Efficient Conditional Pre-training for Transfer Learning

Shuvam Chakraborty
Stanford University

Burak Uzkent*
Stanford University

Kumar Ayush*
Stanford University

Kumar Tanmay
IIT Kharagpur

Evan Sheehan
Stanford University

Stefano Ermon
Stanford University

Abstract

Almost all the state-of-the-art neural networks for computer vision tasks are trained by (1) pre-training on a large-scale dataset and (2) finetuning on the target dataset. This strategy helps reduce dependence on the target dataset and improves convergence rate and generalization on the target task. Although pre-training on large-scale datasets is very useful for new methods or models, its foremost disadvantage is high training cost. To address this, we propose efficient filtering methods to select relevant subsets from the pre-training dataset. Additionally, we discover that lowering image resolutions in the pre-training step offers a great trade-off between cost and performance. We validate our techniques by pre-training on ImageNet in both the unsupervised and supervised settings and finetuning on a diverse collection of target datasets and tasks. Our proposed methods drastically reduce pre-training cost and provide strong performance boosts. Finally, we improve the current standard of ImageNet pre-training by 1-3% *by tuning available models on our subsets and pre-training on a dataset filtered from a larger scale dataset.*

1. Introduction

Recent success of modern computer vision methods relies heavily on large-scale labelled datasets, which are often costly to collect [4, 14, 23]. Alternatives to large-scale labelled data include pre-training a network on the publicly available ImageNet dataset with labels [8] and performing transfer learning on target tasks [16, 18, 19, 27, 32]. On the other hand, unsupervised learning has received tremendous attention recently with the availability of extremely large-scale data with no labels, as such data is costly to obtain [2–4, 12, 14, 15, 23].

*Equal Contribution. Contact: {shuvamc, buzkent, kayush}@cs.stanford.edu

The explosion of data quantity and improvement of unsupervised learning with contrastive learning portends that the standard approach in future tasks will be to (1) learn weights a on a very large-scale dataset with unsupervised learning and (2) fine-tune them on a small-scale target dataset. A major problem with this approach is the large amount of computational resources required to train a network on a very large scale dataset [23]. For example, a recent contrastive learning method, MoCo-v2 [14, 15], uses 8 GPUs to train on ImageNet for 53 hours, which can cost thousands of dollars. Extrapolating, this forebodes pre-training costs on the order of millions of dollars on larger-scale datasets. Those without access to such computation power will require selecting relevant subsets of those datasets specific to their task.

Cognizant of these pressing issues, we propose novel methods to efficiently filter a user defined number of pre-training images conditioned on a target dataset. We also find that the use of low resolution images during pre-training provides a great cost to performance trade-off. Our approach consistently outperforms other methods by 2-9% and are both flexible, translating to both supervised and unsupervised settings, and adaptable, translating to a wide range of target tasks including image recognition, object detection and semantic segmentation. Our methods perform especially well in the more relevant unsupervised setting where pre-training on a 12% subset of data can achieve within 1-4% of full pre-training when considering target task performance. Next, we use our methods to improve standard ImageNet (1.28M images) pre-training. In this direction, we construct a large scale dataset (6.7M images) from multiple datasets and filter 1.28M images conditioned on a target task. Our results show that we improve standard ImageNet pre-training by 1-3% on downstream tasks. Thus, when needing to pre-train from scratch on large scale data for a specific application, our methods can replace the standard ImageNet pre-training with conditional pre-training.

2. Related Work

Active Learning Active Learning fits a function by selectively querying labels for samples where the function is currently uncertain. In a basic greedy setup, the samples with the highest entropies are chosen for annotation [1,10,26,30]. Active learning typically assumes similar data distributions for candidate samples, whereas our data distributions can potentially have large shifts. Furthermore, active learning, due to its iterative nature, can be quite costly, hard to tune, and can require prior distributions [25].

Unconditional Transfer Learning Pre-training networks on ImageNet has been shown to be a very effective way of initializing weights for a target task with small sample size [16, 18, 19, 27, 32]. However, all these studies use unconditional pre-training as they employ the weights pre-trained on the full source dataset, which can be computationally infeasible for future large scale datasets.

Conditional Transfer Learning [7, 24, 33], on the other hand, filter the pre-training dataset conditioned on target tasks. [7, 11] use greedy class-specific clustering based and learn image representations with an encoder trained on the massive JFT-300M dataset [17], which dramatically increases cost. [33] trains a number of expert models on many subsets of the pre-training dataset and uses their performance to weight source images, however this method is naturally quite computationally expensive. Many of these methods also require labelled pre-training data and are not well suited for target tasks such as object detection and semantic segmentation. Our methods differ from these works as we take into account pre-training dataset filtering efficiency, adaptability to different target tasks and settings, and target task performance.

