SimLTD: Simple Supervised and Semi-Supervised Long-Tailed Object Detection

Phi Vu Tran
LexisNexis Risk Solutions

Abstract

While modern visual recognition systems have made significant advancements, many continue to struggle with the open problem of learning from few exemplars. This paper focuses on the task of object detection in the setting where object classes follow a natural long-tailed distribution. Existing methods for long-tailed detection resort to external ImageNet labels to augment the low-shot training instances. However, such dependency on a large labeled database has limited utility in practical scenarios. We propose a versatile and scalable approach to leverage optional unlabeled images, which are easy to collect without the burden of human annotations. Our SimLTD framework is straightforward and intuitive, and consists of three simple steps: (1) pre-training on abundant head classes; (2) transfer learning on scarce tail classes; and (3) fine-tuning on a sampled set of both head and tail classes. Our approach can be viewed as an improved head-to-tail model transfer paradigm without the added complexities of meta-learning or knowledge distillation, as was required in past research. By harnessing supplementary unlabeled images, without extra image labels, SimLTD establishes new record results on the challenging LVIS v1 benchmark across both supervised and semi-supervised settings.

1. Introduction

The task of detecting, localizing, and classifying object instances from image and video is a long-standing problem in computer vision. Recent years have seen unprecedented progress on modern object detection systems, mostly driven by powerful neural architectures. Much of this success is measured on the relatively balanced, small-vocabulary benchmarks like PASCAL VOC [13] and MS-COCO [22]. When evaluated on a more complex and imbalanced dataset with a much larger vocabulary, however, the same models exhibit a considerable drop in detection accuracy [15].

This paper explores ways to enhance the capability of commodity detection systems, with a particular evaluation emphasis on the challenging large-vocabulary LVIS benchmark [15]. LVIS represents a realistic application, a scenario in which object classes follow a natural long-tailed distribution. It is in the tail that most data-hungry models struggle with performance, a distribution characterized by many rare classes having as few as a single training exemplar.

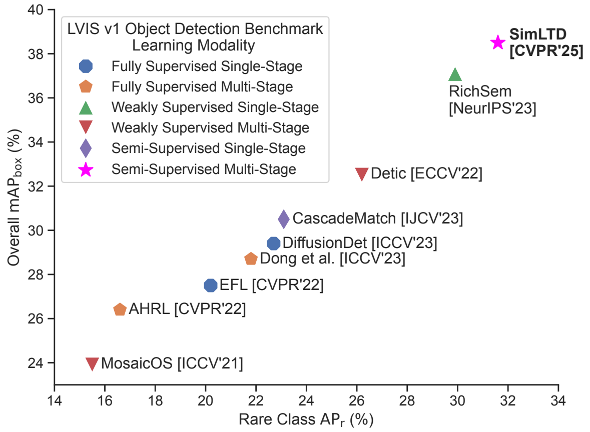


Figure 1. A survey comparing our SimLTD to the state of the art for long-tailed detection. We combine unlabeled data with an intuitive multi-stage training strategy to deliver the best overall performance while also optimizing for accuracy on rare classes. Our simple approach achieves superior results on the challenging LVIS v1 benchmark without requiring auxiliary image-level supervision.

Figure 1 plots several recent state-of-the-art methods to address the extreme disparity of class distributions in long-tailed detection (LTD). One promising direction is to divide the problem into multiple manageable parts to help alleviate the severity of the class imbalance. LST [19] proposes to segment the overall dataset into seven phases, each phase containing progressively smaller but balanced data samples, and train a model in an incremental manner via network expansion and knowledge distillation. While LST adopts innovative ideas from class-incremental learning [4, 31, 40], the method is overly complex with the maintenance of many sub-parts. Moreover, the method requires numerous stages of knowledge transfer that can lead to catastrophic forgetting and result in an inferior solution [12].

Another interesting direction is to leverage external data to augment the training instances in LVIS. The intuition is that while the rare objects may not appear frequently in natural scenes, they can be found in abundance from *object-*

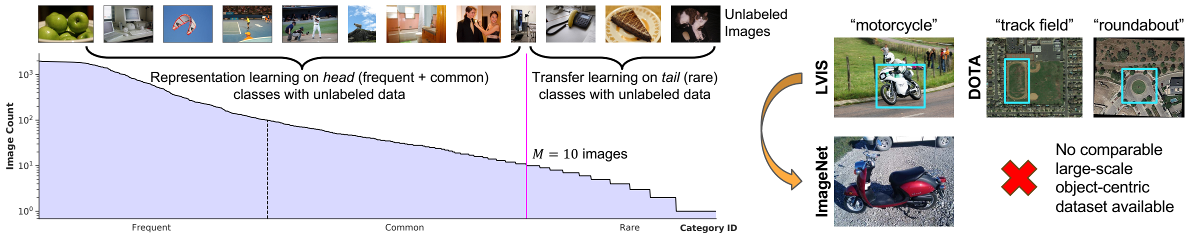


Figure 2. The motivation to our approach. **Left:** We propose an improved head-to-tail model transfer framework for long-tailed detection by incorporating *unlabeled images* in both representation and transfer learning stages. **Right:** While it is possible to find more samples of LVIS instances from ImageNet, such an auxiliary database may not exist in another scenario like aerial imagery. Our framework does not depend on using extra image-level labels to advance long-tailed detection, which expands the applicability of our approach beyond LVIS.

centric sources such as ImageNet [9]. By training on the union of scene-centric LVIS and object-centric ImageNet images, these weakly supervised methods [27, 45, 50] overcome the skewed class distribution by enriching the tail instances with additional whole-image labels. Although appealing, this approach places a hard requirement on having access to a large database of ∼14M *labeled* images across ∼21K object classes, an assumption not necessarily suitable for many practical settings outside of LVIS.

Consider Figure 2 as an example for aerial imagery; there is no comparable object-centric dataset available to match the semantic concepts of "track field", "roundabout", or other categories in DOTA [11]. Moreover, for an industry application focusing on bespoke concepts beyond general objects, building a new object-centric dataset to supplement the main training dataset can be costly and time-consuming. There is also the open question of how to handle the *genuinely rare* classes (*e.g.*, a rare fish species), which are strictly limited in observation and cannot be collected in more quantity from the open Internet. *Can we still advance long-tailed detection without additional hand-labeling?*

We propose a simple and scalable framework, aptly named SimLTD, to answer this question. We deconstruct the LTD problem into three stages consisting of (i) pre-training on *head* classes, (ii) transfer learning on *tail* classes, and (iii) fine-tuning on a small sampled set composed of both head and tail classes. Unlike previous research on multi-stage training [12, 19], our framework allows for the optional use of *unlabeled images* to further boost LTD. We learn with supplementary unlabeled data in a semi-supervised manner via pseudo-labeling, and therefore do not explicitly rely on any additional instance-level or image-level supervision.

