Frozen Feature Augmentation for Few-Shot Image Classification

Supplementary Material

S1. Introduction

We give additional details and results to complement the main paper. All included citations refer to the main paper's references.

S2. Detailed Experimental Setup

In the following, we provide additional details to our experimental setup.

S2.1. Datasets for Few-Shot Transfer Learning

In this section, we focus on details regarding our few-shot transfer datasets. As stated in the main paper, Sec. 4.2, our experiments concentrate around few-shot transfer learning on ILSVRC-2012 [57]. We also provide results on CIFAR10 [1], CIFAR100 [1], DMLab [3, 72], DTD [11], Resisc45 [7], SUN397 [70, 71], and SVHN [47]. When official test and validation splits are available, we use them for evaluation across all datasets. In general, we use the versions in `TensorFlow Datasets`[3]. Our exact splits are given in Tab. 6.

CIFAR10 contains 60,000 images of 10 equally distributed classes split into 50,000 training images and 10,000 test images. We further split the official training dataset into 45,000 training images and 5,000 validation images.

CIFAR100 is a superset of CIFAR10 with 100 equally distributed classes and 60,000 images. Similar to CIFAR10, we use 45,000 images for training, 5,000 images for validation and 10,000 images for test.

DMLab consists of frames collected from the DeepMind Lab environment. Each frame is annotated with one out of six classes. We use 65,550 images for training, 22,628 images for validation, and 22,735 for test.

DTD is a collection of 5,640 textural images categorized into 47 distinct classes. Each of the three splits, *i.e.*, training, validation, and test, has exactly 1,880 images.

ILSVRC-2012[4], also known as 'ImageNet-1k' or just 'ImageNet', is a slimmed version of 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.

Resisc45 is a benchmark with 31,500 images for image scene classification in remote sensing scenarios. In total, 47 different categories for scenes are defined. We use the first 23,000 images for training, the subsequent 2,000 images for validation and the last 6,300 images for test.

SUN397 is a 397-category database of 108,753 images for scene understanding. We use 76,128 images for training, 10,875 images for validation, and 21,750 images for test.

SVHN is a Google Street View dataset with a large collection of house number images. In total, 10 distinct classes exist. We use the cropped version with 73,257 images for training and 26,032 images for test. Further, we create a validation subset by only using the first 70,000 out of 73,257 training images for actual training and the remaining 3,257 images for validation.

S2.2. Data Augmentation

In this section, we provide additional details on the used data augmentation techniques and protocols.

(c/c^2)FroFA: In Tab. 8, we give detailed descriptions of each FroFA, cFroFA, and c^{2}FroFA setting. We mostly build upon an AutoAugment [12] implementation from `Big Vision`[5]. To keep it simple, we use v or v_1, v_2 as sweep parameter(s) for all augmentations. By default, we first reshape the two-dimensional features $\boldsymbol{f}$ to three-dimensional features $\boldsymbol{f}^*$ (1) of shape $\sqrt{N} \times \sqrt{N} \times C$, with $N = 196$ and $C \in \{192, 768, 1024\}$ in all our experiments. Note that the value of C depends on the architecture. We further want to point out, while some augmentations heavily rely on the three-dimensional representation, *e.g.*, all geometric ones, some others are also transferable to a two-dimensional representation, *e.g.*, brightness or contrast.

As pointed out in the main paper, Tab. 3, brightness c^{2}FroFA, contrast FroFA, and posterize cFroFA are our best FroFAs. For all three, we list the best sweep settings in Tab. 7.

Advanced protocols: As mentioned in the main paper, Sec. 4.3, besides our fixed sequential protocol (*cf*. Tab. 4) we also tested variations of RandAugment [13] and TrivialAugment [46]. In all protocols, we sample from the best settings of brightness c^{2}FroFA, contrast FroFA, and posterize cFroFA. In particular, we use $v = 1.0$ for brightness c^{2}FroFA, $v = 5$ for contrast FroFA, and $v_1 = 1, v_2 = 8$ for posterize cFroFA (*cf*. Tab. 8). We re-use the abbreviations from Tab. 4 in the following, *i.e.*, Bc^2, C, and Pc,

[3] https://www.tensorflow.org/datasets

[4] For the sake of completeness, we copied this paragraph from the main paper (unaltered).

