grow, their computational demands are expected to increase, potentially creating a digital divide where advanced models are accessible only to well-resourced organizations, excluding smaller entities and individual researchers. This trend underscores the importance of developing efficient methods to utilize these models, ensuring broad access and fostering progress in various application areas. There is significant potential to impact fields like medical imaging and remote sensing, which can greatly benefit developing regions and marginalized groups. But these applications often require high-resolution image processing, which is resource-intensive. LOFF-TA facilitates wider, fairer access to sophisticated foundation models in resource-limited settings.

Limitations: LOFF-TA is a training strategy meant for low resource settings. Given enough computational resources, full fine-tuning will generally outperform LOFF-TA or adaptation methods like SSF [31]. Another limitation of LOFF-TA is that at inference time it is less efficient than standalone foundation models. However, we note that due to their latency, foundation models are costly to be deployed at scale without distillation [16] (or some other measure) to limit cost which would also alleviate the limitation for LOFF-TA.

7 Conclusion

Foundation models have significantly impacted the community and will likely become more crucial as they grow in size and capability. However, adapting these models for specific tasks is increasingly challenging due to their high resource demands. Revisiting the classical machine learning paradigm of *perceive, then reason* is pragmatic in this context. Using foundation models as powerful feature extractors allows us to benefit from their rich representations, while at the same time mitigating computational costs by training a lightweight classifier on their cached features. LOFF-TA achieves a low memory footprint and high throughput during training, regardless of the chosen foundation model or input image size. It is particularly beneficial for high-resolution image processing when pooling is applied, allowing LOFF-TA to outperform linear classifiers using frozen foundation models and compete with or surpasses fine-tuned foundation models.

Acknowledgements. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) and the Development and Promotion of Science and Technology Talents Project. We acknowledge the Berzelius computational resources provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

 https://doi.org/10.52202/079017-3817

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[2] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. Springer, 2006.

[3] Rishi Bommasani et al. "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258* (2021).

[4] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[5] Lawrence Cayton et al. *Algorithms for manifold learning*. eScholarship, University of California, 2008.

[6] Shoufa Chen et al. "Adaptformer: Adapting vision transformers for scalable visual recognition". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 16664–16678.

[7] Tianqi Chen et al. "Training deep nets with sublinear memory cost". In: *arXiv preprint arXiv:1604.06174* (2016).

[8] Noel CF Codella et al. "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)". In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE. 2018, pp. 168–172.

[9] Marc Combalia et al. "BCN20000: Dermoscopic lesions in the wild". In: *arXiv preprint arXiv:1908.02288* (2019).

[10] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[11] Terrance DeVries and Graham W Taylor. "Dataset augmentation in feature space". In: *arXiv preprint arXiv:1702.05538* (2017).

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". In: *2004 conference on computer vision and pattern recognition workshop*. IEEE. 2004, pp. 178–178.

[13] Xu Han et al. "Pre-trained models: Past, present and future". In: *AI Open* 2 (2021), pp. 225–250.

[14] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[15] John L Hennessy and David A Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[17] Neil Houlsby et al. "Parameter-efficient transfer learning for NLP". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2790–2799.

[18] Edward J Hu et al. "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685* (2021).

[19] Gabriel Ilharco et al. *OpenCLIP*. If you use this software, please cite it as below. July 2021. DOI: 10.5281/zenodo.5143773. URL: https://doi.org/10.5281/zenodo.5143773.

[20] Menglin Jia et al. "Visual prompt tuning". In: *European Conference on Computer Vision*. Springer. 2022, pp. 709–727.

[21] Sohier Dane Karthik Maggie. *APTOS 2019 Blindness Detection*. 2019. URL: https://kaggle.com/competitions/aptos2019-blindness-detection.

[22] Aditya Khosla et al. "Novel dataset for fine-grained image categorization: Stanford dogs". In: *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*. Vol. 2. Citeseer. 2011.

[23] Alexander Kirillov et al. "Segment anything". In: *arXiv preprint arXiv:2304.02643* (2023).

[24] Simon Kornblith, Jonathon Shlens, and Quoc V Le. "Do better imagenet models transfer better?" In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2661–2671.