Our work addresses several challenges associated with the head-to-tail model transfer paradigm [18, 39]. For one, we find that the vanilla transfer of data-rich head representations to data-scarce tail classes does not yield sufficient performance improvement because the distribution of head classes is also skewed. Moreover, the naïve application of unlabeled data can aggravate the model's inductive bias because the unlabeled samples may follow a similar long-tailed distribution. In such case, the trained model will tend to generate more pseudo labels for the head classes, resulting in a larger degree of pseudo-class imbalance. We overcome these obstacles by incorporating data augmentations specifically designed to mitigate class imbalances in both head and tail training stages. Extensive experiments in §4.3 show that stronger pre-trained models on head classes, with and without unlabeled images, transfer well to the tail classes, leading to a desirable solution for long-tailed detection.

Main Contributions **(1)** We present SimLTD, a general framework for effective supervised and semi-supervised LTD with unlabeled images. **(2)** The design of SimLTD is straightforward and intuitive, and is compatible with a range of backbones and detectors based on both classical convolutional and modern transformer architectures. **(3)** When put to the test against competing methods on the challenging LVIS v1 benchmark, SimLTD demonstrates excellent performance and scalability by establishing new state-of-the-art results with compelling margins. We hope SimLTD serves as a strong baseline for future research to tackle realistic long-tailed problems in the community.

2. Related Work

Multi-Stage Learning for LTD Multi-stage approaches typically begin with a step on representation learning followed by a transfer learning or fine-tuning routine. The earlier work of LST [19] pre-trained a model on a subset of many-shot head classes then transferred its representations to progressively smaller but balanced scarce tail samples via knowledge distillation. A more recent method [12] pre-trains a model on the entire long-tailed dataset, and opts to fine-tune the resulting model on "smooth-tail" data to mitigate the effects of class imbalance. These approaches assume pre-trained representations provide better initialization than random weights for fine-tuning on tail classes.

In §3.1, we analyze that assumption to be generally true. We leverage this finding and take a different but related approach to pre-train on the frequent and common head instances. Then we simply copy the head representations

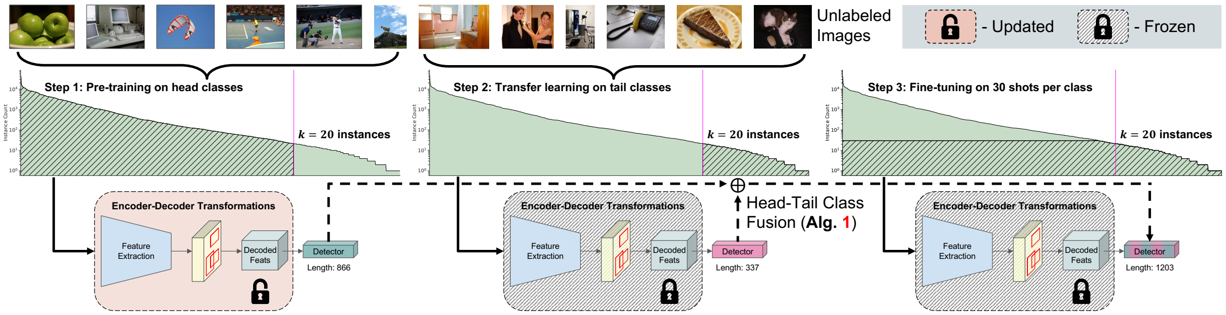

Unlabeled Images
- Updated
- Frozen
Step 1: Pre-training on head classes
Step 2: Transfer learning on tail classes
Step 3: Fine-tuning on 30 shots per class
Instance Count
10^0
10^1
10^2
10^3
10^4
k = 20 instances
Encoder-Decoder Transformations
Feature Extraction
Decoded Feats
Detector
Length: 866
Length: 337
Length: 1203
Head-Tail Class Fusion (Alg. 1)

Figure 3. Overview of SimLTD. **Step 1** pre-trains the model and detector on head classes with unlabeled images. **Step 2** transfers the head representations to tail classes. **Step 3** fine-tunes the detector on a sampled set of head and tail classes. The "model" is an abstract block of encoder-decoder transformations based on either a convolutional or transformer network. The model is updated only during pre-training.

and fine-tune them on the tail classes without resorting to knowledge distillation or meta-learning, as was required in prior studies [19, 39]. Our approach takes three stages of learning compared to the seven stages of LST, which can exacerbate catastrophic forgetting. A novel component to our approach is the introduction of unlabeled images in both pre-training and transfer learning stages for enhanced LTD.

Leveraging Extra Data for LTD There is an emerging trend of using the abundant weak-supervision of ImageNet labels to solve the LVIS problem. Although LVIS is a detection dataset of natural scenes and ImageNet is a classification dataset of object images, they both share 997 overlapping classes between their vocabularies, the intersection of which provides a rich source of ∼1.5M extra images to sample additional LVIS instances. To effectively learn from a mixed LVIS-ImageNet dataset with a domain gap, MosaicOS [45] leverages mosaic augmentation [1] whereas Detic [50] and RichSem [27] rely on the CLIP [30] classifier to map the semantic concepts between LVIS and ImageNet targets.

As discussed in §1, these methods put a strict dependency on a large auxiliary labeled database to augment the main training dataset, which is infeasible for bespoke applications outside of LVIS. Alternatively, we propose a more general and flexible solution to use unlabeled data as a source of auxiliary supervision, which is easy to collect without the burden of human annotations. While our framework is not the first to leverage unlabeled data for LTD, ours is more effective when compared to the competing method of CascadeMatch [44], thanks to our multi-stage training strategy.

Connection to Few-Shot Detection Long-tailed detection is related to the task of few-shot detection (FSD), the purpose of which is to adapt a base detector (trained on many-shot instances) to learn new concepts from few-shot exemplars. The commonality between the two is obvious—both tasks aim to boost detection on categories with very few training instances. However, LTD has unique challenges that extend beyond FSD. For LTD, the tail classes are authentically rare that follow a Zipf distribution [52] in natural scenes. By contrast, the novel few-shot exemplars in FSD datasets are randomly sampled and are not necessarily rare but can include objects of varying degrees of observational frequency. As such, the multi-stage training methods that work well for FSD [35, 38] cannot be directly applied to LTD with the same expected level of effectiveness. We need to devise ways to adapt these methods to the LTD problem to bring improvement over the state of the art.

