Frozen Feature Augmentation for Few-Shot Image Classification

Andreas Bär[1 2 *] Neil Houlsby[1] Mostafa Dehghani[1] Manoj Kumar[1 †]
[1]Google DeepMind [2]Technische Universität Braunschweig
{andreasbaer, neilhoulsby, dehghani, mechcoder}@google andreas.baer@tu-bs.de

Project website: https://frozen-feature-augmentation.github.io

Abstract

Training a linear classifier or lightweight model on top of pretrained vision model outputs, so-called 'frozen features', leads to impressive performance on a number of downstream few-shot tasks. Currently, frozen features are not modified during training. On the other hand, when networks are trained directly on images, data augmentation is a standard recipe that improves performance with no substantial overhead. In this paper, we conduct an extensive pilot study on few-shot image classification that explores applying data augmentations in the frozen feature space, dubbed 'frozen feature augmentation (FroFA)', covering twenty augmentations in total. Our study demonstrates that adopting a deceptively simple pointwise FroFA, such as brightness, can improve few-shot performance consistently across three network architectures, three large pretraining datasets, and eight transfer datasets.

1. Introduction

Vision transformers (ViTs) [19] achieve remarkable performance on ImageNet-sized [43, 69] and smaller [21, 38, 41] datasets. In this setup, *data augmentation*, *i.e.*, a predefined set of stochastic input transformations, is a crucial ingredient. Examples for *image augmentations* are random cropping or pixel-wise modifications that change brightness or contrast. These are complemented by more advanced strategies [13, 46, 75], such as AutoAugment [12].

A more prevalent trend is to first pretrain vision models on large-scale datasets and then adapt them downstream [6, 8, 49, 73]. Notable, even training a simple linear classifier or lightweight model on top of ViT outputs, also known as *frozen features*, can yield remarkable performance across a number of diverse downstream few-shot tasks [16, 25, 52]. Given the success of *image augmentations* and *frozen features*, we ask: *Can we effectively combine image augmentations and frozen features to train a lightweight model?*

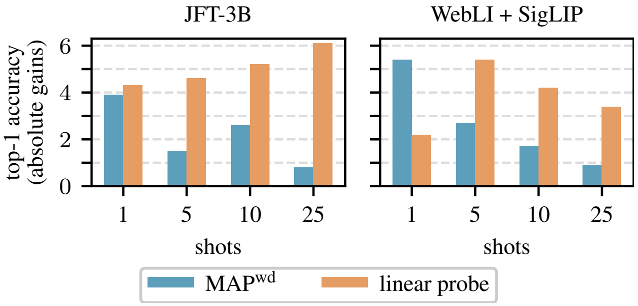


Figure 1. Average top-1 accuracy gains across seven few-shot test sets (CIFAR100 [1], SUN397 [71], ...) on various few-shot settings. We train on frozen features from an L/16 ViT [19] with JFT-3B pretraining [73] or WebLI sigmoid language-image pretraining (SigLIP) [6, 74]. Our proposed frozen feature augmentation (FroFA) method gives consistent gains over a weight decay-regularized multi-head attention pooling [37] (MAP$^{\text{wd}}$) and an L2-regularized linear probe baseline, both without FroFA.

In this paper, we revisit standard image augmentation techniques and apply them on top of frozen features in a data-constrained, few-shot setting. We dub this type of augmentation *frozen feature augmentation (FroFA)*. Inspired directly by image augmentations, we first stochastically transform frozen features and then train a lightweight model on top. Our only modification before applying image augmentations on top of frozen features is a point-wise scaling such that each feature value lies in $[0, 1]$ or $[0, 255]$.

We investigate eight (few-shotted) image classification datasets using ViTs pretrained on JFT-3B [73], ImageNet-21k [17], or WebLI [6]. After extracting features from each few-shot dataset we apply twenty different frozen feature augmentations and train a lightweight multi-head attention pooling (MAP) [37] on top. Our major insights are:

1. Geometric augmentations that modify the shape and structure of two-dimensional frozen features always lead to worse performance on ILSVRC-2012 [57]. On the other hand, simple stylistic (point-wise) augmentations, such as brightness, contrast, and posterize, give steady improvements on 1-, 5-, and 10-shot settings.

[*] Work done as Research Intern at Google DeepMind. [†] Project lead.

2. Additional per-channel stochasticity by sampling independent values for each frozen feature channel works surprisingly well: On ILSVRC-2012 5-shot we improve over an MAP baseline by 1.6% absolute and exceed a well-tuned linear probe baseline by 0.8% absolute.
3. While FroFA provides modest but significant gains on ILSVRC-2012, it excels on seven smaller few-shot datasets. In particular, FroFA outperforms the mean 10-shot accuracy of an MAP baseline by 2.6% and the linear probe baseline by 5.2% absolute (*cf*. Fig. 1, left).
4. Results on the same seven few-shot datasets using a WebLI sigmoid language-image pretrained model [74] further emphasize the transfer capabilities of FroFA. We observe absolute gains ranging from 5.4% on 1-shot to 0.9% on 25-shot compared to an MAP baseline while outperforming a linear probe baseline by over 2% on 1-shot and at least 3% on 5- to 25-shot. (*cf*. Fig. 1, right).

2. Related Works

Few-shot transfer learning: State-of-the-art vision models [6, 16, 19, 32, 55, 73] are typically pretrained on large-scale datasets, *e.g*., ImageNet-21k [17] or JFT [27, 73], before transferred to other smaller-scale ones, *e.g*., CIFAR10 [1], SUN397 [70, 71], or ILSVRC-2012 [57]. Depending on the model size, efficient transfer learning becomes a challenge. Many methods have been proposed for large language models (LLMs), *e.g*., adapters [28], low-rank adaptation [29], or prompt tuning [39], of which some have been successfully adapted to computer vision [5, 22, 30, 76]. CLIP-Adapter [22] builds on contrastive language-image pretraining [52] and combines it with adapters [28]. A follow-up work [76] proposes TiP-Adapter which uses a query-key cache model [24, 51] instead of a gradient descent approach. Inspired by the success of prompt tuning in LLMs [39], Jia *et al*. propose visual prompt tuning at the model input [30]. On the other hand, AdaptFormer [5] uses additional intermediate trainable layers to finetune a frozen vision transformer [19].

In contrast, we do not introduce additional prompts [30] or intermediate parameters [5, 22] that require backpropagating through the network. Instead, we train a small network on top of frozen features from a ViT. This aligns with linear probing [52] which is typically used to transfer vision models to other tasks [16, 25, 73] — our objective.

Further, we focus on few-shot transfer learning [36, 68] in contrast to meta- or metric-based few-shot learning [2, 9, 48, 50, 54, 56, 59]. Kolesnikov *et al*. [32] and Dehghani *et al*. [16] reveal that training a lightweight model on frozen features from a large-scale pretrained backbone yields high performance across various downstream (few-shot) tasks. Similarly, Vasconcelos *et al*. [65] show that training on frozen features gives strong performance on object detection and segmentation. In addition, transfer learning has also shown to be competitive or slightly better than meta-learning approaches [20, 63]. Building on these works, we propose frozen feature augmentation to improve few-shot transfer learning for image classification tasks.