3. Problem Definition and Setup

We assume a target task dataset represented as $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$ where $\mathcal{X}_t = \{x_t^1, x_t^2, \ldots, x_t^M\}$ represents a set of M images with their ground truth labels $\mathcal{Y}_t$. Our goal is to train a function f_t parameterized by θ_t on the dataset $\mathcal{D}_t$ to learn $f_t : x_t^i \mapsto y_t^i$. To transfer learn, we first pre-train θ_t on a large-scale source dataset $\mathcal{D}_s$ and fine-tune θ_t on $\mathcal{D}_t$. This strategy reduces the amount of labeled samples needed in $\mathcal{D}_t$ and boosts the accuracy in comparison to the randomly initialized weights [23, 28]. For the pre-training dataset, we can have either labelled or unlabelled setups: (1) $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$ and (2) $\mathcal{D}_s = (\mathcal{X}_s)$ where $\mathcal{X}_s = \{x_s^1, x_s^2, \ldots, x_s^N\}$. However, it is tough to label vast amounts of publicly available images, and with the increasing popularity of unsupervised learning methods [3–5, 14, 15], it is easy to see that unsupervised pre-training on very large $\mathcal{D}_s$ with no ground-truth labels will be the standard and preferred practice in the future.

A major problem with learning θ_t on a very large-scale dataset $\mathcal{D}_s$ is the computational cost, and using the whole dataset may be impossible for most. One way to reduce costs is to filter out images deemed less relevant for $\mathcal{D}_t$ to create a dataset $\mathcal{D}_s^{'} \in \mathcal{D}_s$ where $\mathcal{X}_s = \{x_s^1, x_s^2, \ldots, x_s^{N^{'}}\}$ represents a filtered version of $\mathcal{D}_s$ with $N^{'} \ll N$. Our approach conditions the filtering step on the target dataset $\mathcal{D}_t$. In this study, we propose flexible and adaptable methods to perform *efficient conditional pre-training*, which reduces the computational costs of pre-training and maintains high performance on the target task.

4. Methods

We investigate a variety of methods to perform efficient pre-training while maintaining high performance on the target dataset. We visualize our overall procedure in Figure 1 and explain our techniques below.

4.1. Conditional Data Filtering

We propose novel methods to perform conditional filtering efficiently. Our methods score every image in the source domain and select the best scoring images according to a pre-specified data budget N'. Our methods are fast, requiring only one forward pass through $\mathcal{D}_s$ to get the filtered dataset $\mathcal{D}_s^{'}$ and can work on both $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$ and $\mathcal{D}_s = (\mathcal{X}_s)$. The fact that we consider *data features not labels* perfectly lends our methods to the latter, more relevant, unsupervised setting. This is in contrast to previous work such as [7, 11, 24] which do not consider efficiency and are designed primarily for the supervised setting and thus will be more difficult to apply to large scale datasets.

Algorithm 1 Clustering Based Filtering

procedure CLUSTERFILTER($\mathcal{D}_s, \mathcal{D}_t, N^{'}, K, AggOp$)
  $f_h \leftarrow TRAIN(\mathcal{D}_t)$ ▷ Train Feature Extractor
  $\mathcal{Z}_t \leftarrow \{f_h(x_t^i)\}_{i=1}^{M}$ ▷ Target Representations
  $\{\hat{z}\}_{k=1}^{K} \leftarrow K\text{-}Means(\mathcal{Z}_t, K)$ ▷ Cluster Target
  $d_k^i \leftarrow ||f_h(x_s^i) - \hat{z}_k||_2$ ▷ Source Distances
  $c_s \leftarrow \{AggOp(\{d_k^i\}_{k=1}^{K})\}_{i=1}^{N}$ ▷ Score Source
  $\mathcal{D}_s^{'} \leftarrow BOTTOM(\mathcal{D}_s, N^{'}, c_s)$ ▷ Filter Source
  return $\mathcal{D}_s^{'}$ ▷ Return the Filtered Subset

4.1.1 Conditional Filtering by Clustering

Selecting an appropriate subset $\mathcal{D}_s^{'}$ of pre-training data $\mathcal{D}_s$ can be viewed as selecting a set of data that minimizes some distance metric between $\mathcal{D}_s^{'}$ and the target dataset $\mathcal{D}_t$, as explored in [7,11]. This is accomplished by taking feature representations $\mathcal{Z}_s$ of the set of images $\mathcal{X}_s$ and selecting pre-training image classes which are close (by some distance metric) to the representations of the target dataset classes. Building on this, we make several significant modifications to account for our goals of efficiency and application to unsupervised settings.

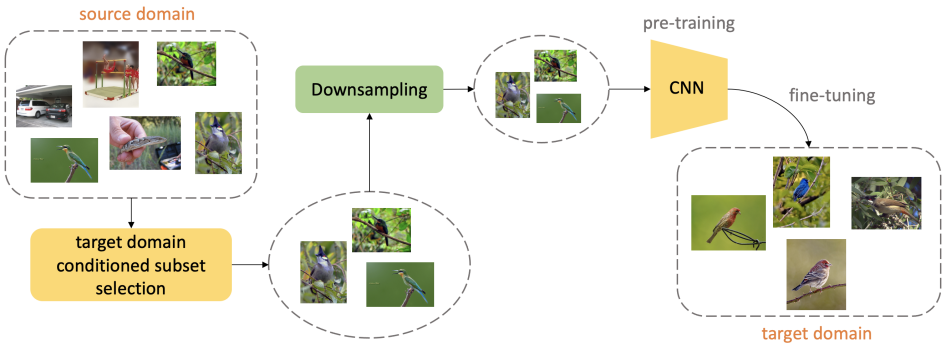

Figure 1. We first perform a conditional filtering method on the source dataset and downsample image resolution on this filtered subset. Finally, we perform pre-training on the subset and finetuning on the target task.