3. The SimLTD Framework

As illustrated in Figure 2, given a long-tailed dataset $\mathcal{D}_{\text{ltd}}$ with C categories, we split it into two disjoint subsets: $\mathcal{D}_{\text{head}}$ with C_{head} frequent and common categories appearing in $> M$ images and $\mathcal{D}_{\text{tail}}$ with C_{tail} rare classes appearing in $\leq M$ images. Furthermore, we have access to an unlabeled dataset $\mathcal{D}_{\text{u}}$ of unknown class distribution. Our goal is to use a combination of labeled and unlabeled images to train a unified model optimized for accuracy on a test set comprising both classes in $C_{\text{head}} \cup C_{\text{tail}}$.

Our SimLTD framework consists of three easy steps: (i) representation learning on $\mathcal{D}_{\text{head}}$, (ii) transfer learning on $\mathcal{D}_{\text{tail}}$, and (iii) fine-tuning on $\mathcal{D}_k$, a reduced dataset composed of k instances per class randomly sampled from $\mathcal{D}_{\text{ltd}}$. Note that $\mathcal{D}_{\text{head}}$, $\mathcal{D}_{\text{tail}}$, and $\mathcal{D}_k$ are all still imbalanced, but not as severe as the original long-tailed $\mathcal{D}_{\text{ltd}}$. We leverage *optional* unlabeled images in both Steps 1 and 2 but do not explicitly need them for effective LTD. Indeed, experiments in §4.3 show that our fully supervised baselines exhibit excellent performance and scalability even without unlabeled data. A diagram of SimLTD is depicted in Figure 3.

3.1. The Devil is in the $\mathcal{D}_{\text{tail}}$

The open challenge of the LTD problem remains in learning an effective model on the few exemplars associated with the tail classes. We draw inspiration from existing empirical evidence that few-shot learning can vastly benefit from pre-trained representations [35, 38]. With that in mind, we

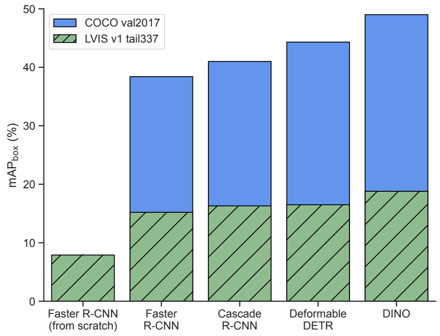


Figure 4. Transfer learning from COCO representations (solid bars) helps improve rare class detection on LVIS (hatched green bars).

Augmentation	Supervised Training	Semi-Supervised Training
Resize	short edge $\in [400, 1200]$	short edge $\in [400, 1200]$
Flip	horizontal	horizontal
SCP with RFS	sample threshold $= 0.001$	sample threshold $= 0.001$
AutoContrast	✓	✓
Equalize	✓	✓
Solarize	✓	✓
Color Jittering	✓	✓
Contrast	✓	✓
Brightness	✓	✓
Sharpness	✓	✓
Posterize	✓	✓
Translation	✗	$(x, y) \in (-0.1, 0.1)$
Shearing	✗	$(x, y) \in (-30°, 30°)$
Rotation	✗	angle $\in (-30°, 30°)$
Cutout	✗	$n \in [1, 5]$, size $\in [0.0, 0.2]$

Table 1. Summary of the data augmentations explored in this study to improve supervised and semi-supervised training.

conduct an empirical analysis to verify whether our intuition extends to the LVIS setting. Following conventional LVIS protocol, we partition the dataset into $C_{\text{head}} = 866$ common classes appearing in $M > 10$ images and $C_{\text{tail}} = 337$ rare classes appearing in $M \leq 10$ images. We aim to improve the detection performance on C_{tail} using various commodity detectors pre-trained on the COCO dataset.

Figure 4 quantifies the effectiveness of pre-trained representations on C_{tail} detection. For each model, we chop off the detection head consisting of the bounding box classifier and regressor modules learned on the COCO dataset, re-initialize them with random values, and perform transfer learning on the LVIS tail classes. We update only the box classifier and regressor while keeping the rest of the architecture frozen, essentially treating the pre-trained model as a fixed detector. Besides the pre-trained networks, we also assess the Faster R-CNN detector [32] initialized from scratch, except for the pre-trained backbone, as the lower-bound baseline.

Discussion We observe a clear trend indicating stronger pre-trained representations, as measured by the AP score on COCO, generally lead to improved rare class detection. This result is both intriguing and encouraging since the models were pre-trained on COCO, a dataset of different scope and size than LVIS. Training from scratch brings out the worst performance with half of the accuracy. The implication of this simple experiment is two-fold: (1) we corroborate prior studies by showing low-shot learning can be improved with transferred representations; and (2) our framework opens opportunities to self-supervised, semi-supervised, and multi-modal learning, all of which have demonstrated significantly better performance than supervised pre-training. Motivated by these insights, we propose to learn powerful representations on $\mathcal{D}_{\text{head}}$ and transfer them to $\mathcal{D}_{\text{tail}}$ for long-tailed detection. While the past attempts at head-to-tail model transfer [12, 19, 39] could only work by incorporating an extra module for meta-learning or knowledge distillation, we now describe our three-step approach to accomplish this goal without the unnecessary complexities.

3.2. Step 1: Representation Learning on $\mathcal{D}_{\text{head}}$

We begin with the supervised setting in which we have training data points $(x_i, y_i) \in \mathcal{D}_{\text{head}}$, where x_i denotes the i-th input image and y_i is the i-th ground truth annotation containing the box label and coordinates. Let $\Psi_{\text{det}}(\mathcal{D}_{\text{head}})$ be a learnable detection function training on the image-target pairs to produce the supervised loss $\mathcal{L}_{\text{sup}}$:

$$\Psi_{\text{det}}(\mathcal{D}_{\text{head}}) \leftarrow \mathcal{L}_{\text{sup}} = \sum L_{\text{cls}}(h(x_i), y_i) + L_{\text{reg}}(h(x_i), y_i). \tag{1}$$

Here, $h(x_i)$ denotes the forward pass on the input image and $(L_{\text{cls}}, L_{\text{reg}})$ are classification (*e.g.*, cross-entropy) and regression (*e.g.*, L_1) losses for the detector. We consider commodity convolutional and transformer-based networks for Ψ_{det}. For the convolutional network, we experiment with Faster R-CNN to compare against previous studies using similar detectors (*i.e.*, Mask R-CNN [17] and RetinaNet [24]). For the contemporary transformer-based network, we adopt the improved variants of the Detection Transformer (DETR) [3], namely Deformable DETR [51] and DINO [47].

