# MULTI-DATASET PRETRAINING: A UNIFIED MODEL FOR SEMANTIC SEGMENTATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

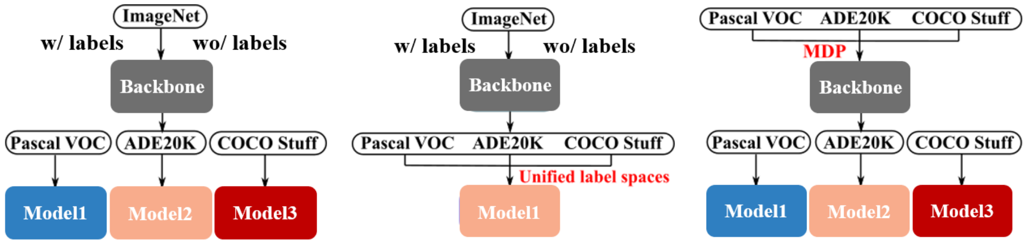
Collecting annotated data for semantic segmentation is time-consuming and hard to scale up. In this paper, we propose a unified framework, termed as **M**ulti-**D**ataset **P**retraining, to efficiently integrate the fragmented annotations of different datasets. The highlight is that the annotations from different datasets can be shared and consistently boost performance for each specific one. Towards this goal, we propose a pixel-to-prototype contrastive learning strategy over multiple datasets regardless of their taxonomy labels. In this way, the pixel level embeddings with the same labels are well clustered, which we find is beneficial for downstream tasks. In order to model the relationship among images and classes from different datasets, we extend the pixel level embeddings via cross-dataset mixing and propose a pixel-to-prototype consistency regularization for better transferability. MDP can be seamlessly extended to semi-supervised setting and utilize the widely available unlabeled data to further boost the feature representation. Experiments conducted on several benchmarks demonstrate its superior performance, and MDP consistently outperforms the pretrained models over ImageNet by a considerable margin.

## 1 INTRODUCTION

As a basic computer vision task, semantic segmentation has experienced remarkable progress over the past decades, mainly benefiting from the growth of the available annotations. However, annotating images at pixel level granularity is time-consuming and difficult to scale up. In order to alleviate the dense annotation requirement, semantic segmentation is usually fine-tuned based on a pretrained model, *e.g.,* training on a large-scale ImageNet classification dataset (Russakovsky et al., 2015). While ImageNet pretraining can *de facto* lead to performance gain, it suffers from the task gap that the pretraining is based on global classification while the downstream task is for local pixel level prediction. In this paper, we arise a critical issue, can we solve the task gap via making use of the available annotations off-the-shelf from diverse segmentation datasets for better performance?

Though promising it is, a major challenge of unifying multi-datasets for training is to tackle the label inconsistency, where taxonomy from different datasets differs, ranging from class definition and class granularity. For example, the classes *'wall-brick'*, *'wall-concrete'* and *'wall-panel'* in COCO-Stuff (Caesar et al., 2018) are simply labeled as *'wall'* in ADE20K (Zhou et al., 2019), and as *'background'* in Pascal VOC (Everingham et al., 2015). This makes integrating different datasets into a common taxonomy time-consuming and error-prone. In this paper, we propose a novel training framework that is able to directly unify different datasets for training regardless of its taxonomy labels. In Fig. 1, we illustrate several typical settings for semantic segmentation. The advantage of MDP is that we are able to conveniently combine multiple datasets for jointly training without any human intervention, and the pretrained model can be used as a backbone for downstream fine-tuning as usual.

