ImageNet-21K Pretraining for the Masses

Tal Ridnik
DAMO Academy, Alibaba Group
tal.ridnik@alibaba-inc.com

Emanuel Ben-Baruch
DAMO Academy, Alibaba Group
emanuel.benbaruch@alibaba-inc.com

Asaf Noy
DAMO Academy, Alibaba Group
asaf.noy@alibaba-inc.com

Lihi Zelnik-Manor
DAMO Academy, Alibaba Group
lihi.zelnik@alibaba-inc.com

Abstract

ImageNet-1K serves as the primary dataset for pretraining deep learning models for computer vision tasks. ImageNet-21K dataset, which is bigger and more diverse, is used less frequently for pretraining, mainly due to its complexity, low accessibility, and underestimation of its added value. This paper aims to close this gap, and make high-quality efficient pretraining on ImageNet-21K available for everyone. Via a dedicated preprocessing stage, utilization of WordNet hierarchical structure, and a novel training scheme called semantic softmax, we show that various models significantly benefit from ImageNet-21K pretraining on numerous datasets and tasks, including small mobile-oriented models. We also show that we outperform previous ImageNet-21K pretraining schemes for prominent new models like ViT and Mixer. Our proposed pretraining pipeline is efficient, accessible, and leads to SoTA reproducible results, from a publicly available dataset. The training code and pretrained models are available at: https://github.com/Alibaba-MIIL/ImageNet21K

1 Introduction

ImageNet-1K dataset, introduced for the ILSVRC2012 visual recognition challenge [45], has been at the center of modern advances in deep learning [30, 20, 46]. ImageNet-1K serves as the main dataset for pretraining of models for computer-vision transfer learning [51, 33, 21], and improving performances on ImageNet-1K is often seen as a litmus test for general applicability on downstream tasks [28, 62, 44]. ImageNet-1K is a subset of the full ImageNet dataset [11], which consists of 14,197,122 images, divided into 21,841 classes. We shall refer to the full dataset as ImageNet-21K, following [27] (although other papers sometimes described it as ImageNet-22K [8]). ImageNet-1K was created by selecting a subset of 1.2M images from ImageNet-21K, that belong to 1000 mutually exclusive classes.

Even though some previous works showed that pretraining on ImageNet-21K could provide better downstream results for large models [27, 14], pretraining on ImageNet-1K remained far more popular. A main reason for this discrepancy is that ImageNet-21K labels are not mutually exclusive - the labels are taken from WordNet [38], where each image is labeled with one label only, not necessarily at the highest possible hierarchy of WordNet semantic tree. For example, ImageNet-21K dataset contains the labels "chair" and "furniture". A picture, with an actual chair, can sometimes be labeled as "chair", but sometimes be labeled as the semantic parent of "chair", "furniture". This kind of tagging methodology complicates the training process, and makes evaluating models on ImageNet-21K less accurate. Other challenges of ImageNet-21K dataset are the lack of official train-validation split, the fact that training is longer than ImageNet-1K and requires highly efficient training schemes, and that the raw dataset is large - 1.3TB.

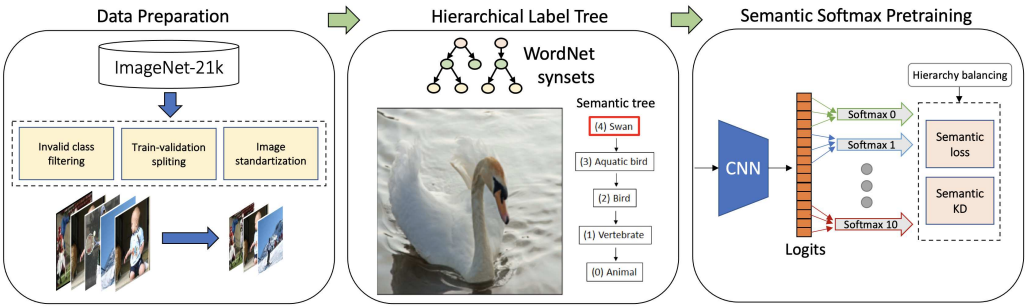

Figure 1: Our end-to-end pretraining pipeline on ImageNet-21K. We start with a dataset preparation and preprocessing stage. Via WordNet's synsets, we convert all the single-label inputs to semantic multi-labels, resulting in a semantic structure for ImageNet-21K, with 11 possible hierarchies. For each hierarchy, we apply a dedicated softmax activation, and aggregate the losses with hierarchy balancing.

Several past works have used ImageNet-21K for pretraining, mostly in comparison to larger datasets, which are not publicly available, such as JFT-300M [49]. [40] and [43] used ImageNet-21K and JFT-300M to train expert models according to the datasets hierarchies, and combined them to ensembles on downstream tasks; [27] and [14] compared pretraining JFT-300M to ImageNet-21K on large models such as ViT and ResNet-50x4. Many papers used these pretrained models for downstream tasks (e.g., [63, 41, 36, 1]). There are also works on ImageNet-21K that did not focus on pretraining: [61] used extra (unlabled) data from ImageNet-21K to improve knowledge-distillation training on ImageNet-1K; [13] used ImageNet-21k for testing few-shot learning; [56] tested efficient softmax schemes on ImageNet-21k; [17] tested pooling operations schemes on animal-oriented subset of ImageNet-21k.

However, previous works have not methodologically studied and optimized a pretraining process specifically for ImageNet-21K. Since this is a large-scale, high-quality, publicly available dataset, this kind of study can be highly beneficial to the community. We wish to close this gap in this work, and make efficient top-quality pretraining on ImageNet-21K accessible to all deep learning practitioners.

Our pretraining pipeline starts by preprocessing ImageNet-21K to ensure all classes have enough images for a meaningful learning, splitting the dataset to a standardized train-validation split, and resizing all images to reduce memory footprint. Using WordNet semantic tree [38], we show that ImageNet-21K can be transformed into a (semantic) multi-label dataset. We thoroughly analyze the advantages and disadvantages of single-label and multi-label training. Extensive tests on downstream tasks show that multi-label pretraining does not improve results on downstream tasks, despite having more information per image. To effectively utilize the semantic data, we develop a novel training method, called *semantic softmax*, which exploits the hierarchical structure of ImageNet-21K tagging to train the network over several semantic softmax layers, instead of the single layer. Using semantic softmax pretraining, we consistently outperform both single-label and multi-label pretraining on downstream tasks. By integrating semantic softmax into a dedicated semantic knowledge distillation loss, we further improved results. The complete end-to-end pretraining pipeline appears in Figure 1.

Using semantic softmax pretraining on ImageNet-21K we achieve significant improvement on numerous downstream tasks, compared to standard ImageNet-1K pretraining. Unlike previous works, which focused on pretraining of large models only [27], we show that ImageNet-21K pretraining benefits a wide variety of models, from larger models like TResNet-L [44], through medium-sized models like ResNet50 [20], and even small mobile-dedicated models like OFA-595 [5] and MobileNetV3 [21]. Our proposed pretraining scheme also outperforms previous ImageNet-21K pretraining schemes that were used to trained MLP-based models like Vision-Transformer (ViT) [14] and Mixer [53].