As summarized in Table 1, we leverage well-known data augmentations to train Ψ_{det}, which include random image resizing, horizontal flipping, and photometric distortion. We also combine Simple Copy-Paste (SCP) [14] together with Repeat Factor Sampling (RFS) [15] to combat the class imbalance in $\mathcal{D}_{\text{head}}$ at both the image and instance levels, analogous to prior research on the importance of image and instance resampling for LTD [37, 42]. Taking a step further to learn even stronger representations on $\mathcal{D}_{\text{head}}$, we propose to train Ψ_{det} in a semi-supervised manner with unlabeled images by way of pseudo-labeling, which is currently leading the literature on semi-supervised detection. We explore three successful methods of increasing effectiveness to advance semi-supervised representation learning on $\mathcal{D}_{\text{head}}$: SoftER Teacher [35], MixTeacher [25], and MixPL [7].

SoftER Teacher builds upon the end-to-end pseudo-labeling approach of Soft Teacher [41] to include an auxiliary loss for consistency learning on region proposals, and was

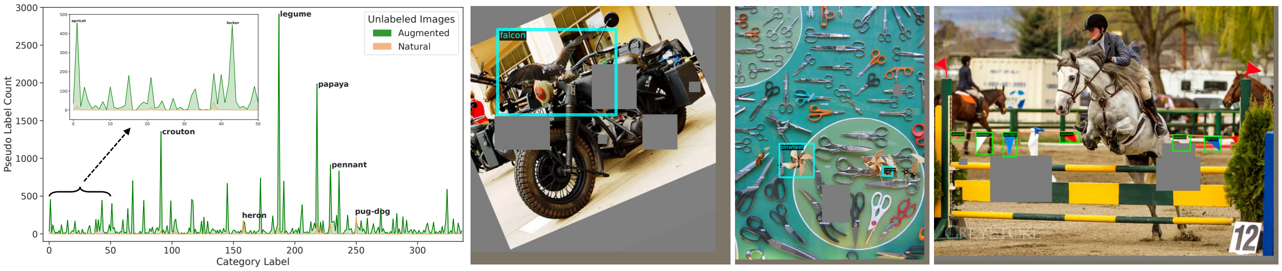


Figure 5. **Left:** Augmenting unlabeled images by randomly pasting rare instances from the training set helps promote pseudo-labeling for semi-supervised LTD. **Middle:** The pasted objects (cyan boxes) often come from contrasting environments to create complex *fake* scenes for the student to learn, which in turn can improve robustness and generalization on *natural* scenes (green boxes) shown on the **Right**.

shown to work particularly well with semi-supervised few-shot detection. MixTeacher introduces a mixed scale feature pyramid to generate more accurate pseudo labels on objects with extreme scale variations, resulting in an overall robust detector. While SoftER Teacher and MixTeacher primarily work with two-stage detectors like Faster R-CNN, MixPL opens the door to semi-supervised learning with single-stage and DETR-based models by integrating mixup [46] and mosaic [1] augmentations with pseudo labels. All three methods follow the student-teacher semi-supervised training paradigm [34], in which the teacher is an exponential moving average of the student. In the semi-supervised setting, the models learn from a joint dataset of labeled $\mathcal{D}_{\text{head}}$ and unlabeled $\mathcal{D}_{\text{u}}$ images via the following compound objective:

$$\Psi_{\text{semi-det}}(\mathcal{D}_{\text{head}}, \mathcal{D}_{\text{u}}) \leftarrow \mathcal{L} = \mathcal{L}_{\text{sup}} + \alpha \mathcal{L}_{\text{pseudo}}, \tag{2}$$

where $\alpha > 0$ controls the contribution of the pseudo-label loss derived from unlabeled data. The functional form of $\mathcal{L}_{\text{pseudo}}$ is the same as $\mathcal{L}_{\text{sup}}$ in Equation (1), except the ground-truth targets y are replaced by pseudo targets $\hat{y}$ predicted by the teacher model during self-training.

3.3. Step 2: Transfer Learning on $\mathcal{D}_{\text{tail}}$

We instantiate the tail models for transfer learning by copying the parameters from the pre-trained head models. Let $\Psi'_{\text{det}}(\mathcal{D}_{\text{tail}}) \leftarrow \Psi_{\text{det}}(\mathcal{D}_{\text{head}})$ be the supervised tail model and $\Psi'_{\text{semi-det}}(\mathcal{D}_{\text{tail}}, \mathcal{D}_{\text{u}}) \leftarrow \Psi_{\text{semi-det}}$ be the semi-supervised counterpart. We train the tail models the same way per Equations (1) and (2), except we update only the classifier and regressor modules, to adapt them to tail classes, while freezing the rest of the networks. The intuition is that pre-trained representations serve as a bootstrapped initializer to train an accurate tail model, according to our analysis in §3.1.

Recall that, unlike common head objects, the tail classes are intrinsically rare so one cannot expect to find abundant occurrences of them in either labeled or unlabeled source. This raises a hurdle for when we wish to train $\Psi'_{\text{semi-det}}$ with unlabeled images: there are very few instances of the rare classes in the unlabeled scenes for the teacher model to propose reliable pseudo targets. We sidestep this hurdle by copying and pasting a random subset of rare instances from the labeled training set to the unlabeled images—a new procedure unique to this work. At each training iteration, the teacher model is guaranteed to see an augmented view of sampled rare objects, amid diverse background scenes, which promotes pseudo-label supervision for the student model. Figure 5 illustrates the impact of this technique on pseudo-labeling along with some examples of the augmented unlabeled images, which are subjected to strong photometric and geometric perturbations with cutout [10, 49]. Although the procedure inevitably leads to redundancy and overfitting, our ablation experiments in §4.5 show that it is surprisingly helpful in adapting head representations to the tail models.