[5] https://github.com/google-research/big_vision/blob/main/big_vision/pp/autoaugment.py

Dataset	Training split	Validation split	Test split
CIFAR10	`train[:45000]`	`train[45000:]`	`test`
CIFAR100	`train[:45000]`	`train[45000:]`	`test`
DMLAB	`train`	`validation`	`test`
DTD	`train`	`validation`	`test`
ILSVRC-2012[†]	`train[:10%],train[10%:20%]` `train[20%:30%],train[30%:40%],train[40%:50%]`	`train[99%:]`	`validation`
Resisc45	`train[:23200]`	`train[23200:25200]`	`train[25200:]`
SUN397	`train`	`validation`	`test`
SVHN	`train[:70000]`	`train[70000:]`	`test`

Table 6. `TensorFlow Datasets`[3] splits used for few-shot transfer learning. Note that before training, we first sample few-shot versions from the respective training split. [†]We don't use all five ILSVRC-2012 training subsplits at the same time but rather average results across the five (few-shotted) training splits (*cf*. Sec. 4.2).

FroFA	Shots	Base learning rate	Batch size	Training steps	v or v_1, v_2
Bc^2	1	0.01	512	4,000	1.0
	10	0.01	64	16,000	1.0
	15	0.01	256	8,000	0.9
	25	0.01	512	8,000	0.8
C	1	0.01	32	16,000	6.0
	10	0.01	128	8,000	6.0
	15	0.01	512	2,000	6.0
	25	0.01	256	4,000	7.0
Pc	1	0.01	512	8,000	1, 8
	10	0.03	512	8,000	1, 8
	15	0.03	512	16,000	1, 8
	25	0.03	64	16,000	2, 8

Table 7. **Our best sweep settings for our best three FroFAs**, namely, brightness c^2FroFA (Bc^2), contrast (C), and posterize cFroFA (Pc), based on the JFT-3B L/16 base setup (*cf*. Sec. 5). We list the shots, base learning rate, batch size, number of training steps, and the augmentation parameter, denoted as v or v_1, v_2 (see Tab. 8 for a detailed explanation of v and v_1, v_2). The best sweep settings are found using *our* ILSVRC-2012 validation set.

respectively. For the RandAugment and TrivialAugment variations, we uniformly sample from either the best three FroFAs, *i.e*., $\mathcal{A}_{\text{top3}} = \{\text{Bc}^2, \text{C}, \text{Pc}\}$, or the best two FroFAs, *i.e*., $\mathcal{A}_{\text{top2}} = \mathcal{A}_3 \setminus \{\text{C}\}$. Further, our RandAugment variation randomly constructs a sequence of augmentations by uniformly sampling the integer sequence length from 1 to $|\mathcal{A}|$, with $\mathcal{A} \in \{\mathcal{A}_{\text{top2}}, \mathcal{A}_{\text{top3}}\}$ depending on whether $\mathcal{A}_{\text{top2}}$ or $\mathcal{A}_{\text{top3}}$ is used.

S2.3. Training Details

Pretraining: In the JFT-3B setup, we use pretrained models from Zhai *et al*. [73]. The models are pretrained using a sigmoid cross-entropy loss. The weights are optimized by Adafactor [58], however, with slight modifications, including the use of the first momentum (in half-precision) by setting $\beta_1 = 0.9$ (instead of discarding it by $\beta_1 = 0$), disabling weight norm-based learning rate scaling, and limiting the second momentum decay to $\beta_2 = 0.999$. Further, weight decay is applied with 3.0 on the head and 0.03 for the rest of the remaining network weights. The learning rate is adapted by a reciprocal square-root schedule for 4,000,000 steps with a linear warm-up phase of 10,000 steps and a linear cool-down phase of 50,000 steps. The starting learning rate is set to 0.0008 for all model sizes (Ti/16, B/16, and L/16). The images are preprocessed by an 224×224 inception-style crop and a random horizontal flip. We set the batch size to 4,096. To stabilize training, a global norm clipping of 1.0 is used.

In the ImageNet-21k setup, we follow settings from Steiner *et al*. [60] and use a sigmoid cross-entropy loss for multi-label pretraining. We use the Adam optimizer [31] in half-precision mode and set $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Further, we apply (decoupled) weight decay [45] with either 0.03 for Ti/16 or 0.1 for B/16 and L/16. We adapt the learning rate using a cosine schedule for roughly 930,000 steps (300 epochs) with a linear warm-up phase of 10,000 steps. We set the starting learning rate to 0.001 for all models. During preprocessing, we crop the images to 224×224 following an inception-style crop and a random horizontal flip. While we don't use any additional augmentation for Ti/16, we follow suggestions by Steiner *et al*. [60] and use the '`light1`' and '`medium2`' augmentation settings for B/16 and L/16 ViTs, respectively. Finally, we use a batch size of 4,096 and stabilize training by using a global norm clipping of 1.0.