The paper's contribution can be summarized as follows:

- We develop a methodical preprocess procedure to transform raw ImageNet-21K into a viable dataset for efficient, high-quality pretraining.
- Using WordNet semantic tree, we convert each (single) label to semantic multi labels, and compare the pretrain quality of two baseline methods: single-label and multi-label pretraining. We

show that while a multi-label approach provides more information per image, it can have significant optimization drawbacks, resulting in inferior results on downstream tasks.
- We develop a novel training scheme called semantic softmax, which exploits the hierarchical structure of ImageNet-21K. With semantic softmax pretraining, we outperform both single-label and multi-label pretraining on downstream tasks. We further improve results by integrating semantic softmax into a dedicated semantic knowledge distillation scheme.
- Via extensive experimentations, we show that compared to ImageNet-1K pretraining, ImageNet-21K pretraining significantly improves downstream results for a wide variety of architectures, include mobile-oriented ones. In addition, our ImageNet-21K pretraining scheme consistently outperforms previous ImageNet-21K pretraining schemes for prominent new models like ViT and Mixer.

2 Dataset Preparation

2.1 Preprocessing ImageNet-21K

Our preprocessing stage consists of three steps, as described in Figure 1 (leftmost image): (1) invalid classes cleaning, (2) creating a validation set, (3) image resizing. Details are as follows:
Step 1 - cleaning invalid classes: the original ImageNet-21K dataset [11] consists of 14,197,122 images, each tagged in a single-label fashion by one of 21,841 possible classes. The dataset has no official train-validation split, and the classes are not well-balanced - some classes contain only 1-10 samples, while others contain thousands of samples. Classes with few samples cannot be learned efficiently, and may hinder the entire training process and hurt the pretrain quality [23]. Hence we start our preprocessing stage by removing infrequent classes, with less than 500 labels. After this stage, the dataset contains 12,358,688 images from 11,221 classes. Notice that the cleaning process reduced the number of total classes by half, but removed only 13% of the original pictures.
Step 2 - validation split: we allocate 50 images per class for a standardized validation split, that can be used for future benchmarks and comparisons.
Step 3 - image resizing: ImageNet-1K training usually uses *crop-resizing* [22] which favours loading the original images at full resolution and resizing them on-the-fly. To make ImageNet-21K dataset more accessible and accelerate training, we resized during the preprocessing stage all the images to 224 resolution (equivalent to *squish-resizing* [22]). While somewhat limiting scale augmentations, this stage significantly reduces the dataset's memory footprint, from 1.3TB to 250GB, and makes loading the data during training faster.

After finishing the preprocessing stage, we kept only valid classes, produced a standardized train-validation split, and significantly reduced the dataset size. We shall name this processed dataset **ImageNet-21K-P** (P for Processed).

2.2 Utilizing Semantic Data

We now wish to analyze the semantic structure of ImageNet-21K-P dataset. This structure will enable us to better understand ImageNet-21K-P tagging methodology, and employ and compare different pretraining schemes.

From single labels to semantic multi labels Each image in the original ImageNet-21K dataset was labeled with a single label, that belongs to WordNet synset [38]. Using the WordNet synset hyponym (subtype) and hypernym (supertype) relations, we can obtain for each class its parent class, if exists, and a list of child classes, if exists. When applying the parenthood relation recursively, we can build a semantic tree, that enables us to transform ImageNet-21K-P dataset into a multi-label dataset, where each image is associated with several labels - the original label, and also its parent class, parent-of-parent class, and so on. Example is given in Figure 1 (middle image) - the original image was labeled as 'swan', but by utilizing the semantic tree, we can produce a list of semantic labels for the image - 'animal, vertebrate, bird, aquatic bird, swan'. Notice that the labels are sorted by hierarchy: 'animal' label belongs to hierarchy 0, while 'swan' label belongs to hierarchy 4. A label from hierarchy k has k ancestors.

Understanding the inconsistent tagging methodology The semantic structure of ImageNet-21K enables us to understand its tagging methodology better. According to the stated tagging methodology of ImageNet-21K [11], we are not guaranteed that each image was labeled at the highest

Figure 2: Example of inconsistent tagging in ImageNet-21K dataset. Two pictures containing the same animal were labeled differently.

Hierarchy	Example Classes
0	person, animal, plant, food, artifact
1	domestic animal, basketball court, clothing
...	
6	whitetip shark, ortolan, grey kingbird

Table 1: Examples of classes from different ImageNet-21K-P hierarchies.

possible hierarchy. An example is given in Figure 2. Two pictures, that contain the animal cow, were labeled differently - one with the label 'animal', the other with the label 'cow'. Notice that 'animal' is a semantic ancestor of 'cow' (cow $\rightarrow$ placental $\rightarrow$ mammal $\rightarrow$ vertebrate $\rightarrow$ animal). This kind of incomplete tagging methodology, which is common in large datasets [32, 42], hinders and complicates the training process. A dedicated scheme that tackles this tagging methodology will be presented in section 3.3.

Semantic statistics By using WordNet synsets, we can calculate for each class the number of ancestors it has - its hierarchy. In total, our processed dataset, ImageNet-21K-P, has 11 possible hierarchies. Example of classes from different hierarchies appears in Table 1. In Figure 4 in appendix A we present the number of classes per hierarchy. We see that while there are 11 possible hierarchies, the vast majority of classes belong to the lower hierarchies.

3 Pretraining Schemes

In this section, we will review and analyze two baseline schemes for pretraining on ImageNet-21K-P: single-label and multi-label training. We will also develop a novel new scheme for pretraining on ImageNet-21K-P, *semantic softmax*, and analyze its advantages over the baseline schemes.

3.1 Single-label Training Scheme

The straightforward way to pretrain on ImageNet-21K-P is to use the original (single) labels, apply softmax on the output logits, and use cross-entropy loss. Our single-label training scheme is similar to common efficient training schemes on ImageNet-1K [44], with minor adaptations to better handle the inconsistent tagging (Full training details appear in appendix B.1). Since we aim for an efficient scheme with maximal throughput, we don't incorporate any tricks that might significantly increase training times. To further shorten the training times, we propose to initialize the models from standard ImageNet-1K training, and train on ImageNet-21K-P for 80 epochs. On 8xV100 NVIDIA GPU machine, mixed-precision training takes 40 minutes per epoch for ResNet50 and TResNet-M architectures ($\sim 5000\frac{\text{img}}{\text{sec}}$), leading to a total training time of 54 hours. Similar accuracies are obtained when doing random initialization, but training the models longer - 140 epochs.

Pros of using single-label training

- **Well-balanced dataset** - with single-label training on ImageNet-21K-P, the dataset is well-balanced, meaning each class appears, roughly, the same number of times.
- **Single-loss training** - training with a softmax (a single loss) makes convergence easy and efficient, and avoids many optimization problems associated with multi-loss learning, such as different gradient magnitudes and gradient interference [60, 7, 9].

