SEPT: Towards Scalable and Efficient Visual Pre-training

Yiqi Lin[1*], Huabin Zheng[2], Huaping Zhong[2], Jinjing Zhu[1], Weijia Li[3†], Conghui He[2], Lin Wang[1, 4†]

[1]AI Thrust, Information Hub, HKUST (Guangzhou), Guangzhou, China
[2]SenseTime Research
[3]Sun Yat-Sen University
[4]Department of Computer Science and Engineering, HKUST, Hong Kong, China
{ylin933, jzhu706}@connect.hkust-gz.edu.cn, {zhenghuabin,zhonghuaping,heconghui}@sensetime.com
liweij29@mail.sysu.edu.cn, linwang@ust.hk

Abstract

Recently, the self-supervised pre-training paradigm has shown great potential in leveraging large-scale unlabeled data to improve downstream task performance. However, increasing the scale of unlabeled pre-training data in real-world scenarios requires prohibitive computational costs and faces the challenge of uncurated samples. To address these issues, we build a task-specific self-supervised pre-training framework from a data selection perspective based on a simple hypothesis that pre-training on the unlabeled samples with similar distribution to the target task can bring substantial performance gains. Buttressed by the hypothesis, we propose the first yet novel framework for **S**calable and **E**fficient visual **P**re-**T**raining (**SEPT**) by introducing a retrieval pipeline for data selection. SEPT first leverage a self-supervised pre-trained model to extract the features of the entire unlabeled dataset for retrieval pipeline initialization. Then, for a specific target task, SEPT retrievals the most similar samples from the unlabeled dataset based on feature similarity for each target instance for pre-training. Finally, SEPT pre-trains the target model with the selected unlabeled samples in a self-supervised manner for target data finetuning. By decoupling the scale of pre-training and available upstream data for a target task, SEPT achieves high scalability of the upstream dataset and high efficiency of pre-training, resulting in high model architecture flexibility. Results on various downstream tasks demonstrate that SEPT can achieve competitive or even better performance compared with ImageNet pre-training while reducing the size of training samples by *one magnitude without resorting to any extra annotations*.

Introduction

To reduce the demand for collecting large-scale labeled data for each target application, supervised pre-training on large-scale datasets, *e.g.*, ImageNet (Deng et al. 2009) and then finetuning on the target tasks have become a successful and standard paradigm in many applications (He et al. 2017; Long, Shelhamer, and Darrell 2015; Sun et al. 2017). To avoid such expensive annotation costs, self-supervised learning (SSL) methods (Doersch, Gupta, and Efros 2015; Zhang, Isola, and Efros 2016; Noroozi and Favaro 2016; Komodakis and Gidaris 2018; Kingma and Welling 2013; Bao et al. 2021) have shown that models can be trained with freely available supervision from raw data itself. Recent advances of SSL methods (Caron et al. 2021; He et al. 2022; Zhou et al. 2021) can achieve comparable or even superior performance to their supervised pre-training version in solving the downstream task.

Several works (Goyal et al. 2021; Kolesnikov et al. 2020; Pham et al. 2021; Dumoulin et al. 2021) suggest that scaling up the pre-training dataset can bring continuous ad-

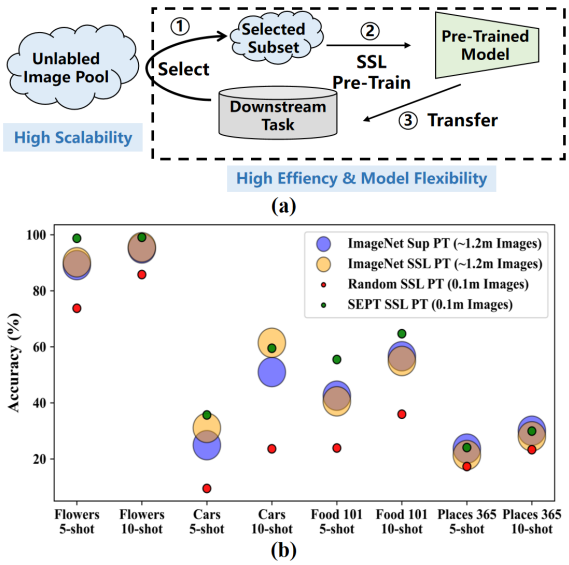


Figure 1: (a) The general pipelines of our proposed SEPT. (b) The down-stream task classification accuracy of SEPT with 0.1m images and ImageNet 1k baselines. The ratio denotes the number of pre-training samples under the same training epoch setting.

*This work was done during the internship at SenseTime when his affiliation is Sun Yat-Sen University.
†Corresponding authors.

vancement of the state-of-the-art (SoTA) performance on the downstream tasks. Despite the promising prospects in collecting unlabeled data, the scalability of unlabeled datasets for SSL methods still suffers from three aspects of challenges in real-world applications.

Firstly, linearly increasing pre-training computation overhead limits the flexibility of model architecture as different application scenarios may request the model with different architectures and scales; thus, it is prohibitively expensive to perform large-scale pre-training for each customized model. Secondly, training on the entire massive unlabeled data raises the risk of introducing potential bias (Caron et al. 2019; Tian, Henaff, and van den Oord 2021) because the unlabeled samples collected from the real-world scenario might be low-quality or without specific semantic information. Lastly, some studies (Ngiam et al. 2018; Ge and Yu 2017) point out the redundancy in the large-scale pre-training for downstream tasks by showing that models pre-trained on a subset can achieve comparable even better to the ones pre-trained on the entire dataset. Therefore, some unlabeled samples can be inessential to a downstream task in self-supervised learning.