Algorithm 1 PyTorch Pseudocode for Head-Tail Class Fusion.

```
# HEAD_IDS : sorted list of head IDs, length 866
# TAIL_IDS : sorted list of tail IDs, length 337
# head_ckpt: model checkpoint on head classes
# tail_ckpt: model checkpoint on tail classes

ALL_IDS = sorted(HEAD_IDS + TAIL_IDS) # length 1203
ID2LABEL = {
   ID: label for label, ID in enumerate(ALL_IDS)
} # mapping from category ID to integer label
head_det = head_ckpt["state_dict"]["detector"]
tail_det = tail_ckpt["state_dict"]["detector"]
fused_det = torch.randn(len(ALL_IDS))

for label, ID in enumerate(HEAD_IDS):
   fused_det[ID2LABEL[ID]] = head_det[label]
for label, ID in enumerate(TAIL_IDS):
   fused_det[ID2LABEL[ID]] = tail_det[label]

head_ckpt["state_dict"]["detector"] = fused_det
torch.save(head_ckpt, save_filename) # to fine-tune
```

3.4. Step 3: Fine-Tuning on $\mathcal{D}_k$

At this stage, we have two separate models with a shared representation, one optimized on head classes and the other on tail classes. We wish to unify the two models into one for efficient single-pass inference on test samples containing both head and tail categories. Algorithm 1 provides the pseudocode for the merging scheme referenced in Figure 3.

Note that we merge parameters at the detector module and

Method	Extra External Data	Base Detector	Backbone	GPU Hrs	mAP$_{\text{box}}$	AP$_{\text{r}}$	AP$_{\text{c}}$	AP$_{\text{f}}$
Seesaw Loss [CVPR'21] [36]	N/A	Faster R-CNN	R101-FPN	–	27.8	18.7	27.0	32.8
NorCal [NeurIPS'21] [28]		Mask R-CNN	R101-FPN	–	28.1	20.8	26.5	30.9
AHRL [CVPR'22] [20]		Mask R-CNN	R101-FPN	–	28.7	19.3	27.6	31.4
EFL [CVPR'22] [21]		RetinaNet	R101-FPN	–	29.2	23.5	27.4	33.8
LogN [IJCV'24] [48]		Mask R-CNN	R101-FPN	–	29.8	22.9	28.8	31.8
SimLTD Supervised Baseline		Faster R-CNN	R101-FPN	83	**31.7**	**24.3**	**32.1**	**34.6**
SimLTD Supervised Baseline		Deformable DETR	R101	234	**37.0**	**31.9**	**36.6**	**39.6**
Dong et al. [ICCV'23] [12]	N/A	Deformable DETR	R50	–	28.7	21.8	28.4	32.0
Detic [ECCV'22] [50]		Deformable DETR	R50	–	31.7	21.4	30.7	37.5
RichSem [NeurIPS'23] [27]		DINO	R50	–	**35.1**	26.0	32.6	**41.8**
SimLTD Supervised Baseline		Faster R-CNN	R50-FPN	81	29.0	20.9	29.7	31.9
SimLTD Supervised Baseline		Deformable DETR	R50	215	35.0	**32.0**	**34.0**	37.5
RichSem [NeurIPS'23] [27]	N/A	DINO	Swin-T	–	38.8	30.8	36.4	45.0
SimLTD Supervised Baseline		DINO	Swin-T	310	**41.1**	**33.6**	**40.1**	**45.4**
DiffusionDet [ICCV'23] [6]		4 @ 300	Swin-B	–	42.0	34.8	40.9	46.4
RichSem [NeurIPS'23] [27]		DINO	Swin-B	–	46.4	38.5	45.1	**51.3**
SimLTD Supervised Baseline		DINO	Swin-B	414	**47.2**	**42.7**	**46.7**	49.9
MosaicOS [ICCV'21] [45]	ImageNet-1K Labels	Faster R-CNN	R50-FPN	–	23.9	15.5	22.4	29.3
RichSem [NeurIPS'23] [27]	ImageNet-21K Labels	Faster R-CNN + CLIP	R50-FPN	–	30.6	**27.6**	29.7	32.9
SimLTD SoftER Teacher [35]	COCO-unlabeled2017	Faster R-CNN	R50-FPN	392	30.3	23.3	30.3	33.3
SimLTD MixTeacher [25]	COCO-unlabeled2017	Faster R-CNN	R50-FPN	434	**31.8**	23.4	**32.1**	**35.1**
Detic [ECCV'22] [50]	ImageNet-21K Labels	DeformDETR + CLIP	R50	–	32.5	26.2	31.3	36.6
RichSem [NeurIPS'23] [27]	ImageNet-21K Labels	DINO + CLIP	R50	–	37.1	29.9	35.6	42.0
SimLTD MixPL [7]	Objects365-unlabeled	DINO	R50	447	38.3	30.8	**37.5**	**42.6**
SimLTD MixPL [7]	COCO-unlabeled2017	DINO	R50	446	**38.5**	**31.6**	**37.5**	42.5
RichSem [NeurIPS'23] [27]	ImageNet-21K Labels	DINO + CLIP	Swin-T	–	41.6	**37.3**	39.7	45.5
SimLTD MixPL [7]	COCO-unlabeled2017	DINO	Swin-T	482	**42.5**	35.7	**42.4**	**45.6**
RichSem [NeurIPS'23] [27]	ImageNet-21K Labels	DINO + CLIP	Swin-B	–	48.2	**46.5**	46.5	51.0
SimLTD MixPL [7]	COCO-unlabeled2017	DINO	Swin-B	794	**49.0**	43.4	**49.0**	**51.5**

Table 2. Main results on LVIS v1 validation. GPU hours denote the wall clock time to train for a total of 640K iterations and are a proxy measure of model complexity. The ResNet and Swin backbones were pre-trained on ImageNet-1K and ImageNet-22K, respectively. The results of Seesaw Loss and Detic are borrowed from EFL and RichSem, respectively. Shaded rows indicate our implemented models.

reuse the pre-trained head representations for the rest of the network. We fine-tune the unified detector on $\mathcal{D}_k$ composed of k instances, or shots, per class sampled from the long-tailed training set. We update only the box classifier and regressor with a reduced learning rate to slowly adapt them to tail classes while preserving the pre-trained accuracy on head classes. Analogous to class-incremental learning, we include both head and tail classes in $\mathcal{D}_k$ for exemplar replay to avoid catastrophic forgetting. We form $\mathcal{D}_k$ via random sampling whereas prior work resorted to a complicated scheme of confidence-guided exemplar replay [12].