Cons of using single-label training

- **Inconsistent tagging** - due to the tagging methodology of ImageNet-21K-P, where we are not guaranteed that an image was labeled at the highest possible hierarchy, ground-truth labels are inherently inconsistent. Pictures, containing the same object, can appear with different single-label tagging (see Figure 2 for example).

- **No semantic data** - during training, we are not presenting semantic data via the single-label ground-truth.

3.2 Multi-label Training Scheme

Using the semantic tree, we can convert any (single) label to semantic multi labels, and train our models on ImageNet-21K-P in a multi-label fashion, expecting that the additional semantic information per image will improve the pretrain quality. As commonly done in multi-label classification [3], we reduce the problem to a series of binary classification tasks. Given N labels, the base network outputs one logit per label, z_n, and each logit is independently activated by a sigmoid function $\sigma(z_n)$. Let's denote y_n as the ground-truth for class n. The total classification loss, L_{tot}, is obtained by aggregating a binary loss from the N labels:

$$L_{\text{tot}} = \sum_{n=1}^{N} L\left(\sigma(z_n), y_n\right). \tag{1}$$

Eq. 1 formalizes multi-label classification as a multi-task problem. Since we have a large number of classes $(11,221)$, this is an extreme multi-task case. For training, we adopted the high-quality training scheme described in [3], that provided state-of-the-art results on large-scale multi-label datasets such as Open Images [32]. Full training details appear in appendix B.2.

Pros of using multi-label training

- **More information per image** - we present for each image all the available semantic labels.
- **Tagging and metrics are more accurate** - if an image was originally given a single label at hierarchy k, with multi-label training we are guaranteed that all ground-truth labels at hierarchies 0 to k are accurate. Hence, multi-label training partly mitigates the inconsistent tagging problem, and makes training metrics more accurate and reflective than single-label training.

Cons of using multi-label training

- **Extreme multi-tasking** - with multi-label training, each class is learned separately (sigmoids instead of softmax). This extreme multi-task learning makes the optimization process harder and less efficient, and may cause convergences to a local minimum [60, 7, 15].
- **Extreme imbalancing** - as a multi-label dataset with many classes, ImageNet-21K-P suffers from a large positive-negative imbalance [3]. In addition, due to the semantic structure, multi-label training is hindered by a large class imbalance [24] - on average, classes from a lower hierarchy will appear far more frequent than classes from a higher hierarchy.

In appendices C.2 and E we show that for multi-label training, ASL loss [3], that was designed to cope with large positive-negative imbalancing, significantly outperforms cross-entropy loss, both on upstream and downstream tasks. This supports our analysis of extreme imbalancing as a major optimization challenge of multi-label training. Notice that we also list extreme multi-tasking as another optimization pitfall of multi-label training, and a dedicated scheme for dealing with it might further improve results. However, most methods that tackle multi-task learning, such as GradNorm [7] and PCGrad [60], require computation of gradients for each class separately. This is computationally infeasible for a dataset with a large number of classes, such as ImageNet-21K-P.

3.3 Semantic Softmax Training Scheme

Our goal is to develop a dedicated training scheme that utilizes the advantages of both the single-label and the multi-label training. Specifically, our scheme should present for each input image all the available semantic labels, but use softmax activations instead of independent sigmoids to avoid extreme multi-tasking. We also want to have fully accurate ground-truth and training metrics, and provide the network direct data on the semantic hierarchies (this is not achieved even in multi-label training, the hierarchical structure there is implicit). In addition, the scheme should remain efficient in terms of training times.

Semantic softmax formulation To meet these goals, we develop a new training scheme called *semantic softmax* training. As we saw in section 2.2, each label in ImageNet-21K-P can belong to

one of 11 possible hierarchies. By definition, for each hierarchy there can be only one ground-truth label per input image. Hence, instead of single-label training with a single softmax, we shall have 11 softmax layers, for the 11 different hierarchies. Each softmax will sample the relevant logits from the corresponding hierarchy, as shown in Figure 1 (rightmost image). To deal with the partial tagging of ImageNet-21K-P, not all softmax layers will propagate gradients from each sample. Instead, we will activate only softmax layers from the relevant hierarchies. An example is given in Figure 3 - the original image had a label from hierarchy 5. We transform it to 6 semantic ground-truth labels, for hierarchies 0-5, and activate only the 6 first semantic softmax layers (only activated layers will propagate gradients). Compared to single-label and multi-label schemes, semantic softmax training scheme has the following advantages:

1. We avoid extreme multi-tasking ($11,221$ uncoupled losses in multi-label training). Instead, we have only 11 losses, as the number of softmax layers.
2. We present for each input image all the possible semantic labels. The loss scheme even provides direct data on the hierarchical structure.
3. Unlike single-label and multi-label training, semantic softmax ground-truth and training metrics are fully accurate. If a sample has no labels at hierarchy k, we don't propagate gradients from the kth softmax during training, and ignore that hierarchy for metrics calculation (A dedicated metrics for semantic softmax training is defined in appendix C.3).
4. Calculating several softmax activations instead of a single one has negligible overhead, and in practice training times are similar to single-label training.

Weighting the different softmax layers For each input image we have K losses (11). As commonly done in multi-task training [7], we need to aggregate them to a single loss. A naive solution will be to sum them: $L_{\text{tot}} = \sum_{k=0}^{K-1} L_k$ where L_k, the loss per softmax layer, is zero when the layer is not activated. However, this formulation ignores the fact that softmax layers at lower hierarchies will be activated much more frequently than softmax layers at higher hierarchies, resulting in over-emphasizing of classes from lower hierarchies. To account for this imbalancing, we propose a balancing logic: let N_j be the total number of classes in hierarchy j (as presented in Figure 4). Due to the semantic structure, the relative number of occurrences of hierarchy k in the loss function will be:

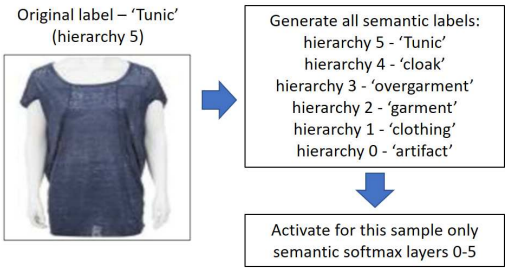

Figure 3: Gradient propagation logic of semantic softmax training.

