Supplementary: Self-Supervised Learning of Video-Induced Visual Invariances

Michael Tschannen Josip Djolonga Marvin Ritter Aravindh Mahendran
Neil Houlsby Sylvain Gelly Mario Lucic

Google Research, Brain Team

The supplementary material discusses details of our model architecture in Appendix A. We specify training details for our model in Appendix B. In Appendix C we provide details regarding how the baseline methods (prior work) were evaluated on the VTAB. Additional per-dataset results contrasting various methods as well as an evaluation of the proposed framework as a pre-training step for object detection are provided in Appendix D. Lastly, an evaluation of our methods and the baselines on Version 2 (arXiv:1910.04867v2) of the VTAB can be found in Appendix E.

		Caltech101	CIFAR-100	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DM-Lab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	Mean
1000	MS	50.4	17.2	34.9	34.7	18.7	80.7	7.0	79.7	90.4	45.5	73.6	45.0	56.9	34.8	60.6	77.8	46.6	48.6	35.3	49.4
	TI	51.6	13.4	37.7	15.5	31.1	78.8	7.2	83.2	85.7	33.6	74.3	61.2	63.9	33.1	61.6	97.4	60.4	36.5	26.1	50.1
	Jigsaw	66.7	18.9	51.4	66.1	37.5	55.1	12.1	76.0	91.5	66.2	72.4	42.8	55.9	30.5	68.2	69.5	35.0	44.9	36.3	52.5
	Rel.Pat.Loc	68.5	19.1	52.2	69.0	41.3	60.9	11.1	77.5	92.6	65.4	70.7	43.5	59.6	33.6	68.2	70.7	29.3	47.2	35.2	53.5
	Rot-YT-F	70.2	21.5	48.0	48.0	35.7	88.3	8.4	83.4	93.0	61.7	73.6	48.1	57.9	39.3	73.4	90.1	51.2	50.5	38.3	56.9
	VIVI-Rot(2)	73.8	27.0	51.2	54.4	35.7	88.2	11.6	77.7	93.1	68.6	73.6	44.0	58.0	39.1	72.5	80.0	49.8	53.0	38.8	57.4
	VIVI-Rot(4)	73.4	25.8	52.0	53.1	42.2	88.5	10.3	84.1	93.7	66.3	73.6	49.9	58.3	38.7	73.6	88.9	52.4	52.8	40.6	58.9
	Rot-YT-F-AA	77.8	24.0	51.2	56.1	33.3	89.1	10.0	85.1	93.9	66.5	73.6	48.8	59.2	39.0	71.2	92.4	55.1	54.6	38.7	58.9
	Ex-YT-F	73.0	24.3	49.8	64.6	48.1	88.4	13.4	83.0	95.5	70.4	73.6	50.4	59.0	38.6	71.0	90.8	46.9	43.5	44.7	59.4
	Ex-ImageNet	72.0	20.1	52.8	54.5	51.0	87.5	15.5	83.8	95.2	72.5	74.2	49.9	60.9	36.9	75.6	92.2	45.6	48.8	41.7	59.5
	MT-SSL	76.1	27.1	52.9	63.2	48.2	89.6	13.6	81.5	93.5	71.0	73.6	56.6	59.4	37.3	72.9	94.7	47.4	52.3	40.7	60.6
	Rot-ImageNet	80.6	25.9	56.5	72.6	47.1	88.6	16.0	81.9	94.2	69.8	73.6	49.1	58.6	37.9	73.1	92.6	50.4	51.6	37.7	60.9
	Ex-YT-S	76.2	28.4	50.4	74.9	53.1	88.0	14.3	81.7	94.8	74.2	73.7	52.8	58.9	39.3	70.9	91.1	50.8	49.7	42.1	61.3
	Ex-YT-F-AA	78.7	28.1	54.6	64.7	52.7	89.0	16.5	83.5	95.5	73.1	73.6	52.2	60.3	39.0	74.4	93.4	54.6	45.9	44.5	61.8
	VIVI-Ex(2)-Ord	76.0	29.0	49.0	77.7	54.7	88.5	13.6	80.5	94.2	73.2	73.6	55.9	60.2	39.1	72.0	91.5	52.1	51.5	42.8	61.9
	VIVI-Ex(2)	75.3	28.8	48.7	77.5	55.5	87.9	12.4	81.6	94.1	73.6	73.6	56.3	60.6	38.9	73.1	91.9	52.2	50.6	44.6	62.0
	VIVI-Ex(4)	76.3	29.0	50.1	77.9	55.6	88.0	14.1	82.4	94.4	73.1	73.6	55.3	60.9	38.6	72.9	95.3	53.0	52.4	44.1	62.5
	Ex-YT-S-AA	79.3	30.1	53.9	75.4	55.3	88.4	14.7	83.4	94.8	75.7	73.6	55.3	59.5	40.7	76.6	91.3	52.9	51.2	41.4	62.8
	VIVI-Ex(4)-AA	78.6	30.3	51.5	75.0	56.1	88.6	14.4	83.0	94.7	75.2	73.6	56.3	60.6	41.6	74.2	94.6	55.5	52.3	41.0	63.0
	VIVI-Ex(4)-Big	77.5	32.8	51.3	79.4	56.6	88.3	16.6	79.8	95.1	75.3	73.6	54.7	57.9	40.4	74.4	92.0	56.8	52.4	47.0	63.3
	VIVI-Ex(4)-Big-AA	77.5	34.8	54.2	76.9	59.5	89.7	16.2	84.3	94.8	77.2	73.6	53.3	60.7	40.5	78.0	93.4	59.2	52.9	47.0	64.4
	Semi-Ex-10%	88.6	53.2	60.8	86.8	85.3	88.0	29.0	83.2	95.2	77.3	71.7	42.3	57.4	36.7	71.4	74.9	53.9	52.7	32.3	65.3
	Sup-100%	91.0	57.0	66.0	88.6	89.9	87.3	34.4	80.6	95.3	80.8	73.2	41.0	56.1	36.3	70.6	85.7	46.0	45.7	35.4	66.4
