Sample- *and* Parameter-Efficient Auto-Regressive Image Models

Supplementary Material

10. Benchmark Datasets

Dataset	train	test	classes
Imagenet-1k [22]	1,281,167	50,000	1000
iNAT-18 [46]	437,513	24,426	8142
CIFAR-10 [34]	50,000	10,000	10
CIFAR-100 [34]	50,000	10,000	100
Food101 [9]	75,750	25,250	101
DTD [21]	3,760	1,880	47
Pets [36]	3,680	3,669	37
Cars [33]	8,144	8,041	196
iWildCam [8]	129,809	14961	182
Camelyon17 [5]	302,436	34904	2
PCAM [47]	262,144	32768	2
RxRx1 [42]	40,612	9854	1139
EuroSAT [30]	16,200	5400	10
fMoW [20]	76,863	19915	62
Infograph [37]	36,023	15,582	345

Table 7. **Evaluation benchmarks.** We provide the references, the number of images in the train and test sets, and the number of categories of all the 15 recognition benchmarks used in this work. Table taken from [26].

11. Computational Cost Estimation

In Table 8, we *estimate* the computational cost of each method using the following simplified formula:

$$\text{Cost} = \text{Parameters} \times \text{Samples} \times \text{Epochs} \times \text{Views}^2 \times \text{Tokens}^2 \tag{2}$$

This formula provides an approximate scaling relationship rather than an exact measurement, as it does not account for hardware optimizations, model-specific efficiencies, or parallelization effects.

- **Parameters (Linear):** The number of parameters in the model determines the size of weight matrices involved in computation. Since most architectures perform matrix multiplications that scale with the number of parameters, computation cost is approximately proportional to this term.
- **Samples (Linear):** The number of training samples contributes linearly since each sample requires a forward and backward pass.
- **Epochs (Linear):** The number of epochs scales cost linearly because training for more epochs means repeating the entire dataset multiple times.
- **Views (Squared):** If a method processes multiple views of the same data (e.g., contrastive learning with augmentations), the computational cost increases quadratically. This is because each pairwise interaction between views often involves computing similarities or attention across all view combinations.
- **Tokens (Squared):** The number of tokens per sample affects cost quadratically because self-attention mechanisms in transformers require $\mathcal{O}(\text{Tokens}^2)$ operations per forward pass.

Name	Parameters	Samples	Epochs	Views	Tokens	Cost	Acc.
DINO	85M	1.2M	800	2	768	19.2e22	75.0
iBOT	307M	1.2M	250	2	196	1.4e22	77.6
BEiT	307M	14M	150	1	256	4.2e22	65.4
MAE	632M	1.2M	1600	1	256	8.0e22	75.3
AIM	632M	2B	2.5	1	256	20.7e22	75.6
XTRA	632M	14M	100	1	256	5.8e22	76.2

Table 8. **Computational cost comparison.**

For clarity, the numbers reported in Table 3 are divided by 10^{22}, as the absolute units of computation do not impact the relative comparisons between methods.

While this formula captures key scaling behaviors, it does not precisely reflect real-world training cost due to factors like activation memory usage, hardware acceleration, and mixed precision training. Nonetheless, it serves as a useful proxy for comparing methods at scale.

Compared to MAE, AIM, and DINO, XTRA demonstrates greater effectiveness. While BEiT is slightly more efficient, XTRA outperforms it by +10.8% in accuracy. iBOT, which integrates contrastive learning with masked image modeling, achieves better accuracy at a lower computational cost.