$$O_{\text{k}} = \sum_{j=0}^{k-1} N_j \tag{2}$$

Hence, to balance the contribution of different hierarchies we can use a normalization factor $W_k = \frac{1}{O_k}$, and obtain a balanced aggregation loss, that will be used for semantic softmax training:

$$L_{\text{tot}} = \sum_{k=0}^{K-1} W_k L_k \tag{3}$$

3.4 Semantic Knowledge Distillation

Knowledge distillation (KD) is a known method to improve not only upstream, but also downstream results [58, 59, 61]. We want to combine our semantic softmax scheme with KD training - *semantic KD*. In addition to the general benefit from KD training [19], for ImageNet-21K-P semantic KD has an additional benefit - it can predict the missing tags that arise from the inconsistent tagging. For example, for the left picture in Figure 2, the teacher model can predict the missing labels - 'cow, placental, mammal, vertebrate'. To implement semantic KD loss, for each hierarchy we will calculate both the teacher and the student the corresponding probability distributions $\{T_i\}_{i=0}^{K-1}$, $\{S_i\}_{i=0}^{K-1}$. The KD loss of hierarchy i will be:

$$L_{\text{KD}_i} = \text{KDLoss}(T_i, S_i) \tag{4}$$

where KDLoss is a standard measurement for the distance between distributions, that can be chosen as Kullback-Leibler divergence [19, 58], or as MSE loss [2, 52]. We have found that the latter converges faster, and used it. A vanilla implementation for the total loss will be a simple sum of the losses from different hierarchies: $L_{\text{KD}} = \sum_{i=0}^{K-1} L_{\text{KD}i}$. However, this formulation assumes that all the hierarchies are relevant for each image. This is inaccurate - usually higher hierarchies represent subspecies of animals or plants, and are not applicable for a picture of a chair, for example. So we need to determine from the teacher predictions which hierarchies are relevant, and weigh the different losses accordingly. Let's assume that for each hierarchy we can calculate the teacher confidence level, P_i. A confidence-weighted KD loss will be:

$$L_{\text{KD}} = \sum_{i=0}^{K-1} P_i L_{\text{KD}i} \tag{5}$$

Eq. 5 is our proposed semantic KD loss. In appendix F we present a method to calculate the teacher confidence level, P_i, from the teacher predictions, similar to [58].

4 Experimental Study

In this section, we will present upstream and downstream results for the different training schemes, and show that semantic softmax pretraining outperforms single-label and multi-label pretraining. We will also demonstrate how semantic KD further improves results on downstream tasks.

4.1 Upstream Results

In appendix C we provide upstream results for the three training schemes. Since each scheme has different training metrics, we cannot use these results to directly compare (pre)training quality.

4.2 Downstream Results

To compare the pretrain quality of different training schemes, we will test our models via transfer learning. To ensure that we are not overfitting a specific dataset or task, we chose a wide variety of downstream datasets, from different computer-vision tasks. We also ensured that our downstream datasets represent a variety of domains, and have diverse sizes - from small datasets of thousands of images, to larger datasets with more than a million images. For single-label classification, we transferred our models to ImageNet-1K [30], iNaturalist 2019 [55], CIFAR-100 [29] and Food 251 [25]. For multi-label classification, we transferred our models to MS-COCO [34] and Pascal-VOC [16] datasets. For video action recognition, we transferred our models to Kinetics 200 dataset [26]. In appendix D we provide full training details on all downstream datasets.

Comparing different pretraining schemes In Table 2 we compare downstream results for three pretraining schemes: single-label, multi-label and semantic softmax. We see that on 6 out of 7

Dataset	Single Label Pretrain	Mutli Label Pretrain	Semantic Softmax Pretrain	Semantic Softmax Pretrain + KD
ImageNet1K$^{(1)}$	81.1	81.0	81.4	82.2
iNaturalist$^{(1)}$	71.5	71.0	72.0	72.7
Food 251$^{(1)}$	75.4	75.2	75.8	76.1
CIFAR 100$^{(1)}$	89.5	90.6	90.4	91.7
MS-COCO$^{(2)}$	80.8	80.6	81.3	82.2
Pascal-VOC$^{(2)}$	88.1	87.9	89.7	89.8
Kinetics 200$^{(3)}$	81.9	81.9	83.0	84.4

Table 2: Comparing downstream results for different pretraining schemes. Darker cell color means better score. Dataset types and metrics: (1) - single-label, top-1 Acc.[%] ; (2) - multi-label, mAP [%]; (3) - action recognition, top-1 Acc. [%].

datasets tested, semantic softmax pretraining outperforms both single-label and multi-label pretrain-

ing. In addition, we see from Table 2 that single-label pretraining performs better than multi-label pretraining (scores are higher on 5 out of 7 datasets tested).

These results support our analysis of the pros and cons of the different pretraining schemes from Section 3: with multi-label training, we have more information per input image, but the optimization process is less efficient due to extreme multi-tasking and extreme imbalancing. All-in-all, multi-label training does not improve downstream results. Single-label training, despite its shortcomings from the partial tagging methodology and the minimal information per image, provides a better pre-training baseline. Semantic softmax scheme, which utilizes semantic data without the optimization pitfalls of extreme multi-label training, outperforms both single-label and multi-label training.

Semantic KD In Table 2 we also compare the downstream results of semantic softmax pretraining, with and without semantic KD. We see that on all tasks and datasets tested, adding semantic KD to our pretraining process improves downstream results. Indeed the ability of semantic KD to fill in the missing tags and provide a smoother and more informative ground-truth is translated to better downstream results. In appendix G we compare single-label pretraining with KD, to semantic softmax pretraining with semantic KD, and show that the latter achieves better results on downstream tasks.

5 Results

In the previous chapters we developed a dedicated pretraining scheme for ImageNet-21K-P dataset, semantic softmax, and showed that it outperforms two baseline pretraining schemes, single-label and multi-label, in terms of downstream results. Now we wish to compare our semantic softmax pretraining on ImageNet-21K-P to other known pretraining schemes and pretraining datasets.

5.1 Comparison to Other ImageNet-21K Pretraining Schemes

We want to compare our proposed training scheme to other ImageNet-21K training schemes from the literature. However, to the best of our knowledge, no previous works have published their upstream results on ImageNet-21K, or shared thorough details about their training scheme or preprocessing stage. Recently, prominent new models called ViT [14] and Mixer [53] were published, and official pretrained weights were released [18]. In Table 3 we compare downstream results when using the official ImageNet-21K weights, and when using weights from semantic softmax pretraining.

	ViT-B-16		Mixer-B-16	
Dataset	Official ImageNet-21K Pretrain	Our ImageNet-21K Pretrain	Official ImageNet-21K Pretrain	Our ImageNet-21K Pretrain
ImageNet1K[(1)]	83.3	**83.9**	79.7	**82.0**
iNaturalist[(1)]	71.7	**73.1**	62.2	**66.6**
Food 251[(1)]	74.6	**76.0**	69.9	**74.5**
CIFAR 100[(1)]	92.7	**94.2**	85.5	**92.3**
MS-COCO[(2)]	81.1	**82.6**	74.1	**80.9**
Pascal-VOC[(2)]	78.7	**93.1**	63.1	**88.6**
Kinetics 200[(3)]	82.7	**84.1**	79.3	**82.1**

Table 3: Comparing downstream results for different pretraining schemes. Dataset types and metrics: (1) - single-label, top-1 Acc.[%] ; (2) - multi-label, mAP [%]; (3) - action recognition, top-1 Acc. [%].

We see from Table 3 that our pretraining scheme significantly outperforms the official pretrain, on all downstream tasks tested. Previous works have observed that MLP-based models can be harder and less stable to use in transfer learning since they don't have inherent translation inductive bias [6, 39, 35]. When using the official weights, we also noticed this phenomenon on some datasets (Pascal-VOC, for example). Using semantic softmax pretraining, the transfer learning training was more stable and robust, and reached higher accuracy.