	VIVI-Ex(4)-Co(10%)	82.8	36.6	58.1	82.7	76.9	81.9	24.1	85.6	94.7	76.4	73.6	79.4	63.9	38.0	76.6	95.3	61.3	42.4	46.3	67.2
	Sup-Rot-100%	91.7	53.7	69.5	90.8	88.1	88.5	32.8	83.4	96.0	82.0	71.1	47.3	57.2	36.6	77.1	88.3	52.1	51.6	33.7	68.0
	VIVI-Ex(4)-Co(100%)	86.1	51.5	64.5	88.7	87.1	79.4	31.7	83.9	95.1	80.8	73.6	78.9	61.7	36.4	78.2	93.8	61.0	43.1	43.6	69.4
	VIVI-Ex(4)-Co(100%)-Big	88.0	53.3	69.0	90.4	88.4	84.4	34.1	86.2	95.9	81.7	73.6	79.9	63.5	37.3	82.9	95.3	67.4	46.2	44.9	71.7
full	MS	68.4	69.6	48.1	52.7	49.2	96.7	56.9	85.5	97.5	88.3	76.8	99.8	90.4	71.7	75.3	100.0	96.3	99.9	97.4	80.0
	TI	76.5	68.5	56.4	66.3	52.0	96.2	59.4	89.8	97.6	90.1	81.0	94.0	91.6	72.3	61.2	100.0	96.4	97.0	86.2	80.7
	Jigsaw	79.1	65.3	63.9	77.9	65.4	93.9	59.2	83.0	97.9	92.0	80.1	99.6	88.6	72.0	74.7	100.0	90.3	99.9	93.6	83.0
	Rel.Pat.Loc	79.9	65.7	65.2	78.8	66.8	93.7	58.0	85.3	97.8	91.5	79.8	99.5	87.7	71.5	75.0	100.0	90.4	99.7	92.6	83.1
	Rot-YT-F	81.8	72.6	60.7	66.5	65.7	96.9	59.4	86.7	98.3	92.2	76.8	99.8	92.1	76.0	81.3	100.0	96.6	99.8	98.0	84.3
	VIVI-Rot(4)	87.1	74.2	62.4	73.5	68.6	97.0	61.1	86.8	98.3	92.8	76.9	99.8	92.1	76.3	79.1	100.0	96.5	100.0	97.7	85.3
	VIVI-Rot(2)	86.7	74.1	61.6	75.1	67.6	97.0	61.9	86.7	98.4	92.6	77.7	99.8	92.5	76.4	81.3	100.0	96.6	99.9	97.1	85.4
	Rot-YT-F-AA	86.8	72.5	63.0	74.7	68.4	96.9	60.1	86.4	98.4	92.8	78.5	99.8	92.2	76.4	81.5	100.0	96.6	99.7	98.1	85.4
	Ex-YT-F	85.0	73.6	63.8	84.9	70.5	96.8	60.6	87.2	98.6	94.3	78.9	99.8	93.3	76.8	80.9	100.0	96.6	99.9	97.3	86.2
	MT-SSL	88.0	76.1	64.4	80.0	72.3	97.2	63.0	85.8	98.3	93.7	78.6	99.7	93.0	75.4	80.4	100.0	96.5	100.0	98.1	86.3
	Ex-ImageNet	83.5	74.2	65.4	83.4	74.9	96.8	60.4	85.5	98.7	94.5	79.8	99.8	93.5	75.5	80.4	100.0	96.5	99.9	98.0	86.4
	Rot-ImageNet	88.5	76.4	67.7	83.0	73.1	97.0	63.2	85.4	98.5	93.9	79.1	99.9	92.2	76.0	82.0	100.0	96.6	100.0	98.3	86.9
	VIVI-Ex(2)-Ord	86.0	75.7	62.1	87.1	76.1	96.9	63.7	87.2	98.6	94.6	79.9	99.8	93.5	76.5	80.9	100.0	96.5	99.8	97.9	87.0
	Ex-YT-S	87.4	75.9	64.8	85.7	75.0	96.9	63.2	87.0	98.6	94.5	80.1	99.8	93.4	77.4	80.4	100.0	96.6	99.9	97.3	87.1
	VIVI-Ex(4)	86.1	76.3	61.8	87.3	76.7	97.0	64.0	86.9	98.6	94.7	80.2	99.8	93.5	76.8	81.3	100.0	96.6	99.8	98.2	87.1
	Ex-YT-F-AA	88.1	75.1	67.7	86.1	73.5	96.9	62.2	86.8	98.8	94.6	79.0	99.9	93.5	76.5	82.9	100.0	96.6	99.9	97.9	87.2
	VIVI-Ex(2)	86.6	76.1	63.4	88.2	74.4	97.0	64.1	88.4	98.6	94.7	79.2	99.8	93.4	77.1	80.9	100.0	96.5	99.9	97.6	87.2
	Ex-YT-S-AA	89.0	76.5	67.3	86.2	75.9	97.0	63.6	86.9	98.8	94.6	80.3	99.8	93.3	77.1	82.0	100.0	96.6	99.9	97.6	87.5
	VIVI-Ex(4)-AA	88.8	76.8	64.0	87.1	75.9	97.2	63.9	88.6	98.6	94.5	79.5	99.8	93.2	76.7	84.0	100.0	96.6	99.8	97.6	87.5
	VIVI-Ex(4)-Co(10%)	89.3	79.1	67.6	89.1	83.2	96.9	66.5	90.1	98.4	93.0	79.6	99.5	92.1	74.8	83.1	100.0	96.5	99.8	93.6	88.0
	VIVI-Ex(4)-Big	89.1	79.4	64.7	89.6	78.7	97.1	69.2	86.9	98.6	95.6	80.2	99.8	93.6	77.2	81.8	100.0	96.6	99.9	98.6	88.3
	VIVI-Ex(4)-Big-AA	90.5	80.4	68.5	87.5	78.3	97.3	68.7	88.7	98.7	95.3	80.5	99.9	92.8	77.8	81.0	100.0	96.7	100.0	98.1	88.5
	Semi-Ex-10%	85.3	82.7	70.5	92.2	89.0	97.0	67.4	86.0	98.6	94.7	78.8	99.8	93.1	76.8	81.5	100.0	96.5	100.0	97.8	88.8
	VIVI-Ex(4)-Co(100%)	92.5	82.0	73.2	92.7	90.9	96.8	70.7	87.4	98.5	93.7	80.2	99.4	91.2	73.4	82.1	100.0	96.5	98.9	96.5	89.3
	Sup-100%	94.1	83.8	74.0	93.2	91.9	97.0	70.7	83.9	98.8	95.3	79.3	99.8	92.1	76.4	80.7	100.0	96.4	99.8	97.7	89.7
	Sup-Rot-100%	94.6	84.8	75.9	94.7	91.5	97.0	70.2	85.9	98.8	94.9	79.5	99.8	92.5	76.5	82.3	100.0	96.5	100.0	98.4	90.2
	VIVI-Ex(4)-Co(100%)-Big	93.5	85.9	77.2	94.4	91.6	97.3	73.7	89.4	98.8	95.1	81.0	99.7	92.5	76.7	84.8	100.0	96.6	99.7	94.6	90.7