Data augmentation: One go-to method to improve performance while training in a low-data regime is data augmentation [60]. Some prominent candidates in computer vision are AutoAugment [12], AugMix [26], RandAugment [12], and TrivialAugment [46]. These methods typically combine low-level image augmentations together to augment the input. Works on augmentations in feature space exist [18, 35, 40, 44, 67], but lack a large-scale empirical study on *frozen features* of single-modal vision models.

To this end, we investigate frozen feature augmentation by reformulating twenty image augmentations, including a subset used in AutoAugment [12], inception crop [62], mixup [67, 75], and patch dropout [42].

3. Framework Overview

We introduce our notations in Sec. 3.1 followed by our caching and training pipeline in Sec. 3.2 and a description of frozen feature augmentations (FroFAs) in Sec. 3.3.

3.1. Notation

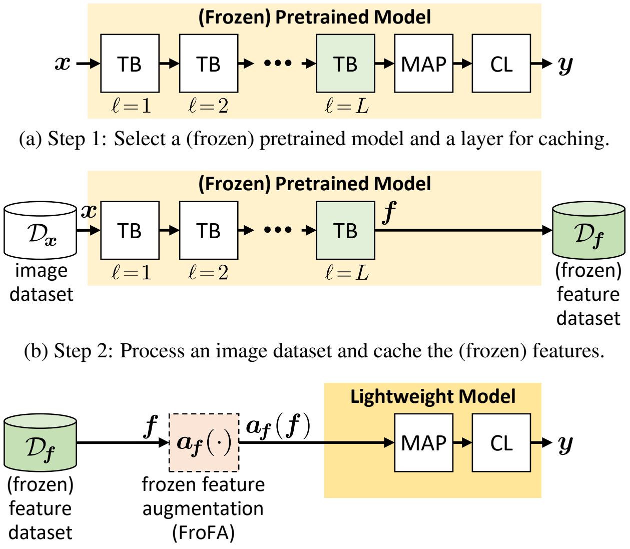
Let $\boldsymbol{x} \in \mathbb{I}^{H \times W \times 3}$ be an RGB image of height H, width W, and $\mathbb{I} = [0, 1]$. A classification model processes $\boldsymbol{x}$ and outputs class scores $\boldsymbol{y} \in [0, 1]^S$ for each class in a predefined set of classes $\mathcal{S}$, with $S = |\mathcal{S}|$. Let L and D be the number of intermediate layers and the number of features of a multi-layer classification model, respectively. We describe the intermediate feature representations of $\boldsymbol{x}$ as $\boldsymbol{f} = \boldsymbol{f}^{(\ell)} = (f_d^{(\ell)}) \in \mathbb{R}^D$, with layer index $\ell \in \{1, ..., L\}$ and feature index $d \in \{1, ..., D\}$. In vision transformers [19], $\boldsymbol{f} = \boldsymbol{f}^{(\ell)} = (f_{n,c}^{(\ell)}) \in \mathbb{R}^{N \times C}$ is typically two-dimensional, where N and C are the number of patches and number of per-patch channels, respectively. Finally, we introduce the patch index $n \in \{1, ..., N\}$ and the per-patch channel index $c \in \{1, ..., C\}$.

3.2. Training on Cached Frozen Features

We investigate pretrained vision transformers with L transformer blocks (TBs) followed by a multi-head attention pooling (MAP) [37] and a classification layer (CL). Fig. 2a presents a simplified illustration. For simplicity, we neglect all operations before the first transformer block (e.g., patchifying, positional embedding, etc.).

To cache intermediate features, we process each image $\boldsymbol{x}$ from an image dataset $\mathcal{D}_{\boldsymbol{x}}$ through the network up until transformer block L. Next, we store the resulting features $\boldsymbol{f}$. After processing the entire image dataset $\mathcal{D}_{\boldsymbol{x}}$ we obtain a (frozen) feature dataset $\mathcal{D}_{\boldsymbol{f}}$, with $\boldsymbol{f} \in \mathcal{D}_{\boldsymbol{f}}$ (Fig. 2b).

Lastly, we train a lightweight model using the cached (frozen) features. Fig. 2c shows an example where a single



(a) Step 1: Select a (frozen) pretrained model and a layer for caching.

(b) Step 2: Process an image dataset and cache the (frozen) features.

(c) Step 3: Train on (augmented) frozen features.

Figure 2. **Pipeline for caching and training on (frozen) features**. (2a): Given a (frozen) pretrained vision transformer, with L transformer blocks (TBs), a multi-head attention pooling (MAP) layer, and a classification layer (CL), we select its L-th transformer block for caching. (2b): Next, we feed images $\boldsymbol{x} \in \mathcal{D}_{\boldsymbol{x}}$ to cache (frozen) features $\boldsymbol{f} \in \mathcal{D}_{\boldsymbol{f}}$. (2c): Finally, we use $\mathcal{D}_{\boldsymbol{f}}$ to train a lightweight model on top. We investigate frozen feature augmentation (FroFA) $\boldsymbol{a_f} \in \mathcal{A}_{\boldsymbol{f}}$ in this scenario.

MAP layer followed by a classification layer is trained using the feature dataset $\mathcal{D}_{\boldsymbol{f}}$. Since our focus is fast training, we defer a detailed analysis on larger models to future work.

3.3. Frozen Feature Augmentation (FroFA)

Data augmentation is a common tool to improve generalization. However, it is typically applied on the input, or in our case: images. How can we map such image augmentations to intermediate transformer feature representations?

Recall that the feature representation $\boldsymbol{f} = (f_{n,c}) \in \mathbb{R}^{N\times C}$ (layer index ℓ omitted) is two-dimensional. We first reshape it to a three-dimensional representation, *i.e.*,

$$\boldsymbol{f}^* = (f^*_{n_1,n_2,c}) \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}\times C}. \tag{1}$$

We further define

$$\boldsymbol{f}^*_c = \boldsymbol{f}^*_{:,:,c} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}\times 1} \tag{2}$$

as a reshaped two-dimensional representation of the c-th channel. Since images and features differ in two fundamental aspects, *i.e.*, channel dimensionality and value range, we address this next.

Channel dimensionality: RGB images have just three channels while features can possess an arbitrary number of channels. To address this, we simply ignore image augmentations that rely on having three color channels, such as color jitter, and include only augmentations which can have an arbitrary number of channels instead, denoted as $C_{\boldsymbol{a}}$. This already covers a majority of commonly applied image augmentations.