An intuitive way to address these issues is to decouple the scales of the pre-training dataset by performing the data selection. Accordingly, in the supervised pre-training pipelines, NDS (Yan, Acuna, and Fidler 2020) and SNDS (Cao et al. 2021) have been proposed to select the relevant subset according to the downstream task data for supervised pre-training using a data recommendation system. ***However, existing research seldom explored such a problem in self-supervised pre-training.*** In this work, we explore an alternative to the existing pre-training paradigm and propose a novel self-supervised Scalable and Efficient visual Pre-Training (**SEPT**) framework (See Fig. 1(a)). It aims at optimizing the pre-training efficiency while maintaining the scalability of the pre-training dataset scale. SEPT is based on a hypothesis that ***training self-supervised learning methods on a subset having similar distribution to the downstream target task should improve performance on the target task***, inspired by the domain adaptation theory (Ben-David et al. 2010; Ganin and Lempitsky 2015).

Supported by the hypothesis, we use the target dataset to perform the instance search via feature similarity for collecting a relatively small task-specific self-supervised pre-training dataset. Specifically, SEPT consists of three steps: retrieval pipeline initialization, task-specific instance search, and task-specific self-supervised learning. SEPT first builds a retrieval pipeline using a self-supervised pre-trained model on a subset of the dataset. Notably, the retrieval pipeline can be easily reused for different target tasks. Intuitively, the retrieval model serves as a distribution discrepancy metric to minimize the distribution gap between pre-training and the target dataset. In task-specific instance search, SEPT uses the retrieval model to search the most similar samples for each task sample from the entire upstream dataset to construct a small subset according to the computational constraints. Finally, SEPT self-supervised pre-train a target model with the retrieved unlabeled subset and then finetune the target model on the task data.

We conduct experiments on **seven** classification and **three** detection tasks with limited labeled samples. Compared with ImageNet 1k pre-training baselines, our SEPT achieves ***competitive or better performance*** on classification tasks, Flowers, Food101, Stanford Cars, and Places365 with only 100k unlabeled images for pre-training and few shot images for finetuning (See Fig. 1(b)).

Related Work

Self-Supervised Representation Learning Self-supervised representation learning has shown promising results in model pre-training with freely available supervision from raw data recently (Jing and Tian 2020; Liu et al. 2021a). Early works mainly focus on designing handcrafted pretext tasks (Doersch, Gupta, and Efros 2015; Zhang, Isola, and Efros 2016; Noroozi and Favaro 2016; Komodakis and Gidaris 2018) using prior knowledge. More recent works can be categorized in discriminative (Dosovitskiy et al. 2014; Bachman, Hjelm, and Buchwalter 2019; He et al. 2020; Chen et al. 2020a) or generative (Kingma and Welling 2013; Xie et al. 2021b; Bao et al. 2021; He et al. 2022) fashion. In the discriminative fashion, contrastive methods (Dosovitskiy et al. 2014; Bachman, Hjelm, and Buchwalter 2019; He et al. 2020; Chen et al. 2020a) force the representation of different views of the same image closer and push representations of views from different images away, which achieves comparable performance to its counterpart of supervised pre-training. We notice that the models pre-trained with contrastive learning have strong instance discriminative power. Therefore we exploit such ability for building the retrieval pipeline to help downstream tasks find visually similar samples from the general data pool.

Data Selection for Pre-training In the context of image pre-training, there are also several works dedicated to improving the performance (Ngiam et al. 2018; Ge and Yu 2017) or efficiency (Yan, Acuna, and Fidler 2020; Cao et al. 2021) from the perspective of selecting the appropriate training data instead of the entire dataset. (Ge and Yu 2017) greedily selects the most similar categories from the source dataset to be used for pre-training using a proposed similarity metric between the source and target categories. (Ngiam et al. 2018) proposes to use a model pre-trained on the source dataset to obtain pseudo labels for target images and uses the pseudo to re-weight the source examples. NDS (Yan, Acuna, and Fidler 2020) represents source datasets with the mixture-of-experts model and employs it to perform data search by finding the dataset with similar behavior to the mixture-of-experts. SNDS (Cao et al. 2021) proposes to use intermediary datasets to train mixture-of-experts for indexing the growing scale of the source dataset.

Despite their promising results, the scalability of source datasets in these supervised pre-training methods is still limited by the extensive human labeling (Cao et al. 2021) and high computation cost on upstream dataset (Ngiam et al. 2018; Ge and Yu 2017; Yan, Acuna, and Fidler 2020). Differently, our work explores a new self-supervised pre-training paradigm with a highly scalable unlabeled dataset and efficient pre-training computation overhead on a task-specific subset.