Dataset	MobileNetV3		OFA595		ResNet50		TResNet-M		TResNet-L	
	1K	21K	1K	21K	1K	21K	1K	21K	1K	21K
iNaturalist[(1)]	62.4	**65.0**	69.0	**71.5**	66.8	**71.4**	70.1	**72.7**	72.4	**74.8**
CIFAR100[(1)]	86.7	**88.5**	88.3	**90.3**	86.8	**90.3**	89.5	**91.7**	90.2	**92.5**
Food 251[(1)]	70.1	**70.3**	72.9	**73.5**	72.2	**74.0**	75.1	**76.1**	76.3	**77.0**
MS-COCO[(2)]	73.0	**74.9**	74.9	**77.7**	76.7	**80.5**	79.5	**82.2**	81.1	**83.7**
Pascal-VOC[(2)]	72.1	**72.4**	72.4	**81.5**	86.9	**87.9**	85.8	**89.8**	88.2	**92.5**
Kinetics200[(3)]	72.2	**74.3**	73.2	**78.1**	78.2	**81.3**	80.5	**84.3**	82.1	**84.6**

Table 4: Comparing downstream results for ImageNet-1K standard pretraining, and our proposed ImageNet-21K-P pretraining scheme. (1) - single-label dataset, top-1 Acc [%] metric; (2) - multi-label dataset, mAP [%] metric; (3) - action recognition dataset, top-1 Acc [%] metric.

5.2 Comparison to ImageNet-1K Pretraining

In Table 4 we compare downstream results, for different models, when using ImageNet-1K pretraining (taken from [57]), and when using our ImageNet-21K-P pertraining. We can see that our pretraining scheme significantly outperforms standard ImageNet-1K pretraining on all datasets, for all models tested. For example, on iNaturalist dataset we improve the average top-1 accuracy by 2.9%.

Notice that some previous works stated that pretraining on a large dataset benefits only large models [27, 49]. MobileNetV3 backbone, for example, has only 4.2M parameters, while ViT-B model has 85.6M parameters. Previous works assumed that a large number of parameters, like ViT has, is needed to properly utilize pretraining on large datasets. However, we show consistently and significantly that even small mobile-oriented models, like MobileNetV3 and OFA-595, can benefit from pretraining on a large (publicly available) dataset like ImageNet-21K-P. Due to their fast inference times and reduced heating, mobile-oriented models are used frequently for deployment. Hence, improving their downstream results by using better pretrain weights can enhance real-world products, without increasing training complexity or inference times.

5.3 ImageNet-1K SoTA Results

In Table 10 in appendix H we bring downstream results on ImageNet-1K for different models, when using ImageNet-21K-P semantic softmax pretraining. To achieve top results, similar to previous works [33, 61, 58], we added standard knowledge distillation loss into our ImageNet-1K training. To the best of our knowledge, for all the models in Table 10 we achieve a new SoTA record (for input resolution 224). Unlike previous top works, which used private datasets [49], we are using a publicly available dataset for pretraining. Note that the gap from the original reported accuracies is significant. For example, MobileNetV3 reported accuracy was 75.2% [21] - we achieved 78.0%; ResNet50 reported accuracy was 76.0% [20] - we achieved 82.0%.

5.4 Additional Comparisons and Results

In appendix J we bring additional comparisons: (1) Comparison to Open Images pretraining; (2) Downstream results comparison on additional non-classification computer-vision tasks; (3) Impact of different number of training samples on upstream results.

6 Conclusion

In this paper, we presented an end-to-end scheme for high-quality efficient pretraining on ImageNet-21K dataset. We start by standardizing the dataset preprocessing stage. Then we show how we can transform ImageNet-21K dataset into a multi-label one, using WordNet semantics. Via extensive tests on downstream tasks, we demonstrate how single-label training outperforms multi-label training, despite having less information per image. We then develop a new training scheme, called semantic softmax, which utilizes ImageNet-21K hierarchical structure to outperform both single-label and multi-label training. We also integrate the semantic softmax scheme into a dedicated knowledge distillation loss to further improve results. On a variety of computer vision datasets and tasks, dif-

ferent architectures significantly and consistently benefit from our pretraining scheme, compared to ImageNet-1K pretraining and previous ImageNet-21K pretraining schemes.

Broader Impact In the past, pretraining on ImageNet-21K was out of scope for the common deep learning practitioner. With our proposed pipeline, high-quality efficient pretraining on ImageNet-21K will be more accessible to the deep learning community, enabling researchers to design new architectures and pretrain them to top results, without the need for massive computing resources or large-scale private datasets. In addition, our findings that even small mobile-oriented models significantly benefit from large-scale pretraining can be used to enhance real-world products. Finally, our improved pretraining scheme on ImageNet-21K can support prominent MLP-based models that require large-scale pretraining, like ViT and Mixer.

References

[1] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, Vivek Natarajan, and Mohammad Norouzi. Big self-supervised models advance medical image classification, 2021.

[2] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *arXiv preprint arXiv:1312.6184*, 2013.

[3] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. *arXiv preprint arXiv:2009.14119*, 2020.

[4] Clemens-Alexander Brust and Joachim Denzler. Integrating domain knowledge: using hierarchies to improve deep classifiers. In *Asian Conference on Pattern Recognition*, pages 3–16. Springer, 2019.

[5] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[6] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[7] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[8] Valeriu Codreanu, Damian Podareanu, and Vikram Saletore. Scale out for large minibatch sgd: Residual network training on imagenet-1k with improved accuracy and reduced time to train. *arXiv preprint arXiv:1711.04291*, 2017.

[9] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[10] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[13] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

[16] 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.

[17] Wonjoon Goo, Juyong Kim, Gunhee Kim, and Sung Ju Hwang. Taxonomy-regularized semantic deep convolutional neural networks. In *European Conference on Computer Vision*, pages 86–101. Springer, 2016.

[18] Google. vit pretrained weights. https://console.cloud.google.com/storage/browser/vit_models, 2021.

[19] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, pages 1–31, 2021.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[22] Jeremy Howard and Sylvain Gugger. Fastai: A layered api for deep learning. *Information*, 11(2):108, 2020.

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

[24] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[25] Parneet Kaur, Karan Sikka, Weijun Wang, Serge Belongie, and Ajay Divakaran. Foodx-251: a dataset for fine-grained food classification. *arXiv preprint arXiv:1907.06167*, 2019.

[26] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[27] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6(2):8, 2019.

[28] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *arXiv preprint*, 2009.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[31] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[32] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

[33] Jungkyu Lee, Taeryun Won, Tae Kwan Lee, Hyemin Lee, Geonmo Gu, and Kiho Hong. Compounding the performance improvements of assembled techniques in a convolutional neural network. *arXiv preprint arXiv:2001.06268*, 2020.

[34] 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.

[35] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*, 2020.

