Compact Backbone

**Higher pre-training accuracy on ImageNet does not translate to higher fine-tuning accuracy**. From Table 2 we can see that among all the backbones we used for our experiments, EfficientNetV2-S has the highest pre-training ImageNet-1k accuracy (84.23%). Even though the fine-tuning performance of EfficientNet was high, it did not perform the best in any of the domains or datasets on which we evaluated the models. Even among datasets whose images are sourced from ImageNet dataset, such as Tiny ImageNet and Stanford Dogs, we find that ConvNeXt outperforms EfficientNet. Therefore, we recommend the practitioners to not use the pre-training accuracy as an ironclad criterion to choose the backbone.

**Convolutional models strongly outperform transformers for resource-efficient low-data finetuning tasks.** Even though Swin transformer has a hierarchical structure capable of exploiting the spacial inductive bias (Goldblum et al., 2023), it still performed poorly in almost all our tasks compared to modern CNN architectures such as ConvNeXt (which was designed based on macro architectural insights from the Swin transformer). Hence, we recommend that for fine-tuning on small datasets, it is better to avoid using transformer architectures, such as Swin, and use pure CNN backbones, such as ConvNeXt, EfficientNet or RegNet.

**ConvNeXt architecture consistently outperforms other models when fine-tuning on natural image datasets.** This superior performance can be attributed to the thought design of ConvNeXt, which integrates architectural advancements that bridge the gap between traditional convolutional networks, such as ResNet, and modern Swin transformer models. ConvNeXt retains the beneficial convolutional inductive bias, enabling it to learn more effectively than attention-based transformer models. Our data clearly indicates that for natural images, ConvNeXt stands out as the best model, delivering exceptional fine-tuning performance

**RegNet and EfficientNet models are excellent choices for fine-tuning across a wide range of image domains.** While ConvNeXt excels predominantly with natural images, EfficientNet closely follows in performance, and RegNet also shows strong results in this domain. However, the versatility of RegNet and EfficientNet extends beyond natural images. Our experiments reveal that these models also perform exceptionally well on diverse domains, including remote sensing images, plant datasets, and medical images, such as histopathology images. Therefore, we recommend practitioners to consider RegNet and EfficientNet when working with datasets beyond natural images, as their adaptability and robust performance across various domains make them valuable tools for fine-tuning tasks.

**ShuffleNet is a better choice than MobileNet when very light-weight models are needed.** Among very lightweight models, specifically those with a model size of less than 50 MB, which are ideal for on-device applications, we find that ShuffleNetV2 generally outperforms MobileNetV3 across multiple domains. Although MobileNetV3 has shown better performance on medical domain, ShuffleNetV2 demonstrates more consistent and slightly superior performance across a broader range of image domains. Therefore, we recommend ShuffleNetV2 as a better choice for practitioners dealing with on-device applications, where fine-tuning a model on a domain-specific dataset is required.

**WaveMix performs well in datasets where multi-resolution token-mixing aids in learning** WaveMix outperforms all other models (9% increase from the second best, ConvNeXt) in galaxy morphology classification. WaveMix also performs well in medical domain performing better than all other datasets, and also maintaining the performance across different magnification. WaveMix, which uses 2D-DWT might possess inductive bias that can analyse the domains of astronomy and medical images better than other convolutional models due its multi-resolution token-mixing. Multiple levels of 2D-DWT also gives more significance to low frequency components (shapes) compared to regular convolutions which are biased towards higher frequency features such as textures. We recommend using WaveMx in domains where features across different resolutions are needed for better performance. WaveMix also performs better than ConvNeXt in CIFAR-10 and CIFAR-100 datasets, which were actually low resolution natural images (32 × 32) which were resized (to 256 × 256) for our training. Similarly, it gives best performance in remote-sensing dataset, EuroSAT, whose images (64 × 64) were resized (to 256 × 256) for our training. WaveMix is also state-of-theart in many low-resolution datasets such as EMNIST (28×28). The only low resolution image dataset where WaveMix did not perform well is Tiny ImageNet whose images (64 × 64) were also resized (to 256 × 256). We attribute this to the fact that Tiny ImageNet is an ImageNet-1k subset and the other models which has better performance in ImageNet-1k naturally performed better. So, we recommend using WaveMix for low-resolution image datasets.

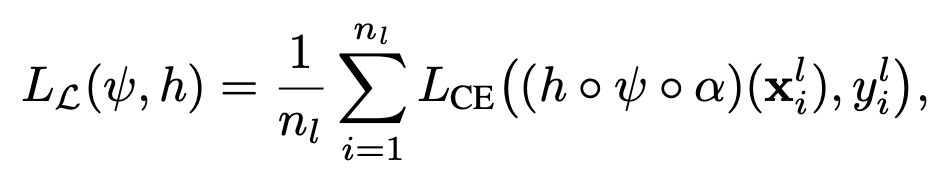
**Most of the top models in every domain retain their higher performance even with less training data.** We find that even when we fine-tune with a small percentage of training data (even 1% ∼ 1000 images), the models which performed well with full training set still retained their superiority. This points to the presence of a domain specific inductive bias present in these models since they can learn better representations with very less data. The failure of these models to perform well on other domains similarly with less training data also alludes to this inductive bias.

Self-training method

**Debiased Self-Training**

In semi-supervised learning (SSL), we have a labeled dataset of nl labeled samples

and an unlabeled dataset of nu unlabeled samples, where the size of the labeled dataset is usually much smaller than that of the unlabeled dataset. The standard cross-entropy loss on weakly augmented labeled examples is:



where α is the weak augmentation function. Since there are few labeled samples, the feature generator

and the task-specific head will easily over-fit, and typical SSL methods use these pseudo labels on

plenty of unlabeled data to decrease the generalization error. Different SSL methods design different

pseudo labeling function f [30, 59, 42]