Methodology

Preliminaries

Problem Definition. Consider a general unlabeled dataset $\mathcal{U} = \{x_i\}_{i=0}^{N}$ where x_i is an image and a labeled task dataset $\mathcal{T} = (x_i, y_i)_i$ where x_i is an image and y_i is the ground truth, we assume $|\mathcal{U}| \gg |\mathcal{T}|$. Our goal is to train a model θ that fits well on the task $\mathcal{T}$ with the help of an unlabeled dataset $\mathcal{U}$ by finding the most informative subset. Formally, we seek an optimal unlabeled subset $\mathcal{U}^*$ that can help enhance the target task performance with the labeled task dataset $\mathcal{T}$:

$$\mathcal{U}^* = \underset{\hat{\mathcal{U}} \subseteq \mathcal{U}}{\arg\min} \mathbb{E}_{x \sim \mathcal{T}} \left[\mathcal{L} \left(\boldsymbol{y} | \theta_{\hat{\mathcal{U}} \cup \mathcal{T}}(\boldsymbol{x}) \right) \right] \tag{1}$$

where $\theta_{\hat{\mathcal{U}} \cup \mathcal{T}}$ denotes that θ is trained with the union of selected unlabeled subset $\hat{\mathcal{U}}$ and labeled task data. Intuitively, the whole learning system does not involve any annotation from a general unlabeled dataset $\mathcal{U}$, which means $\mathcal{U}$ can be simply scaled up by raw data itself without any human prior.

Theoretical Insight. SEPT is based on a hypothesis that ***training self-supervised learning methods on a subset having similar distribution to the downstream target task should improve performance on the target task***. Therefore, our research question is also related to the domain adaptation problem (Pan and Yang 2009), where the general unlabeled dataset can be viewed as a source domain covering various unknown semantics categories while the target domain respects the specific task. Unlike the existing domain adaptation methods with well-defined source and target datasets, the general dataset in our problem setting is more uncurated and realistic. From the perspective of domain adaptation, SEPT focuses on sampling the informative source samples to proximate the target distribution, *i.e.*, finding the desired source distribution for better domain adaptation.

Given by (Ben-David et al. 2010), let $\mathcal{H}$ be a hypothesis space, the generalization errors of a function $f \in \mathcal{H}$ on the target domain $\mathcal{T}$ and the source domain $\mathcal{S}$ as ϵ_t and ϵ_s, respectively. Then, for any function $f \in \mathcal{H}$, we have the following generalization bound,

$$\epsilon_t(f) \leq \epsilon_s(f) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \epsilon^* \tag{2}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ is $\mathcal{H}\Delta\mathcal{H}$-Divergence which measures the discrepancy between the two domains:

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) = \sup_{f, f' \in \mathcal{H}} \big| & \mathbb{E}_{x \sim S}[f(x) \neq f'(x)] \\ & - \mathbb{E}_{x \sim T}[f(x) \neq f'(x)] \big| \end{aligned} \tag{3}$$

The key problem of domain adaptation is to minimize the distribution discrepancy $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$. However, in our setting, the source domain $\mathcal{S}$ is a subset of the unlabeled dataset $\mathcal{S} \in P(\mathcal{U})$ required to select. Therefore, we first need to introduce a pre-defined and generalized metric for measuring the distribution discrepancy. In (Ganin et al. 2016), a domain classifier is introduced to measure the distance between two distributions by recognizing the domain category of samples, and the $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ is reformulated as follow,

$$d_{\mathcal{H}_p\Delta\mathcal{H}_p}(\mathcal{S}, \mathcal{T}) \leq 2 \sup_{h \in \mathcal{H}_d} \big| \alpha(h) - 1 \big| \tag{4}$$

Algorithm 1: Task-specific Instance Search

Input: an unlabeled dataset $\mathcal{U}$, a target dataset $\mathcal{T}$, a budget (number of images) of pre-training dataset $\mathcal{K}$ and a feature extractor θ_R well trained on subset $\hat{\mathcal{U}} \subseteq \mathcal{U}$.
Output: a task-specific pre-training subset D_{search}.

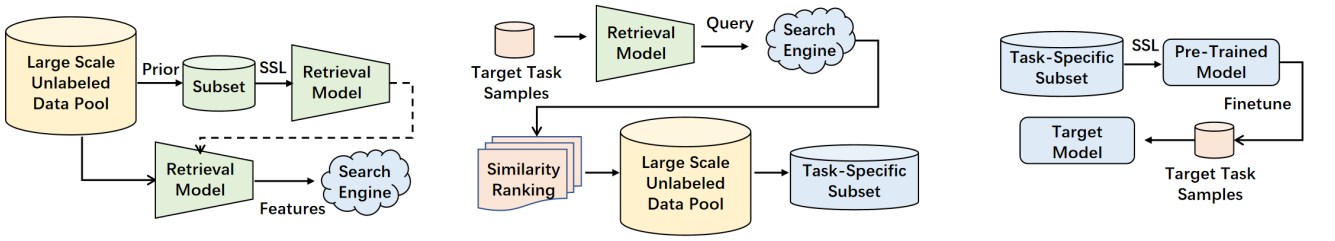
$\mathcal{F}_u \leftarrow \theta_R(x^*), \forall x^* \in \mathcal{U}$
for x_i *in* $\mathcal{T}$ **do**
  $F_i \leftarrow \theta_R(x_i)$
  $RankedList(x_i) \leftarrow Sorted(sim(F_i, F_u))$
end for
$D_{search} \leftarrow \emptyset$
for j *in* $\mathcal{K}$ **do**
  for x_i *in* $\mathcal{T}$ **do**
    if $D_{search} \cap RankedList(x_i)[j]$ *is* $\emptyset$ **then**
      $D_{search} \leftarrow D_{search} \cup RankedList(x_i)[j]$
      if $|D_{search}| \geq \mathcal{K}$ **then**
        Exit
      end if
    end if
  end for
end for
return selected subset D_{search}