Value range: RGB values lie within a specific range $\mathbb{I}$, *e.g.*, $\mathbb{I} = [0,1]$ or $\mathbb{I} = \{0, ..., 255\} \subset \mathbb{N}_0$, while in theory features have no such constraints. Assuming $H = \sqrt{N}$ and $W = \sqrt{N}$, we define an image augmentation as

$$\boldsymbol{a_x} : \mathbb{I}^{\sqrt{N}\times\sqrt{N}\times C_{\boldsymbol{a}}} \rightarrow \mathbb{I}^{\sqrt{N}\times\sqrt{N}\times C_{\boldsymbol{a}}}, \boldsymbol{a_x} \in \mathcal{A}_{\boldsymbol{x}}, \tag{3}$$

where $\mathcal{A}_{\boldsymbol{x}}$ is the set of image augmentations. To also address the value range mismatch, we introduce a deterministic feature-to-image mapping

$$\boldsymbol{t}_{\boldsymbol{f}\rightarrow\boldsymbol{x}} : \mathbb{R}^{\sqrt{N}\times\sqrt{N}\times C_{\boldsymbol{t}}} \rightarrow \mathbb{I}^{\sqrt{N}\times\sqrt{N}\times C_{\boldsymbol{t}}} \tag{4}$$

that maps each element of $\boldsymbol{f}^*$ (1) from $\mathbb{R}$ to $\mathbb{I}$, with $C_{\boldsymbol{t}}$ as the number of channels of $\boldsymbol{f}^*$. We use

$$\boldsymbol{x_f} = \boldsymbol{t}_{\boldsymbol{f}\rightarrow\boldsymbol{x}}(\boldsymbol{f}^*) = \frac{\boldsymbol{f}^* - f_{\min}}{f_{\max} - f_{\min}}, \tag{5}$$

where $f_{\min}$ and $f_{\max}$ are the minimum and maximum value of $\boldsymbol{f}^*$, respectively, with elements of $\boldsymbol{x_f}$ now in $\mathbb{I} = [0,1]$. We further define an image-to-feature mapping

$$\boldsymbol{t}_{\boldsymbol{f}\leftarrow\boldsymbol{x}} : \mathbb{I}^{\sqrt{N}\times\sqrt{N}\times C_{\boldsymbol{t}}} \rightarrow \mathbb{R}^{\sqrt{N}\times\sqrt{N}\times C_{\boldsymbol{t}}} \tag{6}$$

that maps $\boldsymbol{x_f}$ back to the original feature value range. In this case, we invert (4) and use

$$\boldsymbol{f}^* = \boldsymbol{t}_{\boldsymbol{f}\leftarrow\boldsymbol{x}}(\boldsymbol{x_f}) = \boldsymbol{x_f} \cdot (f_{\max} - f_{\min}) + f_{\min}. \tag{7}$$

Combining (3), (4), and (6), we obtain a generic (frozen) feature augmentation as a function composition

$$\boldsymbol{a_f} = \boldsymbol{t}_{\boldsymbol{f}\leftarrow\boldsymbol{x}} \circ \boldsymbol{a_x} \circ \boldsymbol{t}_{\boldsymbol{f}\rightarrow\boldsymbol{x}}. \tag{8}$$

We now define three variations of $\boldsymbol{a_f}$:

1. **(Default) FroFA**: We apply $\boldsymbol{a_f}$ (8) once across the entire feature. We set $C_{\boldsymbol{a}} = C_{\boldsymbol{t}} = C$ and compute $f_{\min}$ and $f_{\max}$ in (5), (7) across all elements of $\boldsymbol{f}^*$. Further, as normally done in pixel space, $\boldsymbol{a_x}$ (3) samples a random augmentation value and changes all elements of $\boldsymbol{x_f}$ using the same value. For example, employing random contrast in a FroFA fashion scales each element of $\boldsymbol{x_f}$ by the *exact same randomly sampled factor*.
2. **Channel FroFA (cFroFA)**: For each channel in the mapped features $\boldsymbol{x_f}$ (5), $\boldsymbol{a_x}$ (3) samples a random augmentation value *per channel* and applies that value to all elements in that channel ($C_{\boldsymbol{a}} = 1$ while $C_{\boldsymbol{t}} = C$). By using cFroFA for our random contrast example, we obtain C *independently sampled scaling factors, one for each channel*.

3. **Channel^2 FroFA (c^{2}FroFA)**: In addition to applying augmentations per channel ($C_{\boldsymbol{a}} = 1$) as done in cFroFA, $\boldsymbol{t}_{\boldsymbol{f}\to\boldsymbol{x}}$ (4) and $\boldsymbol{t}_{\boldsymbol{x}\leftarrow\boldsymbol{f}}$ (6) also operate per channel ($C_{\boldsymbol{t}} = 1$), *i.e*., on $\boldsymbol{f}^*_c$ (2). In this case, $f_{\min}$ and $f_{\max}$ are the per-channel maximum and minimum, respectively. In contrast, FroFA and cFroFA use the maximum and minimum across the entire feature. We denote this variant as c^{2}FroFA since both the mappings (4), (6) and the augmentation (3) are applied on a per-channel basis. Although not adding additional stochasticity, we found that for random brightness this variant gives more stable results across a range of augmentation hyper parameters.

While an element-wise FroFA might seem like a natural next step, our initial experiments lead to significantly worse results. We hypothesize that per-element augmentations might lead to substantial changes in the feature appearance.

4. Experimental Setup

In this section, we describe our experimental setup.

4.1. Network Architectures

We employ pretrained Ti/16 [64], B/16 [19], and L/16 [19] vision transformers. Further, we follow Zhai *et al*. [73] and use a lightweight multi-head attention pooling (MAP) [37] before the final classification layer for training on top of frozen features (*cf*. Sec. 3.3).

4.2. Datasets

Pretraining: We consider three pretraining datasets, *i.e*., JFT-3B [73], ImageNet-21k [17], and WebLI [6]. First introduced by Hinton *et al*. [27], JFT is now a widely used proprietary, large-scale dataset [6, 10, 14, 19, 32, 33, 61]. We use JFT-3B [73] which consists of nearly 3 billion multi-labeled images following a class-hierarchy of 29,593 labels. The images are annotated with noisy labels by using a semi-automated pipeline. We follow common practice [16, 73] and ignore the hierarchical aspect of the labels.

ImageNet-21k contains 14,197,122 (multi)-labeled images with 21,841 distinct labels. We equally split the first 51,200 images into a validation and test set and use the remaining 14,145,922 images for training.

Lastly, WebLI is a web-scale multilingual image-text dataset for vision-language training. It encompasses text in 109 languages with 10 billion images and roughly 31 billion image-text pairs.

Few-shot transfer learning: We investigate eight datasets for few-shot transfer learning, *i.e*., ILSVRC-2012 [57], CIFAR10 [1], CIFAR100 [1], DMLab [3, 72], DTD [11], Resisc45 [7], SUN397 [70, 71], and SVHN [47].

ILSVRC-2012, alias 'ImageNet-1k' or just 'ImageNet', stems from ImageNet-21k and contains 1,281,167 training images of 1,000 classes. We randomly sample 1-, 5-, 10-, and 25-shot versions from the first 10% of the training set. We further create additional disjoint sets by using the next four 10% fractions of the training set. In addition, we follow previous works [4] and create a 'minival' set using the last 1% (12,811 images) of the ILSVRC-2012 training set. The 'minival' set is used for hyperparameter tuning and design decisions while the official ILSVRC-2012 validation set is used as a test set. In summary, our setup consists of 1,000, 5,000, 10,000, or 25,000 training images, 12,811 validation images ('minival'), and 50,000 test images ('validation').

For the other seven datasets, we also select a training, validation, and test split and create few-shot versions of the respective training set. Similar to ILSVRC-2012, we use the validation sets to tune hyperparameters and report final results on the test sets. A short description of each dataset and more details can be found in the Supplementary, Sec. S2.1.

4.3. Data Augmentation

We reuse the set of augmentations first defined in AutoAugment [12] and adopted in later works [13, 46]. In addition, we consider a few other image augmentations [42, 62, 75]. We select *five geometric* augmentations, *i.e*., rotate, shear-x, shear-y, translate-x, and translate-y; *four crop & drop* augmentations, *i.e*., crop, resized crop, inception crop [62], and patch dropout [42]; *seven stylistic* augmentations, *i.e*., brightness, contrast, equalize, invert, posterize, sharpness, and solarize; and *two other* augmentations, *i.e*., JPEG and mixup [75]. In Supplementary, Sec. S3.7, we also test two additional augmentations.