Training Only with Target Dataset. We do not train a network f_h on a large scale dataset, i.e. JFT-300M [7], as this defeats the entire goal of pre-training efficiency. Therefore, we first train a model f_h with parameters θ_h using the target dataset $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$ and use the learned θ_h to filter the source dataset $\mathcal{D}_s$.

Consider Source Images Individually. Selecting entire classes of pre-training data can be suboptimal when limited to selecting a small subset of the data. For example, if limited to 6% of ImageNet, (a reasonable budget for massive datasets), we can only select 75 of the 1000 classes, which may prohibit the model from having the breadth of data needed to learn transferable features. Instead, we treat each image x_s^i from $\mathcal{D}_s$ separately to flexibly over-represent relevant classes while not being forced to select full set of images from different classes. Additionally, very large scale datasets may not have class labels $\mathcal{Y}_s$. For this reason, we develop methods that work with unsupervised learning, and treating source images independently accomplishes this.

Scoring and Filtering. Finally, we choose to perform K-Means clustering on the representations $\mathcal{Z}_t$ learned by f_h to get K cluster centers $\{\hat{z}\}_{k=1}^K$. We then compute the distances between $\mathcal{X}_s$ and $\{\hat{z}\}_{k=1}^K$ as

$$d_k^i(x_s^i, k) = ||f_h(x_s^i; \theta_h) - \hat{z}_k||_p \tag{1}$$

where p is typically 1 or 2 (L1 or L2 distance). We can score x_s^i by considering an *Aggregation Operator(AggOp)* of either average distance to the cluster centers

$$c_s^i = \frac{1}{K}\sum_{k=1}^{K} d_k^i \tag{2}$$

or minimum distance

$$c_s^i = \min(\{d_k^i\}_{k=1}^K). \tag{3}$$

To filter, we sort by c_s^i in ascending order and select N' images to create $\mathcal{D}_s^{'} \in \mathcal{D}_s$ and pre-train θ_t on it.

Advantages of our Method Performing unsupervised clustering ensures that our method is not fundamentally limited to image recognition target tasks and also does not assume that source dataset images in the same class should be grouped together. Furthermore, our method requires only a single forward pass through the much smaller pre-training dataset. It attains our goals of efficiency and flexibility, in contrast to prior work such as [7, 11]. We outline the algorithm step-by-step in Algorithm 1.

Algorithm 2 Domain Classifier Filtering

procedure DOMAINCLSFILTER($\mathcal{D}_s, \mathcal{D}_t, N^{'}$)
  SAMPLE $\{x_s^i\}_{i=1}^M \in \mathcal{D}_s$
  $\mathcal{X}_h \leftarrow \{\{x_s^i\}_{i=1}^M, \{x_t^i\}_{i=1}^M\}$
  $\mathcal{Y}_h \leftarrow \{\{0\}_{i=1}^M, \{1\}_{i=1}^M\}$ ▷ Domain Labels
  $\mathcal{D}_h \leftarrow (\mathcal{X}_h, \mathcal{Y}_h)$ ▷ Training Data
  $f_h(x; \theta_h) \leftarrow_{\theta_h} CELoss(\mathcal{D}_h)$ ▷ Fit Model
  $c_s \leftarrow \{f_h(x_s^i; \theta_h)\}_{i=1}^N$ ▷ Score
  $\mathcal{D}_s^{'} \leftarrow TOP(\mathcal{D}_s, N^{'}, c_s)$ ▷ Filter Source
  return $\mathcal{D}_s^{'}$ ▷ Return the Filtered Subset

4.1.2 Conditional Filtering with Domain Classifier

In this section, we propose a novel domain classifier to filter $\mathcal{D}_s$ with several desirable attributes. We outline the algorithm step-by-step in Algorithm 2.

Training. In this method, we propose to learn θ_h to ascertain whether an image belongs to $\mathcal{D}_s$ or $\mathcal{D}_t$. θ_h is learned on a third dataset $\mathcal{D}_h = (\mathcal{X}_h, \mathcal{Y}_h)$ where $\mathcal{X}_h = \{\{x_s^i\}_{i=1}^M, \{x_t^i\}_{i=1}^M\}, M = |\mathcal{D}_t|$, consisting of full set of $\mathcal{D}_t$ and a small random subset of $\mathcal{D}_s$. Each source image $x_s^i \in \mathcal{X}_s^{'}$ receives a negative label and each target image $x_t^i \in \mathcal{X}_t^{'}$ receives a positive label giving us the label set $\mathcal{Y}_h = \{\{0\}_{i=1}^M, \{1\}_{i=1}^M\}$. We then learn θ_h on $\mathcal{D}_h$ using cross entropy loss as

$$_{\theta_h} \sum_{i=1}^{2M} y_h^i log(f_h(x_h^i; \theta_h)) + (1 - y_h^i) log(1 - f_h(x_h^i; \theta_h)). \tag{4}$$

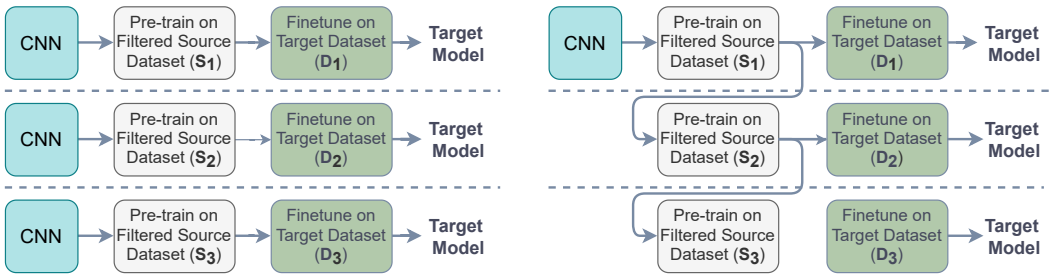

Figure 2. Independent conditional pre-training (**Left**) and sequential conditional pre-training (**Right**) with $P = 3$ target tasks. Sequential pre-training reduces the number of epochs required to pre-train models to accomplish these tasks.