Towards this goal, we rely on contrastive loss that is widely used in self-supervised learning (Chen et al., 2020a; He et al., 2020) for pretraining. The core idea is to aggregate semantically similar pixel level embeddings via contrastive clustering. In particular, we adjust the global contrastive loss to a supervised pixel level one and construct a pixel-to-prototype contrastive mapping such that the pixel embeddings with the same labels enjoy better intra-class compactness and inter-class separability. Here, the prototype denotes the embedding of each specific class. Considering that classes from different datasets may share similar embeddings, we extend the pixel-to-prototype mapping in two folds: first, we enrich the pixel level embeddings via cross-dataset mixing to bridge different datasets;



(a) Standard pipeline (b) Multi-dataset training (c) Multi-dataset pretraining

Figure 1: Typical settings for semantic segmentation. a) The standard pipeline, which performs supervised or self-supervised pretraining on ImageNet, and trains segmentation models over specific datasets. b) A simple multi-dataset training method that **manually** unifies label spaces of different datasets. c) Our proposed MDP, which automatically integrates multiple datasets for pretraining.

second, we extend the pixel-to-prototype contrastive mapping, which can be treated as hard encoding with the ground truth labels, with consistency regularization, *i.e.*, the pixel-prototype similarity of a pixel should be consistent with its corresponding pixel on another view. We find that such consistency regularization is beneficial for better transferability.

Benefit from the pixel-to-prototype mapping, MDP can be seamlessly extended to unlabeled data to further boost the feature representation for pretraining. In this setting, the pixel level labels of the unlabeled data are simply replaced with pseudo labels obtained with the current model, and followed by the same learning procedure. Note that different from traditional semi-supervised settings that require the unlabeled data come from the same domain with the labeled one, MDP can take advantage of the available unlabeled data from diverse domains, which is widely applicable and importantly, *MDP transfers most of the computation budget to the pretraining stage, while is able to avoid cumbersome data mining for each downstream task and is beneficial for fast deployment*. Experiments conducted on several widely used semantic segmentation benchmarks demonstrate the effectiveness of our proposed multi-dataset pretraining mechanism.
In a nutshell, this paper makes the following contributions:

- We propose a novel pretraining framework to make use of the available annotations from diverse datasets. As far as we know, we are the first to unify multiple semantically labeled datasets for pretraining, while not be bothered by the chaotic labels from diverse datasets.
- We propose a pixel-to-prototype contrastive learning strategy to effectively model intra-class compactness and inter-class separability. To better make use of cross-dataset samples, we enrich the pixel embeddings via cross-dataset mixing and extend the pixel-to-prototype hard mapping with a smoother consistency constraint.
- MDP can be seamlessly extended to unlabeled data to further improve the feature representation, and the computation budget is transferred to the pretraining stage, which is beneficial for fast deployment over different downstream tasks. This is different from conventional semi-supervised settings that require data mining over each specific domain.
- MDP consistently outperforms the pretrained models over ImageNet by a considerable margin. We hope that our findings may shed light on future research to design appropriate pretext tasks for semantic segmentation.

## 2 Related Work

**Contrastive learning.** Contrastive learning-based methods learn representations by contrasting positive pairs against negative pairs in a discriminative fashion. Recent works mainly benefit from instance discrimination (Wu et al., 2018), which regards each image and its augmentations as one separate class and others are negatives (He et al., 2020; Chen et al., 2020a; Dosovitskiy et al., 2015; Chen et al., 2020b; Hjelm et al., 2018; Oord et al., 2018; Tian et al., 2019; Grill et al., 2020). Since using a large number of negatives is crucial for the success of contrastive loss-based representation learning, (Wu et al., 2018) uses a memory bank to store the pre-computed representations from