In the WebLI setup, we use a pretrained vision-language model from Zhai *et al*. [74]. The model consists of an L/16 ViT, later used in our experiments for few-shot transfer learning, and an L-sized transformer [66] for text embeddings. Similar to the JFT-3B training setup, the Adafactor optimizer is used with first momentum (in half-precision) and $\beta_1 = 0.9$, disabled weight norm-based learning rate scaling, and limitation of the second momentum decay to $\beta_2 = 0.999$. Further, weight decay is applied with 0.0001 and the learning rate is adapted by a reciprocal square-root schedule with a linear warm-up phase of 50,000 steps and a linear cool-down phase of 50,000 steps. The starting learn-

	Augmentation	Description
Geometric	rotate	We rotate each of the C feature channels by $z \sim U(-v, v)$. We sweep across $v \in \{15, 30, 45, 60, 75, 90\}$ representing the maximum positive and negative rotation angle in degrees.
	shear-{x,y}	We (horizontally/vertically) shear each of the C feature channels by $z \sim U(0, v)$. We sweep across $v \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ representing the maximum level of horizontal or vertical shearing.
	translate-{x,y}	We (horizontally/vertically) translate each of the C feature channels by uniformly sampling z from $\{0, 1, ..., v\}$. We sweep across integer values $1 \leq v \leq 7$ representing the maximum horizontal or vertical translation.
Crop & drop	crop	We randomly crop each of the C feature channels to $v \times v$ at the same spatial position. We sweep across integer values $1 \leq v \leq 13$ representing the square crop size.
	resized crop	We resize each of the C feature channels to $v \times v$ and then randomly crop each to 14×14 at the same spatial position. We sweep across $v \in \{16, 18, 20, 22, 24, 26, 28, 35, 42\}$ representing the resized squared spatial resolution.
	inception crop	We apply an inception crop with probability v. We sweep across $v \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.
	channel dropout†	We apply a channel dropout mask at the input with probability v. We sweep across $v \in \{0.1, 0.3, 0.5, 0.99\}$.
	patch dropout	We randomly keep v out of N patches of $\boldsymbol{f}$ having shape $N \times C$. Note that the patch ordering is also randomized. We sweep across $v \in \{1, 2, 4, 12, 20, 28, 36, 44, 52, 60, 68, 76, 84, 92, 100, 116, 132, 148, 164, 180\}$.
Stylistic	brightness	We randomly add a value $z \sim U(-v, v)$ to each of the C feature channels. We sweep across $v \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. In the default FroFA and the cFroFA variants, the features are scaled by (5) taking the minimum $f_{\min}$ and maximum $f_{\max}$ across all channels into account. In the c^{2}FroFA variant, each channel $\boldsymbol{f}_c^*$ (2) is shifted individually and uses the channel minimum and maximum instead. Further, in the cFroFA and c^{2}FroFA variants we sample z exactly C times, *i.e.*, each channel has its individual z.
	contrast	We randomly scale each of the C feature channels by $z \sim U(\frac{1}{v}, v)$. We sweep across $v \in \{1.25, 1.5, 2, 3, 4, 5, 6, 7, 9, 10\}$. We test this method using the default FroFA as well as cFroFA. Note that in the cFroFA variant we sample z exactly C times, *i.e.*, each channel has its individual z.
	equalize	We first map the features from value range $\mathbb{R}$ to the integer subset $\mathbb{I} = \{0, 1, ..., 195\}$, *i.e.*, executing (5) followed up by a discretization step. We choose this value range as preliminary results mapping from $\mathbb{R}$ to the more commonly used $\mathbb{I} = \{0, 1, ..., 255\}$ didn't show any effects. We continue by equalizing 196 bins and then transforming the results back to the original space using (7). We apply equalize with probability v. In particular, we sweep across $v \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.
	invert	We change the sign of the features with probability v. We sweep across $v \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.
	posterize	We first map the features $\boldsymbol{f}^*$ from value range $\mathbb{R}$ to the integer subset $\mathbb{I} = \{0, 1, ..., 255\}$, *i.e.*, executing (5) followed up by a discretization step. In other words, we use an 8-bit representation for features $\boldsymbol{f}^*$. Posterize performs a quantization by a bit-wise left and right shift. We uniformly sample the shift value z between integer values v_1 and v_2. In our sweep, we test a subset of all possible combinations. In particular, we first set $v_2 = 8$ and reduce v_1 from 7 to 1. We then fix $v_1 = 1$ and increase v_2 from 2 to 7 again. We test this method using the default FroFA as well as cFroFA. Note that in the cFroFA variant we sample z exactly C times, *i.e.*, each channel has its individual z.
	sharpness	We first apply a two-dimensional convolution using a 3×3 smoothing filter. Next, we mix the original features with the resulting 'smoothed' features using a randomly sampled blending factor $z \sim U(0, v)$. We sweep across $v \in \{0.2, 0.4, 0.6, 0.8, 1.0, 1.5, 2.0, 3.0\}$.
	solarize	We do not map features from $\mathbb{R}$ to $\mathbb{I} = [0, 1]$, but stay in $\mathbb{R}$. We compute the minimum $f_{\min}$ and maximum $f_{\max}$ across features $\boldsymbol{f}^*$. We conditionally subtract all values smaller than $0.5 \cdot f_{\min}$ from $f_{\min}$ or larger than $0.5 \cdot f_{\max}$ from $f_{\max}$. We apply this method with a probability v and sweep across $v \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.
	uniform noise†	We randomly add $z \sim U(-v, v)$ to each element independently. We sweep across $v \in \{0.1, 0.3, 0.5, 0.7\}$.
Other	JPEG	We first map the features from value range $\mathbb{R}$ to the integer subset $\mathbb{I} = \{0, 1, ..., 255\}$, *i.e.*, executing (5) followed up by a discretization step. We then perform a JPEG compression of each channel by randomly sampling a JPEG quality $z \sim U(v_1, v_2)$. We sweep across combinations of $v_1 \in \{10, 25, 50, 75\}$ and $v_2 \in \{25, 50, 75, 100\}$, with $v_2 > v_1$.
	mixup	We do not map features from $\mathbb{R}$ to $[0, 1]$, but stay in $\mathbb{R}$. We mix two features $\boldsymbol{f}_i^*, \boldsymbol{f}_j^*$ according to $z \cdot \boldsymbol{f}_i^* + (1 - z) \cdot \boldsymbol{f}_j^*$ by sampling a random value $z \sim B(\alpha, \alpha)$, with Beta distribution $B(\alpha, \alpha)$ parameterized by $\alpha = v$. The labels are mixed using the same procedure. We sweep across $v \in \{0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