Scoring and Filtering. Once we learn θ_h we obtain the confidence score $p(y_h = 1|x_s^i; \theta_h)$ for each image $x_s^i \in \mathcal{X}_s$. We then sort the source images $\mathcal{X}_s$ in descending order based on $p(y_h = 1|x_s^i; \theta_h)$ and choose the top N' images to create the subset $\mathcal{D}_s^{'} \in \mathcal{D}_s$.

Interpretation. Our method can be interpreted as selecting images from the pre-training dataset with high probability of belonging to the target domain. It can be shown [13] that the Bayes Optimal binary classifier $\hat{f}_h$ assigns probability

$$p(y_h = 1|x_s^i; \theta_h) = \frac{p_t(x_s^i)}{p_s(x_s^i) + p_t(x_s^i)} \tag{5}$$

for an image $x_s^i \in \mathcal{X}_s$ to belong to the target domain, where p_t and p_s are the true data probability distributions for the target and source domains respectively.

Algorithm 3 Sequential Pre-training

procedure SEQUENTIALPRE-TRAIN($\mathcal{T}, \mathcal{T}_{sem}, \mathcal{S}, N^{'}$)
  f = RAND() ▷ Randomly Initialize Model
  while $True$ **do** ▷ Handle All Tasks
    $\mathcal{T}_{sem}$.wait() ▷ Wait for Task Semaphore
    $\mathcal{S}, D_i = \mathcal{T}$.pop() ▷ Current Task from Queue
    $\mathcal{S}_i'$ = FILTER($\mathcal{D}_i, \mathcal{S}, N^{'}$)
    f = TRAIN($f, \mathcal{S}'_i$) ▷ Update Model
    TASK($f, \mathcal{D}_t, T_i$) ▷ Perform Current Task

4.2. Sequential Pre-training

Previously, we treated pre-training for different target tasks independently by pre-training a model from scratch on each conditionally filtered source dataset. In practice, we may be interested in many different target tasks over time, and performing separate pre-training from scratch for each one may hinder efficiency by re-learning basic image features. To avoid it, we propose sequential pre-training where we leverage previously trained models to more quickly pre-train on the next conditionally filtered source dataset.

Formally, we assume that we have a large scale source dataset S (which can potentially grow over time) and want to perform tasks on P target datasets, which we receive sequentially over time as $((S, D_1, t_1), (S, D_2, t_2), \ldots, (S, D_P, t_P))$. We receive our first target task with target dataset D_1 at time t_1, and we conditionally filter S into S_1' based on our data budget. Then, we pre-train a model from scratch on S_1' and finetune it on D_1 to get target model f_{t_1}. Generally, when we receive D_i at time t_i, we filter S conditioned on D_i to obtain S_i'. Then, we take our last pre-training model, trained on S_{i-1}', and update its weights by pre-training on S_i' and finetune on D_i to obtain f_{t_i} to accomplish the current task. Subsequent tasks require smaller and smaller amounts of additional pre-training, thus drastically reducing the total number of pre-training epochs required to accomplish these tasks. We lay out this procedure step by step in Algorithm 3 and visual comparison between independent and sequential conditional pre-training is shown in Figure 2.

4.3. Adjusting Pre-training Spatial Resolution

To further increase the efficiency of pre-training, we propose lowering the spatial resolution of images $\mathcal{X}_s$ in the source dataset $\mathcal{D}_s$ while pre-training. We assume that an image is represented as $x_s^i \in \mathbb{R}^{W_s \times H_s}$ or $x_t^i \in \mathbb{R}^{W_t \times H_t}$ where W_s and W_t represent image width in source and target dataset whereas H_s and H_t represent image height in source and target dataset. Traditionally, after augmentations, we use $W_s, W_t = 224$ and $H_s, H_t = 224$. Here, we consider decreasing W_s and H_s on the pre-training task while maintaining $W_t, H_t = 224$ on the target task. Reducing image resolution while pre-training can provide significant speedups by decreasing FLOPs required by convolution operations, and our experiments show that downsizing image resolution by half $W_s, H_s = 112$ almost halves the pre-training time with negligible loss on the target dataset.

5. Experiments

5.1. Datasets

Source Dataset For our primary source dataset, we utilize ImageNet-2012 [8], with ∼1.28M images over 1000 classes. While full ImageNet is commonly used to pre-train,

Figure 3. High scoring ImageNet samples selected by our conditional filtering methods for Stanford Cars and Caltech Birds.