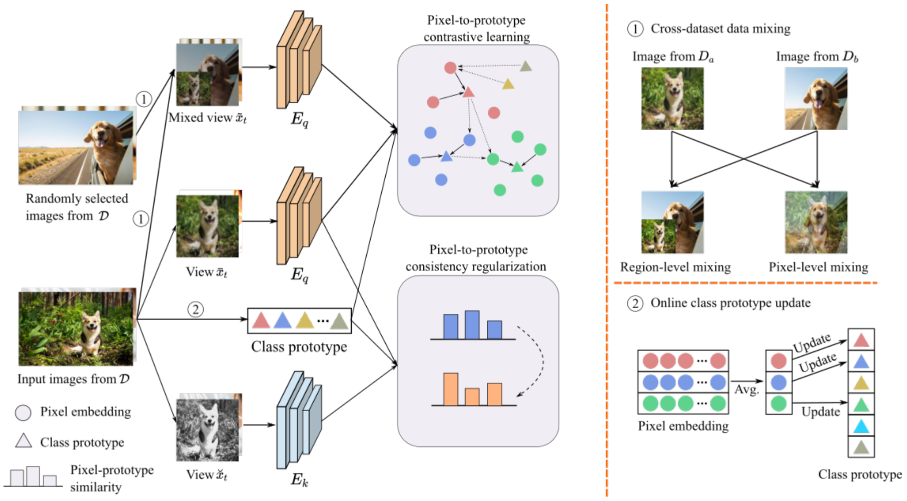

Pixel-to-prototype contrastive learning
①
Mixed view $\tilde{x}_t$
$E_q$
Randomly selected images from $\mathcal{D}$
①
View $\bar{x}_t$
$E_q$
Pixel-to-prototype consistency regularization
②
Class prototype
Input images from $\mathcal{D}$
Pixel embedding
Class prototype
Pixel-prototype similarity
View $\breve{x}_t$
$E_k$
① Cross-dataset data mixing
Image from $D_a$
Image from $D_b$
Region-level mixing
Pixel-level mixing
② Online class prototype update
Avg.
Update
Update
Update
Pixel embedding
Class prototype

Figure 2: The pipeline of MDP. Given a labeled image $x_t$ randomly sampled from the collection of multiple datasets $\mathcal{D}$, we first obtain three different views $\hat{x}_t$, $\breve{x}_t$ and $\bar{x}_t$ (not shown in the figure) through data augmentation as well as a mixed view $\tilde{x}_t$ through cross-dataset mixing. Then we conduct pixel-to-prototype contrastive learning and consistency regularization to model intra-class compactness and inter-class separability, as well as considering inter-class similarity. The right part illustrates two components of MDP, *i.e.,* cross-dataset mixing and online class prototype update.

which positive examples are retrieved given a query. Based on it, (He et al., 2020) uses a momentum update mechanism to maintain a long queue of negative examples for contrastive learning. Some recent works introduce contrastive learning for semantic segmentation (Zhao et al., 2020b; Liu et al., 2021; Alonso et al., 2021; Wang et al., 2021; Gansbeke et al., 2021) by pulling close the pixel level embeddings with the same labels and pushing apart the embedding of pixels with different labels. Different from these works, our method focuses on the scenario of multi-dataset pretraining, and proposes pixel-to-prototype loss to effectively model the intra-class and inter-class relationships.

**Multiple dataset training.** For recognition tasks like object detection and semantic segmentation, training on naively combined datasets yields low accuracy and poor generalization (Lambert et al., 2020) since different datasets have different class definitions and class granularity. Dataset unification, which involves merging different semantic concepts, is important for multi-dataset training. Liang et al. (2018) manually builds a semantic concept hierarchy by combining labels from all four popular datasets and explicitly incorporates the hierarchy into network construction. Lambert et al. (2020) manually unifies the taxonomies of 7 semantic segmentation datasets and uses Amazon Mechanical Turk to resolve inconsistent annotations between datasets. However, these methods need heavily manual effort. Zhao et al. (2020a) and Zhou et al. (2021) trains a universal detector by first training a single partitioned detector on multiple datasets with shared backbone dataset-specific outputs, and loss and then unifies the outputs of the partitioned detector in a common taxonomy. These methods still rely on partitioned learning on their respective datasets. Unlike these works, we do not unify their label spaces but make the pixel level features distinguishable by multi-dataset pretraining.

# 3 Method

In this section, we elaborate on our proposed multi-dataset pretraining strategy. The whole procedure is shown in Fig. 2. The core modules consist of three parts, 1) pixel-to-prototype contrastive loss, 2) cross-dataset learning, and 3) MDP with semi-supervised extension, which would be explained in detail in the following sections.

## 3.1 Pixel-to-pixel Contrastive Learning

We first present a simple baseline that directly extends the contrastive learning to pixel level, guided by the pixel level annotations. In this setting, pixels with the same label are treated as positive pairs, while those with different labels are regarded as negative pairs. Note that since we only constrain the positive samples within the same label, it is conveniently applicable for multiple datasets.