In total, we end up with *twenty distinct augmentations*. Note that all data augmentations incorporate random operations, *e.g*., a random shift in x- and y-direction (translate-x and translate-y, respectively), a randomly selected set of patches (patch dropout), a random additive value to each feature (brightness), or a random mix of two features and their respective classes (mixup). Please refer to the Supplementary, Sec. S2.2, for more details. We focus on the following set of experiments:

1. We investigate FroFA for all eighteen augmentations (and two additional ones in Supplementary, Sec. S3.7).
2. For our top-performing FroFAs, namely, brightness, contrast, and posterize, we incorporate additional stochasticity using cFroFA and c^{2}FroFA (*cf*. Sec. 3.3).
3. We investigate a sequential protocol where two of the best three (c/c^2)FroFA are arranged sequentially, namely, brightness c^{2}FroFA, contrast FroFA, and posterize cFroFA. We test all six possible combinations.
4. Finally, we also apply variations of RandAugment [13] and TrivialAugment [46] directly on top of cached frozen features. More details and results can be found in the Supplementary, Secs. S2.2 and S3.2, respectively.

In Supplementary, Sec. S3.6, we complement our study by comparing our best FroFA to input data augmentations.

4.4. Training & Evaluation Details

We describe some base settings for pretraining, few-shot learning, and evaluation. Please refer to Supplementary, Sec. S2.3 for more training details.

Pretraining: Models are pretrained on `Big Vision`[1]. We re-use the Ti/16, B/16, and L/16 ViTs pretrained on JFT-3B from Zhai *et al*. [73]. In addition, we pretrain Ti/16, B/16, and L/16 ViTs on ImageNet-21k following the settings described by Steiner *et al*. [60]. We further use a pretrained L/16 ViT image encoder stemming from a vision-language model from Zhai *et al*. [74] which follows their sigmoid language-image pretraining (SigLIP) on WebLI.

Few-shot transfer learning: Models are transferred using `Scenic`[2] [15]. We train the lightweight MAP-based head by sweeping across five batch sizes (32, 64, 128, 256, and 512), four learning rates (0.01, 0.03, 0.06, and 0.1), and five training step sizes (1,000; 2000; 4,000; 8,000; and 16,000). In total, we obtain 100 configurations for each shot, but also investigate hyperparameter sensitivity on a smaller sweep in Supplementary, Sec. S3.5. For our experiments in Secs. 6 and 7, we also sweep four weight decay settings (0.01, 0.001, 0.0001, and 0.0, *i.e*., 'no weight decay'), highlighted by a 'wd' superscript. We use the validation set for early stopping and to find the best setting across the sweep. Our cached-feature setup (*cf*. Fig. 2) fits on a single-host TPUv2 platform where our experiments run in the order of minutes.

Evaluation: We report the top-1 accuracy across all our few-shot datasets. Although we mainly report test performance, we tune all hyperparameters and base all of our design decisions on the validation set.

4.5. Baseline Models

We establish two baselines: MAP and linear probe.

MAP: We first cache the $N \times C$-shaped frozen features from the last transformer block. Afterwards, we train a lightweight MAP head (*cf*. Fig. 2) from scratch following the training protocol in Sec. 4.4. We add a 'wd' superscript, *i.e*., MAP$^{\text{wd}}$, whenever we include the weight decay sweep. For simplicity, the MAP head employs the same architectural design as the underlying pretrained model.

Linear probe: We use cached $1 \times C$-shaped frozen features from the pretrained MAP head to solve an L2-regularized regression problem with a closed-form solution [73]. We sweep the L2 decay factor using exponents of 2 ranging from -20 up to 10. This setting is our auxiliary baseline.

Baseline	1-shot	5-shot	10-shot	25-shot
MAP	57.9	78.8	80.9	**83.2**
Linear probe	**66.5**	**79.6**	**81.5**	82.4

Table 1. **Baseline average top-1 accuracy** on *our* ILSVRC-2012 test set. We use the JFT-3B L/16 base setup (*cf*. Sec. 5) and follow the respective baseline setting (*cf*. Sec. 4.5). Each shot is sampled five times. The best result per shot is boldfaced.

5. Finding the Optimal FroFA Setup

We focus our first investigations on an L/16 ViT pretrained on JFT-3B, *i.e*., our largest model and largest pure image classification pretraining dataset, followed by few-shot transfer learning on subsets of the ILSVRC-2012 training set, *i.e*., our largest few-shot transfer dataset. We will refer to this setup as *our JFT-3B L/16 base setup*.

5.1. Baseline Performance

We first report the baseline performance in Tab. 1. We observe a large gap between MAP and linear probe on 1-shot (-8.6% absolute) which significantly decreases on 5-, 10-, and 25-shot settings to -0.8%, -0.6%, and $+0.8$% absolute, respectively.

In the following, our main point of comparison is the MAP baseline. This might be counter-intuitive since the performance is worse than linear probe in most cases. However, the higher input dimensionality in the MAP-based setting (*cf*. Sec. 4.5) gives us the option of input reshaping (*cf*. Sec. 3.3) which opens up more room and variety for frozen feature augmentations (FroFAs). Later in Sec. 6.3, we compare the performance of our best FroFA to the linear probe.

5.2. Default FroFA

We now investigate the effect of adding a single FroFA to the MAP baseline and start with the default FroFA formulation. Recall that we only use a single randomly sampled value per input (*cf*. Sec. 3.3). In Tab. 2, we report gains w.r.t. the MAP baseline on eighteen distinct FroFAs, categorized into geometric, crop & drop, stylistic, and other. In Supplementary, Sec. S3.7, we report on two additional FroFAs.

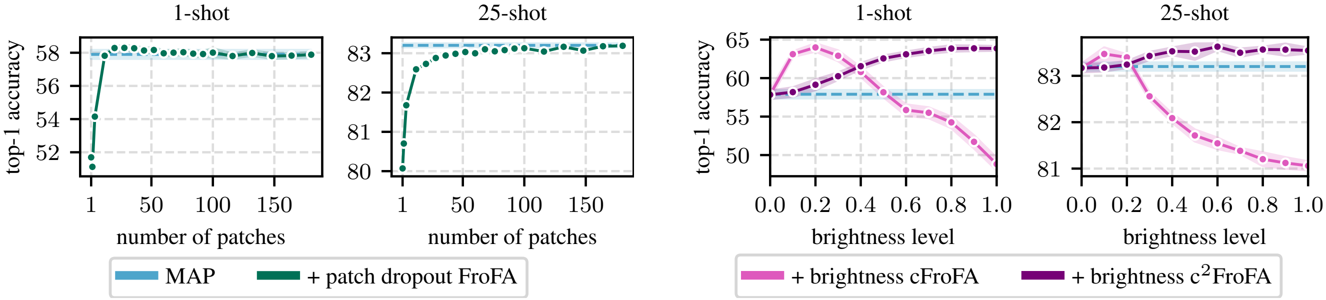
Geometric: Interestingly, all geometric augmentations consistently lead to worse performance across all settings.

