A. Datasets, Models and Fine-tuning Hyperparameters

Datasets The **19 Fine-grained datasets** contain 19 commonly used fine-grained visual classification datasets, covering a wide range of domains, including *objects*, *scene*, *plants*, *animals*, *food*, *texture*, *medical*, *logo* and *art*. The **DomainNet** [45] benchmark is designed for evaluating multi-source domain adaptation in object recognition. It contains 0.6 million images across 6 domains (*clipart*, *infograph*, *painting*, *quickdraw*, *real*, and *sketch*). All domains include 345 categories (classes) of objects. We use the official train/test splits in our experiments. The **VTAB** [61] benchmark is designed for evaluating the transferability of pre-trained models. It consists of 19 datasets and the tasks are categorized into *natural*, *structured* and *special*. Some of the datasets can also be categorized as in the 19 fine-grained datasets. Note that there are some datasets also exists in the 19 fine-grained datasets. We include them when reporting the VTAB performance. Currently we used 15 of the 19 datasets for VTAB, covering all the categories. More detailed information of each dataset can be found in Table 3.

Table 3. Datasets statistics. For the *aircrafts, flowers* and *surface* dataset, the original training set and validation set are combined following the practice. Note datasets noted with * are not included in our experiments.

Benchmark	Dataset Names	Alias	Domain	Classes	Training	Test
19 Fine-grained	Stanford Dogs [30]	dogs	animals	120	12,000	8,580
	CUB-Birds 200 [56]	birds	animals	200	5,994	5,794
	Oxford Flowers [43]	flowers	plants	102	2,040	6,149
	VegFru [19]	vegfru	plants	290	29,000	116,156
	Herbarium 2019 [52]	herbarium	plants	683	31, 546	2,679
	FGVC Aircrafts [38]	aircrafts	objects	100	6,667	3,333
	Stanford Cars [32]	cars	objects	196	8,144	8,041
	MIT Indoor-67 [50]	mit67	scene	67	5,360	1,340
	European Flood Depth [3]	flood	scene	2	3153	557
	NWPU Resisc45 [7]	resisc45	scene	45	25,200	6,300
	Food-101 [6]	food101	food	101	12,000	8,500
	iFood [28]	ifood	food	251	118,475	11,994
	Describable Textures [8]	dtd	texture	47	4,230	1,410
	Open Surface-2500 [4]	surface	texture	23	48,875	8,625
	Magnetic Tile [23]	tile	texture	5	1,008	336
	Pneumonia [29]	pneumonia	medical	2	25,216	624
	Malaria Cell Images [48]	cell	medical	2	20,668	6,890
	BelgaLogos [25]	logo	logo	27	7,500	2,500
	SemArt [15]	semart	art	26	18,174	3,208
DomainNet	Clipart	clipart	clipart	345	33,525	14,604
	Real	real	real	345	120,906	52,041
	Quickdraw	quickdraw	quickdraw	345	120,750	51,750
	Painting	painting	painting	345	50,416	21,850
	Inforgraph	inforgraph	inforgraph	345	36,023	15,582
	Sketch	sketch	sketch	345	48,212	20,916
VTAB	Caltech101* [13]	caltech101	natural - objects	101	3,060	6,084
	SUN397* [58]	sun397	natural - scene	397	73,257	26,032
	Oxford Flowers [43]	flowers	natural - plants	102	2,040	6,149
	CIFAR-100 [33]	cifar100	natural - objects	100	50,000	10,000
	SVHN [41]	svhn	natural - object	10	73,257	26,032
	Oxford IIIT Pet [44]	pets	natural - animal	37	3,680	3,669
	Describable Textures [8]	dtd	natural - texture	47	4,230	1,410
	NWPU Resisc45 [7]	resisc45	specialized - scene	45	25,200	6,300
	EuroSAT [18]	eurosat	specialized - scene	10	20,250	6,750
	Diabetic Rethinopathy [27]	retinopathy	specialized - medical	5	35,126	53,576
	PatchCamelyon [55]	pcam	specialized - medical	2	262,145	32,769
	CLEVR distance [24]	clevr_dist	structured	7	70,000	15,000
	CLEVR counting [24]	clevr_dist	structured	8	70,000	15,000
	Dmlab Frames*	dmlab	structured	6	65,550	22,628
	dSprites orientation [39]	dsprites_ori	structured	40	663,552	73,728
	dSprites location [39]	desprites_loc	structured	6	663,552	73,728
	KITTI distance [16]*	kitti_dist	structured	4	7,481	7,518
	Small NORB azimuth [34]	smallnorb_elevation	structured	18	24,300	24,300
	Small NORB elevation [34]	smallnorb_azimuth	structured	9	24,300	24,300