Specifically, given multiple labeled datasets $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_N\}$ along with label spaces $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, ..., \mathcal{Y}_N\}$, we randomly select $n$ samples $\{x_t\}_{t=1}^{n}$ from $\mathcal{D}$ regardless of which domain each sample comes from. Denote $\hat{x}_t$ and $\bar{x}_t$ be two different augmented views of image $x_t$, with ground truth pixel level class label map $\hat{Y} = [\hat{y}_i \in \mathcal{Y}]$ and $\bar{Y} = [\bar{y}_i \in \mathcal{Y}]$, where $i$ denotes pixel in the image. $\hat{x}_t$ and $\bar{x}_t$ are separately sent to feature extractor $E_q$ and $E_k$ to obtain $d$-dimensional per-pixel unit-normalized features $\hat{F} = [\hat{f}_i]$ and $\bar{F} = [\bar{f}_i]$, where $E_k$ is a momentum update version of $E_q$. For pixel $i$ in $\hat{F}$, the pixel $j$ in $\bar{F}$ with the same class label is considered as positive sample of $i$, and the pixel level contrastive loss is computed by:

$$\mathcal{L}_{pixel} = -\frac{1}{\bar{N}_{y_i}} \sum_{j=1}^{N} \mathbb{1}\left[\hat{y}_i = \bar{y}_j\right] \log \left( \frac{\exp\left(\hat{f}_i \cdot \bar{f}_j / \tau\right)}{\sum_{k=1}^{N} \exp\left(\hat{f}_i \cdot \bar{f}_k / \tau\right)} \right), \tag{1}$$

where $N = H \times W$ denotes the total number of pixels in each view, $H$ and $W$ mean the height and width of each view, respectively. $\bar{N}_{y_i}$ denotes the number of pixels with class $y_i$ in $\bar{x}_t$ and $\tau$ is the temperature parameter. The loss is averaged over all pixels on the first view. Similarly, the contrastive loss for pixel $j$ on the second view is also computed and averaged. Since the calculation of pixel level contrastive loss is independent for each image, Eq. 1 enables multi-dataset training and does not need to consider the label mapping. However, such a pixel-to-pixel optimization strategy is sensitive to noisy annotations and more importantly, it does not consider the relationship across datasets, which limits its representation ability. In the following, we extend pixel-to-pixel contrastive loss to pixel-to-prototype one.

### 3.2 Pixel-to-prototype Contrastive Learning

In this section, we adjust the pixel-to-pixel contrastive learning with more robust pixel-to-prototype mapping. The motivation is that the class-level representation is more stable and memory-efficient compared with pixel level embeddings. In particular, we merge the label spaces of the collection of all datasets $\mathcal{D}$ and obtain class set $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup ... \cup \mathcal{Y}_N$. We maintain a prototype for each class $j \in \mathcal{Y}$ in the memory bank. Then we learn the embeddings of all the pixels in each input image by pulling them close to the same class prototype and pushing them apart to different class prototypes.

**Class prototype generation.** A key component for pixel-to-prototype mapping is to maintain the prototype for each class. Considering that the labeling granularity is at pixel level, which is huge and memory-consuming, we propose an efficient prototype maintenance strategy to dynamically update the class embeddings. Specifically, suppose we have a total of $n$ training images and $|\mathcal{Y}|$ semantic classes. Let $P_j$ be the prototype of class $j$, we first calculate the embedding of class $j$ for $i$-th image, represented by $p_{ij}$, by average pooling all the embeddings of pixels labeled as $j$ in the $i$-th image. Then, $P_j$ is obtained by averaging all embeddings among $n$ images that contain class $j$:

$$P_j = \frac{\sum_{i=1}^{n} m_{ij} p_{ij}}{\sum_{i=1}^{n} m_{ij}}, \tag{2}$$

where $m_{ij} \in [0, 1]$ is a binary mask indicating whether the $i$-th image contains pixels with class $j$. In practice, in order to realize the dynamic update of $P_j$, we store the embedding sum of class $j$ and the number of images that contain pixels with class $j$ for each batch $b$. Comparing with pixel level embeddings, the benefits of utilizing class prototype are two-folds:

- **Reducing the storage and GPU memory requirements**. Extending the pixel-to-pixel loss across images needs a memory bank with a size of $H \times W \times N_b \times Dim$, where $Dim$ indicates the dimension of pixel embedding and $N_b$ indicates the number of image embeddings saved in the memory bank. Its storage and GPU calculation consumption will become intolerable under the multi-dataset setting since $H \times W$ and $N_b$ is large. While using the class prototype, we can reduce the storage size to $|\mathcal{Y}| \times B \times Dim$ while maintaining the dynamic update of $P$, where $B = N_b/bs$ is the total number of batches and $bs$ means the batch size, and only $P$ is involved in the GPU calculation.

- **Alleviating class imbalance especially for multiple datasets**. In multiple datasets jointly training, the categories of small-scale datasets will be obscured by the categories of large-scale ones. The pixel level comparison will obviously exacerbate this problem. Our method obtains one prototype for each class $j$, regardless of the number of images containing $j$. This ensures the contribution of small-scale datasets and classes with rare data be not ignored. Please see the appendix for category results on ADE20k, MDP significantly benefits for rare categories.

**Loss function.** In pixel-to-prototype contrastive learning, we first update the prototype in the memory bank using $\bar{F}$ according to Eq. (2). Then, for pixel $i$ with class $j$ in $\hat{x}$, we maximize the agreement between its embedding $\hat{f}_i$ and the prototype of class $j$. Additionally, we require a push-force to avoid collapse in the embedding space. This can be achieved by pushing $\hat{f}_i$ and other class prototypes apart. The pixel-to-prototype contrastive loss for pixel $i$ is computed by:

$$\mathcal{L}_{class} = -\mathbb{1}\left[\hat{y}_i = j\right] \log \left( \frac{\exp\left(\hat{f}_i \cdot P_j / \tau\right)}{\sum_{k=1}^{|Y|} \exp\left(\hat{f}_i \cdot P_k / \tau\right)} \right) \tag{3}$$

The final loss is also computed for pixels from two different views and then averaged. In this way, pixel embeddings in the current image are not only influenced by the same class pixels in other images but also interact with the classes which do not occur in the current image, which greatly improves the embedding quality compared to pixel-to-pixel contrastive learning.

### 3.3 Cross-Dataset Learning

In real applications, classes from different datasets may share similar embeddings. In order to better model the inter-class relationship defined by the provided labels, we propose two cross-dataset interaction operations. First, we enrich the pixel level embeddings via cross-dataset mixing. Second, we extend the pixel-to-prototype contrastive mapping with a more general consistency regularization.

**Cross-image pixel representation.** We adopt two different levels of data mixing methods, *i.e.,* region-level mixing and pixel-level mixing, to interact representation across different datasets. These two mixing methods alleviate the problem of input inconsistency and class inconsistency from different perspectives, and they are complementary to improve the feature representation.

*Region-level mixing.* Given two labeled images $\{x_i,y_i\}$ and $\{x_j, y_j\}$ from $\mathcal{D}$, we conduct cross-dataset cutmix (Yun et al., 2019) operation by:

$$\begin{aligned} \tilde{x} &= M \odot x_i + (1 - M) \odot x_j \\ \tilde{y} &= M \odot y_i + (1 - M) \odot y_j \end{aligned} \tag{4}$$

where $M$ denotes a binary mask indicating where to drop out and fill in from two images, which is obtained by uniform sampling, and $\odot$ is element-wise multiplication. Region-level mixing enables the model to see different regions of different datasets at the same time, and can reduce the domain gap between datasets. At the same time, region-level mixing destroys the regional continuity of the original image, making the model pay more attention to pixel level details and also increase the diversity of features.