[25] Simon Kornblith et al. "Similarity of neural network representations revisited". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3519–3529.

[26] Jonathan Krause et al. "3d object representations for fine-grained categorization". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2013, pp. 554–561.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90.

[28] Manoj Kumar, Mostafa Dehghani, and Neil Houlsby. "Dual PatchNorm". In: *arXiv preprint arXiv:2302.01327* (2023).

[29] Rebecca Sawyer Lee et al. "A curated mammography data set for use in computer-aided detection and diagnosis research". In: *Scientific data* 4.1 (2017), pp. 1–9.

[30] Junnan Li et al. *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation*. 2022. arXiv: 2201.12086 [cs.CV].

[31] Dongze Lian et al. "Scaling & shifting your features: A new baseline for efficient model tuning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 109–123.

[32] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[33] Samuel G Müller and Frank Hutter. "Trivialaugment: Tuning-free yet state-of-the-art data augmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 774–782.

[34] Maria-Elena Nilsback and Andrew Zisserman. "Automated flower classification over a large number of classes". In: *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE. 2008, pp. 722–729.

[35] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[36] Omkar M Parkhi et al. "Cats and dogs". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3498–3505.

[37] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[38] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *arXiv preprint arXiv:2302.13971* (2023). URL: https://arxiv.org/abs/2302.13971.

[39] Hugo Touvron et al. "Training data-efficient image transformers & distillation through attention. arXiv 2020". In: *arXiv preprint arXiv:2012.12877* (2020).

[40] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific data* 5.1 (2018), pp. 1–9.

[41] Grant Van Horn et al. "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 595–604.

[42] Vikas Verma et al. "Manifold mixup: Better representations by interpolating hidden states". In: *International conference on machine learning*. PMLR. 2019, pp. 6438–6447.

[43] Gui-Song Xia et al. "AID: A benchmark data set for performance evaluation of aerial scene classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7 (2017), pp. 3965–3981.

[44] Jianxiong Xiao et al. "Sun database: Large-scale scene recognition from abbey to zoo". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 3485–3492.

[45] Suorong Yang et al. "Image data augmentation for deep learning: A survey". In: *arXiv preprint arXiv:2204.08610* (2022).

 https://doi.org/10.52202/079017-3817

[46] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. “Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models”. In: *arXiv preprint arXiv:2106.10199* (2021).

[47] Xueyan Zou et al. “Segment everything everywhere all at once”. In: *arXiv preprint arXiv:2304.06718* (2023).

Appendix

A Representation similarities

To examine the similarities between classifiers trained on images and trained on tensors using LOFF, we employed Centered Kernel Alignment (CKA) [25], a technique commonly used to analyze representational similarity in neural networks. Figure 3 presents the CKA analysis results for classifiers trained on Oxford-III Pet. In the top panels, we observe that the LOFF classifier's low-to-mid layers exhibit similarity to the higher layers of the image-trained classifier, indicating that LOFF learns features similar to the image-trained classifier's high-level features. This suggests that LOFF's offline features are sufficiently informative for learning high-level features earlier in the network, with subsequent layers adapting to novel features. The middle panels demonstrate that the cascaded frozen foundation model classifier displays similar behavior to the LOFF classifier. The right panel reveals high similarities between corresponding layers of the frozen foundation and LOFF classifiers, particularly in the earlier layers, indicating shared learned features. The bottom panel illustrates the internal representational similarity of each classifier with itself before and after fine-tuning. The left panel shows strong similarity between the image-trained classifier's layers, indicating the retention of pretrained features during fine-tuning. Conversely, the middle and right panels indicate that the frozen foundation and LOFF classifiers retain similarity in the low-to-mid layers while learning new features in the higher layers.

Figure 3: *CKA similarities between different models. Left*: Representation similarity of different classifiers after fine-tuning on Oxford-III-Pet. *Right*: Representation similarity of the internal layers of each classifier with itself before and after fine-tuning.