Models The `TIMM` and `torchvision` model zoo collected over 590 ImageNet pre-trained models in various architectures and training recipes. We filtered out 409 models that can be fine-tuned with batch size 32 and evaluated the single image inference latency of the 400+ models on single GPU. The scatter plot of latency and accuracy can be seen in Fig. 6(a). We can identify the Pareto frontier models of the 400+ models, spanning the latency from 3 ms to 120 ms. We select 22 widely used models that are near the Pareto Front curve, which covers a wide range of architecture families, including ReseNet [17], DenseNet [21], MobileNet [20], EfficientNet [53], ViTs [11], Swin-T [36] and ConvNeXt [37]. The detailed statistics of the selected 22 models can be seen in Table 4.

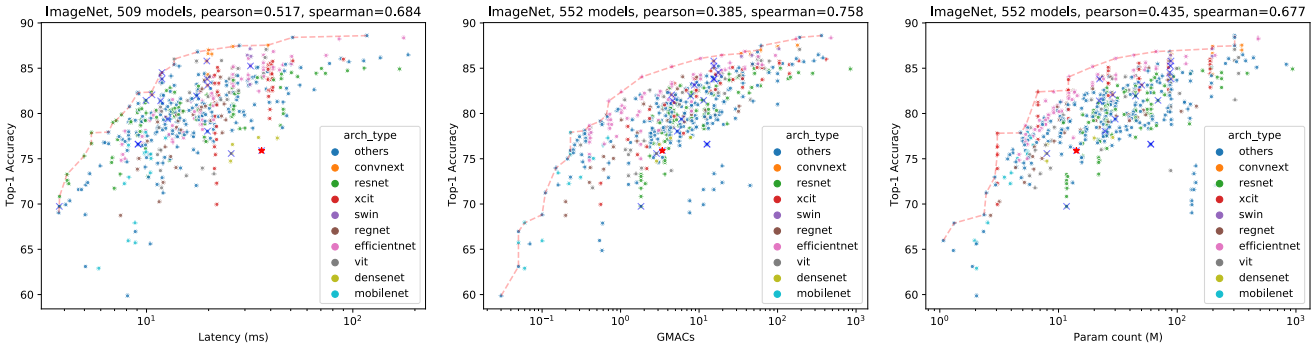

Figure 6. The statistics of the 500+ ImageNet pre-trained models. The latency is measured on V100 GPU with batch size 1. The dashed line connects the Pareto Frontier models. The blue crossed models are our selected 22 models and the red crossed dot is the reference model - DenseNet-169.

Table 4. The statistics of the 22 models, which are ranked by their single image inference latency (ms) on the V100 GPU.

	Model Name	Arch Family	Acc	Pretrain	Img Size	Latency	FLOPs	#Params
1	resnet18	resnet	69.74	IN-1K	224	3.78	1.82	11.69
2	mobilenet_v2	mobilenet	71.88	IN-1K	224	7.33	0.31	3.50
3	mixer_b16_224	others	76.61	IN-1K	224	9.10	12.62	59.88
4	mixer_b16_224_in21k	others	-	IN-21K	224	9.10	12.62	59.88
5	wide_resnet50_2	resnet	81.45	IN-1K	224	9.94	11.43	68.88
6	convnext_tiny	convnext	82.06	IN-1K	224	10.63	4.47	28.59
7	vit_small_patch16_224	vit	81.40	IN-1K	224	11.71	4.61	22.05
8	vit_small_patch16_384	vit	83.81	IN-1K	384	11.88	15.52	22.2
9	vit_base_patch16_224	vit	84.53	IN-1K	224	11.88	17.58	86.57
10	vit_base_patch16_224_in22k	vit	-	IN-21K	224	11.88	17.58	86.57
11	resmlp_24_224	others	79.38	IN-1K	224	12.67	5.96	30.02
12	efficientnet_b0	efficientnet	76.30	IN-1K	224	15.06	0.40	5.29
13	resnet101	resnet	81.93	IN-1K	224	17.48	7.83	44.55
14	convnext_base	convnext	83.82	IN-1K	224	19.68	15.38	88.59
15	convnext_base_in22ft1k	convnext	85.80	IN-21K-1K	224	19.60	14.38	88.59
16	gmixer_24_224	others	78.04	IN-1K	224	19.74	5.28	24.72
17	convnext_small	convnext	83.13	IN-1K	224	19.80	8.70	50.22
18	efficientnet_b3	efficientnet	81.10	IN-1K	300	24.27	2.01	12.23
19	densenet121	densenet	75.58	IN-1K	224	25.73	2.87	7.98
20	swin_base_patch4_window7_224	swin	85.25	IN-1K	224	31.90	15.47	87.77
21	swin_base_patch4_window7_224_in22k	swin	-	IN-21K	224	31.90	15.47	87.77
22	densenet169	densenet	75.90	IN-1K	224	36.13	3.40	14.15