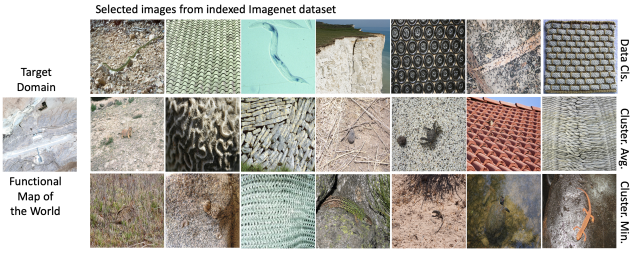

Figure 4. High scoring ImageNet samples selected by our conditional filtering methods for fMoW.

we use it as a proxy for a larger scale dataset that must be filtered from to thoroughly test our methods' finetuning performance under various settings. Thus, we experiment under two data budgets, limiting filtered subsets of ImageNet to 75K ($\sim$6%) and 150K ($\sim$12%) images. This is an appropriate proportion when dealing with pre-training datasets on the scale of tens of millions or more images. We also test our methods in larger scale settings and compile 6.71M images from the Places, OpenImages, ImageNet, and COCO datasets [21, 22, 34] and perform filtering on this source dataset to perform conditional pre-training.

Image Recognition For image recognition tasks, we utilize the Stanford Cars [31], the Caltech Birds [20], and a subset of the Functional Map of the World [6] (fMoW) datasets as target datasets. These datasets lend important diversity to validate the flexibility of our methods. Cars has a fairly small distribution shift from ImageNet, and pre-training on ImageNet performs well on it, but Birds contains a larger shift and pre-training datasets emphasizing natural settings such as iNat perform better [7, 29]. Finally, fMoW, consisting of overhead satellite images, contains images very dissimilar to ImageNet. Additionally, Birds and Cars are fine grained tasks, discriminating between different species of birds or models of cars, respectively. In contrast, fMoW contains more general categories, i.e., buildings and landmarks.

Object Detection and Image Segmentation [14, 15] show that unsupervised ImageNet pre-training is most effective when paired with more challenging low level downstream tasks. Therefore, we also perform experiments in the object detection and semantic segmentation setting to validate the flexibility and adaptability of our methods. To this end, we utilize the Pascal VOC 2007 [9] dataset with unsupervised ImageNet pre-training of the backbone.

5.2. Analyzing Source Dataset Filtering Methods

Domain Classifier Accuracy We typically train the domain classifier to 92-95% accuracy. We empirically find this is the *sweet spot* as classifiers with 88-90% accuracy, perhaps due to not learning relevant features, and 98+% accuracy, perhaps due to over-discriminating minor differences between domains such as noise or color/contrast, do not perform as well.

Efficiency and Adaptability Comparison. The domain classifier trains a simple binary classifier and bypasses full representation learning on a target dataset, computing distances, or clustering. However, this difference in efficiency is small compared to pre-training cost. More importantly, when the target task is not image level classification, the representation learning step for clustering based filtering must be modified in a non-trivial manner. This can involve a global pooling over spatial feature maps while performing object detection or an entirely different setup like unsupervised learning. The domain classifier is more adaptable than clustering as it does not require modification for any type of target task.

Qualitative Analysis. In Figures 3 and 4, we visualize some of the highest scoring filtered images for all our methods on image classification tasks and verify that our filtering methods do select images with *relevant features* to the target task. Unsurprisingly, more interpretable images are selected for Birds and Cars, as there are no satellite images in ImageNet. Nevertheless, we see that the selected images for fMoW stil contain *relevant features* such as color, texture, and shapes.

Supervised Pre-train.		Target Dataset			Cost (hrs)
224 x 224		Small Shift		Large Shift	
Pre-train. Sel. Method		Cars	Birds	fMow	
0%	**Random Init.**	52.89	42.17	43.35	0
100%	**Entire Dataset**	82.63	74.87	59.05	160-180
6%	**Random**	72.2	57.87	50.25	30-35
	Domain Cls.	**74.37**	**59.73**	**51.17**	35-40
	Clustering (Avg)	73.64	56.33	**51.14**	40-45
	Clustering (Min)	**74.23**	57.67	50.27	40-45
12%	**Random**	76.12	62.73	**53.28**	45-50
	Domain Cls.	76.18	**64**	**53.41**	50-55
	Clustering (Avg)	**77.12**	61.73	53.12	55-60
	Clustering (Min)	75.81	**64.07**	52.91	55-60

Supervised Pre-train.		Target Dataset			Cost (hrs)
112 x 112		Small Shift		Large Shift	
Pre-train. Sel. Method		Cars	Birds	fMow	
0%	**Random Init**	52.89	42.17	43.35	0
100%	**Entire Dataset**	83.78	73.47	57.39	90-110
6%	**Random**	72.76	57.4	49.73	15-20
	Domain Cls.	73.66	**58.73**	50.66	20-25
	Clustering (Avg)	**74.53**	56.97	**51.32**	25-30
	Clustering (Min)	71.72	**58.73**	49.06	25-30
12%	**Random**	75.4	62.63	52.59	30-35
	Domain Cls.	76.36	**63.5**	**53.37**	35-40
	Clustering (Avg)	**77.53**	61.23	52.67	40-45
	Clustering (Min)	76.36	63.13	51.6	40-45

Table 1. Target task accuracy and approximate filtering and pre-training cost (time in hrs on 1 GPU) on 3 visual categorization datasets obtained by pre-training on different subsets of the source dataset (ImageNet) with different filtering methods at different resolutions. **Left**: Pre-training with 224×224 pixels images, **Right**: Pre-training with 112×112 pixels images.