where $\mathcal{H}_d$ is the hypothesis space of the domain classifier and $\alpha(h)$ is the optimal classifier. Inspired by the design of the domain classifier, we argue that it is possible to find a general domain discriminator for an arbitrary target task. In practice, SEPT uses a retrieval pipeline based on feature similarity to proximate the domain discrimination process. Intuitively, for each target instance, SEPT selects the most similar source sample that mostly confuses the general domain discriminator and thus minimizes the $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$ with a measurement selected by prior knowledge. After selecting the source samples, SEPT adapts the most straightforward strategy, pre-training and fine-tuning, to perform domain adaptation for simplicity.

Proposed Framework

Overview. SEPT is a pre-training framework aiming to find a task-relevant subset from a massive unlabeled image pool given a specific task with limited samples. The framework consists of three steps: retrieval pipeline initialization, task-specific instance search, and task-specific self-supervised learning. In the first step, we extract and store features of the entire unlabeled dataset $\mathcal{U}$ with a self-supervised model. In the second step, we perform the instance search based on feature similarity for each sample in $\mathcal{T}$ to aggregate the most similar subset $\hat{\mathcal{U}}$ from $\mathcal{U}$. In the last step, a downstream model is self-supervised, pre-trained on the selected subset $\hat{\mathcal{U}}$, and then fine-tuned with target data $\mathcal{T}$ only with the task objective. Fig. 2 shows the overall pipeline.

Retrieval Pipeline Initialization. To be consistent with our self-supervised pre-training framework, we use a self-supervised pre-trained model as the retrieval model for feature extraction. Theoretically, pre-training a giant model across the entire $\mathcal{U}$ should be the optimal solution. However,



(a).Retrieval Pipeline Initialization **(b).Task-specific Instance Search** **(c).Task-specific Self-supervised Learning**

Figure 2: Framework Overview. SEPT consists of three steps: a) retrieval pipeline initialization, b) task-specific instance search, and c) task-specific self-supervised learning. SEPT first trains a self-supervised model on a subset from a large-scale unlabeled data pool. Then, given a few task data, SEPT retrievals are the most similar subset. Finally, SEPT self-supervised pre-trains a target model on the selected subset and fine-tunes the pre-trained model with task samples and their annotations.

we argue that it is critical to introduce certain prior knowledge for building retrieval models for two reasons. Firstly, the retrieval model only needs to be trained once during initialization, which makes the computational cost of initialization not contribute to the marginal cost of indexing or queries and scaling up the data pool. Secondly, training on the entire massive unlabeled data raises the risk of introducing potential bias to retrieval models because it is impossible to know the visual concept distribution of the data pool. Recent self-supervised methods (Chen et al. 2020b; Caron et al. 2021; Zhou et al. 2021) have shown promising instance discrimination on the unlabeled images and provided lots of successful practice on the ImageNet with various architectures. Therefore, we use ImageNet self-supervised pre-trained model f as the feature extractor for retrieval to avoid the computation cost and the risk of introducing noise. The retrieval model can be viewed as a replaceable module in our framework and can be integrated with the latest techniques in this area as they emerge. In our experiments, we train the feature extractor using ibot (Zhou et al. 2021) with ViT-S (Dosovitskiy et al. 2020).

Task-specific Instance Search. As depicted in **Algorithm 1**, for each image in the task data $x_i \in \mathcal{T}$, we retrieve a set of image $\hat{\mathcal{U}}_i$ from the given general data pool $\mathcal{U}$. The set $\hat{\mathcal{U}}_i$ represents the top-K similar images to x_i in $\mathcal{U}$. Retrieved data for all images x_i are combined to obtain a subset $\hat{\mathcal{U}} = Union(\hat{\mathcal{U}}_0, ..., \hat{\mathcal{U}}_i)$. Considering the computation cost produced during the retrieval on a large-scale data pool,*e.g.*, 150 million in our setting, we keep our retrieval approach as simple as possible by calculating the similarity between two samples. Specifically, we use a self-supervised retrieval model f to measure the similarity $sim(f(x_i), f(x_u)), x_u \in \mathcal{U}$ between the task sample and unlabeled samples. SEPT can handle the different budget constraints of the model pre-training by setting the scale of retrieved samples for real-world applications. Our data retrieval method only depends on the raw images x_i itself and does not leverage any task-specific labels.

Task-specific Self-Supervised Learning. After obtaining the retrieval dataset, we conduct self-supervised pre-training on them by optimizing the following objective:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{x \sim \hat{\mathcal{U}}} \left[\mathcal{L}_{self} \left(\boldsymbol{x} \right) \right] \tag{5}$$

Note that task-specific self-supervised learning is decoupled from the self-supervised retrieval models. Therefore, the former can use the latest techniques or various backbones in this area as they emerge.