[36] Wei Liu, Sihan Chen, Longteng Guo, Xinxin Zhu, and Jing Liu. Cptr: Full transformer network for image captioning. *arXiv preprint arXiv:2101.10804*, 2021.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[38] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[39] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*, 2020.

[40] Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Daniel Keysers, Neil Houlsby, et al. Deep ensembles for low-data transfer learning. *arXiv preprint arXiv:2010.06866*, 2020.

[41] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

[42] 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.

[43] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, Cedric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. Scalable transfer learning with expert models. *arXiv preprint arXiv:2009.13239*, 2020.

[44] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1400–1409, 2021.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[47] Gilad Sharir, Asaf Noy, and Lihi Zelnik-Manor. An image is worth 16x16 words, what is a video worth?, 2021.

[48] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[49] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[50] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[51] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[52] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[53] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[54] Michael J Trammell, Priyanka Oberoi, James Egenrieder, and John Kaufhold. Contextual label smoothing with a phylogenetic tree on the inaturalist 2018 challenge dataset. *Washington Academy of Sciences. Journal of the Washington Academy of Sciences*, 105(1):23–45, 2019.

[55] 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.

[56] Sudheendra Vijayanarasimhan, Jonathon Shlens, Rajat Monga, and Jay Yagnik. Deep networks with large output spaces. *arXiv preprint arXiv:1412.7479*, 2014.

[57] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[58] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[59] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. Clusterfit: Improving generalization of visual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6509–6518, 2020.

[60] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[61] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. *arXiv preprint arXiv:2101.05022*, 2021.

[62] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[63] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020.

Appendices

A Number of Classes in Different Hierarchies

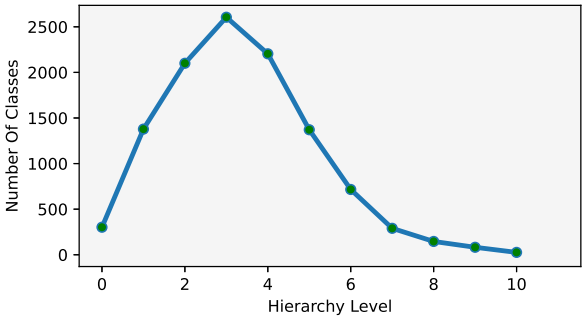

Figure 4: Number of classes in different hierarchies.

B Training Details

B.1 Single-label ImageNet-21K-P Training Details

To better handle the ground-truth inconsistencies of ImageNet-21K-P, we increase the label-smooth factor from the common value of 0.1 to 0.2. As explained in section 2.1, we also use squish-resizing instead of crop-resizing. We trained the models with input resolution 224, using an Adam optimizer with learning rate of 3e-4 and one-cycle policy [48]. When initializing our models from standard ImageNet-1K pretraining (pretraining weights taken from [57]), we found that 80 epochs are enough for achieving strong pretrain results on ImageNet-21K-P. For regularization, we used RandAugment [10], Cutout [12], Label-smoothing [50] and True-weight-decay [37]. We observed that the common ImageNet statistics normalization [33, 51] does not improve the training accuracy, and instead normalized all the RGB channels to be between 0 and 1. Unless stated otherwise, all runs and tests were done on TResNet-M architecture. On an 8xV100 NVIDIA GPU machine, training with mixed-precision takes 40 minutes per epoch on ResNet50 and TResNet-M architectures ($\sim 5000\frac{\text{img}}{\text{sec}}$).

B.2 Multi-label ImageNet-21K-P Training Details

For multi-label training, we convert each image single label input to semantic multi labels, as described in section 2.2. Multi-label training details are similar to single-label training (number of epochs, optimizer, augmentations, learning rate, models initialization and so on), and training times are also similar. The main difference between single-label and multi-label training relies in the loss function: for multi-label training we tested 3 loss functions, following [3]: cross-entropy ($\gamma_- = \gamma_+ = 0$), focal loss ($\gamma_- = \gamma_+ = 2$) and ASL ($\gamma_- = 4, \gamma_+ = 0$). For ASL, we tried different values of γ_- to obtain the best mAP scores.

C Upstream Results

As we have a standardized dataset with a fixed train-validation split, the training metrics for each pretraining method can be used for future benchmark and comparisons.

C.1 Singe-label Upstream Results

For single-label training, regular top-1 accuracy metric becomes somewhat irrelevant - if pictures with similar content have different ground-truth labels, the network has no clear "correct" answer. Top-5 accuracy metric is more representative, but still limited. Upstream results of single-label training are given in Table 5. We can see that the top-1 accuracies obtained on ImageNet-21K-P,

$37\% - 46\%$, are significantly lower than the ones obtained on ImageNet-1K, $75\% - 85\%$. This accuracy drop is mainly due to the semantic structure and inconsistent tagging methodology of ImageNet-21K-P. However, as we take bigger and better architectures, we see from Table 5 that the accuracies continue to improve, so we are not completely hindered by the inconsistent tagging.

Model Name	Top-1 Acc. [%]	Top-5 Acc. [%]
MobileNetV3	37.8	66.3
ResNet50	42.2	72.0
TResNet-M	45.3	75.2
TResNet-L	45.5	75.6

Table 5: Accuracy of different models in single-label training.

C.2 Multi-label Upstream Results

For multi-label training, we will use the common micro and macro mAP accuracy [3] as training metrics. However, due to the missing labels in the validation (and train) set, this metric also is not fully accurate. In Table 7 we compare the results for three possible loss functions for multi-label classification - cross-entropy, focal loss and ASL. We see that ASL loss [3], that was designed to

Loss Type	Micro-mAP [%]	Macro-mAP [%]
Cross-Entropy	47.3	73.9
Focal loss	47.4	74.1
ASL	48.5	74.7

Table 6: Comparing different loss functions for multi-label classification on ImageNet-21K-P.

cope with large positive-negative imbalancing, outperform cross-entropy and focal loss. This is in agreement with our analysis in section 3.2, where we identify extreme imbalancing as one of the optimization challenges that stems from multi-label training.

C.3 Semantic Softmax Upstream Results

With semantic softmax training, we can calculate for each hierarchy its top-1 accuracy metric. We can also calculate the total accuracy by weighting the different accuracies by the number of classes in each hierarchy (see Figure 4). Notice that we are not using classes above the maximal hierarchy for our metrics calculation. Hence, and unlike single-label and multi-label training, with semantic softmax our training metrics are fully accurate.

In Figure 5 we present the top-1 accuracies achieved by different models on different hierarchy levels, when trained with semantic softmax (with KD).

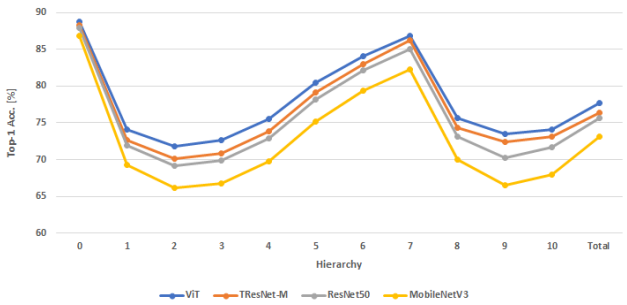

Figure 5: Top-1 accuracies on different hierarchies.