B Unfreezing the Cascaded Models

We evaluate the model's performance in the cascaded setting (*foundation model + DeiT-S classifier*) when the whole model is fine-tuned. We present our findings in Table 5, comparing the use of DINOv2 and OpenCLIP as foundation models. Once again, we find that DINOv2 outperforms OpenCLIP, in most cases. When DINOv2 is unfrozen, it either surpasses or matches the performance of its frozen alternative (see Table 2), though the margin of improvement is rather small. OpenCLIP shows a similar trend, but its performance seems more dependent on the dataset it is evaluated on. Overall, unfrozen foundation models appear to outperform their frozen alternatives. However, their computational cost makes them prohibitive for larger models or image sizes.

Table 5: *Fine-tuning a foundation model + DeiT-S classifier*. We append a DeiT-S classifier after a DINOv2 [35] or a OpenCLIP [19] foundation model and the whole cascaded model.

Foundation	**Oxford-III Pet** n = 7,349	**Flowers102** n = 8,189	**Caltech-101** n = 8,677	**StanfordCars** n = 16,185	**StanfordDogs** n = 20,580	**SUN397** n = 39,700	**NABirds** n = 48,562
DINOv2 ViT-B	95.2 ± 0.2	99.3 ± 0.1	97.4 ± 0.2	93.9 ± 0.4	87.0 ± 0.3	76.2 ± 0.7	86.3 ± 0.2
OpenCLIP ViT-B	91.8 ± 0.5	95.4 ± 1.2	96.8 ± 0.2	94.5 ± 0.2	81.3 ± 0.3	74.8 ± 0.2	77.8 ± 0.1

https://doi.org/10.52202/079017-3817

C Tensor augmentations

In our experiments, we consistently observe a performance improvement when applying tensor augmentations. However, the need for task-specific and domain-appropriate augmentation strategies should be emphasized. For example, in scene classification using the SUN397 dataset, applying vertical flips to images when training DeiT-S (pretrained on ImageNet) led to decreased accuracy (−2%), likely due to the unrealistic expectation of upside-down building facades during testing. One might expect a similar effect with tensor augmentations, since the feature space preserves the spatial orientations of objects (see Figure 4). Interestingly we see a much smaller performance drop when we apply vertical flips as tensor augmentations on this dataset (−0.2%). This suggests LOFF-TA's tensor augmentations exhibit greater resilience to *improper* augmentations compared to the image domain. This observation merits further exploration to understand its underlying mechanisms and consequences.

Intriguingly, LOFF-TA demonstrates benefits from spatial tensor augmentations despite their potential conflict with the spatial information present in feature tokens. Feature tokens in foundation models carry positional data, informed by positional embeddings and attention mechanisms. Spatial augmentations, such as horizontal flips, disrupt this positional context, yet our experiments, especially with Trivial Augment, show a notable performance enhancement (see Table 3). This paradox suggests that these disruptions may actually bolster the classifier's ability to learn robust features, thereby improving classification outcomes. The precise dynamics of this phenomenon remain unclear, presenting an exciting avenue for future research, particularly in the realm of auto augment strategies for foundation features.

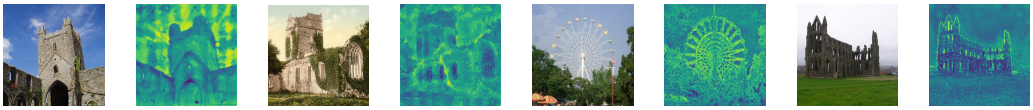


Figure 4: *Robustness and spatial consistency of features.* Images along with a random channel of the corresponding foundation features reveal the spatial consistency between objects in the image and feature spaces. This consistency allows insights from the image space to guide tensor augmentation choice, *e.g.* if vertical flips are harmful for a building facade dataset in image space, they are likely to be harmful in feature space. However, we observe that training with LOFF-TA is more robust against 'incorrect' augmentation choices compared to standard classifier training on images.

D Pooling ablation

Table 6: *Pooling enables larger resolution*. We compare different approaches to pooling used for LOFF and LOFF-TA: *no pooling*, *max pooling*, and *average pooling*. Results are reported for foundation features extracted by DINOv2 [35] ViT-B and ViT-G models from standard 256×256 without pooling versus larger 512×512 images with pooling (note that the compute costs are equivalent).