Table 4: Testing accuracy for every data set in the VTAB benchmark using 1000 and all samples for fine-tuning. Each number is the median of three fine-tuning runs. The proposed methods have the prefix Video-Induced Visual Invariances (VIVI). "Ex" and "Rot" stand for exemplar [3] and rotation prediction [4] frame-level self-supervision, respectively. These identifiers are followed with the number of shots in parentheses if an InfoNCE prediction loss across shots is used (except methods using shot order prediction have the suffix "-Ord"). Baseline methods only using frames and shots have the suffix "YT-F" and "YT-S", respectively. The suffix "-AA" denotes methods that use AutoAugment [1].

A. Architectures

Here we expand on the short description in Section 4. The frame encoder f is modelled using the ResNet-50 v2 [5] architecture with BatchNorm [6]. We also investigate in several experiments the effect of model capacity by widening the network by a factor of three. To avoid mismatch in batch statistics between the two data sources, in the co-training experiments we replace the BatchNorm with GroupNorm [14] and also standardize [10] the weights of the convolutions.

For each prediction task, we attach a different linear head to the 2048-dimensional pre-logits ResNet representation before applying the respective loss or prediction function. For exemplar, following [7], we use a linear head with 1000 outputs with L2-normalization of the features before feeding into the triplet-loss. For rotation prediction we rely on a linear head with 4 outputs. For the video-level loss (prediction across shots using $\mathcal{L}_{\text{NCE}}$ and temporal order prediction) we project the pre-logits, average-pooled across the frames of the same shot, to 512 dimensions using a linear head, and feed this representation to the prediction functions g_m. Finally, in the experiments with co-training, we rely on an additional linear classification head with 1000 outputs.

For the $\mathcal{L}_{\text{NCE}}$ loss, when we sample 2 shots, we predict one from the other using an multilayer perceptron (MLP), i.e., the function g in (2) has the form $g(e, e') = \phi_1(e)^\top \phi_2(e')$, where ϕ_1, ϕ_2 are MLPs with a single hidden layer with 256 units and 128 outputs. In the experiments with 4 shots, we use a 2-layer Long Short-Term Memory (LSTM) prediction function with 256 hidden units to predict every shot embedding from the previous ones. To match the dimension of the LSTM output (256) and that of the future shot embeddings (512) we employ another linear layer. We use temporal order prediction only together with exemplar-based self-supervised learning (SSL) and for data with 2 shots per video, relying on a single-hidden-layer MLP with 512 hidden units as prediction function.

For both frame and shot-level SSL approaches we use the augmentation mechanism from [12]. For models co-trained with a supervised loss based on a fraction of ImageNet we additionally use the same HSV-space color randomization as [15]. We also perform experiments where we replace the augmentation mechanism from [12] with AutoAugment (AA), which is an augmentation policy learned using a reinforcement learning algorithm from the full ImageNet data set. More specifically, we rely on the TF-Hub module publicly available at https://tfhub.dev/google/image_augmentation/nas_imagenet/1.

B. Training details

Table 6 provides details about the schedules, batch size, loss weights, etc. used for the individual methods. When exploring the effect of AA we reduce the weight of the video-level loss, λ, by a factor of 2. The schedule for VIVI-Ex(4)-Co(10%) is motivated as follows. We take the schedule and batch size used for the ImageNet exemplar co-training experiments for 10% labeled ImageNet examples from [15], stretch the schedule to 100k iterations and reduce the batch size (as well as the learning rate) so that number of epochs over the 10% (128k example) data set matches that of [15]. Table 5 shows some statistics of the YouTube-8M (YT8M) subset we use for training.

We set the margin parameter in the semi-hard triplet loss [11] to 0.5. For rotation-based SSL, following common practice [4, 7], we compute the predicted rotation after appending to the mini-batch 3 rotated copies of the mini-batch along the batch dimension and compute the rotation loss for all rotated copies.

We train all models on 128 cores of a Google TPU v3 Pod. For exemplar SSL the triplet loss is computed per core. For all frame/shot level loss variants, $\mathcal{L}_{\text{NCE}}$ is computed across all cores when prediction is across 4 shots, and computed per core when prediction is across 2 shots as computing the loss across all cores led to instabilities for that case.

	Mean	Std.
Number of shots per video	25.5	30.3
Shot duration	9.0 sec.	25.3 sec.
Video duration	230.7 sec.	61.7 sec.

Table 5: Statistics of the YT8M subset we use for training.