Fine-tuning Hyperparameters All models are trained with a single GPU with the same settings with the hyperparameter search ranges. We performed fine-tuning with following hyper-parameters: we fine-tune 30 epochs with SGD with Nesterov momentum 0.9, batch size of 32 and weight decay of 10^{-4}. The learning rate η decays by $0.1\times$ at 15th and 25th epochs. We performed a grid search of with various initial learning rates and data augmentation strategies, i.e., $\eta_0 \in \{0.05, 0.01, 0.005, 0.001\}$ and data_aug $\in \{\texttt{rrcrop}, \texttt{rcrop}\}$. Here `rrcrop` stands for random resized cropping, which randomly crops ratio ranging from 0.08 to 1.0 with random aspect ratio between [3/4, 4/3], which is adopted in [51]. And `rcrop` stands for random

cropping, which differs with with `rrcrop` in that it uses fixed cropping ratio (0.875). We report the best top-1 test accuracy of the 8 trials.

B. Feature-Based Model Selection

More Model Selection Failure Cases In additional to the model selection results on the 19 fine-grained datasets (Fig. 2 in Sec. 2.2, here we show the existing MS results on DomainNet and VTAB in Fig. 7. The feature-based MS methods perform relatively well on DomainNet datasets (*clipart*, *infograph*, *painting*, *quickdraw*, *real*, and *sketch*). However, the MS methods have weak or even negative correlations for some of the VTAB tasks. For example, the structured tasks including *smallnorb*, *clevr* and *dsprites*. Even the simple task SVHN has negative correlations, which indicates that the feature-based MS methods can fail to estimate the relative performance with only feature information.

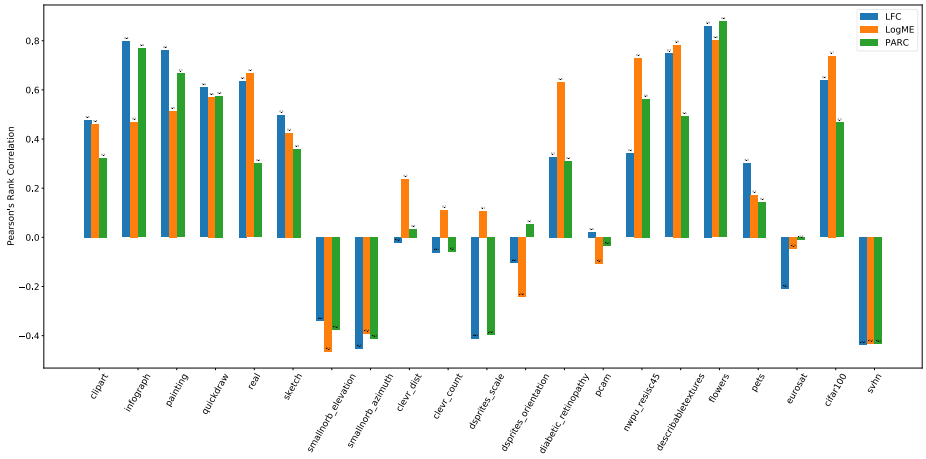

Figure 7. Comparison of MS methods on DomainNet and VTAB. The first 6 datasets (*clipart*, *infograph*, *painting*, *quickdraw*, *real*, and *sketch*) are from DomainNet, and the rest of the datasets are from VTAB.

C. Model Recommendation

Feature Embedding Fig. 8 illustrates how the learning history is represented in the embedding space. Table 5 lists the details of the features used in learning based MS. The categorical features are converted to one-hot vectors. And all features are normalized with their minimum and maximum values across all datasets, so that the maximums and minimum values are 1 and 0 after normalization. The visualization of the real embedding of all available fine-tuning tasks can be seen in Fig 9.