Crop & drop: Applying a simple crop or a combination of resizing and crop yield a significant performance boost in the 1-shot setting of 3.0% and 1.9% absolute, respectively. Patch dropout, on the other hand, provides modest gains in the 1-shot regime. Dropping patches is directly related to training efficiency, so we investigate this further. Fig. 3a shows the top-1 accuracy on 1- and 25-shot as a function of number of patches. Results across other shots are similar (*cf*. Supplementary, Sec. S3.1). Similar to observations

[1] https://github.com/google-research/big_vision
[2] https://github.com/google-research/scenic

		Geometric					Crop & drop				Stylistic							Other	
Shots	MAP	rotate	shear-x	shear-y	translate-x	translate-y	crop	res. crop	incept. crop	patch drop.	**brightness**	**contrast**	equalize	invert	**posterize**	sharpness†	solarize†	JPEG†	mixup
1	57.9	−1.3	−0.6	−0.8	−1.2	−1.4	+3.0	+1.9	+0.0	+0.4	+4.8	+2.8	+1.0	+2.7	+3.7	−0.1	+1.0	−0.1	−1.4
5	78.8	−0.3	−0.2	−0.2	−0.3	−0.3	+0.0	−0.2	+0.0	+0.0	+1.1	+0.8	+0.5	−0.3	+0.8	+0.1	−0.1	−0.3	−0.3
10	80.9	−0.2	−0.1	−0.1	−0.2	−0.2	+0.0	−0.2	+0.0	+0.0	+0.6	+0.6	+0.4	+0.0	+0.6	+0.1	+0.0	−0.1	+0.2
25	83.2	−0.2	−0.1	−0.2	−0.1	−0.2	+0.0	−0.1	−0.1	+0.0	+0.1	+0.1	+0.0	−0.2	+0.0	+0.0	+0.0	+0.0	+0.1

Table 2. **(Average) top-1 accuracy for default FroFA** on *our* ILSVRC-2012 test set. Absolute gains to the MAP baseline are reported. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). In total, we investigate eighteen FroFAs, categorized into *geometric*, *crop & drop*, *stylistic*, and *other*. We highlight deterioration by shades of red and improvement by shades of green. Each shot is sampled five times, except for augmentations marked with '†'. Best three FroFAs are boldfaced.



(a) Patch dropout FroFA

(b) Channel variants (c/c^2) of brightness FroFA

Figure 3. **Average top-1 accuracy for FroFA variants** on *our* ILSVRC-2012 test set. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). We sweep across a base sweep (*cf*. Sec. 4.4) to first find the best setting on *our* ILSVRC-2012 validation set for each FroFA operation point (*cf*. Supplementary, Sec. S2.2). Shaded areas indicate standard errors collected via sampling each shot five times.

by Liu *et al*. [42] we can randomly drop a large fraction of patches (>50%) without loosing performance. A key difference is that Liu *et al*. only investigated the effect in the image space, while we provide evidence that patch dropout also transfers to the feature space. Finally, inception crop does not improve performance.

Stylistic: The largest gains can be observed when employing a stylistic FroFA, in particular brightness, contrast, and posterize. We identified brightness as the best performing FroFA with absolute gains of 4.8% on 1-shot, 1.1% on 5-shot, and up to 0.6% on 10-shot.

Other: Neither JPEG nor mixup yield performance gains but rather more or less worsen the performance.

		Brightness			Contrast		Posterize	
Shots	MAP	-	c	c^2	-	c	-	c
1	57.9	+4.8	+**5.9**	+**6.1**	+**2.8**	+2.5	+3.7	+**5.9**
5	78.8	+1.1	+**1.5**	+**1.6**	+**0.8**	+0.0	+**0.8**	+**0.8**
10	80.9	+0.6	+**1.1**	+**0.9**	+**0.6**	+0.0	+**0.6**	+**0.5**
25	83.2	+0.1	+**0.4**	+**0.3**	+**0.1**	−**0.1**	+**0.0**	+**0.0**

Table 3. **Average top-1 accuracy for a selection of default (-) and channel (c/c^2) FroFA** on *our* ILSVRC-2012 test set. Absolute gains to the MAP baseline are reported. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). Each shot is sampled five times. The best results per shot *and* FroFA are boldfaced (multiple ones if close, *i.e*., ±0.2).

5.3. Channel FroFA

We continue with channel FroFA (cFroFA) using three stylistic augmentations: brightness, contrast, and posterize. In Tab. 3, we report absolute gains w.r.t. the MAP baseline and incorporate channel (c) and non-channel (-) variants. First, contrast cFroFA does not improve upon its non-channel variant across all shots. Second, posterize cFroFA improves performance on 1-shot from 3.7% to 5.9% while maintaining performance on all other shots. Lastly, brightness cFroFA significantly improves performance across all shots: 4.8% → 5.9% on 1-shot, 1.1% → 1.5% on 5-shot, 0.6% → 1.1% on 10-shot, and 0.1% → 0.4% on 25-shot.

Given the strong improvements for brightness cFroFA, we further test brightness c^{2}FroFA (c^2 in Tab. 3). On a first look, the c^{2}FroFA variant performs comparable to the cFroFA variant. In Fig. 3b, we report top-1 accuracy on 1- and 25-shot as a function of the brightness level. Results across other shots are similar and can be found in Supplementary, Sec. S3.1. Now we clearly observe that brightness cFroFA is more sensitive to the brightness level than brightness c^{2}FroFA. In general, brightness cFroFA only works well for small brightness levels (0.1 to 0.5), while its c^{2}FroFA counterpart performs better than the MAP baseline across the board. We attribute the better sensitivity properties of brightness c^{2}FroFA to the channel-wise mappings (5), (7) on $\boldsymbol{f}_c^*$ (2) since this is the only change compared to cFroFA. We did not observe similar effects for posterize when switching from cFroFA to c^{2}FroFA.

5.4. Sequential FroFA

Finally, out of our best three augmentations, *i.e.*, brightness c^{2}FroFA (Bc^2), contrast FroFA (C), and posterize cFroFA (Pc), we combine two of them sequentially (→) yielding six combinations. In Tab. 4, we compare all six combinations to our prior best (Bc^2). On 1-shot, 'Bc^2→Pc' significantly outperforms 'Bc^2', improving absolute gains from 6.1% to 7.7%, while maintaining performance on other shots. We conclude that advanced FroFA protocols may further improve performance. As an initial investigation, we applied variations of RandAugment and TrivialAugment using our best three FroFAs (*cf*. Tab. 3), however, with limited success. We include results in the Supplementary, Sec. S3.2, and leave a deeper investigation to future works.

6. Results on More Model Architectures and Pretraining Datasets

How well does our best non-sequential FroFA strategy, *i.e.*, brightness c^{2}FroFA, transfer across multiple architecture and pretraining setups? We address this question in Secs. 6.1 and 6.2 and explore FroFA on ILSVRC-2012 frozen features from Ti/16, B/16, and L/16 ViTs pretrained on JFT-3B or ImageNet-21k, respectively. We further provide a comparison to linear probe in Sec. 6.3. Throughout this section, we report results on ILSVRC-2012. Further, in this section and Sec. 7, *all MAP-based models employ a weight decay sweep* denoted as MAP$^{\text{wd}}$ (Sec. 4.4).