C. Baseline fine-tuning details

As mentioned in the main manuscript we compared against two baseline methods: MT-SSL (Multi-Task Self-Supervised Learning) [2], and TI (Transitive Invariance) [13]. For MT-SSL we considered two variants: MS which was pre-trained on

	LR	#it.	w. #it.	LR schedule	WD	λ	γ	batch size	#exemp.
Ex-ImageNet	0.8	120k	17k	$\times$0.1@52k;86k	10^{-4}	-	-	2048	8
Ex-YT-F	0.8	120k	17k	$\times$0.1@52k;86k	10^{-4}	-	-	2048	8
Ex-YT-S	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	-	-	2048	8 (sh.)
VIVI-Ex(2)-Ord	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	$\{2.0, 1.0, \underline{0.5}\}$	-	$1024 \cdot 2$ sh.	8 (sh.)
VIVI-Ex(2)	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	$\{0.08, 0.04, \underline{0.02}\}$	-	$1024 \cdot 2$ sh.	8 (sh.)
VIVI-Ex(4)	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	$\{\underline{0.04}, 0.02, 0.01\}$	-	$512 \cdot 4$ sh.	8 (sh.)
VIV-Ex(4)-Big	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	0.04	-	$512 \cdot 4$ sh.	8 (sh.)
VIVI-Ex(4)-Co(10%)	0.1	100k	3k	$\times$0.1@76k;88k;96k	10^{-3}	0.04	$\{1.0, \underline{4.0}, 8.0, 16.0\}$	$512 \cdot 4$ sh., 256 im.	8 (sh.)
VIVI-Ex(4)-Co(100%)	0.8	100k	3k	$\times$0.1@70k;85k;95k	10^{-4}	0.04	$\{0.1, 0.5, \underline{1.0}, 5.0\}$	$512 \cdot 4$ sh., 2048 im.	8 (sh.)
VIVI-Ex(4)-Co(100%)-Big	0.8	100k	3k	$\times$0.1@70k;85k;95k	10^{-4}	0.04	$\{0.1, 0.5, 1.0, \underline{5.0}\}$	$512 \cdot 4$ sh., 2048 im.	8 (sh.)
Rot-ImageNet	0.8	120k	17k	$\times$0.1@52k;86k	10^{-4}	-	-	2048	1
Rot-YT-F	0.8	120k	17k	$\times$0.1@52k;86k	10^{-4}	-	-	2048	1
VIVI-Rot(2)	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	$\{0.2, 0.1, \underline{0.05}, 0.025\}$	-	$1024 \cdot 2$ sh.	4 (sh.)
VIVI-Rot(4)	0.8	120k	5k	$\times$0.1@90k;110k	10^{-4}	$\{0.32, 0.16, \underline{0.08}, 0.04\}$	-	$512 \cdot 4$ sh.	4 (sh.)

Table 6: Learning rate (LR), number of training iterations (#it.), number of linear warm-up iterations (w. #it.), learning rate schedule (LR schedule), weight decay (WD), video-level loss weight (λ), supervised cross-entropy loss weight (γ), batch size, and the number of exemplars (#exemp.) for the different models considered in this paper. Lists of values indicate values explored in the parameter sweep, with the optimal value (in terms of validation VTAB 1000 example score) underlined. For the co-training methods we indicate video (suffix "sh.") and image (suffix "im.") batch size. If the number of exemplars is followed by "(sh.)" we use consecutive frames of the same shot to create exemplars.

motion segmentation only, and MT-SSL which combined MS with three other tasks in a multi-task setting. We obtained pre-trained checkpoints for all three methods (MS, MT-SSL, and TI) from the authors of their respective prior works.

C.1. Fine-tuning motion segmentation and multi-task SSL baselines

MS and MT-SSL pre-trained a ResNet-101 up to block3. The representation at block3 is $7 \times 7 \times 1024$, which is too big. In [2], the authors used max-pooling to down-sample this to $3 \times 3 \times 1024$ and then trained a linear predictor for ImageNet classification. We experimented with this approach for VTAB evaluation. The default evaluation protocol for VTAB is to sweep over initial learning rates: 0.1 and 0.01. These were too high for the MS and MT-SSL models. For several downstream evaluation tasks fine-tuning diverged. We therefore modified the evaluation sweep minimally to sweep over initial learning rates: $0.1, 0.05, 0.01$. We also evaluated a simpler alternative: Global average pooling the block3 representation into a $1 \times 1 \times 1024$ dimensional vector. We found that global average pooling the representation achieved best results on the VTAB validation set. It also did not diverge at higher learning rates, so we could use the default learning rate schedule in this case. We therefore used this setting for the final evaluation on test data.

C.2. Fine-tuning the transitive invariance baseline

We exported the pre-trained caffe checkpoint into TensorFlow using the Caffe-TensorFlow tool[1]. We found that the pre-trained VGG-16 backbone diverges at higher learning rates when fine-tuning downstream on VTAB tasks. We therefore manually adjusted the sweep over initial learning rates and found $0.01, 0.005, 0.001$ to work well. Another challenge with transferring this baseline model to several downstream data sets was that it is a patch-based model that expects 96×96 dimensional input, whereas the VTAB benchmark scales all images to 224×224. We experimented with three ways of deploying this downstream: (a) Resize the input image from 224×224 into 96×96, (b) apply the model fully convolutionally and compute a global average pool at the end, and (c) crop patches of size 96×96 at stride 32 from the input image and then average the representations across all of these. We found that (c) was computationally extremely expensive. (b) performed best and we report results for that approach on the VTAB test set.

[1] https://github.com/ethereon/caffe-tensorflow

D. Additional results

Object detection We evaluate the proposed framework as a pre-training step for object detection. Specifically, we use different VIVI models and baselines as backbone for RetinaNet [8] and evaluate it on the COCO-2017 data set [9]. For all experiments, we rely on the standard ResNet-50 v2 architecture. We use the standard RetinaNet training protocol with fine-tuning and only adapt the learning rate in cases where the default learning rate is too high by halving it until the training loss decays smoothly (i.e., we tune the learning rate based on the training loss). Table 7 shows the standard COCO-2017 detection metric for different models. It can be seen that VIVI-Ex(4) outperforms the Ex-YT-F, MS, and MT-SSL baselines, and VIVI-Ex(4)-Co(100%) improves over supervised ImageNet pre-training (Sup-100%).