When transferring to the downstream target task, only task data is used for fine-tuning the pre-trained backbone. *The exploration of improving performance by joint training with retrieval and task data is left for future work*. Given the task-specific self-supervised models, we train a task model using the pre-training model as initialization with the following loss, formulated as:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{x \sim \mathcal{T}} \left[\mathcal{L}_{sup} \left(y | \boldsymbol{x} \right) \right] \tag{6}$$

Experiments Setup

Datasets and Setup

Datasets for Retrieval Pipeline. We combine three large-scale datasets, ImageNet-22k(IN22k) (Deng et al. 2009), INTERN (Shao et al. 2021) and YFCC-100m (Thomee et al. 2016), to construct an unlabeled data pool with totally 155 million images, called SEPT-155m. IN22k collected 14.2 million images from 21,841 classes and has a widely used subset ILSVRC2012 (IN1k), which consists of 1.2 million images from 1,000 classes. The INTERN classification dataset consists of 40m images with more than 115K concepts. YFCC-100m contains approximately 99.2 million photos. The default retrieval model is self-supervised pre-trained on IN1k, which can be viewed as a subset of the entire unlabeled data pool. Note that we do not access any annotation information during the experiments.

Datasets of Target Task. We provide an extensive evaluation of our approach on seven classification datasets. We randomly sample 5-shot or 10-shot from each category for all datasets to construct our target tasks. The first four tasks are built from Flowers(Nilsback and Zisserman 2008), Stanford Cars(Krause et al. 2013), Food101(Bossard, Guillaumin, and Gool 2014), Places365(Zhou et al. 2017), respectively. To comprehensively simulate complex tasks on a

Task Dataset	Number Classes	Scale	Cross SC	Testing Samples
Flowers102	102	Small	✗	6149
Stanford Cars	196		✗	8041
Food101	101		✗	25250
Place365	365		✗	36500
iNat-P1k	1000	Large	✗	17157
iNat-I1k	1000		✗	17598
iNat-M1k	1000		✓	18646

Table 1: Statistical information of target task datasets.

scale of IN1k (1k categories), the last three tasks, INat-P1k, INat-I1k, and INat-M1k, are all constructed from iNaturalist 2017(Van Horn et al. 2018) by sampling different 1,000 categories to build a subset. All categories in INat-P1k/INat-I1k are sampled from the same super-category of Plantae/Insecta. Differently, categories in INat-M1k are evenly sampled from 13 different supercategories. The statistical information of classification datasets is summarized in Table 1. *More details about these datasets are summarized in the appendix*. In each of them, the SEPT's goal is to improve the performance of its target task by transferring knowledge from a set of relevant unlabeled images.

Baselines and Evaluation. In this regime, we compare our framework with the random initialization, IN1k supervised pre-training, and IN1k self-supervised pre-training models since they are widely adopted standard practices in various vision applications. To demonstrate the effectiveness of data selection, we also provide the baseline pre-trained with randomly sampled images from an unlabeled data pool. For a fair comparison, we evaluate all the pre-training baselines and SEPT under 300 epochs pre-training protocol and report the performance under the same finetuning setting expect different pre-trained model initialization.

Implementation Details

Retrieval Pipeline. The retrieval model uses ViT-S (Dosovitskiy et al. 2020) pre-trained on IN1k using self-supervised methods ibot(Zhou et al. 2021) for 800 epochs. We use the combination of the [CLS] token and patch tokens as features for an image with a dimension of 768. To perform efficient searching, we use milvus (Wang et al. 2021) to build the retrieval pipeline. Specifically, we use IVF_SQ8H as index type, set the *nlist* of index to 16384, and use 256 *nprobe* for searching. The images used for feature extraction are resized to 256×256 and then center cropped with 224×224.

Target Task Pre-training. All experiments are conducted on Swin-T (Liu et al. 2021b). The self-supervised pre-training follows MoBy (Xie et al. 2021a) in 300 epochs setting with batch size 512 on 8 Tesla V100 GPUs. The pre-training adopts AdamW (Loshchilov and Hutter 2018) with a fixed learning rate of 0.001 and a fixed weight decay of 0.05. The key queue size is set to 4096, the temperature is set to 0.2, and the drop path rate is set to 0.2.

Target Task Finetuning. All finetuning experiments use the same 100-epoch finetuning setting on single Tesla V100 GPU. In finetuning, we set the batch size to 64 and employ an AdamW optimizer with a base learning rate of 5e-3, weight decay of 0.05, a stochastic depth ratio of 0.1, and a layer-wise learning rate decay of 0.9. We also adopt a cosine learning rate scheduler with 10 epochs warm-up and follow the same data augmentation used in (Xie et al. 2021b).

Main Results

Pre-training with 100k Images. Table 2 shows the main results that compare SEPT in scales of 100k pre-training images and the according baselines. In conclusion, the pre-training samples selected by SEPT provide helpful knowledge to all target datasets compared to the random sampling baseline. In all small-scale downstream datasets, SEPT can achieve results that are better than or comparable to the supervised or self-supervised IN1k baselines with 1/12 of the scale of pre-training samples. As the large-scale downstream dataset with one thousand visual concepts, the results indicate that general knowledge learned from IN1k pre-training still is a plus for these complex tasks. Moreover, the performance on three large-scale datasets indicates that SEPT has the edge over fine-grained classification tasks. Specifically, the performance gaps between SEPT and IN1k baselines on datasets collected from single super categories iNat-P1k and iNat-I1k are relatively small than iNat-M1k, including multiple super categories images. Note that 100k images might not be sufficient enough to solve such complex tasks with intensive categories. Moreover, the performance gap between supervised and self-supervised IN1k baseline indicates that the ground truth also plays a crucial role in transferring knowledge for different downstream tasks.