MoCo-v2 [15]		Target Dataset			Cost (hrs)
224 x 224		Small Shift		Large Shift	
Pre-train. Sel. Method		Cars	Birds	fMow	
0%	**Random Init.**	52.89	42.17	43.35	0
100%	**Entire Dataset**	83.52	67.49	56.11	210-220
6%	**Random**	75.70	56.82	52.53	20-25
	Domain Cls.	78.67	**61.55**	52.96	23-28
	Clustering (Avg)	78.66	60.88	53.19	25-30
	Clustering (Min)	**79.45**	59.36	**53.5**	25-30
12%	**Random**	75.66	61.70	53.56	30-35
	Domain Cls.	78.68	63.08	54.01	33-38
	Clustering (Avg)	78.68	62.53	**54.4**	35-40
	Clustering (Min)	**79.55**	**63.6**	**54.26**	35-40

MoCo-v2 [15]		Target Dataset			Cost (hrs)
112 x 112		Small Shift		Large Shift	
Pre-train. Sel. Method		Cars	Birds	fMow	
0%	**Random Init**	52.89	42.17	43.35	0
100%	**Entire Dataset**	84.09	66.57	56.83	110-120
6%	**Random**	75.38	56.63	52.59	10-15
	Domain Cls.	76.84	57.93	53.3	13-18
	Clustering (Avg)	76.86	**58.4**	**53.75**	15-20
	Clustering (Min)	**77.53**	57.1	**53.83**	15-20
12%	**Random**	78.35	61.50	54.28	15-20
	Domain Cls.	**80.38**	**63.93**	54.53	18-23
	Clustering (Avg)	80.21	63.50	**55.06**	20-25
	Clustering (Min)	79.63	62.77	**55.03**	20-25

Table 2. Target task accuracy and approximate filtering and pre-training cost (time in hrs on 4 GPUs) on 3 visual categorization datasets obtained by pre-training on different subsets of the source dataset (ImageNet) with different filtering methods at different resolutions. **Left**: Pre-training with 224×224 pixels images, **Right**: Pre-training with 112×112 pixels images.

5.3. Transfer Learning for Image Recognition

5.3.1 Supervised Pre-training Results

We present target task accuracy for all our methods on Cars, Birds, and fMoW along with approximate pre-training and filtering time in Table 1.

Effect of Image Resolution. We see that downsizing pre-training resolution produces gains of up to .5% in classification accuracy on Cars and less than 1% drop in accuracy on Birds and fMoW, while being 30-50% faster than full pre-training. These trends suggest that training on lower resolution images can help the model learn more generalizeable features for similar source and target distributions. This effect erodes slightly as we move out of distribution, however pre-training on lower resolution images offers an attractive trade-off between efficiency and accuracy in all settings.

Impact of Filtering. We find that our filtering techniques consistently provide up to a 2.5% performance increase over random selection, with a relatively small increase in cost. Unsurprisingly, filtering provides the most gains on Cars and Birds where the target dataset has a smaller shift. On fMoW, it is very hard to detect *similar* images to ImageNet, as the two distributions have very little overlap. Nevertheless, in this setting, our filtering methods can still select enough relevant features to provide a 1-2% boost.

Comparison of Filtering Methods. While all our methods perform well, we see that the domain classifier is less variable than clustering and always outperforms random selection. On the other hand, average clustering performs well on Cars or fMoW, but does worse than random on Birds and vice versa for min clustering. These methods rely on computing high dimensional vector distances to assign a measure of similarity, which may explain their volatility since such high dimensional distances are not considered in supervised pre-training.

5.3.2 Unsupervised Pre-training Results

We observe promising results in the supervised setting, but a more realistic and useful setting is the unsupervised set-

ting due to the difficulties inherent in labeling large-scale data. Thus, we use MoCo-v2 [15], a state of the art unsupervised learning method, to pre-train on ImageNet and present results for Cars, Birds, and fMoW in Table 2.

Effect of Image Resolution. We find that in the unsupervised setting, with 150K pre-training images, lower resolution pre-training largely maintains or even improves performance as the target distribution shifts. Unsupervised pre-training relies more on high level features and thus may be better suited than supervised methods for lower resolution pre-training, since higher resolution images may be needed to infer fine grained label boundaries.

Increased Consistency of Clustering. Relative to the supervised setting, clustering based filtering provides more consistent boosts across the different settings and datasets. It is possible that clustering based filtering may be well suited for unsupervised contrastive learning techniques, which also rely on high dimensional feature distances.

Impact of Filtering. Our filtering techniques aim to separate the image distributions based on the true image distributions and feature similarity, not label distribution (which may not be observable). Unsupervised learning naturally takes advantage of our filtering methods, and we see gains of up to 5% over random filtering in the 75K setting and up to 4% in the 150K setting, a larger boost than during supervised pre-training. This leads to performance that is within 1-4% of full unsupervised pre-training but close to 10 times faster, due to using a 12% subset. These results are notable, because we anticipate that unsupervised learning will be the default method for large-scale pre-training and our methods can approach full pre-training while significantly reducing cost.