METHOD	AP
RetinaNet (Random init.)	26.9
RetinaNet (MS)	32.3
RetinaNet (Ex-YT-F)	32.6
RetinaNet (MT-SSL)	32.7
RetinaNet (VIVI-Ex(4))	33.6
RetinaNet (Sup-100%)	35.3
RetinaNet (VIVI-Ex(4)-Co(100%))	36.5

Table 7: Object detection performance of RetinaNet [8] on the COCO-2017 data set [9] for different ways of pre-training the ResNet-50 v2 backbone (Sup-100% corresponds to standard supervised ImageNet pre-training). The reported AP values are the median over 3 runs.

Additional figures In Fig. 5 to 9 we provide per-data set comparisons of different model pairs to better understand the effect of increasing the model size, using AA, and co-training with different amounts of labeled images. All numbers are accuracies when using 1000 labels for fine-tuning.

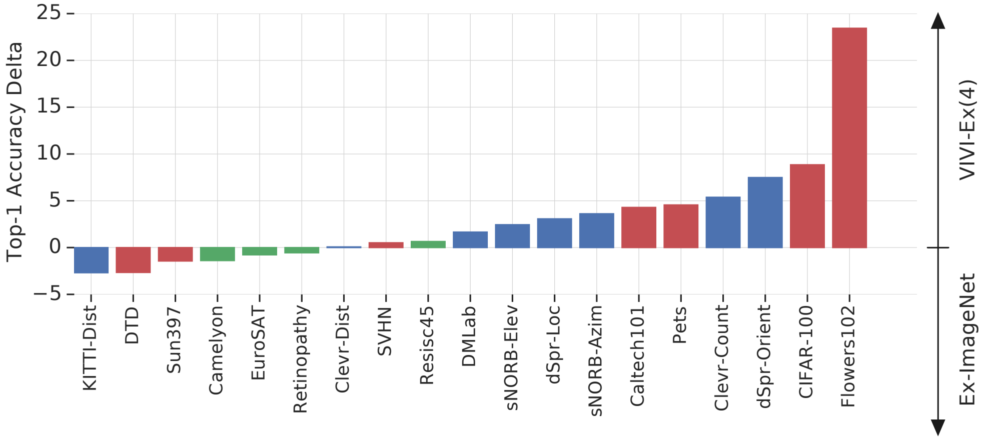

Figure 5: Per-data set comparison of ImageNet-based exemplar SSL (Ex-ImageNet) with VIVI-Ex(4). Training on YT8M rather than ImageNet and exploiting temporal information mostly helps on natural (red) and structured (blue) data sets, and slightly hurts for some specialized (green) data sets.

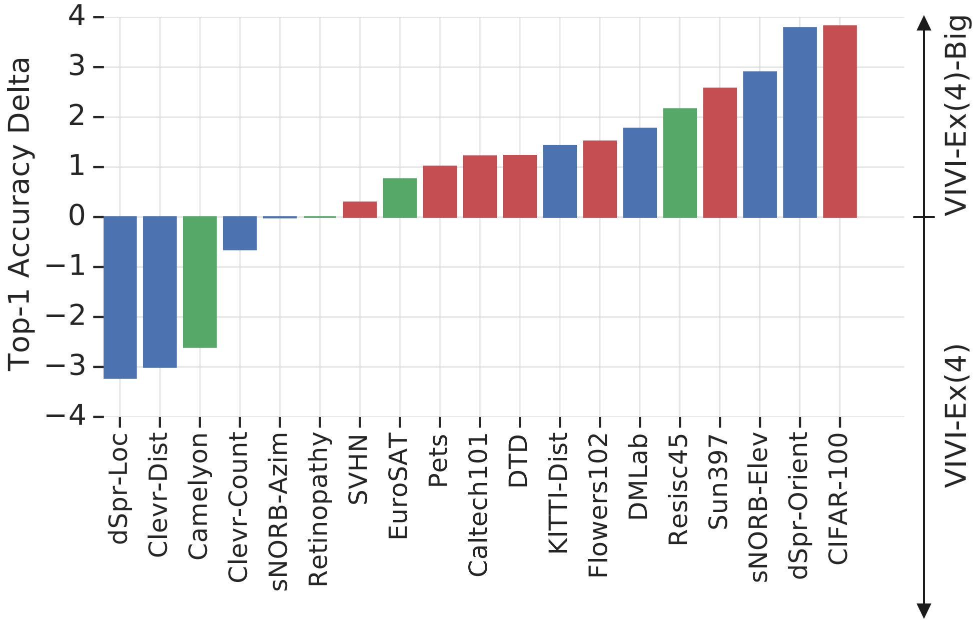

Figure 6: Per-data set comparison of VIVI-Ex(4) and a 3× wider counterpart (VIVI-Ex(4)-Big). Increasing model capacity leads to an increase in accuracy for all natural (red) data sets and some structured (blue) and specialized (green) data sets. However, some structured and specialized data sets also incur a reduction in accuracy.

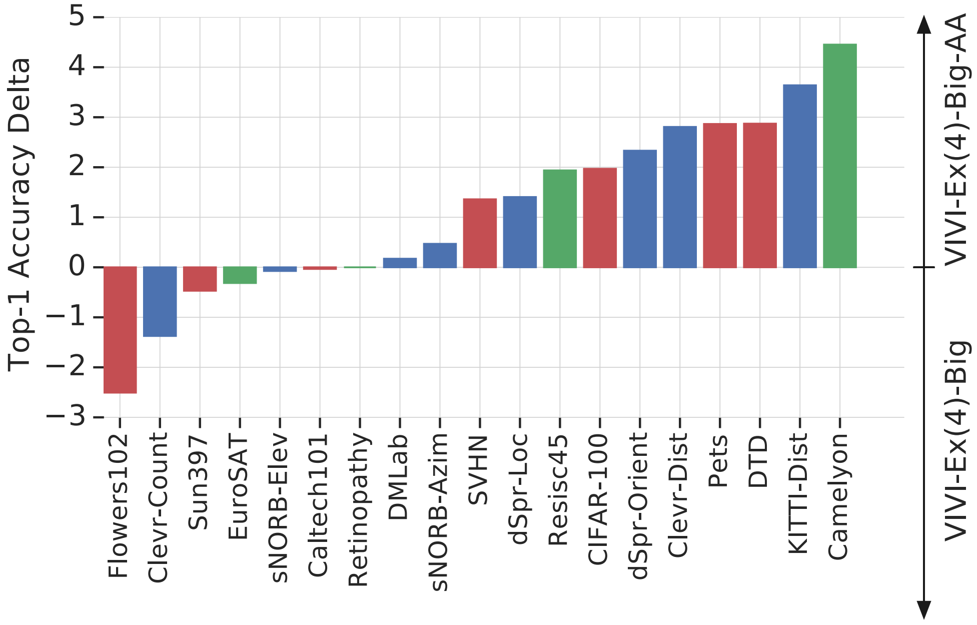

Figure 7: Per-data set comparison of VIVI-Ex(4) and a variant using AA. AA seems to benefit all data set categories similarly, and also leads to reductions in accuracy for a few data sets from all categories.