	Method	Pooling	Size	**APTOS**, $\kappa\uparrow$ n = 3,662	**AID**, Acc. ↑ n = 10,000	**DDSM**, AUC ↑ n = 10,239	**ISIC**, Rec. ↑ n = 25,333	**NABirds**, Acc. ↑ n = 48,562
ViT-B	LOFF	✗	256	89.6 ± 0.2	91.9 ± 0.3	94.2 ± 1.2	70.8 ± 2.1	83.0 ± 0.1
	LOFF-TA			90.4 ± 0.6	92.3 ± 0.7	94.4 ± 0.1	72.8 ± 1.7	83.5 ± 0.3
	LOFF	Average	512	90.3 ± 0.2	92.7 ± 0.7	95.7 ± 0.3	73.8 ± 0.1	86.3 ± 0.3
	LOFF-TA			**90.7 ± 0.8**	**93.7 ± 0.5**	**96.1 ± 0.1**	**77.9 ± 1.9**	86.7 ± 0.3
	LOFF	Max	512	89.4 ± 1.5.	93.2 ± 0.6	95.3 ± 0.5	74.3 ± 1.5	86.2 ± 0.3
	LOFF-TA			90.5 ± 1.0	**93.7 ± 0.3**	95.5 ± 0.1	77.4 ± 0.0	**86.8 ± 0.4**
ViT-G	LOFF	✗	256	88.6 ± 1.5	93.3 ± 0.5	94.8 ± 1.6	73.1 ± 0.5	87.4 ± 0.2
	LOFF-TA			89.9 ± 0.4	94.0 ± 0.2	95.3 ± 0.1	76.0 ± 0.7	88.5 ± 0.2
	LOFF	Average	512	89.0 ± 0.6	94.0 ± 0.3	96.0 ± 0.5	77.5 ± 0.7	89.0 ± 0.2
	LOFF-TA			90.1 ± 0.7	94.3 ± 0.2	96.1 ± 0.6	79.4 ± 2.8	90.0 ± 0.2
	LOFF	Max	512	90.3 ± 0.6	94.1 ± 0.2	95.4 ± 0.4	74.0 ± 1.6	88.8 ± 0.1
	LOFF-TA			**91.8 ± 0.3**	**94.6 ± 0.2**	**96.3 ± 0.6**	**79.9 ± 0.2**	**90.1 ± 0.2**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We provided a list of contributions in Section 1 Introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We have listed the limitations of our proposed method and under which assumptions it works as described.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

https://doi.org/10.52202/079017-3817

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

Answer: [NA]

Justification: This is an empirical study and we do not report any theoretical results.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

Answer: [Yes]

Justification: We provided the architecture details, hyperparameter selection details i.e. default hyperparameters except the learning rate and schedule for which we provided the selection strategy. We also listed the set of augmentations we utilized. We will provide the full training logs along with the configuration files (stored online at wandb.com) after the review process.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

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

   Answer: [No]

   Justification: The datasets are all public and sources are referenced in the manuscript. We will provide the code after publication.

   Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: We have used the default settings defined in the original papers. We referenced which dataset splitting guidelines we used in the absence of official splits. We listed the optimizer we used and described our learning rate selection strategy.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

https://doi.org/10.52202/079017-3817

Answer: [Yes]

Justification: We repeated our experiments with different random seeds and reported the standard deviation of our results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We provided the GPU memory and throughput for both training and inference for all experiments, along with the hardware specifications.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: No human subjects or crowdsourcing were used during the study. We adhered to the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: We provided a societal impact statement addressing potential positive and negative impacts of the study.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our study is about efficient training strategies and it does not pose a high risk for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

https://doi.org/10.52202/079017-3817

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

Answer: [Yes]

Justification: We cited all datasets and pretrained weights we used in this study. We referenced both the original versions of these assets and modifications to them.

Guidelines:

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

13. **New Assets**

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

Answer: [NA]

Justification: We will release the code after publication. No other assets will be released.

Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

Answer: [NA]

Justification: No human subjects or crowdsourcing were used during the study.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: No human subjects or crowdsourcing were used during the study.

    Guidelines:

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

https://doi.org/10.52202/079017-3817