5.3.3 Sequential Pre-training

Cognizant of the inefficiencies of independent pre-training conditionally on the target tasks, we assume a practical scenario where over time we receive three tasks, D_1, D_2, D_3 representing Cars/Birds/fMoW respectively, with S being ImageNet. We use the domain classifier to filter 150K images, obtain S'_1, S'_2, S'_3, and sequentially pre-train f_p for 100, 40, and 20 epochs respectively with MoCo-v2. In contrast, during independent pre-training we pre-train a separate f_p for 100 epochs for each target task.

We present results in Figure 5. Naturally, for Cars the results do not change, but since learned features are leveraged, not discarded, for subsequent tasks, we observe gains of up to 1% on Birds and 2% on fMoW over Table 2 while using 160 total pre-training epochs vs 300 for independent pre-training. Our sequential pre-training method augments the effectiveness of our filtering methods in settings with many target tasks over time and drastically reduces the number of epochs required. We leave the application of this technique for object detection and segmentation as future work.

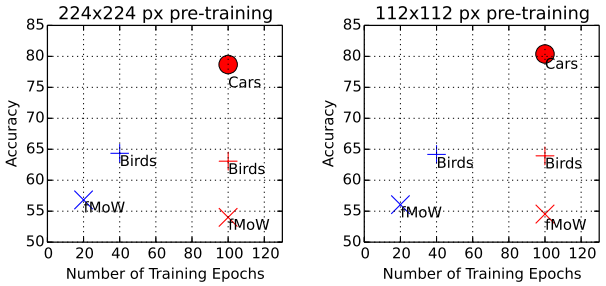

Figure 5. Results for sequential pre-training (blue) vs independent pre-training (red) where pre-training with 224×224 pixels images is shown on the **Left** and with 112×112 pixels images is shown on the **Right**. Our sequential method requires fewer epochs over time with improved accuracy.

5.4. Transfer Learning for Low Level Tasks

Previously we explored image level classification target tasks for conditional pre-training. In this section, we perform experiments on transfer learning for object detection and semantic segmentation on the Pascal VOC 2007 dataset.

We present results in Table 3. For filtering, we use the domain classifier with no modifications and for clustering, we use MoCo-v2 on Pascal VOC 2007 to learn representations.

Effect of Image Resolution. Overall, pre-training on low resolution images produces no overall decrease in performance, with the corresponding 30-50% reduction in training time, confirming the adaptability of pre-training on low resolution images for more challenging low level tasks.

Adaptability Comparison Relative to prior work [7, 33], our clustering method is more adaptable and can efficiently be used for detection/segmentation as well as image classification. However, the representation learning step for clustering must be changed for such target tasks, which can hinder downstream performance as a representation learning technique like MoCo-v2 may be more challenging on smaller scale datasets like Pascal VOC 2007. The domain classifier, on the other hand, avoids these challenges and does not have to change when the target task is changed.

Performance Comparison We observe that all of our proposed filtering techniques yield consistent gains of up to 9% over random filtering, confirming their applicability to lower level tasks. In the segmentation setting, pre-training on a 12 % subset can match full pre-training performance. Clustering produces meaningful gains, but the domain classifier outperforms it in almost every object detection scenario and the majority of segmentation metrics. This is especially pronounced with a larger pre-training subset, showing the domain classifier can filter more relevant images.

Detection		224x224			112x112		
Pre-train. Sel.	Method	AP	AP50	AP75	AP	AP50	AP75
0%	Random Init.	14.51	31.00	11.62	14.51	31.00	11.62
100%	Entire Dataset	43.94	73.05	45.96	43.62	72.56	45.52
6%	Random	29.01	54.02	27.26	28.10	52.82	26.39
	Domain Cls.	**30.47**	**56.58**	29.04	**31.19**	**56.90**	**30.43**
	Clustering (Avg)	**30.61**	55.65	28.75	30.13	55.01	29.47
	Clustering (Min)	**30.44**	56.11	**29.46**	30.39	55.89	28.18
12%	Random	30.84	52.07	29.15	30.56	56.1	29.04
	Domain Cls.	**34.41**	**61.85**	**33.36**	**34.98**	**61.83**	**35.02**
	Clustering (Avg)	32.34	56.24	31.28	32.01	57.16	33.48
	Clustering (Min)	32.58	57.77	31.16	32.96	58.25	33.64

Segmentation		224x224			112x112		
Pre-train. Sel.	Method	mIOU	mAcc	allAcc	mIOU	mAcc	allAcc
0%	Random Init.	0.45	0.55	0.82	0.45	0.55	0.82
100%	Entire Dataset	0.65	0.74	0.89	0.63	0.72	0.88
6%	Random	0.55	0.65	0.85	0.58	0.68	0.87
	Domain Cls.	**0.62**	**0.70**	**0.88**	**0.62**	**0.70**	**0.88**
	Clustering (Avg)	0.61	**0.70**	**0.88**	0.59	0.69	0.87
	Clustering (Min)	0.61	**0.70**	**0.88**	0.61	**0.70**	**0.88**
12%	Random	0.56	0.65	0.86	0.59	0.69	0.87
	Domain Cls.	**0.65**	**0.74**	**0.89**	**0.62**	**0.71**	**0.89**
	Clustering (Avg)	0.64	0.73	**0.89**	0.59	0.68	0.87
	Clustering (Min)	0.61	0.70	0.88	0.61	0.70	0.88

Table 3. Comparison of different source dataset filtering methods and pre-training image resolutions on transfer learning on Pascal-VOC object detection (**Left**) and semantic segmentation (**Right**) tasks. For object detection and semantic segmentation, we use unsupervised pre-training method MoCo-v2 [15] .