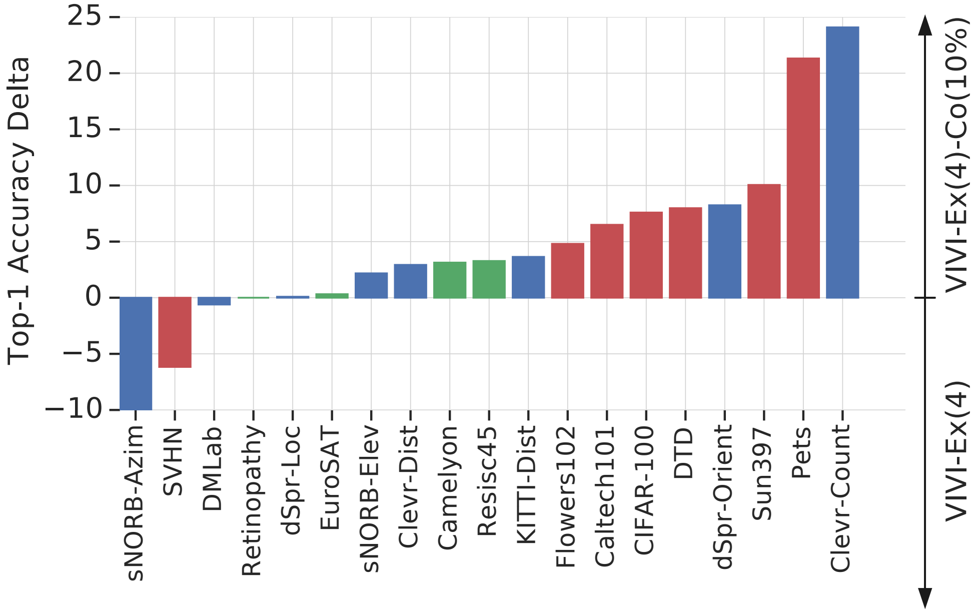

Figure 8: Per-data set comparison of VIVI-Ex(4) and its counterpart co-trained with 10% class-balanced ImageNet data (VIVI-Ex(4)-Co(10%)). Most data sets from each category incur an increase in accuracy, but one data set from each the natural and structured categories suffer a significant loss in accuracy.

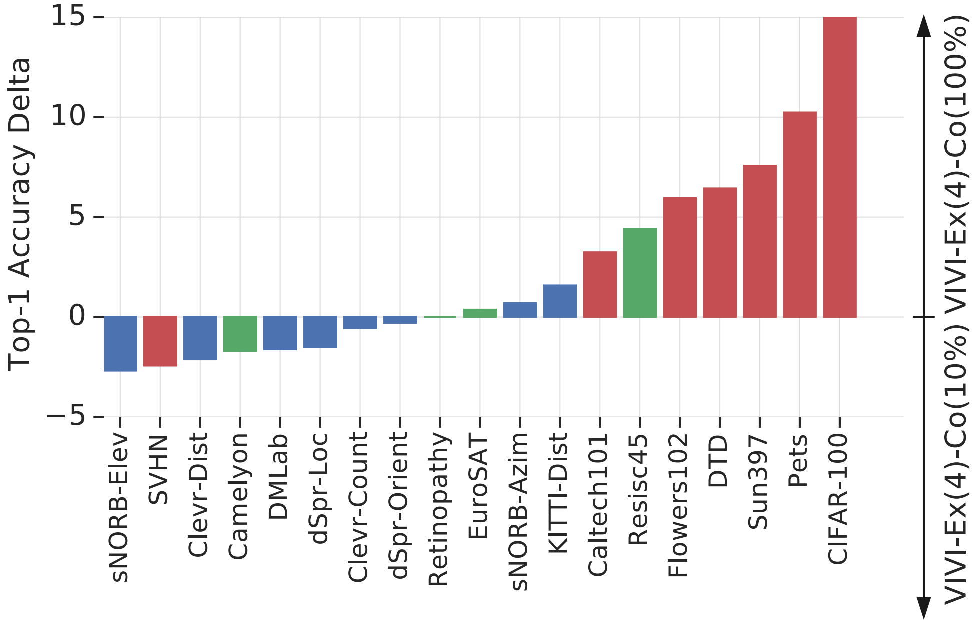

Figure 9: Effect of increasing the number of ImageNet images used for co-training from 10% (VIVI-Ex(4)-Co(10%)) to 100% (VIVI-Ex(4)-Co(100%)). The accuracy on the majority of natural (red) data sets is significantly increased, whereas most of the structured data sets incur a slight drop in accuracy.

E. Evaluation on VTAB Version 2

In Table 8 we report complete testing results for the 1000 example transfer regime of Version 2 (arXiv:1910.04867v2) of the VTAB benchmark [16]. Table 9 shows the mean testing accuracy per data set category, and Table 10 is the Version 2-analog of Table 1. Note that the full data set evaluation is the same in both Versions 1 and 2. The 1000 example regime of Version 2 uses 800 samples for fine-tuning and 200 samples for validation to select the hyper-parameters for fine-tuning (see Section 4; the considered set of fine-tuning hyper-parameters remains the same). In contrast, Version 1 uses all the 1000 samples for fine-tuning, and the standard validation set for each data set. We emphasize that we do not re-select the *training* hyper-parameters (such as the video-level loss weight, see Table 6), but only the hyper-parameters for the transfer step.