4. Empirical Study

4.1. Evaluation Protocol

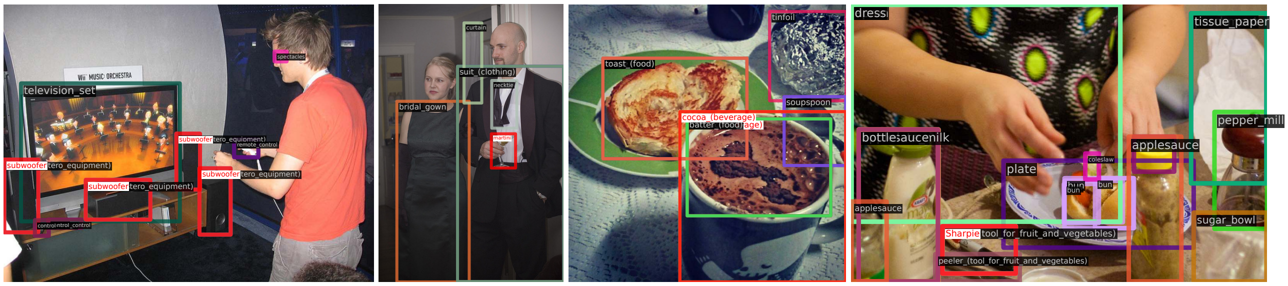
We benchmark our approach on the challenging LVIS v1 dataset, which has 100170 training and 19809 validation images over 1203 classes. We use the standard LVIS evaluator to compute the detection metric mAP$_{\text{box}}$ for all classes, without test-time augmentation, along with AP$_{\text{r}}$, AP$_{\text{c}}$, and AP$_{\text{f}}$ for the rare, common, and frequent categories. We sample $\mathcal{D}_k$ three times with different random seeds and report on the averaged metrics to capture statistical variability. Following prior studies, we focus on the performance gains of mAP$_{\text{box}}$ and AP$_{\text{r}}$ in our comparative analysis.

4.2. Implementation Details

We implement our models using PyTorch [29] and MMDetection [5], and train on 8× A6000 GPUs. We pre-train in Step 1 for up to 540K iterations, perform transfer learning in Step 2 for 40K iterations, and fine-tune for 80K iterations. See our open-source code for the full reproducible details.

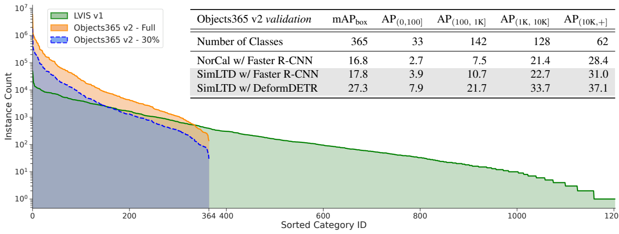
High-Quality Supervised Baseline We construct a stronger supervised baseline than previously by combining our multi-stage training recipe with diverse data augmentations. We explore various ResNet [16] and Swin [26] backbones, along with FPN [23], for feature extraction. The supervised baseline is important to our framework because it serves as the basis for the teacher model to propose reliable pseudo targets for semi-supervised learning.

Semi-Supervised LTD We leverage SoftER Teacher, Mix-



$k = 1$ shot $k = 3$ shots $k = 4$ shots $k = 5$ shots

Figure 6. SimLTD detections on LVIS v1 validation images. We highlight visualizations containing truly rare k-shot exemplars from the training set—drawn with red and white boxes. SimLTD does well in the extremely low-shot regime using as few as a single training instance.



Objects365 v2 *validation*	mAP_{box}	$\text{AP}_{(0,100]}$	$\text{AP}_{(100,\ 1\text{K}]}$	$\text{AP}_{(1\text{K},\ 10\text{K}]}$	$\text{AP}_{(10\text{K},+]}$
Number of Classes	365	33	142	128	62
NorCal w/ Faster R-CNN	16.8	2.7	7.5	21.4	28.4
SimLTD w/ Faster R-CNN	17.8	3.9	10.7	22.7	31.0
SimLTD w/ DeformDETR	27.3	7.9	21.7	33.7	37.1

Figure 7. Evaluation on Objects365 binned by the count of training images per class group. All models use the ResNet-50 backbone.

Method	Base Detector	Backbone	$\text{AP}^{\text{fixed}}_{\text{box}}$	$\text{AP}^{\text{fixed}}_{\text{r}}$	$\text{AP}^{\text{fixed}}_{\text{c}}$	$\text{AP}^{\text{fixed}}_{\text{f}}$
CascadeMatch (Sup)	Cascade R-CNN		27.1	20.3	26.1	31.1
FASA (Supervised)	Mask R-CNN	R101-FPN	28.2	22.0	28.3	30.9
SimLTD Supervised	Faster R-CNN		**32.7**	**24.9**	**32.9**	**35.9**
CascadeMatch	Cascade R-CNN		30.5	23.1	29.7	34.7
SimLTD SoftER Teacher	Faster R-CNN	R50-FPN	31.3	24.1	31.1	34.6
SimLTD MixTeacher	Faster R-CNN		**32.8**	**24.5**	**32.9**	**36.4**
CascadeMatch	Cascade R-CNN		32.9	**26.5**	31.8	36.8
SimLTD SoftER Teacher	Faster R-CNN	R101-FPN	33.0	26.1	32.6	36.4
SimLTD MixTeacher	Faster R-CNN		**34.4**	26.1	**34.2**	**38.2**

Table 3. Performance comparison between SimLTD, FASA [43], and CascadeMatch [44] using the alternative AP^{Fixed} metric [8].

Teacher, and MixPL for semi-supervised LTD, and inherit all hyper-parameters originally tuned on the COCO dataset without changes. We harness ~123K COCO-unlabeled2017 images to improve both representation and transfer learning in Steps 1 and 2. We also experiment with Objects365 [33] to further validate our approach on another related domain with ~1.7M unlabeled images in the wild by removing all label information from the training set.

4.3. Main Results

Table 2 reports on the effectiveness of our approach against existing methods representing the state of the art on LVIS. Our SimLTD supervised baseline with Faster R-CNN outperforms all methods using related detectors. The gains are convincing, with margins up to $+3.9$ AP_{box} and $+5.6$ AP_{r}. We observe a similar trend when comparing our supervised baseline using DETR-based models. SimLTD demonstrates compelling performance and scalability across a multitude of backbones and detectors without the need for extra data.