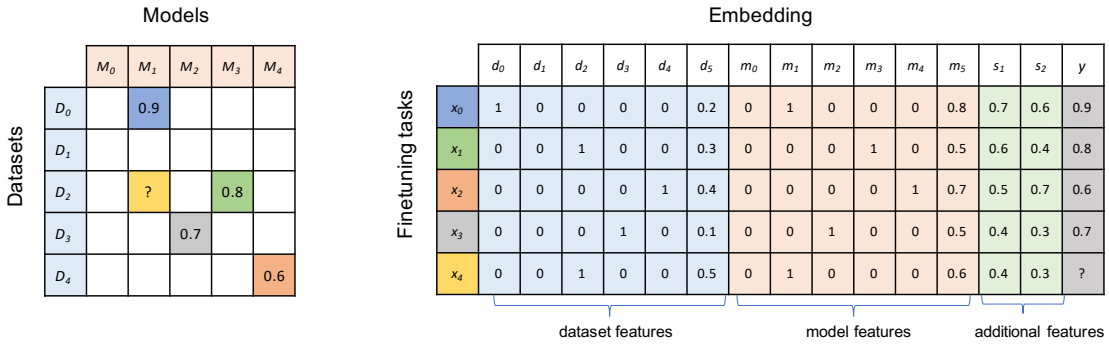

Figure 8. An illustration of learning from the training history and the representation of training data for the recommender system. The left figure shows the matrix of fine-tuning performance with 4 pairs of dataset and model and the goal is to predict the performance of unknown pairs. The right figure shows the encoding of the each training job is concatenated features of the datset, model and others.

Table 5. The complete features for embedding fine-tuning tasks for learning-based MS.

Field idx	Field Name	Feature Name	Type	One-hot	Log	Dimension	Min	Max
1	dataset	dataset id	category	Yes	No	41	0	40
1	dataset	dataset size	scalar	No	Yes	1	1008	1200000
1	dataset	number of classes	scalar	No	Yes	1	2	1000
2	model	architecture id	category	Yes	No	500	0	409
2	model	architecture family id	category	Yes	No	10	0	9
2	model	pre-trained dataset id	category	Yes	No	3	0	2
2	model	input size	scalar	No	Yes	1	106	448
2	model	GMACs (G)	scalar	No	Yes	1	0.03	46.95
2	model	#Parameters (G)	scalar	No	Yes	1	1.88	88.59
3	MS score	LFC	scalar	No	No	1	0.002	0.792
3	MS score	LogME	scalar	No	No	1	-0.905	2.209
3	MS score	PARC	scalar	No	No	1	0.085	80.358