It can be seen that the relative improvements of our methods in terms of the VTAB 1000 example mean accuracy over the baselines obtained for Version 2 are comparable to Version 1, with few exceptions. Accordingly, the ranking of methods according to the mean accuracy remains mostly unchanged.

		Caltech101	CIFAR-100	DTD	Flowers102	Pets	SVHN	Sun397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DM-Lab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	Mean
1000	TI	54.9	7.1	38.3	28.2	32.3	77.0	7.4	50.0	84.1	50.0	63.1	12.7	61.7	35.0	41.6	86.1	59.3	21.1	29.2	44.2
	MS	52.3	12.7	37.3	32.6	15.8	81.8	6.8	76.8	89.7	49.7	57.3	43.2	55.7	38.4	48.4	81.2	46.4	34.8	35.1	47.1
	Rel.Pat.Loc	67.8	17.9	51.0	67.2	38.8	61.7	10.5	73.4	92.6	66.2	59.7	44.3	55.7	39.4	57.8	63.6	32.7	29.9	34.9	50.8
	Jigsaw	66.2	14.8	50.7	65.3	34.0	54.9	11.4	73.0	91.5	66.7	71.3	44.1	56.2	42.2	63.8	66.0	34.2	32.9	31.7	51.1
	Rot-YT-F	69.5	21.5	47.2	44.0	34.2	88.4	9.5	82.5	92.1	59.4	70.5	47.1	57.4	46.2	71.5	92.0	49.9	36.8	37.1	55.6
	VIVI-Rot(4)	72.7	24.6	48.6	48.9	39.1	88.8	11.0	83.1	93.5	65.9	71.2	49.0	58.6	46.5	70.4	89.1	51.7	24.7	39.8	56.7
	VIVI-Rot(2)	73.0	26.7	49.1	55.0	43.2	88.4	12.7	83.0	93.1	66.7	71.6	45.1	57.8	45.5	69.5	88.0	51.1	37.5	38.1	57.6
	Rot-YT-F-AA	77.1	23.0	50.8	55.3	35.2	88.6	9.4	81.9	93.4	64.5	71.4	42.8	58.6	46.9	71.7	91.4	53.9	40.7	39.8	57.7
	Ex-YT-F	72.2	23.9	51.9	50.1	44.7	88.1	15.3	82.5	95.3	69.7	70.6	47.6	59.5	45.7	72.3	90.3	46.8	32.6	44.7	58.1
	BigBiGAN	80.8	39.2	56.6	77.9	44.4	76.8	20.3	77.4	95.6	74.0	69.3	53.9	55.6	38.7	71.4	70.6	46.7	27.2	46.3	59.1
	Ex-ImageNet	72.7	23.2	54.5	68.7	48.7	88.4	14.4	80.3	95.5	73.8	74.1	44.9	60.2	45.7	69.6	89.3	45.8	34.2	40.7	59.2
	MT-SSL	76.2	26.2	49.3	63.5	48.5	89.1	10.6	80.3	93.3	70.2	71.7	55.6	62.1	44.3	76.3	86.6	43.2	39.1	38.9	59.2
	Rot-ImageNet	77.1	25.2	55.9	71.2	44.2	88.5	14.4	81.1	93.5	70.1	64.1	48.0	59.6	46.9	74.5	92.7	49.9	37.4	38.9	59.6
	Ex-YT-S	75.4	28.1	50.2	74.3	52.3	88.2	14.7	82.2	94.5	74.4	74.0	52.7	59.9	45.4	70.7	88.0	48.3	32.9	36.6	60.2
	VIVI-Ex(2)-TimeArrow	74.9	28.5	48.3	77.1	55.3	88.0	14.1	83.0	94.1	73.9	71.5	46.8	58.2	46.4	73.9	92.7	48.1	35.6	42.3	60.7
	VIVI-Ex(4)	75.0	29.3	49.0	77.8	55.3	87.9	15.1	80.3	94.2	73.9	70.4	55.2	58.8	46.4	70.9	91.3	51.4	34.9	42.1	61.0
	VIVI-Ex(2)	74.8	28.8	49.5	77.5	54.0	88.2	14.0	80.0	94.0	73.7	71.9	56.0	61.1	46.5	70.1	91.8	51.0	38.2	45.0	61.4
	Ex-YT-S-AA	78.0	29.4	50.7	75.3	55.3	89.7	14.0	83.9	94.9	76.3	71.7	52.3	58.7	46.5	72.2	91.0	52.6	35.3	39.9	61.5
	Sup-10%	84.9	46.1	55.4	82.8	80.2	86.9	24.9	80.6	95.2	73.1	71.9	39.4	55.7	43.5	63.3	76.0	45.9	30.9	32.9	61.6
	VIVI-Ex(4)-AA	78.6	30.5	50.9	74.8	55.8	88.8	14.5	81.2	94.9	75.5	71.8	54.3	61.0	45.7	76.3	93.0	54.1	31.7	39.7	61.7
	VIVI-Ex(4)-Big	76.5	33.0	51.6	75.2	57.1	87.5	17.1	80.1	94.7	75.1	72.4	55.4	57.1	46.6	73.3	89.4	55.1	35.6	44.1	61.9
	Ex-YT-F-AA	76.5	27.8	54.7	75.0	52.0	89.7	16.7	83.6	95.5	74.1	73.5	51.1	57.9	48.6	74.8	94.2	52.7	34.8	43.9	61.9
	VIVI-Ex(4)-Big-AA	79.7	35.1	56.0	72.3	56.9	89.5	18.0	82.0	95.2	77.2	71.6	54.7	60.3	48.7	75.8	92.4	58.5	35.6	46.2	63.5
	Semi-Ex-10%	87.7	52.8	60.4	84.0	84.0	87.1	29.2	79.1	94.9	77.1	70.1	39.4	56.0	42.3	72.0	73.7	52.3	37.6	33.5	63.9
	Sup-100%	89.8	54.6	65.6	88.4	89.1	86.3	34.5	79.7	95.3	81.0	72.6	41.8	52.5	42.7	75.3	81.0	47.3	32.6	35.8	65.6
	VIVI-Ex(4)-Co(10%)	82.7	36.5	57.9	82.0	76.6	82.2	24.0	84.7	94.8	76.8	73.2	75.5	60.5	46.7	77.4	95.0	58.3	30.5	45.3	66.3
	Sup-Rot-100%	89.9	52.8	68.6	90.3	88.8	88.7	32.5	80.5	95.9	83.4	73.2	48.2	57.0	48.5	79.1	92.5	50.0	30.6	32.8	67.5
	VIVI-Ex(4)-Co(100%)	86.7	51.6	64.8	88.2	86.3	80.4	32.5	84.3	95.1	80.9	72.6	78.0	60.4	45.4	80.2	92.9	61.7	30.2	41.8	69.2
	VIVI-Ex(4)-Co(100%)-Big	87.6	53.6	68.9	89.9	87.0	84.1	34.2	85.9	95.5	81.6	71.9	75.8	62.8	45.9	83.9	95.3	62.1	27.1	45.3	70.4