Table 8. **Details on our used set of augmentations**. For simplicity, instead of introducing a new hyper parameter for each data augmentation, we re-use v as a sweep parameter that is set during a sweep and differs for each augmentation. If not stated otherwise, each method is only applied as default FroFA and we first map features $\boldsymbol{f}$ (two-dimensional representation) or $\boldsymbol{f}^*$ (three-dimensional representation) from value range $\mathbb{R}$ to $\mathbb{I} = [0, 1]$ using (5). By default, we assume a three-dimensional representation $\boldsymbol{f}^*$ although some augmentations would work also in the two-dimensional representation $\boldsymbol{f}$, *i.e.*, a reshaping is not necessary. †FroFAs not present in the main paper.

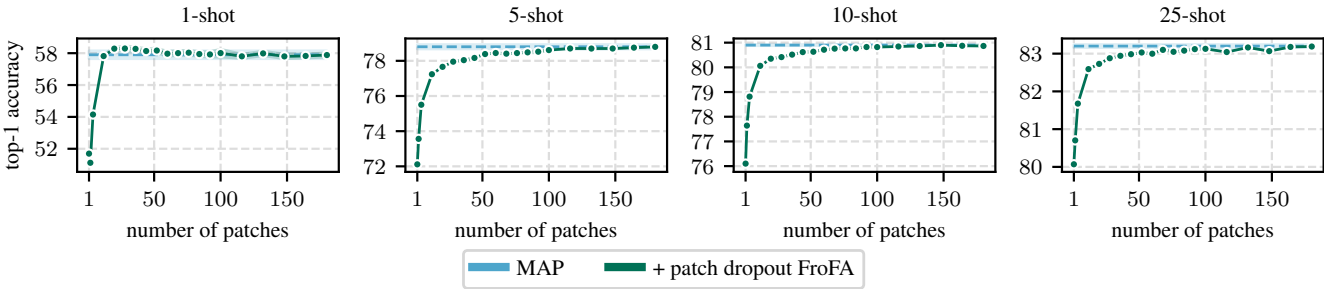

Figure 5. **Average top-1 accuracy for patch dropout FroFA** on *our* ILSVRC-2012 test set. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). We sweep across a base sweep (*cf*. Sec. 4.4) to first find the best setting on *our* ILSVRC-2012 validation set for each number of patches (*cf*. Sec. S2.2).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