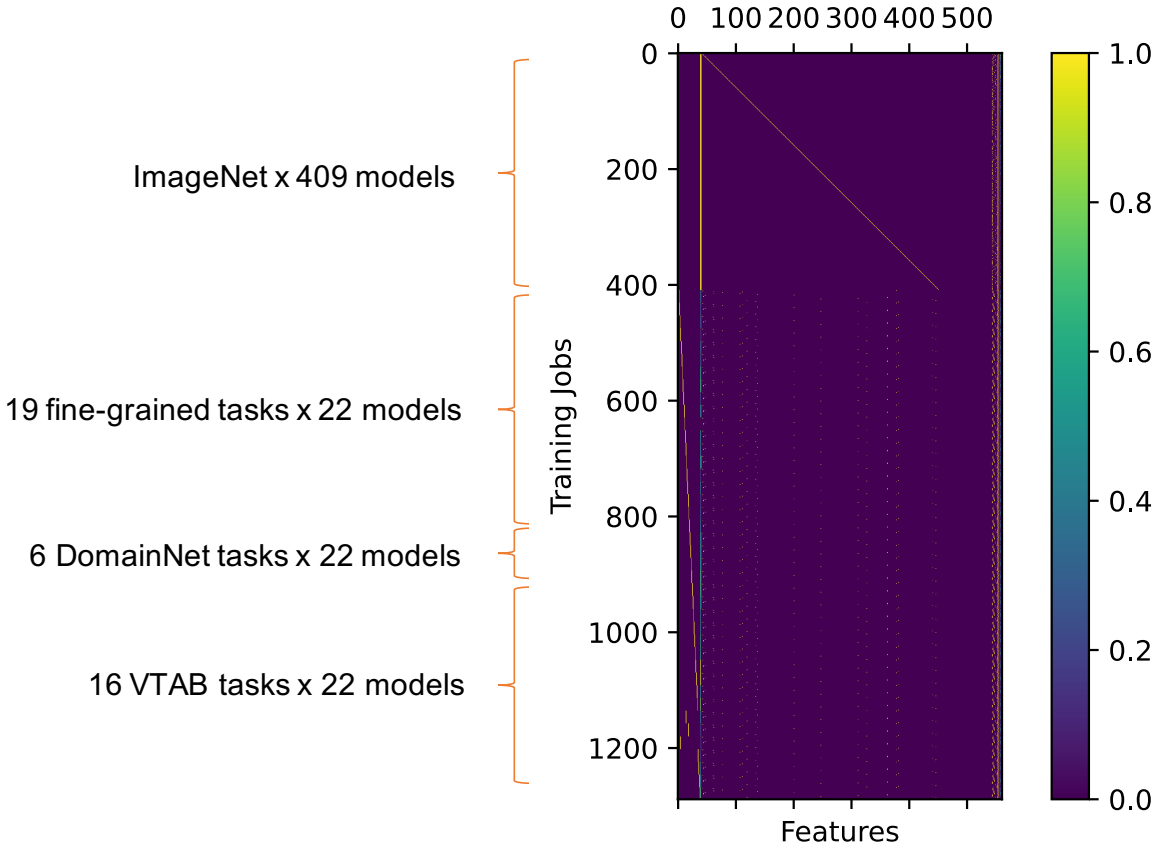

Figure 9. Visualization of the normalized feature embedding of all training jobs, including 400+ ImageNet training jobs, 418 training jobs on the 19 fine-grained tasks, 132 training jobs on 6 DomainNet datasets and 352 training jobs on the VTAB datasets.

Training Details We implement the LR and FM algorithms with the `xlearn` library. The regression models with MAE loss are trained with SGD until the loss converges. The initial learning rate is 0.2 and the regularization λ is 0.002. Instance-wise normalization is disabled.

D. Heterogeneous Model Zoo and Architecture Bias

The existence of inductive bias for different architectures indicates that there is no single best architecture for all tasks with different characteristics. Our hypothesis is that the optimal architecture for transfer learning is task dependent. To verify this, we perform experiments with diverse datasets and architectures. We identify the existence of architecture bias for different datasets, which confirms the need of task dependent architecture selection. Fig 10 shows the fine-tuning performance of the 22 models on 40 downstream tasks, from which we can see the ranking of the model for each task can be significantly different. It also demonstrates that the best performing model can be task dependent.

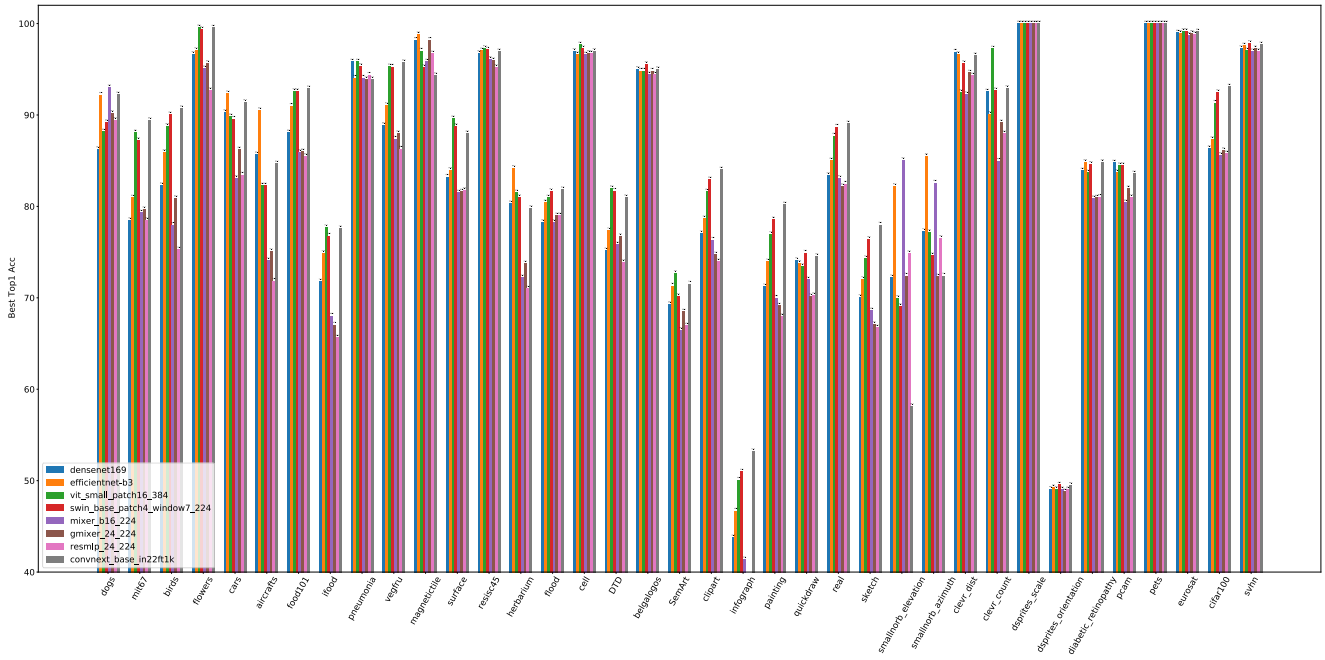

Figure 10. The performance of different architectures on the 19 fine-grained datasets, DomainNet and VTAB. The best performing architecture for different tasks can be different, For example, EfficientNet-B3 with higher resolution input performs best on *aircrafts*, *herbarium* and *smallnorb_azimuth*; Swin-B/16 wins on *surface*, *texture (DTD)* and *semart*.