6.1. JFT-3B Pretraining

In Fig. 4a, we report improvements in top-1 accuracy w.r.t. the MAP$^{\text{wd}}$ baseline for Ti/16, B/16, and L/16 ViTs pretrained on JFT-3B. Across all shots and all architectures incorporating FroFA either *maintains or improves performance* over the MAP$^{\text{wd}}$ baseline. On 1-shot, we further observe increasing improvements from FroFA on scaling the architecture. With higher shots, the improvement over the baseline becomes smaller. We attribute this to the already strong baseline performance leaving lesser headroom for improvements. We refer to the Supplementary, Sec. S3.3, for the exact values.

6.2. ImageNet-21k Pretraining

In Fig. 4b, we again look at improvements in top-1 accuracy w.r.t. the MAP$^{\text{wd}}$ baseline for the same ViTs, but now pretrained on ImageNet-21k. Consistent with our JFT-3B results, the performance either *maintains or improves* over the MAP$^{\text{wd}}$ baseline by incorporating FroFA and the improvements over the baseline become smaller with higher shots. We further observe increasing improvements from FroFA on scaling the architecture on 5- and 10-shot. We refer to the Supplementary, Sec. S3.3, for the exact values.

Shots	MAP	Bc^2	Bc^2→C	C→ Bc^2	Bc^2→Pc	Pc→ Bc^2	C→Pc	Pc→C
1	57.9	+6.1	+4.0	+2.7	+**7.7**	+5.2	+5.0	+3.1
5	78.8	+**1.6**	+**1.5**	+0.2	+**1.5**	+0.4	+1.3	+0.0
10	80.9	+0.9	+**1.2**	+0.1	+**1.0**	+0.1	+0.9	+0.3
25	83.2	+**0.3**	+**0.4**	−0.7	+**0.2**	−0.5	+0.2	−0.4

Table 4. **Average top-1 accuracy for a sequential FroFA protocol** on *our* ILSVRC-2012 test set. Absolute gains to the MAP baseline are reported. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). We combine the best settings of brightness c^{2}FroFA (Bc^2), contrast FroFA (C), and posterize cFroFA (Pc) sequentially (two at a time, order indicated by '↑'). Each shot is sampled five times. The best results per shot are boldfaced (multiple ones if close, *i.e.*, ±0.2).

6.3. Linear Probe Comparison

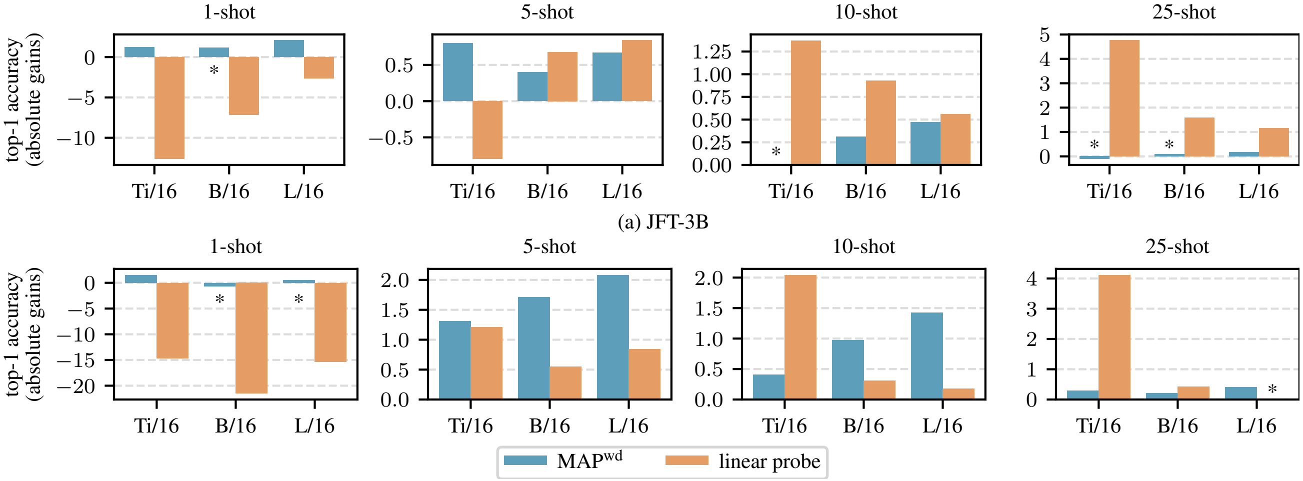
Finally, we revisit Figs. 4a and 4b, but now discuss gains w.r.t. the linear probe baseline. We start with models pretrained on JFT-3B (*cf*. Fig. 4a). On 1-shot, we observe that we lack behind linear probe but can close the gap by scaling up the model size. On 5- to 25-shot, with the exception of Ti/16 on 5-shot, brightness c^{2}FroFA significantly outperforms the linear probe baseline. On ImageNet-21k (*cf*. Fig. 4b), we observe even larger gaps to linear probe on 1-shot (up to −20% absolute). However, similar to results on JFT-3B, performance on 5- to 25-shot improves significantly over linear probe or at worst stays the same.

7. Results on More Few-Shot Datasets and Vision-Language Pretraining

Our study so far explored FroFA on ILSVRC-2012 as a few-shot dataset. In this section, we analyze FroFA on seven additional few-shot datasets, *i.e.*, CIFAR10, CIFAR100, DMLab, DTD, Resisc45, SUN397, and SVHN. In Sec. 7.1, we first use an L/16 ViT pretrained on JFT-3B for our analysis. In Sec. 7.2, we extend this analysis with the L/16 ViT image encoder of a vision-language model which was pretrained with sigmoid language-image pretraining (SigLIP) [74] on WebLI.

7.1. JFT-3B Pretraining

In Tab. 5 (upper half), we report mean results over the seven few-shot datasets using a JFT-3B L/16 ViT. Per dataset and



(a) JFT-3B

(b) ImageNet21k

Figure 4. **Average top-1 accuracy of brightness c^{2}FroFA combined with weight decay for JFT-3B (a) and ImageNet-21k (b) ViTs** on *our* ILSVRC-2012 test set trained on few-shotted ILSVRC-2012 training sets. Absolute gains to the weight-decayed MAP, *i.e.* MAP^{wd}, and L2-regularized linear probe baseline are reported. Each shot is sampled five times. An asterisk (*) indicates that statistical significance is not given under a two-tailed t-test with 95% confidence for that particular 'pretraining, shots, model, baseline'-setting (*e.g.*, 'JFT3-B, 10-shot, Ti/16, MAP' or 'ImageNet-21k, 25-shot, L/16, linear probe').

Pretraining scheme	Method	1-shot	5-shot	10-shot	25-shot
JFT-3B	MAP^{wd}	49.5	65.8	68.3	74.1
	Linear probe	49.1	62.7	65.7	68.8
	MAP^{wd} + FroFA	**53.4**	**67.3**	**70.9**	**74.9**
WebLI + SigLIP	MAP^{wd}	45.9	67.7	71.8	75.1
	Linear probe	49.1	65.0	69.3	72.6
	MAP^{wd} + FroFA	**51.3**	**70.4**	**73.5**	**76.0**

Table 5. **Average top-1 accuracy of our best FroFA computed across seven few-shot datasets** using a JFT-3B or WebLI-SigLIP L/16 ViT with weight decay. We report the mean across all test sets and refer to Supplementary, Tabs. 11 and 12, for more details. Per shot and dataset, the best result is boldfaced.

shot, top-1 accuracy and two-tailed t-tests with 95% confidence are provided in Supplementary, Tab. 11. We compare the MAP^{wd} and linear probe baseline with MAP^{wd} combined with brightness c^{2}FroFA (MAP^{wd} + FroFA). Across all shots, 'MAP^{wd} + FroFA' yields the highest mean results, surpassing the second-best approach (MAP^{wd}) by 3.9%, 1.5%, 2.6%, and 0.8% absolute on 1-, 5-, 10-, and 25-shot, respectively (*cf*. Fig. 1, left). Furthermore, Fig. 1 (left) reveals that while the gains to MAP^{wd} diminish with higher shots, the gains to linear probe actually increase and amount to at least 4.0% absolute across all shots.