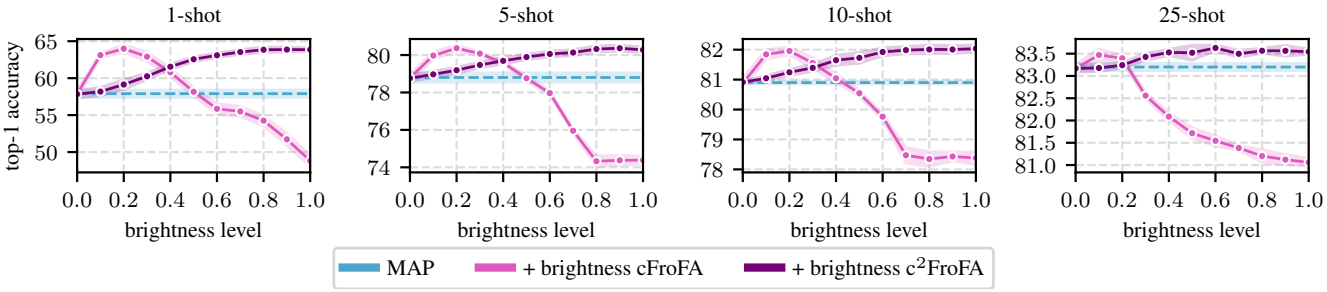

Figure 6. **Average top-1 accuracy for channel variants (c/c^2) of brightness FroFA** on *our* ILSVRC-2012 test set. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). We sweep across a base sweep (*cf*. Sec. 4.4) to first find the best setting on *our* ILSVRC-2012 validation set for each brightness level (*cf*. Sec. S2.2). Shaded areas indicate standard errors collected via sampling each shot five times.

ing rate is set to 0.001. The images are resized to 256×256 while the text is tokenized into 64 tokens by SentencePiece [34] trained on the English C4 dataset [53] using a vocabulary size of 32,000. The training is limited to 40 billion examples and a batch size of 32,768 is used.

Few-shot transfer learning: We first process each few-shot dataset through a pretrained model and store the extracted features (*cf*. Fig. 2). We resize each image to 224×224 before feeding it to the model.

We follow up with a training where we mostly use transfer learning settings from Steiner *et al*. [60]. We use a sigmoid cross-entropy loss. This might be non-intuitive given that all of our few-shot datasets are not multi-labeled. However, we didn't really observe any performance drops compared to using the more common softmax cross-entropy loss, so we stick to the sigmoid cross-entropy loss. We use stochastic gradient descent with momentum of 0.9. Similar to the pretraining setup, we also store internal optimizer states in half-precision. Except for the experiment series in Secs. 6 and 7, we do not apply any weight decay. The learning rate is adapted following a cosine schedule with a linear warm-up phase of 500 steps. In addition, we stabilize training by using a global norm clipping of 1.0. Further, we sweep across batch size, learning rate and number of steps yielding 100 combinations (*cf*. Sec. 4.4) for each shot.

S3. Additional Experiments and Results

In this section, we show additional experimental results.

S3.1. Patch Dropout and Brightness

In Fig. 3, we only report results for 1- and 25-shot settings using patch dropout FroFA and brightness (c/c^2)FroFA. We extend this by also reporting results for 5- and 10-shot settings in Figs. 5 and 6. The observations from Fig. 3 on 1- and 25-shot also transfer to 5- and 10-shot.

S3.2. Advanced FroFA Protocols

In Tab. 10, we report results for our RandAugment (RA*) and TrivialAugment (TA*) variations from Sec. S2.2. We did not average across five runs and thus only report absolute gains with respect to a reference run. Therefore, numbers which are reported in the main paper, *e.g*., in Tab. 4, are slightly different. Overall, we observe that both RA* and TA* do not improve upon the best single augmentation, *i.e*., brightness c^{2}FroFA (Bc^2). We also observe that increasing the set of augmentations from $\mathcal{A}_{\text{top2}}$ to $\mathcal{A}_{\text{top3}}$ rather worsens the performance for both RA* and TA*.

S3.3. ILSVRC-2012 Results

In Tab. 9, we give more detailed results for Fig. 4, *i.e*., Ti/16, B/16, and L/16 pretrained on either ImageNet-21k or JFT-

Model	Method	JFT-3B 1-shot	JFT-3B 5-shot	JFT-3B 10-shot	JFT-3B 25-shot	ImageNet-21k 1-shot	ImageNet-21k 5-shot	ImageNet-21k 10-shot	ImageNet-21k 25-shot
Ti/16	MAP^{wd}	19.1	46.4	**53.6***	**60.2***	20.5	53.6	59.7	64.9
	Linear probe	**33.0**	**48.0**	52.2	55.4	**36.8**	53.7	58.0	61.1
	MAP^{wd} + FroFA	20.3	47.2	**53.6***	60.1*	22.1	**54.9**	**60.1**	**65.2**
B/16	MAP^{wd}	51.3*	74.8	77.5	79.8*	31.3*	71.7	75.3	78.1
	Linear probe	**59.6**	74.5	76.9	78.3	**52.2**	72.9	76.0	77.9
	MAP^{wd} + FroFA	52.4*	**75.2**	**77.8**	**79.9***	30.6*	**73.4**	**76.3**	**78.3**
L/16	MAP^{wd}	61.8	79.8	81.5	83.4	38.8*	75.9	78.6	80.7
	Linear probe	**66.5**	79.6	81.5	82.4	**54.7**	77.1	79.8	81.1*
	MAP^{wd} + FroFA	63.9	**80.4**	**82.0**	**83.6**	39.3*	**78.0**	**80.0**	**81.2***