D.1. Architecture bias

Given a task and a set of well pre-trained models in different architectures, we say there is an architecture bias for a task if one architecture obtains the best performance and outperforms the second best architecture by a large margin (e.g., >2% top-1 accuracy). To justify the significance of architecture bias, we show the following facts for each benchmark: a) the performance distribution of each architecture over a baseline model across all downstream tasks. b) the performance gain of the best model over the second best performing model for each task. Fig. 11 ranks the models by their mean performance, from which we can see ConvNeXt, ViTs, Swin-T, EfficientNet are top ranked models in terms of average performance gain over DenseNet-169. Note that although ConvNeXt has the strongest average performance, it is not always the best for all tasks. Other architectures can outperform ConvNeXt significantly for datasets like *aircrafts*, *magnetictile*, *herbarium*, *dogs*, indicating the existence of architecture bias for those datasets. Similarly, we observe stronger architecture bias on structured tasks in *VTAB*, such as *smallnorb elevation* and *clevr count*. For *DomainNet*, we find ConvNeXt with ImageNet-22K pre-training performs best on 5 out of 6 domains (e.g., *clipart*, *inforgraph*, *painting*) with significant performance gains over DenseNet169 (> 6%). However, the best performing architecture on *quickdraw* is Swin-T and the differences among the architectures are small (<1%) (Fig. 12c).

We have empirically verified the hypothesis that the optimal architecture for transfer learning is task dependent and there is no single best model that performs best on every datasets. Here we performed statistical tests for the hypothesis. We conduct non-parametric paired one-tailed t-test (the Wilcoxon signed-rank test) on whether the selected model's performance is greater than other fine-tuning methods across 19 fine-tuning tasks. The null hypothesis H_0 states that the mean performance difference

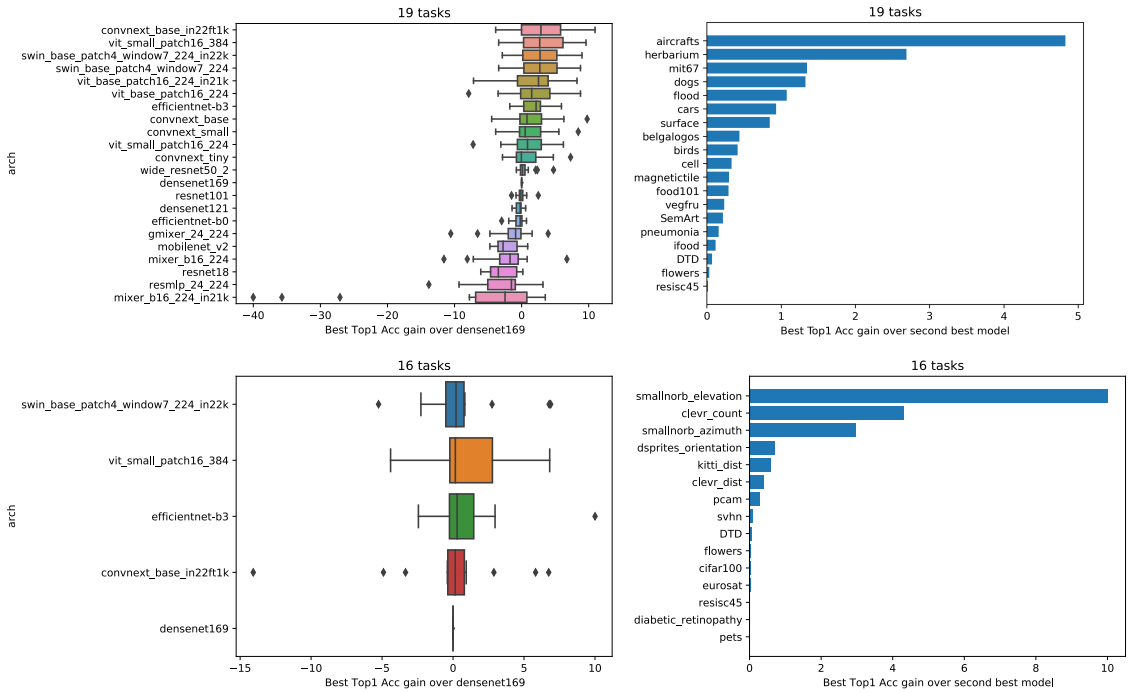

Figure 11. The significance of architecture bias on the 19 fine-grained datasets and VTAB benchmark. The first column shows the performance gain over DenseNet-169 on downstream tasks for each model. The models are ranked by their mean accuracy gain. The second column shows the performance gain of the optimal model over the second best performing model for each dataset.