7.2. WebLI Vision-Language Pretraining

Given the strong performance with the JFT-3B L/16 ViT, we finally ask: Does FroFA also transfer to ViTs with vision-language pretraining?

To answer this question, we train 'MAP^{wd}', 'linear probe', and 'MAP^{wd} + FroFA' using frozen features from the L/16 ViT image encoder of a WebLI-SigLIP vision-language model. In Tab. 5 (lower half), we report mean results over the same seven few-shot datasets from before. We again provide more detailed results and two-tailed t-tests in Supplementary, Tab. 12. Across all shots, 'MAP^{wd} + FroFA' again yields the highest mean results, surpassing the second-best approach on 1-shot (linear probe) by 2.2% absolute and the second-best approach on 5-, 10-, and 25-shot (MAP^{wd}) by 2.7%, 1.7%, and 0.9% absolute, respectively (*cf*. Fig. 1, right). In Fig. 1 (right), we observe that the gains to both MAP^{wd} and linear probe (neglecting 1-shot) diminish with higher shots. Overall, we can confirm that FroFA also transfers to a ViT with vision-language pretraining.

8. Conclusion

We investigated twenty frozen feature augmentations (FroFAs) for few-shot transfer learning along three axes: model size, pretraining and transfer few-shot dataset. We show that a training with FroFAs, in particular stylistic ones, gives large improvements upon a representative baseline across all shots. In addition, per-channel variants further improve performance, e.g., by 1.6% absolute in the ILSVRC-2012 5-shot setting. Finally, we show that FroFA excels on smaller few-shot datasets. For example, averaged results across seven few-shot tasks show that training on cached frozen features from a JFT-3B L/16 vision transformer with a per-channel variant of brightness FroFA gives consistent gains of at least 4.0% absolute upon linear probe across 1- to 25-shot settings.

References

[1] Alex Krizhevsky. Learning Multiple Layers of Features From Tiny Images, 2009. 1, 2, 4, 13

[2] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved Few-Shot Visual Classification. In *Proc. of CVPR*, pages 14481–14490, virtual, 2020. 2

[3] Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. DeepMind Lab. *arXiv*, 1612.03801:1–11, 2016. 4, 13

[4] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better Plain ViT Baselines for ImageNet-1k. *arXiv*, 2205.01580: 1–3, 2022. 4, 13

[5] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition. In *Proc. of NeurIPS*, pages 16664–16678, New Orleans, LA, USA, 2022. 2

[6] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A Jointly Scaled Multilingual Language-Image Model. In *Proc. of ICLR*, pages 1–33, Kigali, Rwanda, 2023. 1, 2, 4

[7] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE*, 105(10):1865–1883, 2017. 4, 13

[8] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible Scaling Laws for Contrastive Language-Image Learning. In *Proc. of CVPR*, pages 2818–2829, Vancouver, BC, Canada, 2023. 1

[9] Tsz-Him Cheung and Dit-Yan Yeung. MODALS: Modality-agnostic Automated Data Augmentation in the Latent Space. In *Proc. of ICLR*, pages 1–18, virtual, 2021. 2

[10] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proc. of CVPR*, pages 1063–6919, Honolulu, HI, USA, 2017. 4

[11] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. In *Proc. of CVPR*, pages 3606–3613, Columbus, OH, USA, 2014. 4, 13

[12] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *Proc. of CVPR*, pages 113–123, Long Beach, CA, USA, 2019. 1, 2, 4, 13

[13] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *Proc. of NeurIPS*, pages 18613–18624, virtual, 2020. 1, 4, 13, 18

[14] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. CoAtNet: Marrying Convolution and Attention for All Data Sizes. In *Proc. of NeurIPS*, pages 3965–3977, virtual, 2021. 4

[15] Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay. Scenic: A JAX Library for Computer Vision Research and Beyond. In *Proc. of CVPR*, pages 21393–21398, New Orleans, LA, USA, 2022. 5

[16] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling Vision Transformers to 22 Billion Parameters. In *Proc. of ICML*, pages 7480–7512, Honolulu, HI, USA, 2023. 1, 2, 4

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. of CVPR*, pages 248–255, Miami, FL, USA, 2009. 1, 2, 4

[18] Terrance DeVries and Graham W. Taylor. Dataset Augmentation in Feature Space. In *Proc. of ICLR - Workshops*, pages 1–12, Toulon, France, 2017. 2

[19] 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 *Proc. of ICLR*, pages 1–21, virtual, 2021. 1, 2, 4

[20] Vincent Dumoulin, Neil Houlsby, Utku Evci, Xiaohua Zhai, Ross Goroshin, Sylvain Gelly, and Hugo Larochelle. A Unified Few-Shot Classification Benchmark to Compare Transfer and Meta Learning Approaches. In *Proc. of NeurIPS - Datasets and Benchmarks Track*, pages 1–14, virtual, 2021. 2

[21] Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. How to Train Vision Transformer on Small-Scale Datasets? In *Proc. of BMVC*, pages 1–16, London, UK, 2022. 1

[22] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. CLIP-Adapter: Better Vision-Language Models with Feature Adapters. *Int. J. Comput. Vis.*, 132(2):581–595, 2023. 2

[23] Raphael Gontijo-Lopes, Sylvia Smullin, Ekin Dogus Cubuk, and Ethan Dyer. Tradeoffs in Data Augmentation: An Empirical Study. In *Proc. of ICLR*, pages 1–27, virtual, 2021. 18

[24] Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving Neural Language Models with a Continuous Cache. In *Proc. of ICLR*, pages 1–9, Toulon, France, 2017. 2

[25] Xuehai He, Chuanyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-Efficient Model Adaptation for Vision Transformers. In *Proc. of AAAI*, pages 817–825, Washington, DC, USA, 2023. 1, 2

[26] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *Proc. of ICLR*, pages 1–15, Virtual, 2020. 2

[27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling Knowledge in a Neural Network. In *Proc. of NIPS - Workshops*, pages 1–9, Montréal, QC, Canada, 2014. (In 2018, 'NIPS' was renamed to 'NeurIPS'). 2, 4

[28] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proc. of ICML*, pages 2790–2799, Long Beach, CA, USA, 2019. 2

[29] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *Proc. of ICLR*, pages 1–13, virtual, 2022. 2

[30] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning. In *Proc. of ECCV*, pages 709–727, Tel Aviv, Israel, 2022. 2

[31] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*, pages 1–15, San Diego, CA, USA, 2015. 14

[32] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. In *Proc. of ECCV*, pages 491–507, virtual, 2020. 2, 4