Scaling Up the Number of Pre-training Images. Table 3 shows the comparison across different scales of pre-training samples. In the cases shown, the model pre-trained on SEPT outperforms the baseline using random sampling images in all scales of pre-training samples at a large margin. Remarkably, when SEPT is pre-trained with 1000k selected images, a comparable size to IN1k, it outperforms IN1k supervised pre-training baseline around 30% in Food101 under 5-shot settings. We observe that SEPT models pre-trained with 500k samples outperform the IN1k self-supervised baseline on three downstream datasets. In addition, the increased performance, along with the growing scale of pre-training samples, provides strong evidence of the scalability of SEPT in the size of pre-training samples. Moreover, SEPT achieves more performance gains on the fine-grained dataset iNat-I1k than iNat-M1k, indicating that SEPT is a fine-grained friendly solution for downstream tasks. Interestingly, IN1k supervised pre-training still is a strong baseline for complex tasks like iNat-M1k, whose classes are across super categories. Nevertheless, our methods can be scaled up by retrieving more unlabeled images.

Ablation Study

Retrieval Models. Table 4 compares different retrieval methods (i.e., ViT-S, ViT-B, and Vit-L) using IN22k as an unlabeled data pool. We find that given the same data pool, the high capacity of the retrieval model trained with more diverse data can usually provide more performance gain.

Pre-Train Samples		Rand Init (0)	IN1k Sup (1200k)	IN1k SSL (1200k)	Random (100k)	SEPT (100k)
Datasets	Fine-tune Samples	Top 1 accuracy				
Flowers	5-shot	19.90	89.17	90.18	73.77	**98.73**
	10-shot	31.20	95.12	95.54	85.80	**99.09**
Stanford Cars	5-shot	2.81	24.92	31.09	9.48	**35.75**
	10-shot	3.74	51.01	**61.45**	23.62	59.52
Food101	5-shot	4.78	42.58	40.57	23.91	**55.59**
	10-shot	7.48	56.66	54.96	35.97	**64.70**
Place365	5-shot	2.87	23.70	21.40	17.30	**24.08**
	10-shot	5.12	30.08	28.06	23.32	**29.94**
iNat-P1k	5-shot	2.73	30.30	29.47	16.50	**32.33**
	10-shot	5.74	47.08	**46.60**	29.54	45.82
iNat-I1k	5-shot	1.67	33.60	**30.70**	13.71	28.58
	10-shot	4.10	49.92	**46.57**	26.12	38.55
iNat-M1k	5-shot	2.85	36.40	**31.90**	15.72	24.93
	10-shot	5.51	48.57	**43.96**	25.03	33.65

Table 2: Results with 100k images pre-training on SEPT-155m. IN1k Sup, IN1k SSL, and Random denote the IN1k supervised pre-training, IN1k self-supervised pre-training, and SSL with the randomly selected sample.

Number of Samples	Method	Food101	iNat-I1k	iNat-M1k
		5-shot	5-shot	5-shot
1200k	IN1k Sup	42.58	33.60	36.4
	IN1k SSL	40.57	30.70	31.90
100k	Random	23.91	13.71	15.72
	SEPT	55.59	28.58	24.93
500k	Random	32.70	20.12	22.88
	SEPT	69.72	44.43	33.74
1000k	Random	34.53	22.76	24.84
	SEPT	**72.17**	**49.95**	**37.19**

Table 3: Results with different scales of pre-training data. IN1k Sup, IN1k SSL, and Random denote the IN1k supervised pre-training, IN1k self-supervised pre-training, and self-supervised pre-training with random samples.

Retrieval Model	Pre-train Setting	IN1k Linear Val	Food101	
			5-shot	10-shot
ViT-S	IN1k 800 epochs	77.9	49.68	60.99
ViT-B		79.5	52.25	62.60
ViT-L		81.0	51.10	61.43
ViT-B	IN22k 80 epochs	79.0	49.83	62.15
ViT-L		**82.3**	**53.64**	**64.66**

Table 4: Results with different retrieval models under 100K pre-training setting using IN22k as unlabeled data pool.

Specifically, ViT-L trained with IN22k can bring 4% improvement in both 5-shot and 10-shot settings compared to the ViT-S trained with IN1k. Our framework can be upgraded using more powerful self-supervised methods trained with higher capacity and a larger dataset scale, depending on the budget for building the retrieval pipeline.