Table 9. **Average top-1 accuracy for JFT-3B and ImageNet-21k ViTs** on *our* ILSVRC-2012 test set trained on few-shotted ILSVRC-2012 training sets, complementing Fig. 4. We report results for the weight-decayed MAP, *i.e*. MAP^{wd}, and L2-regularized linear probe baseline, as well as our best FroFA-based approach, *i.e*., weight-decayed MAP combined with brightness c^{2}FroFA (MAP^{wd} + FroFA). The best results per shot are boldfaced. Each shot is sampled five times and an asterisk (*) indicates that the improvement of 'MAP^{wd} + FroFA' to 'MAP^{wd}'or 'linear probe' is not statistically significant under a two-tailed t-test with 95% confidence.

Shots	MAP	Bc^2	RA* $\mathcal{A}_{\text{top2}}$	RA* $\mathcal{A}_{\text{top3}}$	TA* $\mathcal{A}_{\text{top2}}$	TA* $\mathcal{A}_{\text{top3}}$
1	58.4	+**6.0**	+3.9	+2.4	+4.8	+4.3
5	79.1	+**1.5**	+1.0	+0.4	+**1.4**	+1.2
10	80.7	+**1.3**	+1.0	+0.6	+**1.4**	+1.4
25	83.0	+**0.6**	+**0.4**	+0.0	+**0.5**	+0.4

Table 10. **Top-1 accuracy for advanced FroFA protocols** on *our* ILSVRC-2012 test set. Absolute gains to the MAP baseline (reference run) are reported. We use the JFT-3B L/16 base setup (*cf*. Sec. 5). We compare brightness c^{2}FroFA (Bc^2) with our variations of RandAugment (RA*) and TrivialAugment (TA*), *cf*. Sec. S2.2. For the latter, we either use the top-2 ($\mathcal{A}_{\text{top2}}$) or top-3 ($\mathcal{A}_{\text{top3}}$) augmentations. The best results per shot are boldfaced (multiple ones if close, *i.e*., $\pm$0.2).

3B and subsequently finetuned on few-shotted ILSVRC-2012 training sets. Numbers for the two baselines, *i.e*., weight-decayed MAP (MAP^{wd}) and L2-regularized linear probe, and our best method, *i.e*., MAP^{wd} combined with brightness c^{2}FroFA (MAP^{wd} + FroFA), are reported. As before, we observe that linear probe is particularly strong on 1-shot while our method is *on par or favorable to MAP^{wd} and linear probe* on 5- to 25-shot settings.

S3.4. Results for Seven Other Few-Shot Datasets

In Fig. 1 and Tab. 5 we report mean results across seven few-shot datasets for 'MAP^{wd}', 'linear probe', and 'MAP^{wd} + FroFA' using frozen features from a JFT-3B or WebLI-SigLIP L/16 ViT. In Tabs. 11 and 12 we complement these with exact numbers for each dataset and shot.

We first look at JFT-3B results (Tab. 11). Similar to Tab. 5 (upper half) and Fig. 1 (left), we observe that on average our method, *i.e*., 'MAP^{wd} + FroFA', significantly surpasses both MAP^{wd} and linear probe across all shots. A closer look at the individual datasets reveals that in some settings linear probe is the best (*e.g*., SUN397, 1-shot). Further, DMLab seems to show not a clear trend. However, in most settings we observe that 'MAP^{wd} + FroFA' is either better or at least maintains the performance. In general, a similar observation can be made on the WebLI-SigLIP setting (*cf*. Tab. 12). For example, DMLab seems to be a clear outlier since MAP^{wd} and 'MAP^{wd} + FroFA' more or less perform on par, except for 25-shot. Overall, we observe that 'MAP^{wd} + FroFA' is either better or at least maintains the performance.