Table 6. The Wilcoxon signed-rank test on whether the row model is significantly better than the column model on the 19 fine-grained tasks. The table shows the p-value. The bold values indicate that the row model is statistically better than the column model ($p < 0.05$). No model is statistically better than all other models.

	ViT-S/16-384	Swin-B-P4-W7-in22k	Swin-B-P4-W7	Efficientnet-B3	ViT-B/16-224-in21k
ConvNeXt-B-in22ft1k	0.492	0.384	0.198	0.084	**0.003**
ViT-S/16-384	-	0.147	0.072	0.107	**0.002**
Swin-B-P4-W7-in22k	0.862	-	0.098	0.121	**0.003**
Swin-B-P4-W7	0.928	0.909	-	0.156	**0.016**
Efficientnet-B3	0.893	0.887	0.853	-	0.779

between selected model and baseline model is zero. The alternative hypothesis H_1 states that the selected model outperforms the baseline model. As shown in Fig 11, we pick the top 6 models with the best average performance on the 19 fine-grained tasks and check whether any of them can be significantly better than others. Table 6 presents the p-values of each test, with the number of observations equal to 19 for each model compared. There is no single model that outperforms all other models.

D.2. Why do certain model work well on certain datasets?

We investigate the reason why certain model performs better on certain datasets than others. As shown in Fig. 11, the top 4 best performing models are ConvNeXt-B-in22ft1k, Efficientnet-B3, ViT-S/16-384 and Swin-B-in22k. Here we analyze why they are chosen for certain tasks and what distinguish them from other models.

- ConvNeXt-B-in22ft1k performs best on many downstream tasks, such as *birds* [56] and *food101* [6]. One reason is that this model is obtained with strong pre-training on ImageNet-22k and then fine-tuned on ImageNet-1K, biasing towards datasets that are close to ImageNet.

- Efficientnet-B3 [53] is chosen over ConvNeXt for *aircrafts*. Note that EfficientNet-B3 adopts a *higher resolution* for input images (300 instead of 224). Dataset such as *aircrafts* benefits from the high resolution to make the subtle differences

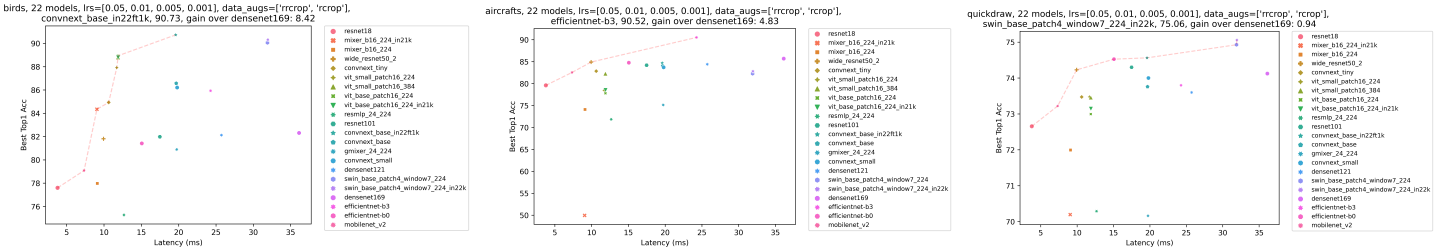

Figure 12. The Pareto front models for *birds*, *aircrafts* and *DomainNet-quickdraw*. ConvNeXt, EfficientNet-B3 and Swin-T are the best performing models and have large margin over the other architectures.

noticeable. EfficientNet-B3 also has significant better performance on structured tasks such as *smallnorb* and *clevr* in VTAB, which are synthetic 3D objects tasks such as counting and angle estimation. Similar observation is also made in [61] that *structured* tasks behaves differently with nature images.

- Swin-B [36] performs best on *quickdraw* (Fig. 12c), which has no color or texture information but only shapes. It suggests that Swin-T has the advantage of capturing the structure information. However the task is so simple the architecture bias or pre-training makes not too much on performance difference (the difference between ResNet-18 and the wining Swin-T is only 2%). Similarly previous work [62] finds that ViTs are better than CNNs on this task, and conclude that ViTs are better preserving shape and structure information.