[33] Jannik Kossen, Mark Collier, Basil Mustafa, Xiao Wang, Xiaohua Zhai, Lucas Beyer, Andreas Steiner, Jesse Berent, Rodolphe Jenatton, and Efi Kokiopoulou. Three Towers: Flexible Contrastive Learning with Pretrained Image Models. In *Proc. of NeurIPS*, pages 31340–31371, New Orleans, LA, USA, 2023. 4

[34] Taku Kudo and John Richardson. SentencePiece: A Simple and Language-Independent Subword Tokenizer and Detokenizer for Neural Text Processing. In *Proc. of EMNLP - System Demonstrations*, pages 66–71, Brussels, Belgium, 2018. 16

[35] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and Wlliam Campbell. A Closer Look At Feature Space Data Augmentation For Few-Shot Intent Classification. In *Proc. of EMNLP - Workshops*, pages 1–10, Hong Kong, China, 2019. 2

[36] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. The Omniglot Challenge: A 3-year Progress Report. *Curr. Opin. Behav. Sci.*, 29:97–104, 2019. 2

[37] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-Based Permutation-Invariant Neural Networks. In *Proc. of ICML*, pages 3744–3753, Long Beach, CA, USA, 2019. 1, 2, 4

[38] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision Transformer for Small-Size Datasets. *arXiv*, 2112.13492:1–11, 2021. 1

[39] Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proc. of EMNLP*, pages 3045–3059, virtual, 2021. 2

[40] Xiaofeng Liu, Yang Zou, Lingsheng Kong, Zhihui Diao, Junliang Yan, Jun Wang, Site Li, Ping Jia, and Jane You. Data Augmentation via Latent Space Interpolation for Image Classification. In *Proc. of ICPR*, pages 728–733, Beijing, China, 2018. 2

[41] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient Training of Visual Transformers with Small Datasets. In *Proc. of NeurIPS*, pages 23818–23830, virtual, 2021. 1

[42] Yue Liu, Christos Matsoukas, Fredrik Strand, Hossein Azizpour, and Kevin Smith. PatchDropout: Economizing Vision Transformers Using Patch Dropout. In *Proc. of WACV*, pages 3942–3951, Waikoloa, HI, USA, 2023. 2, 4, 6

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proc. of ICCV*, pages 10012–10022, virtual, 2021. 1

[44] Zichang Liu, Zhiqiang Tang, Xingjian Shi, Aston Zhang, Mu Li, Anshumali Shrivastava, and Andrew Gordon Wilson. Learning Multimodal Data Augmentation in Feature Space. In *Proc. of ICLR*, pages 1–15, Kigali, Rwanda, 2023. 2

[45] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proc. of ICLR*, pages 1–18, New Orleans, LA, USA, 2019. 14

[46] Samuel G. Müller and Frank Hutter. TrivialAugment: Tuning-Free Yet State-of-the-Art Data Augmentation. In *Proc. of ICCV*, pages 774–782, virtual, 2021. 1, 2, 4, 13

[47] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Proc. of NIPS - Workshops*, pages 1–9, Granada, Spain, 2011. (In 2018, 'NIPS' was renamed to 'NeurIPS'). 4, 13

[48] Alex Nichol, Joshua Achiam, and John Schulman. On First-Order Meta-Learning Algorithms. *arXiv*, 1803.02999:1–15, 2018. 2

[49] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features Without Supervision. *Trans. Mach. Learn. Res.*, 1:1–32, 2024. 1

[50] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task-Dependent Adaptive Metric for Improved Few-Shot Learning. In *Proc. of NeurIPS*, pages 719–729, Montréal, QC, Canada, 2018. 2

[51] Emin Orhan. A Simple Cache Model for Image Recognition. In *Proc. of NeurIPS*, pages 10128–10137, Montréal, Canada, 2018. 2

[52] 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 *Proc. of ICML*, pages 8748–8763, virtual, 2021. 1, 2

[53] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning With a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21 (140):1–67, 2020. 16

[54] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and Flexible Multi-Task Classification Using Conditional Neural Adaptive Processes. In *Proc. of NeurIPS*, pages 7957–7968, Vancouver, BC, Canada, 2019. 2

[55] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses. In *Proc. of NeurIPS - Datasets and Benchmarks Track*, pages 1–12, virtual, 2021. 2

[56] Pau Rodríguez, Issam H. Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding Propagation: Smoother Manifold for Few-Shot Classification. In *Proc. of ECCV*, pages 121–138, virtual, 2020. 2

[57] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. 1, 2, 4, 13

[58] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *Proc. of ICML*, pages 4596–4604, Stockholm, Sweden, 2018. 14

[59] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-Shot Learning. In *Proc. of NIPS*, pages 4077–4087, Long Beach, CA, USA, 2017. (In 2018, 'NIPS' was renamed to 'NeurIPS'). 2

[60] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to Train Your ViT? Data, Augmentation, and Regularization in Vision Transformers. *Trans. Mach. Learn. Res.*, 5:1–16, 2022. 2, 5, 14, 16

[61] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *Proc. of ICCV*, pages 843–852, Venice, Italy, 2017. 4

[62] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proc. of CVPR*, pages 2818–2826, Las Vegas, NV, USA, 2016. 2, 4

[63] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking Few-Shot Image Classification: A Good Embedding Is All You Need? In *Proc. of ECCV*, pages 266–282, virtual, 2020. 2

[64] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training Data-Efficient Image Transformers & Distillation Through Attention. In *Proc. of ICML*, pages 10347–10357, virtual, 2021. 4

[65] Cristina Vasconcelos, Vighnesh Birodkar, and Vincent Dumoulin. Proper Reuse of Image Classification Features Improves Object Detection. In *Proc. of CVPR*, pages 13628–13637, New Orleans, LA, USA, 2022. 2

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Proc. of NIPS*, pages 5998–6008, Long Beach, CA, USA, 2017. (In 2018, 'NIPS' was renamed to 'NeurIPS'). 14

[67] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States. In *Proc. of ICML*, pages 6438–6447, Long Beach, CA, USA, 2019. 2

[68] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching Networks for One-Shot Learning. In *Proc. of NIPS*, pages 3637–3645, Barcelona, Spain, 2016. (In 2018, 'NIPS' was renamed to 'NeurIPS'). 2

[69] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In *Proc. of ICCV*, pages 548–558, virtual, 2021. 1

[70] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN Database: Large-Scale Scene Recognition From Abbey to Zoo. In *Proc. of CVPR*, pages 3485–3492, San Francisco, CA, USA, 2010. 2, 4, 13

[71] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. SUN Database: Exploring a Large Collection of Scene Categories. *Int. J. Comput. Vis.*, 119(1): 3–22, 2016. 1, 2, 4, 13

[72] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, André Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A Large-Scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv*, 1910.04867:1–33, 2020. 4, 13

[73] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. In *Proc. of CVPR*, pages 12104–12113, New Orleans, LA, USA, 2022. 1, 2, 4, 5, 14

[74] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language-Image Pretraining. In *Proc. of ICCV*, pages 11975–11986, Paris, France, 2023. 1, 2, 5, 7, 14

[75] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization. In *Proc. of ICLR*, pages 1–13, Vancouver, BC, Canada, 2018. 1, 2, 4

[76] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-Adapter: Training-Free Adaption of CLIP for Few-Shot Classification. In *Proc. of ECCV*, pages 493–510, Tel Aviv, Israel, 2022. 2