Sizes of Retrieval Unlabeled Data Pool. To study the performance gains brought by the increasing scale of the unlabeled data pool, we verify our method on four different scales of the data pool by combining different datasets. Results of 100k pre-training settings with different scales of unlabeled data pool are shown in Table 5. Our methods surpass the random sampling at a large margin on all scales of data unlabeled data pool. Increasing the data pool size can consistently improve performance on most downstream tasks, demonstrating the advantage of scaling up the pre-training dataset. It also can be observed that the performance gain becomes trivial when we further increase the image pool size from 55 million to 155 million. To further verify SEPT in a more realistic setting, we provide results only on YFCC-100m, significantly less curated datasets. The SEPT and random sampling baseline from YFCC-100m are relatively weak compared to others, which indicates that the distribution shift of unlabeled data can raise the risk of decreasing the representation quality in pre-trained models. Nevertheless, SEPT still can achieve substantial performance compared to a random sampling baseline in YFCC-100m setting.

Analysis of Hypothesis and Generalization

Redundancy in IN1k. Fig. 3 shows results on the Food101 dataset with different proportions of IN1k samples for pre-training. SEPT using 500k images for pre-training achieves comparable performance against the full IN1k dataset setting, meaning that not all the pre-training samples are informative for a specific task. The performance gaps between SEPT and a random sample further verify the superiority of our method across different settings.

Generalization with More Downstream Samples. Table 6 shows the results using 10-shot SEPT pre-trained models under 50% and 100% Food101 dataset (about 70,000 images) finetuning setting. The results show that SEPT surpasses all the random baselines in different scales of pre-training when using more labeled images. Consistent with the few shot setting, SEPT pre-trained with 500k images can achieve even better results against IN1k pre-trained baseline. We also notice that the performance gaps between different pre-trained models will significantly reduce when the model can access more labeled images. It motivates us to propose a more challenging yet practical few-shot setting for evaluation.

Data Pool	Scale	Method	Food101		iNat-I1k		iNat-M1k	
			5-shot	10-shot	5-shot	10-shot	5-shot	10-shot
IN1k	1.2m	Random	26.57	38.50	16.40	28.80	16.98	26.26
		SEPT	33.71	48.80	17.96	31.44	22.39	32.19
IN22k	14m	Random	25.81	38.66	14.57	26.69	16.99	26.43
		SEPT	49.68	60.99	23.44	36.24	24.44	33.34
IN22k+INTERN	55m	Random	26.03	38.70	15.25	28.15	17.46	26.47
		SEPT	54.82	**64.80**	27.46	38.14	24.68	33.40
IN22k+INTERN+YFCC-100m	155m	Random	23.91	35.97	13.71	26.12	15.72	25.03
		SEPT	**55.59**	64.70	**28.58**	**38.55**	**24.93**	**33.65**
YFCC-100m	100m	Random	21.69	33.62	10.38	22.76	14.26	23.33
		SEPT	51.49	61.90	23.48	35.45	24.37	32.77

Table 5: Results on various scales of unlabeled data pool under 100K pre-training setting.

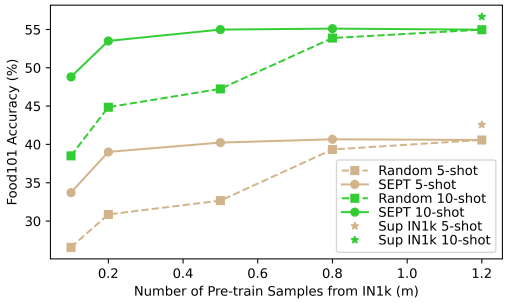


Figure 3: The comparison of using different numbers of IN1k pre-training images in Food101 tasks.

Generalization on Detection. To investigate the generation ability of SEPT on other tasks, we conduct experiments on object detection under similarly limited labeled images setting. The detection datasets include CityScapes (Cordts et al. 2016), VOC (Everingham et al. 2010) and LogoDet-3k (Wang et al. 2022). For all detection tasks, we randomly sample 1,000 images from the original training set for our experiments. In LogoDet-3k task, we create a miniature version of the dataset, called Logo-100, with a 100 class subset for our experiments. Given the annotation of bounding boxes, we evaluate SEPT with two retrieval strategies, including using the whole image as a query and the instance as a query. When using an instance for retrieval, each instance will be cropped and resized to a fixed size for feature extraction. Table 7 shows the results of detection. Our method outperforms the random sampling baseline under 100k and 300k pre-training settings. The results indicate that using an instance as a query can bring more performance gain from SEPT. In CityScapes task, SEPT can achieve comparable performance to IN1k baselines under 300k pre-training. In VOC and Logo dataset, IN1k pre-training still is a strong baseline. On the one hand, all the categories in VOC task are collected in IN1k, resulting in the strong generalization ability of IN1k pre-training models. On the other hand, we also find that some logos are very rare in our unlabeled data pool by visualizing the retrieval result, which suggests that IN1k pre-training still is a decent option when SEPT can not validly collect enough relevant pre-training data.

Pre-train Samples	IN1k SSL	Rand	SEPT	Rand	SEPT
	1200k	100k		500k	
50%	90.08	84.34	88.11	88.57	**91.74**
100%	92.31	88.44	90.64	91.57	**93.15**

Table 6: More downstream samples on Food101.