S3.5. Reducing the Hyperparameter Sweep

Across all experiments, we first tune our baseline extensively on a designated validation set to get the best possible accuracy and then report results on the respective test set. We apply the same protocol to tune FroFA for a fair comparison. However, since our hyperparameter sweeps are considerably large, it might raise concerns of overtuning the hyperparameters. To address this potential concern, we measure the sensitivity of our hyperparameter sweep by repeating the experiment series from Tab. 11 with a smaller sweep of 8 instead of 100 configurations: two batch sizes (32 and 512), two learning rates (0.01 and 0.03), and two training step settings (1,000 and 16,000). The absolute improvements over the MAP baseline averaged across the seven datasets from Tab. 11 are 3.7%, 3.7%, 3.2%, and 2.6% in the 1-, 5-, 10-, and 25-shot, respectively. Thus, our improvements remain consistent even with this much smaller hyperparameter sweep. We did not use weight decay in these experiments but expect a similar conclusion if weight decay is enabled.

S3.6. Comparison to Input Data Augmentations

In the following, we focus on a comparison between input data augmentations (IDAs) and frozen feature aug-

Trans. dataset	Method	1-shot	5-shot	10-shot	25-shot
CIFAR10	MAP^{wd}	81.6	97.0	97.1	97.5
	Linear probe	80.9	94.1	96.7	97.3
	MAP^{wd} + FroFA	**89.7**	**97.4**	**97.7**	**97.8**
CIFAR100	MAP^{wd}	63.4	82.9	85.4	86.7
	Linear probe	58.4	80.9	83.8	85.1
	MAP^{wd} + FroFA	**67.3**	**84.1**	**86.1**	**86.9**
DMLab	MAP^{wd}	24.3	**28.8**	27.5*	**35.7***
	Linear probe	24.0	26.3	25.6	30.9
	MAP^{wd} + FroFA	**25.4**	27.2	**27.8***	35.6*
DTD	MAP^{wd}	47.5	68.6	74.0	80.7
	Linear probe	46.9	65.9	71.3	77.3
	MAP^{wd} + FroFA	**53.0**	**70.8**	**75.3**	**81.7**
Resisc45	MAP^{wd}	61.6	86.7*	89.1*	91.0*
	Linear probe	67.1	85.6	88.2	91.0
	MAP^{wd} + FroFA	**66.0**	**87.0***	**89.4***	**91.1***
SUN397	MAP^{wd}	51.3	74.0	77.5	80.6
	Linear probe	**56.7**	70.9	75.6	78.6
	MAP^{wd} + FroFA	56.3	**75.6**	**78.9**	**81.2**
SVHN	MAP^{wd}	**16.9***	22.9	27.2	46.2
	Linear probe	11.8	15.0	18.7	21.5
	MAP^{wd} + FroFA	16.4*	**29.0**	**40.9**	**50.0**
Mean	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**

Table 11. **Average top-1 accuracy of our best FroFA combined with weight decay for seven transfer datasets using a JFT-3B L/16 ViT**, complementing Fig. 1 (left) and Tab. 5 (upper half). Results are reported on the respective test set (*cf*. Tab. 6). We compare results to a weight-decayed MAP baseline, *i.e.*, MAP^{wd}, and an L2-regularized linear probe. Per shot and dataset, the best result is boldfaced. We run 'MAP^{wd}' and 'MAP^{wd} + FroFA' experiments with five seeds. An asterisk (*) indicates that the improvement of 'MAP^{wd} + FroFA' to 'MAP^{wd}' is not statistically significant under a two-tailed t-test with 95% confidence.

Trans. dataset	Method	1-shot	5-shot	10-shot	25-shot
CIFAR10	MAP^{wd}	71.7	88.7	91.4	93.6
	Linear probe	74.4	88.2	91.5	93.5
	MAP^{wd} + FroFA	**77.9**	**92.6**	**93.4**	**94.2**
CIFAR100	MAP^{wd}	45.1	73.2	75.3	78.7
	Linear probe	52.5	72.4	76.7	77.7
	MAP^{wd} + FroFA	**55.5**	**74.6**	**77.4**	**79.2**
DMLab	MAP^{wd}	**23.3***	**28.1***	29.0*	**35.4**
	Linear probe	21.9	25.5	27.7	30.7
	MAP^{wd} + FroFA	22.6*	25.9*	**29.6***	34.0
DTD	MAP^{wd}	52.7	71.7	77.6	82.9
	Linear probe	50.6	70.6	76.5	81.8
	MAP^{wd} + FroFA	**59.4**	**76.1**	**80.0**	**84.1**
Resisc45	MAP^{wd}	65.2*	83.7	91.0*	92.6
	Linear probe	**70.5**	86.4	89.4	92.2
	MAP^{wd} + FroFA	65.1*	**87.2**	**91.1***	**93.0**
SUN397	MAP^{wd}	42.0	69.5	75.7	79.4
	Linear probe	**50.1**	68.7	74.2	77.4
	MAP^{wd} + FroFA	42.6	**73.9**	**77.3**	**79.9**
SVHN	MAP^{wd}	21.6	58.7	62.7	62.8
	Linear probe	23.5	43.3	48.8	54.6
	MAP^{wd} + FroFA	**36.3**	**62.3**	**65.6**	**67.5**
Mean	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 12. **Average top-1 accuracy of our best FroFA combined with weight decay for all transfer datasets using a WebLI-SigLIP ViT**, complementing Fig. 1 (right) and Tab. 5 (lower half). Results are reported on the respective test set (*cf*. Tab. 6). We compare results to a weight-decayed MAP baseline, *i.e.*, MAP^{wd}, and an L2-regularized linear probe. Per shot and dataset, the best result is boldfaced. We run 'MAP^{wd}' and 'MAP^{wd} + FroFA' experiments with five seeds. An asterisk (*) indicates that the improvement of 'MAP^{wd} + FroFA' to 'MAP^{wd}' is not statistically significant under a two-tailed t-test with 95% confidence.