For the methods requiring external data, our semi-supervised models also deliver impressive performance. When equipped with MixTeacher and Faster R-CNN, SimLTD exceeds the competition by up to $+7.9$ AP_{box} while being competitive on AP_{r}. Furthermore, SimLTD scales well by coupling with MixPL and transformer-based models to achieve new state-of-the-art results from harnessing only unlabeled images. SimLTD works equally well with both COCO and Objects365 unlabeled images, signifying that our model can extract meaningful pseudo-label supervision from a large uncurated database with a distribution different from the training dataset. Figure 6 visualizes example SimLTD detections on select LVIS validation images.

Ancillary to the main LVIS results, we also evaluate on Objects365 to showcase the generality of our framework. Following NorCal [28], we sample 30% of the training set and split it into $C_{\text{head}} = 332$ classes appearing in $M > 100$ images and $C_{\text{tail}} = 33$ classes appearing in $M \leq 100$ images. SimLTD outperforms the existing baseline across the board in Figure 7. We fine-tune SimLTD with $k = 30000$ shots, a parameter that changes by dataset. Section 4.5 gives a detailed ablation analysis regarding the impact on AP from varying the number of shots for fine-tuning with $\mathcal{D}_k$.

4.4. Comparison to the State of the Art

SimLTD *vs*. CascadeMatch [44] Although both SimLTD and CascadeMatch leverage COCO-unlabeled2017 for semi-supervised LTD, there are major differences between the two. First, CascadeMatch is trained end-to-end in a single stage whereas we take the decoupled approach. Second, CascadeMatch adopts the stronger Cascade R-CNN [2] compared to Faster R-CNN in SimLTD. CascadeMatch follows the AP^{Fixed} protocol [8], which replaces the standard maximum 300 detections per image by a cap of 10K detections per class from the entire validation set. Despite the disadvantage of a simpler model, Table 3 shows that SimLTD outperforms CascadeMatch by notable margins in almost every measure.

Configuration	mAP_{box}	AP_{r}
Single-Stage Training +++Copy-Paste	28.1	16.8
Multi-Stage Baseline (w/ Random Resize)	26.8	17.9
Multi-Stage Baseline +Photometric Jittering	26.9	18.2
Multi-Stage Baseline ++Repeat Sampling	27.1	19.3
Multi-Stage Baseline +++Copy-Paste (Ours)	**29.0**	**20.9**

Table 4. Ablation experiments quantifying the effectiveness of each component in our SimLTD supervised baseline. The single-stage model is trained end-to-end on the whole long-tailed dataset.

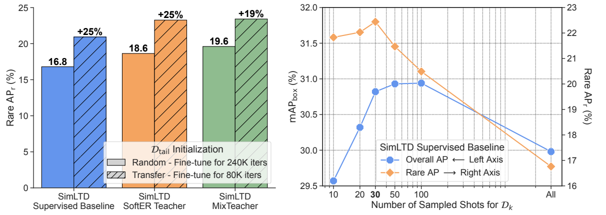


Figure 8. Ablation experiments assessing the impact on AP from transfer learning on tail classes **(left)** and from varying the number of sampled shots for fine-tuning with $\mathcal{D}_k$ **(right)**.

These superior results lend further support to the merit of our multi-stage training strategy.

SimLTD *vs*. Dong et al. [12] on Multi-Stage Learning We compare our multi-stage training strategy to that of Dong et al., which also utilizes a three-step procedure of pre-training followed by fine-tuning and knowledge distillation. We focus our analysis on their powerful Deformable DETR model, which yields similar results to our simpler Faster R-CNN model. When we train with the same capable Deformable DETR architecture, Table 2 shows that SimLTD exceeds their model by outsized margins of $+6.3$ AP_{box} and $+10.2$ AP_{r}. These remarkable gains are directly attributed to our multi-stage learning approach, which is carefully designed to optimize for accuracy on both head and tail classes.

SimLTD *vs*. RichSem [27] on Using Extra Data Recall that RichSem relies on the CLIP classifier (pre-trained on $\sim$400M image-caption pairs) and image-supervision from an additional $\sim$1.5M images to produce the state-of-the-art results reported in Table 2. However, such strict dependencies are impractical when the method is applied to a bespoke dataset outside of generic objects. By contrast, our SimLTD leverages unlabeled images, without resorting to either CLIP or auxiliary ImageNet labels, to deliver better results than RichSem with the ResNet backbone and slightly worse AP_{r} with the Swin backbones. When we remove external data, SimLTD substantially outperforms RichSem by $+6.0$ AP_{r} in the fully supervised setting, implying that the success of RichSem is sensitive to the contributions of CLIP and ImageNet supervision. Our framework carries the benefit of being robust across settings with and without external data.

4.5. Ablation Experiments

Design of SimLTD Baseline The design of SimLTD is centered on an intuitive multi-stage training strategy combined with standard data augmentations, without bells and whistles, to overcome the class imbalances in both head and tail datasets. Table 4 shows the contributions of RFS and Copy-Paste in establishing a more robust baseline than was previously possible in the existing literature, by using the simple Faster R-CNN detector with ResNet-50 FPN. We also showcase the viability of multi-stage learning over the naïve single-stage training procedure on the whole long-tailed dataset, which yields markedly worse results.

The Impact of Transfer Learning on $\mathcal{D}_{\text{tail}}$ As discussed in §3.3, we transfer the pre-trained head representations to optimize on tail classes in Step 2 of our framework. However, this step is optional and may be skipped, because we can initialize the tail classes with random values before fine-tuning. Figure 8 shows that transfer learning on tail classes is a worthwhile task. Across both supervised and semi-supervised settings, transfer learning gives a boost by up to $+4.7$ AP_{r} and comes with the added bonus of shortening the fine-tuning time by $^2/_3$ of the required iterations.

How Many Shots to Sample for $\mathcal{D}_k$? Figure 8 illustrates the impact on mAP_{box} and AP_{r} as a function of sampled shots for fine-tuning with $\mathcal{D}_k$, ranging from 10 to "All" meaning the entire long-tailed training set. The aim of this experiment is to optimize for accuracy on rare classes while mitigating catastrophic forgetting on pre-trained head representations. We analyze the "knee in the curve" and find that 30-shots balance the trade-off between the two metrics. Figure 3 visualizes that with 30-shots, we sample the whole tail distribution containing 20 or fewer instances per class and include a mixture of head categories for exemplar replay. Moving to the left of this "sweet spot" with $\{10, 20\}$-shots, we observe marked reductions in mAP_{box}, indicating adverse forgetting on head classes from insufficient samples. Moving to the right of the sweet spot, we see a precipitous drop in AP_{r} in response to the overwhelming amount of head samples.