ImageNet+	224x224			112x112		
Pre-train. Sel. Method	Cars	Birds	fMow	Cars	Birds	fMow
ImageNet	83.52	67.49	56.11	84.09	66.57	56.83
ImageNet+Domain Cls.	**84.33**	**69.78**	**57.95**	**84.56**	**69.88**	**58.04**

Table 4. Image classification results for ImageNet+ pre-training on three target tasks. By fine-tuning ImageNet weights on our ImageNet filtered subset, we can improve ImageNet pre-training performance on downstream classification tasks.

	POIC-Random@224	POIC-Ours@112	POIC-Ours@224	ImageNet@224
Accuracy	82.96	84.29	**84.51**	83.52
Cost (hrs)	210-220	130-140	230-240	210-220

Table 5. Results on large scale pre-training (MoCo-v2) and fine-tuning on the Stanford Cars dataset, with pre-training resolutions of both 112×112 pixels and 224×224 pixels. Conditionally filtering 1.28M images out of the POIC dataset with the domain classifier improves accuracy on the Stanford Cars dataset over random filtering and ImageNet (1.28M images) pre-training.

5.5. Improving ImageNet Pre-training

Thus far, we have used ImageNet as a proxy for a very large scale dataset to show the promise of our methods in pre-training on task-conditioned subsets. Since pre-trained models on ImageNet (1.28M images) are readily available, we now motivate practical use of our method by showing how they can outperform full ImageNet pre-training.

ImageNet+ Here, we take a model pre-trained on ImageNet (1.28M images) and help it focus on specific examples to our task by tuning its weights for a small number of epochs on our conditionally filtered subset of ImageNet before transfer learning. We apply this method to Cars/Birds/fMoW and tune pre-trained ImageNet weights with MoCo-v2 for 20 additional epochs on 150K domain classifier filtered ImageNet subsets. We present results in Table 4 and report improvements by up to 1-3% over full ImageNet pre-training with minimal extra cost.

Large Scale Filtering Here, we envision a scenario in which a user wants to pre-train a model from scratch for a specific application with access to larger scale data than full ImageNet. To this end, we assemble a large scale dataset, which we call POIC, consisting of 6.71M images from the Places, OpenImages, ImageNet, and COCO datasets [21,22,34]. Next, we filter a subset the size of full ImageNet (1.28M images) using the domain classifier conditioned on the Cars dataset. We pre-train the weights using an unsupervised learning method, MoCo-v2, and present our results on the Cars dataset in Table 5. Our filtering methods improve on the current default of 224 resolution ImageNet pre-training by 1-1.5% with good cost tradeoffs. Interestingly, a random subset of the large scale dataset performs worse than ImageNet, showing that our filtering method is crucial to select relevant examples. Here we are forced to use a 19% subset, but previous experiments showed larger relative gains for 6% in comparison to 12% subsets, so access to even larger scale data ($\gg$6.7M), which should be common in the future, could further improve results. This shows promise that our methods can leverage exponentially growing data scale to replace ImageNet pre-training for specific target tasks.

6. Conclusion

In this work, we proposed filtering methods to efficiently pre-train on large scale datasets conditioned on transfer learning tasks including image recognition, object detection and semantic segmentation. To further improve pre-training efficiency, we proposed decreased image resolution for pre-training and found this shortens pre-training cost by 30-50% with similar transfer learning accuracy. Additionally, we introduced sequential pre-training to improve the efficiency of conditional pre-training with multiple target tasks. Finally, we demonstrated how our methods can improve the standard ImageNet pre-training by focusing models pre-trained on ImageNet on relevant examples and filtering an ImageNet-sized dataset from a larger scale dataset.

References

[1] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018. 2

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[6] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6172–6180, 2018. 5

[7] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018. 2, 3, 5, 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 4

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[10] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017. 2

[11] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1086–1095, 2017. 2, 3

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[13] Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, pages 11058–11070, 2019. 4

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1, 2, 5

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2, 5, 6, 7, 8

[16] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. *arXiv preprint arXiv:1901.09960*, 2019. 1, 2

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[18] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 1, 2

[19] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 1, 2

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5

[21] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, Mar 2020. 5, 8

[22] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 8

[23] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 1, 2

[24] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018. 2

[25] Mijung Park and Jonathan Pillow. Bayesian active learning with localized priors for fast receptive field characterization. *Advances in neural information processing systems*, 25:2348–2356, 2012. 2

[26] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 2

[27] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures,

dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016. 1, 2

[28] Burak Uzkent, Evan Sheehan, Chenlin Meng, Zhongyi Tang, Marshall Burke, David B Lobell, and Stefano Ermon. Learning to interpret satellite images using wikipedia. In *IJCAI*, pages 3620–3626, 2019. 2

[29] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 5

[30] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016. 2

[31] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 5

[32] Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. *arXiv preprint arXiv:1510.00098*, 2015. 1, 2

[33] Xi Yan, David Acuna, and Sanja Fidler. Neural data server: A large-scale search engine for transfer learning data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 7

[34] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5, 8