D Downstream Datasets Training Details

For single-label classification, our downstream datasets were ImageNet-1K [30], iNaturalist 2019 [55], CIFAR-100 [29] and Food-251 [25]. For multi-label classification, our downstream datasets were MS-COCO [34] and Pascal-VOC [16]. For video action recognition, our downstream dataset was Kinetics-200 [26].

General details:

- To minimize statistical uncertainty, for datasets with less than $150,000$ images (CIFAR-100, Food-251, MS-COCO, Pascal-VOC), we report result of averaging 3 runs with different seeds.
- All results are reported for input resolution 224.
- For all downstream datasets we used cutout of 0.5, rand-Augment and true-weight-decay of 1e-4.
- All single-label datasets are trained with label-smooth of 0.1
- Unless stated otherwise, dataset was trained for 40 epochs with Adam optimizer, learning rate of 3e-4, one-cycle policy and and squish-resizing.

Specific dataset details:

- ImageNet-1K - Since the dataset is bigger than the others, we finetuned our networks for 100 epochs using SGD optimizer, and learning rate of 4e-4. We used crop-resizing with the common minimal crop factor of 0.08.
- MS-COCO - We used ASL loss with $\gamma_- = 4$.
- Pascal-VOC - We used ASL loss with $\gamma_- = 4$, and learning rate of 5e-5.
- Kinetics-200 - we trained for 30 epochs with learning rate of 8e-5. We used the training method described in [47], with simple averaging of the embedding from each sample along the video.

E Downstream Results for Different Multi-label Losses

In Table 7 we compare downstream results when using multi-label pretraining with vanilla cross-entropy (CE) loss and ASL loss. We see that on all downstream datasets, pretraining with ASL leads to significantly better results

Dataset	Multi Label Pretrain (CE)	Multi Label Pretrain (ASL)
ImageNet1K[(1)]	79.6	**81.0**
iNaturalist[(1)]	69.4	**71.0**
Food 251[(1)]	74.3	**75.2**
CIFAR 100[(1)]	89.9	**90.6**
MS-COCO[(2)]	79.1	**80.6**
Pascal-VOC[(2)]	87.6	**87.9**
Kinetics 200[(3)]	81.1	**81.9**

Table 7: Comparing downstream results for different losses of multi-label pretraining. Dataset types and metrics: (1) - single-label, top-1 Acc.[%] ; (2) - multi-label, mAP [%]; (3) - action recognition, top-1 Acc. [%].

F Calculating Teacher Confidence

Using the teacher prediction for hierarchy i and the semantic ground-truth, we want to evaluate the teacher confidence level, P_i, so we can weight properly the contribution of different hierarchies in the KD loss. Our proposed logic for calculating the teacher's (semantic) confidence is simple:
- If the ground-truth highest hierarchy is higher than i, set P_i to 1.

- Else, calculate the sum probabilities of the top 5% classes in the teacher prediction (we deliberately don't take only the probability of the highest class, to account for class similarities).

In Figure 6 we present the teacher confidence level for different hierarchies, averaged over an epoch. We can see that lower hierarchies have, in average, higher confidence levels. This stems from the

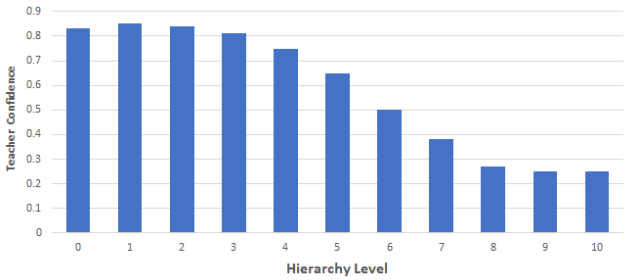

Figure 6: Teacher average confidence levels for different hierarchies.

fact that not all hierarchies are relevant for each image. For the picture in Figure 3, for example, only hierarchies 0-5 are relevant, so we expect the teacher will have low confidence for hierarchies higher than 5.

G Semantic KD Vs Regular KD

Dataset	Single Label + KD Pretrain	Sematic Softmax + Semanic KD Pretrain
ImageNet1K[(1)]	81.5	**82.2**
iNaturalist[(1)]	72.4	**72.7**
Food 251[(1)]	76.0	**76.1**
CIFAR 100[(1)]	91.0	**91.7**
MS-COCO[(2)]	81.6	**82.2**
Pascal-VOC[(2)]	89.0	**89.8**
Kinetics 200[(3)]	83.6	**84.4**

Table 9: Comparing KD with different schemes. Dataset types and metrics: (1) - single-label, top-1 Acc.[%] ; (2) - multi-label, mAP [%]; (3) - action recognition, top-1 Acc. [%].

H ImageNet-1K Transfer Learning Results

Model Name	ImageNet-1K + KD Top-1 Acc. [%]
MobileNetV3	78.0
OFA-595	81.0
ResNet50	82.0
Mixer-B-16	82.2
TResNet-M	83.1
TResNet-L	83.9
ViT-B-16	84.4

Table 10: Transfer learning results On ImageNet-1K, when using ImageNet-21K-P pretraining.

I ImageNet-21K-P - Winter21 Split

For a fair comparison to previous works, the results in the article are based on the original ImageNet-21K images, i.e. we are using Fall11 release of ImageNet-21K (*fall11-whole.tar* file), which contains all the original images and classes of ImageNet-21K. After we processed this release to create ImageNet-21K-P, we are left with a dataset that contains 11221 classes, where the train set has 11797632 samples and the test set has 561052 samples. We shall name this variant *Fall11 ImageNet-21K-P*.

Recently, the official ImageNet site[1] used our pre-processing methodology to offer direct downloading of ImageNet-21K-P, based on a new release of ImageNet-21K - Winter21 (*winter21-whole.tar* file). Compared to the original dataset, the Winter21 release removed some classes and samples. The Winter21 variant of ImageNet-21K-P is a dataset that contains 10450 classes, where the train set has 11060223 samples and the test set has 522500 samples. We shall name this variant *Winter21 ImageNet-21K-P*.

For enabling future comparison and benchmarking, we report the upstream accuracies also on this new variant of ImageNet-21K-P:

	Single-Label Training Acc. [%]	Multi-Label Training Macro-mAP [%]	Semantic Softmax Training Acc. [%]
Fall11 ImageNet-21K-P	45.3	74.7	75.6
Winter21 ImageNet-21K-P	47.3	78.7	77.7

Table 11: Upstream results, with different pretraining methods, for different variants of ImageNet-21K-P. Tested model - TResNet-M.