Method	Pre-train Samples	CityScapes	VOC	Logo-100
Random Init	0	11.5	13.9	33.8
IN1k Sup	1200k	33.2	67.5	69.5
IN1k SSL	1200k	33.4	**67.8**	**69.9**
Random	100k	26.3	47.0	55.3
SEPT-Image	100k	29.9	55.6	64.3
SEPT-Instance	100k	31.4	57.4	64.6
Random	300k	30.0	60.2	61.5
SEPT-Image	300k	33.4	60.6	65.9
SEPT-Instance	300k	**34.9**	63.3	65.2

Table 7: Results for object detection using 1,000 samples for each dataset. Performance is measured in mAP.

Conclusion

In this work, we revisited the scalability and efficiency of transfer learning in the context of scaling up the pre-training datasets. We proposed a pre-training framework, SEPT, which takes full advantage of the enormous scale of datasets without prohibitively expensive annotations by selecting the task-specific subset to perform efficiently pre-training via the similarity search. SEPT only conducts self-supervised pre-training on the retrieval data; thus, it leaves much space for future work to design a more effective algorithm to boost downstream tasks' performance with the retrieval data. It could also be extended to more different tasks, *e.g.*, segmentation, and more modalities, *e.g.*, vision-language. Finally, we hope our work could inspire more research about from the data perspective for the community.

Acknowledgments

This work was supported partially by National Natural Science Foundation of China (No. 42201358).

References

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning representations by maximizing mutual information across views. *NeurIPS*.

Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2021. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine learning*, 79(1): 151–175.

Bossard, L.; Guillaumin, M.; and Gool, L. V. 2014. Food-101–mining discriminative components with random forests. In *ECCV*.

Cao, T.; Doubov, S. A.; Acuna, D.; and Fidler, S. 2021. Scalable Neural Data Server: A Data Recommender for Transfer Learning. *NeurIPS*.

Caron, M.; Bojanowski, P.; Mairal, J.; and Joulin, A. 2019. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2959–2968.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *ICCV*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint arXiv:2003.04297*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *ICCV*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Dosovitskiy, A.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2014. Discriminative unsupervised feature learning with convolutional neural networks. *NeurIPS*.

Dumoulin, V.; Houlsby, N.; Evci, U.; Zhai, X.; Goroshin, R.; Gelly, S.; and Larochelle, H. 2021. Comparing transfer and meta learning approaches on a unified few-shot classification benchmark. *arXiv preprint arXiv:2104.02638*.

Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2): 303–338.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*, 1180–1189. PMLR.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *JMLR*.

Ge, W.; and Yu, Y. 2017. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *CVPR*.

Goyal, P.; Caron, M.; Lefaudeux, B.; Xu, M.; Wang, P.; Pai, V.; Singh, M.; Liptchinsky, V.; Misra, I.; Joulin, A.; et al. 2021. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *CVPR*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *ICCV*.

Jing, L.; and Tian, Y. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *TPAMI*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; and Houlsby, N. 2020. Big transfer (bit): General visual representation learning. In *ECCV*.

Komodakis, N.; and Gidaris, S. 2018. Unsupervised representation learning by predicting image rotations. In *ICLR*.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *ICCVW*.

Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; and Tang, J. 2021a. Self-supervised learning: Generative or contrastive. *TKDE*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *TPAMI*.

Loshchilov, I.; and Hutter, F. 2018. Decoupled Weight Decay Regularization. In *ICLR*.

Ngiam, J.; Peng, D.; Vasudevan, V.; Kornblith, S.; Le, Q. V.; and Pang, R. 2018. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *TKDE*.

Pham, H.; Dai, Z.; Xie, Q.; and Le, Q. V. 2021. Meta pseudo labels. In *ICCV*.

Shao, J.; Chen, S.; Li, Y.; Wang, K.; Yin, Z.; He, Y.; Teng, J.; Sun, Q.; Gao, M.; Liu, J.; et al. 2021. INTERN: A New Learning Paradigm Towards General Vision. *arXiv preprint arXiv:2111.08687*.

Sun, C.; Shrivastava, A.; Singh, S.; and Gupta, A. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 843–852.

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. YFCC100M: The new data in multimedia research. *Communications of the ACM*.

Tian, Y.; Henaff, O. J.; and van den Oord, A. 2021. Divide and contrast: Self-supervised learning from uncurated data. In *ICCV*, 10063–10074.

Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. 2018. The inaturalist species classification and detection dataset. In *CVPR*.

Wang, J.; Min, W.; Hou, S.; Ma, S.; Zheng, Y.; and Jiang, S. 2022. LogoDet-3K: A Large-Scale Image Dataset for Logo Detection. *ACM Trans. Multim. Comput. Commun. Appl.*

Wang, J.; Yi, X.; Guo, R.; Jin, H.; Xu, P.; Li, S.; Wang, X.; Guo, X.; Li, C.; Xu, X.; et al. 2021. Milvus: A purpose-built vector data management system. In *SIGMOD*, 2614–2627.

Xie, Z.; Lin, Y.; Yao, Z.; Zhang, Z.; Dai, Q.; Cao, Y.; and Hu, H. 2021a. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2021b. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*.

Yan, X.; Acuna, D.; and Fidler, S. 2020. Neural data server: A large-scale search engine for transfer learning data. In *CVPR*.

Zhang, R.; Isola, P.; and Efros, A. A. 2016. Colorful image colorization. In *ECCV*.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million Image Database for Scene Recognition. *TPAMI*.

Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A.; and Kong, T. 2021. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*.