5. Conclusion

We introduced SimLTD, a simple and versatile framework for supervised and semi-supervised long-tailed detection. Standing out from existing work, SimLTD delivers excellent performance and scalability by achieving new state-of-the-art results on the challenging LVIS v1 benchmark, without requiring auxiliary image-level supervision. We hope the practitioner finds utility in our multi-stage training approach and for our work to spur future research aimed at pushing the performance envelope of long-tailed detection.

Acknowledgments

The author thanks Victor Palmer for the insightful discussion, Brian Keller, Andrew Melkonian, Mehran Ghandehari, Elan Ding, and anonymous reviewers for their thoughtful and constructive feedback on this paper.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. https://arxiv.org/abs/2004.10934, 2020. 3, 5

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into High Quality Object Detection. In *CVPR*, 2018. 7

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, 2020. 4

[4] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-End Incremental Learning. In *ECCV*, 2018. 1

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. https://arxiv.org/abs/1906.07155, 2019. 6

[6] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. DiffusionDet: Diffusion Model for Object Detection. In *ICCV*, 2023. 6

[7] Zeming Chen, Wenwei Zhang, Xinjiang Wang, Kai Chen, and Zhi Wang. Mixed Pseudo Labels for Semi-Supervised Object Detection. https://arxiv.org/abs/2312.07006, 2023. 4, 6

[8] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross Girshick. Evaluating Large-Vocabulary Object Detectors: The Devil is in the Details. https://arxiv.org/abs/2102.01066, 2021. 7

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2

[10] Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. https://arxiv.org/abs/1708.04552, 2017. 5

[11] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Micheal Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges. *IEEE TPAMI*, 2021. 2

[12] Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Boosting Long-Tailed Object Detection via Step-Wise Learning on Smooth-Tail Data. In *ICCV*, 2023. 1, 2, 4, 6, 8

[13] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 1

[14] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. In *CVPR*, 2021. 4

[15] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *CVPR*, 2019. 1, 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 6

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 4

[18] Grant Van Horn and Pietro Perona. The Devil is in the Tails: Fine-Grained Classification in the Wild. https://arxiv.org/abs/1709.01450, 2017. 2

[19] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to Segment the Tail. In *CVPR*, 2020. 1, 2, 3, 4

[20] Banghuai Li. Adaptive Hierarchical Representation Learning for Long-Tailed Object Detection. In *CVPR*, 2022. 6

[21] Bo Li, Yongqiang Yao, Jingru Tan, Gang Zhang, Fengwei Yu, Jianwei Lu, and Ye Luo. Equalized Focal Loss for Dense Long-Tailed Object Detection. In *CVPR*, 2022. 6

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 1

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 6

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, 2017. 4

[25] Liang Liu, Boshen Zhang, Jiangning Zhang, Wuhao Zhang, Zhenye Gan, Guanzhong Tian, Wenbing Zhu, Yabiao Wang, and Chengjie Wang. MixTeacher: Mining Promising Labels with Mixed Scale Teacher for Semi-Supervised Object Detection. In *CVPR*, 2023. 4, 6

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*, 2021. 6

[27] Lingchen Meng, Xiyang Dai, Jianwei Yang, Dongdong Chen, Yinpeng Chen, Mengchen Liu, Yi-Ling Chen, Zuxuan Wu, Lu Yuan, and Yu-Gang Jiang. Learning from Rich Semantics and Coarse Locations for Long-Tailed Object Detection. In *NeurIPS*, 2023. 2, 3, 6, 8

[28] Tai-Yu Pan, Cheng Zhang, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. On Model Calibration for Long-Tailed Object Detection and Instance Segmentation. In *NeurIPS*, 2021. 6, 7

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. Curran Associates, Inc., 2019. 6

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 3

[31] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*, 2017. 1

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 4

[33] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In *ICCV*, 2019. 7

[34] Antti Tarvainen and Harri Valpola. Mean Teachers are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. In *NeurIPS*, 2017. 5

[35] Phi Vu Tran. LEDetection: A Simple Framework for Semi-Supervised Few-Shot Object Detection. In *AISTATS*, 2024. 3, 4, 6

[36] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw Loss for Long-Tailed Instance Segmentation. In *CVPR*, 2021. 6

[37] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The Devil is in Classification: A Simple Framework for Long-Tail Object Detection and Instance Segmentation. In *ECCV*, 2020. 4

[38] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E. Gonzalez, and Fisher Yu. Frustratingly Simple Few-Shot Object Detection. In *ICML*, 2020. 3

[39] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to Model the Tail. In *NeurIPS*, 2017. 2, 3, 4

[40] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *CVPR*, 2019. 1

[41] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-End Semi-Supervised Object Detection with Soft Teacher. In *ICCV*, 2021. 4

[42] Burhaneddin Yaman, Tanvir Mahmud, and Chun-Hao Liu. Instance-Aware Repeat Factor Sampling for Long-Tailed Object Detection. https://arxiv.org/abs/2305.08069, 2023. 4

[43] Yuhang Zang, Chen Huang, and Chen Change Loy. FASA: Feature Augmentation and Sampling Adaptation for Long-Tailed Instance Segmentation. In *ICCV*, 2021. 7

[44] Yuhang Zang, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Semi-Supervised and Long-Tailed Object Detection with CascadeMatch. *IJCV*, 2023. 3, 7

[45] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. MosaicOS: A Simple and Effective Use of Object-Centric Images for Long-Tailed Object Detection. In *ICCV*, 2021. 2, 3, 6

[46] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *ICLR*, 2018. 5

[47] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *ICLR*, 2023. 4

[48] Liang Zhao, Yao Teng, and Limin Wang. Logit Normalization for Long-Tail Object Detection. *IJCV*, 2024. 6

[49] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. In *AAAI*, 2020. 5

[50] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting Twenty-Thousand Classes Using Image-Level Supervision. In *ECCV*, 2022. 2, 3, 6

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*, 2021. 4

[52] George Kingsley Zipf. The Psycho-Biology of Language: An Introduction to Dynamic Philology. *Routledge*, 2013. 3