Note that the Winter21 variant of ImageNet-21K-P contains 10% fewer classes and 6% fewer images. In Table 12 we compare downstream results when using Winter21 and Fall11 variants of ImageNet-21K-P

Dataset	Fall11 ImageNet-21K-P	Winter21 ImageNet-21K-P
ImageNet1K(1)	**81.4**	81.2
iNaturalist(1)	**72.0**	71.8
Food 251(1)	**75.8**	75.5
CIFAR 100(1)	90.4	**90.5**
MS-COCO(2)	**81.3**	81.1
Pascal-VOC(2)	89.7	**90.1**
Kinetics 200(3)	**83.0**	82.8

Table 12: Comparing downstream results when using different variant of ImageNet-21K-P. All results are for MTResNet model, with semantic softmax pretraining. Dataset types and metrics: (1) - single-label, top-1 Acc.[%] ; (2) - multi-label, mAP [%]; (3) - action recognition, top-1 Acc. [%].

We can see that compared to Fall11 variant, using Winter21 variant leads to a minor reduction in performances on downstream tasks.

J Additional Ablation Tests

In this section we will bring additional ablation tests and comparisons.

[1] www.image-net.org

J.1 Comparison to Pretraining on Open Images Dataset

Open Images (v6) [31] is a large scale multi-label dataset, which consists of 9 million training images and 9600 labels. In Table 13 we compare downstream results when using two different datasets for pretraining: ImageNet-21K (semantic softmax training) and Open Images (multi-label training).

Dataset	ImageNet-21K Pretrain	Open Images Pretrain
ImageNet1K[(1)]	**81.4**	81.0
iNaturalist[(1)]	**72.0**	70.7
Food 251[(1)]	**75.8**	74.8
CIFAR 100[(1)]	**90.4**	89.4
MS-COCO[(2)]	**81.3**	80.5
Pascal-VOC[(2)]	**89.7**	89.6
Kinetics 200[(3)]	**83.0**	81.6

Table 13: Comparing ImageNet-21K pretraining to Open Images pretraining. Downstream dataset types and metrics: (1) - single-label, top-1 Acc.[%] ; (2) - multi-label, mAP [%]; (3) - action recognition, top-1 Acc. [%].

As we can see, ImageNet-21K pretraining consistently provides better downstream results than Open Images. A possible reason is that Open Images, as a multi-label dataset with large number of classes, suffers from the same multi-label optimization pitfalls we described in section 3.2.

J.2 Comparison on Additional Non-Classification Computer-Vision Tasks

In Table 14 and Table 15 we compare 1K and 21K pretraining on two additional computer-vision tasks: object detection (MS-COCO dataset) and image retrieval (INRIA holidays dataset).

	1K Pretraining	21K Pretraining
mAP [%]	42.9	44.3

Table 14: Comparing downstream results on MS-COCO object detection dataset.

	1K Pretraining	21K Pretraining
mAP [%]	81.1	82.1

Table 15: Comparing downstream results on on INRIA Holidays image retrieval dataset.

We can see that also on non-classification tasks such as object detection and image retrieval, pretraining on ImageNet-21K translates to better downstream results than ImageNet-1K pretraining.

J.3 Impact of Different Number of Training Samples

In Figure 7 we test the impact of the number of training samples on on the upstream accuracies. As we can see, there is no saturation - more training images lead to better semantic accuracies.

K Pseudo-code

In the following sections we will bring pseudo-code (PyTorch-style) to some components in our semantic softmax training scheme: logits sampling, KD calculation and estimating teacher confidence.

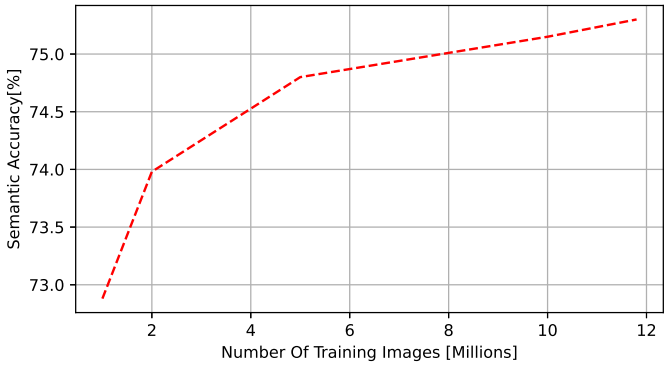

Figure 7: Upstream results for different number of training images.

K.1 Logits Sampling

```python
def split_logits_to_semantic_logits(logits, hierarchy_indices_list):
    semantic_logit_list = []
    for i, ind in enumerate(hierarchy_indices_list):
        logits_i = logits[:, ind]
        semantic_logit_list.append(logits_i)
    return semantic_logit_list
```

K.2 KD Logic

```python
def calculate_KD_loss(input_student, input_teacher, hierarchy_indices_list):
    semantic_input_student = split_logits_to_semantic_logits(
    input_student, hierarchy_indices_list)
    semantic_input_teacher = split_logits_to_semantic_logits(
    input_teacher, hierarchy_indices_list)
    number_of_hierarchies = len(semantic_input_student)

    losses_list = []
    # scanning hirarchy_level_list
    for i in range(number_of_hierarchies):
        # converting to semantic logits
        inputs_student_i = semantic_input_student[i]
        inputs_teacher_i = semantic_input_teacher[i]

        # generating probs
        preds_student_i = stable_softmax(inputs_student_i)
        preds_teacher_i = stable_softmax(inputs_teacher_i)

        # weight MSE-KD distances according to teacher confidence
        loss_non_reduced = torch.nn.MSELoss(reduction='none')(preds_student_i,
        preds_teacher_i)
        weights_batch = estimate_teacher_confidence(preds_teacher_i)
        loss_weighted = loss_non_reduced * weights_batch.unsqueeze(1)
        losses_list.append(torch.sum(loss_weighted))

    return sum(losses_list)
```

K.3 Teacher Confidence

```
def estimate_teacher_confidence(preds_teacher)
    with torch.no_grad():
        num_elements = preds_teacher.shape[1]
        num_elements_topk = int(np.ceil(num_elements / 20))  # top 5%
        weights_batch = torch.sum(torch.topk(preds_teacher,
        num_elements_topk).values, dim=1)
    return weights_batch
```

L Limitations

In this section we will discuss some of the limitations of our proposed pipeline for pretraining on ImageNet-21K:

1) While our work did put a large emphasis on the efficiency of the proposed pretraining pipeline, for reasonable training times we still need an 8-GPUs machine (1 GPU training will be quite long, 2-3 weeks).

2) For creating an efficient pretraining scheme, and also to stay within our inner computing budget, we did not incorporate training tricks that significantly increase training times, although some of these tricks might give additional benefits and improve pretraining quality.

An example - techniques for dealing with extreme multi-tasking, such as GradNorm [7] and PCGrad [60], that would probably improve the pretrain quality of multi-label training, but would significantly increase training times.

Another example of methods from the literature we have not tested - general "semantic" techniques that can be used for training neural networks ([4, 54] for example). We found that most of these techniques are not feasible for large-scale efficient training. In addition, we believe that since our novel method, semantic softmax, is designed and tailored to the specific needs and characterizations of ImageNet-21K, it will significantly outperform general semantic methods.

3) When using private datasets which are larger than ImageNet-21K, such as JFT-300M [49], the pretrain quality that can be achieved is probably still higher than the one we offer.