mentations (FroFAs). As a prerequisite, we first compare the memory requirements of IDAs to FroFAs in a cached-feature setup.

Let $\mathcal{D}$ be a dataset with D images where a cached frozen feature requires memory of size M. Training a model for T epochs on N different IDAs and K different augmentation settings requires $D \times M \times T \times N \times K$ memory, since we need to store *all variations* of the dataset. With FroFA, however, *a single copy* of the dataset is sufficient, since the augmentations are directly applied on the cached frozen features during training. Thus, FroFA is $T \times N \times K$ more efficient compared to IDA in a cached-feature setup.

Next, we evaluate two IDAs, brightness (base augmentation of our best FroFA) and RandAugment [13] (a popular IDA), using a hyperparameter sweep comparable to the brightness c^{2}FroFA sweep (without weight decay). In all our settings, we train the MAP head on the output of the last transformer block, *i.e.*, our standard cached-feature setup (*cf*. Fig. 2). We did not average across five runs and thus only report absolute gains with respect to a reference run. Across all setups, we observe a reduction in accuracy from brightness c^{2}FroFA (*cf*. Tab. 13). Notably, *performance drops by more than 5%* when applying brightness or RandAugment IDA on ILSVRC-2012, 10-shot. This aligns with prior work [23] showing poorer pretrained network performance on diverse augmented images. In summary, we observe that *FroFA strongly outperforms IDAs* in a cached-feature setup.

S3.7. Additional FroFA Techniques

We extend our investigations in Tab. 2 with uniform noise and channel dropout FroFAs (details in Tab. 8) and show the absolute improvements in accuracy to our best FroFA, *i.e.*, brightness c^{2}FroFA, in Tab. 14. We did not average across five runs and thus only report absolute gains with respect to a reference run. While channel dropout performs compa-

Dataset	IDA	1-shot	5-shot	10-shot	25-shot
Mean across 7 SUN397	Brightness	−5.6	−0.7	−0.7	−0.3
	RandAugment	−6.2	−4.6	−3.6	−2.1
ILSVRC-2012	Brightness	−14.1	−9.7	−6.7	−5.2
	RandAugment	−14.2	−10.1	−6.9	−4.5

Table 13. **Ablation on input data augmentations (IDAs)**. We report absolute gains in top-1 accuracy (in %) on *our* ILSVRC-2012 test set w.r.t. our best FroFA setting (Tab. 3, brightness c^{2}FroFA) using a JFT-3B L/16 ViT. Negative numbers indicate that our proposed approach, *i.e.*, brightness c^{2}FroFA, is better. 'Mean across 7' incorporates all few-shot datasets, except ILSVRC-2012.

Dataset	FroFA	1-shot	5-shot	10-shot	25-shot
ILSVRC-2012	Uniform noise	−4.5	−2.3	−1.5	−1.0
	Channel dropout	−4.5	−1.1	−0.8	0.0

Table 14. **Ablation on additional frozen feature augmentations (FroFAs)**. We report absolute gains in top-1 accuracy (in %) on *our* ILSVRC-2012 test set w.r.t. our best FroFA setting (Tab. 3, brightness c^{2}FroFA) using a JFT-3B L/16 ViT. Negative numbers indicate that our proposed approach, *i.e.*, brightness c^{2}FroFA, is better.

rable to brightness c^{2}FroFA on 25-shot, in all other setups, channel dropout and uniform noise perform worse with performance drops ranging from 0.8% to 4.5% absolute.

S4. Final Remarks

We would like to thank the reviewers for suggesting to provide additional comparisons to input data augmentations, statistical significance tests, more details on the hyperparameter sweep, additional feature augmentation techniques, and a discussion on a few missing related works. The main paper already shows a clear tendency of frozen feature augmentations in a cached-feature setup. The additional experiments carried out in the Supplementary further highlight this tendency which makes our case even stronger.