D.3. More Pareto front results

We have shown that the optimal model is dataset dependent. A natural question to ask is that whether the Pareto front models for ImageNet continue to be on the Pareto frontier for other downstream tasks. Similar to Fig. 6(a), we plot the scatter plot of latency and performance for three datasests in Fig 12 and show more results in Fig. 13. We can see that the Pareto frontier models is actually task dependent, which suggests the need to perform dataset dependent search.

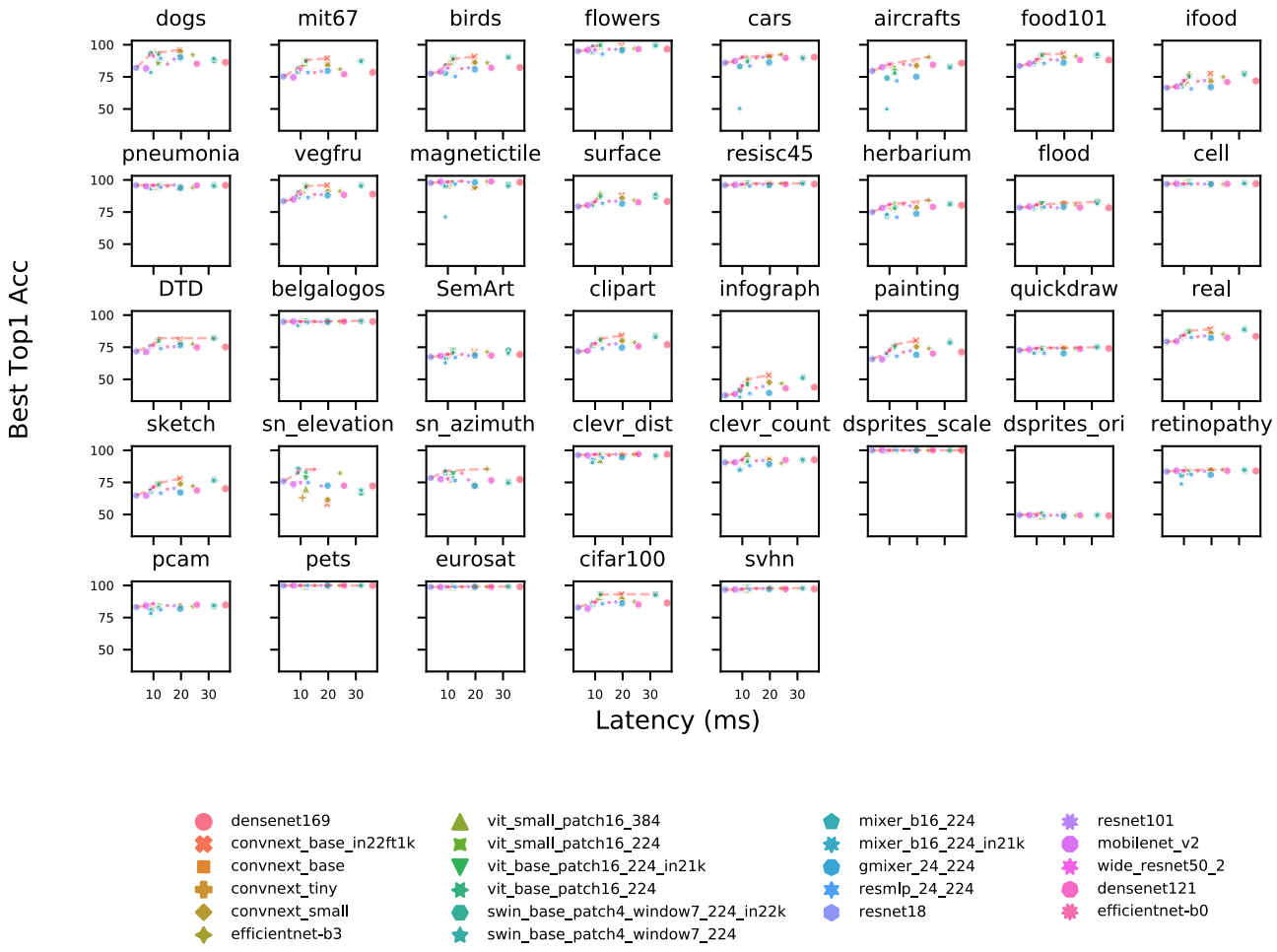

Figure 13. The Pareto front models can be task dependent.