Table 8: Testing accuracy for fine-tuning hyper-parameter selection according to **Version 2 (arXiv:1910.04867v2) of the VTAB benchmark** [16]. Each number is the median of three fine-tuning runs. The proposed methods have the prefix VIVI. “Ex” and “Rot” stand for exemplar [3] and rotation prediction [4] frame-level self-supervision, respectively. These identifiers are followed with the number of shots in parentheses if an InfoNCE prediction loss across shots is used (except methods using shot order prediction have the suffix “-Ord”). Baseline methods only using frames and shots have the suffix “YT-F” and “YT-S”, respectively. The suffix “-AA” denotes methods that use AutoAugment [1].

Method	Mean	Nat.	Spec.	Str.
TI	44.2	35.0	61.8	43.3
MS	47.1	34.2	68.4	47.9
Rel.Pat.Loc	50.8	45.0	73.0	44.8
Jigsaw	51.1	42.5	75.6	46.4
Rot-YT-F	55.6	44.9	76.1	54.8
VIVI-Rot(4)	56.7	47.7	78.4	53.7
VIVI-Rot(2)	57.6	49.7	78.6	54.1
Rot-YT-F-AA	57.7	48.5	77.8	55.7
Ex-YT-F	58.1	49.4	79.5	54.9
BigBiGAN	59.1	56.6	79.1	51.3
Ex-ImageNet	59.2	52.9	80.9	53.8
MT-SSL	59.2	51.9	78.9	55.8
Rot-ImageNet	59.6	53.8	77.2	56.0
Ex-YT-S	60.2	54.8	81.3	54.3
VIVI-Ex(2)-TimeArrow	60.7	55.2	80.6	55.5
VIVI-Ex(4)	61.0	55.6	79.7	56.4
VIVI-Ex(2)	61.4	55.3	79.9	57.5
Ex-YT-S-AA	61.5	56.1	81.7	56.0
Sup-10%	61.6	65.9	80.2	48.5
VIVI-Ex(4)-AA	61.7	56.3	80.9	57.0
VIVI-Ex(4)-Big	61.9	56.9	80.6	57.1
Ex-YT-F-AA	61.9	56.1	81.7	57.2
VIVI-Ex(4)-Big-AA	63.5	58.2	81.5	59.0
Semi-Ex-10%	63.9	69.3	80.3	50.9
Sup-100%	65.6	72.6	82.2	51.1
VIVI-Ex(4)-Co(10%)	66.3	63.1	82.4	61.1
Sup-Rot-100%	67.5	73.1	83.2	54.8
VIVI-Ex(4)-Co(100%)	69.2	70.1	83.2	61.3
VIVI-Ex(4)-Co(100%)-Big	70.4	72.2	83.7	62.3

Table 9: Overall and per group mean testing accuracy for fine-tuning hyper-parameter selection according to **Version 2 (arXiv:1910.04867v2) of the VTAB benchmark**. Each number is the median of three fine-tuning runs. See the caption of Table 8 for a description of the method abbreviations.

Method	Mean		Nat.	Spec.	Str.
Ex-ImageNet	59.2		52.9	**80.9**	53.8
VIVI-Ex(4)	61.0	(+1.8)	55.6	79.7	56.4
VIVI-Ex(2)	61.4	(+2.2)	55.3	79.9	57.5
VIVI-Ex(4)-Big	**61.9**	(+2.7)	**56.9**	80.6	**57.1**
Semi-Ex-10% [16]	63.9		**69.3**	80.3	50.9
VIVI-Ex(4)-Co(10%)	**66.3**	(+2.4)	63.1	**82.4**	**61.1**
Sup-100%	65.6		72.6	82.2	51.1
Sup-Rot-100% [16]	67.5		**73.1**	83.2	54.8
VIVI-Ex(4)-Co(100%)	69.2	(+3.6)	70.1	83.2	61.3
VIVI-Ex(4)-Co(100%)-Big	**70.4**	(+4.8)	72.2	**83.7**	**62.3**

Table 10: Testing result summary as in Table 1 for fine-tuning hyper-parameter selection according to **Version 2 (arXiv:1910.04867v2) of the VTAB benchmark**. Each number is the median of three fine-tuning runs.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *Proc. CVPR*, 2019. 2, 8

[2] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 3, 4

[3] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014. 2, 8

[4] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *Proc. ICLR*, 2018. 2, 3, 8

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. ECCV*, 2016. 3

[6] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proc. ICML*, 2015. 3

[7] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proc. CVPR*, 2019. 3

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. CVPR*, 2017. 5

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, 2014. 5

[10] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv:1903.10520*, 2019. 3

[11] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, 2015. 3

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 3

[13] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *Proc. ICCV*, 2017. 3

[14] Yuxin Wu and Kaiming He. Group normalization. In *Proc. ECCV*, 2018. 3

[15] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proc. ICCV*, 2019. 3

[16] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The Visual Task Adaptation Benchmark. *arXiv:1910.04867*, 2019. 8, 9