*Pixel-level mixing.* Also, given two random samples $\{x_i,y_i\}$ and $\{x_j, y_j\}$, we can conduct cross-dataset pixel-level mixing by simply using mixup (Zhang et al., 2017) operation:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1 - \lambda) x_j \\ \tilde{y} &= \{\lambda, y_i, y_j\} \end{aligned} \tag{5}$$

Where $\lambda \in [0, 1]$ is uniformly sampled. It should be noted that the class label should be an integer, so we do not mix the label but storing the set $\{\lambda, y_i, y_j\}$ in $\tilde{y}$. The final loss $\mathcal{L}(\tilde{x}, \tilde{y}, P)$ for pixel-level mixing can be obtained by considering the contribution of the two groundtruth labels, $y_i$ and $y_j$, using the class prototype $P$ and weight $\lambda$:

$$\mathcal{L}(\tilde{x}, \tilde{y}, P) = \lambda \mathcal{L}(\tilde{x}, y_i, P) + (1 - \lambda) \mathcal{L}(\tilde{x}, y_j, P). \tag{6}$$

Pixel-level mixing can integrate the image information from different datasets in a more fine-grained way. In addition, pixels will contain contents from different classes, which makes up for inconsistencies across datasets at both image and class levels.

**Pixel-to-prototype consistency regularization.** The pixel-to-prototype contrastive learning can be treated as one-hot hard mapping for each pixel since it only pulling pixels close to the prototype corresponding to their ground truth labels. However, since the labels between different datasets may share similar semantics, it is better to model the relationships across categories. Towards this goal, we propose a novel pixel-to-prototype consistency regularization strategy, that regularizes the representation consistency of corresponding pixels at different views. Specifically, given $\hat{x}_t$ as one augmented view of $x_t$, we generate another view of $\hat{x}_t$ with only color transforms and denote it as $\breve{x}_t$. Also, $\hat{F} = [\hat{f}_i]$ and $\breve{F} = [\breve{f}_i]$ are obtained by the feature extractor $E_q$ and $E_k$ but equipped with different projection heads. For pixel $i$ in $\hat{F}$, the consistency regularization constraints its similarity to all the class prototypes be consistent with its corresponding pixel in $\breve{F}$:

$$\mathcal{L}_{con} = -\breve{s}_i \log \hat{s}_i \tag{7}$$

Where $s_i$ is obtained by normalizing the similarity between $f_i$ and $P$ with a softmax function:

$$s_{ij} = \frac{\exp\left(f_i \cdot P_j/\tau\right)}{\sum_{k=1}^{|Y|} \exp\left(f_i \cdot P_k/\tau\right)} \tag{8}$$

The total loss function is a combination of Eq. (3) and Eq. (7), namely:

$$\mathcal{L}_{total} = \mathcal{L}_{class} + \alpha \mathcal{L}_{con}, \tag{9}$$

where $\alpha$ is a balancing factor that controls the contributions of the two terms.

### 3.4 Extending to semi-supervised setting

Considering that obtaining pixel level annotations is time-consuming, we extend MDP to a new semi-supervised setting that supports unlabeled data from diverse domains for pretraining. Given a collection of labeled datasets $\mathcal{D}$ and widely available unlabeled data collection $\mathcal{U}$, it only takes a little effort to extend MDP to this semi-supervised setting. For labeled images sampled from $\mathcal{D}$, we use the same method for feature learning and obtain class prototype $P$. For unlabeled samples, suppose $x_u$ is an image randomly selected from $\mathcal{U}$, and $\hat{x}_u$ is its augmented view, we first send $\hat{x}_u$ to the momentum feature extractor $E_k$ to obtain $\hat{F}_u$. Then we calculate the similarity between pixel $i$ in $\hat{F}_u$ and $P$ and set its pseudo labels $\hat{y}_{u_i}$ to the class with the highest similarity. Since unlabeled data has a wider range of data sources, the class prototype may not cover all pixel classes. So we only consider the pseudo labels with similarity confidence higher than the threshold $T$:

$$\hat{y}_{u_i} = \mathbb{1}[f_i \cdot P_j \geq T] \arg\max_j \left(f_i \cdot P_j\right). \tag{10}$$

Once obtaining pseudo label $\hat{y}_u$, the unlabeled data will follow the same pretraining procedure as labeled ones. It should be noted that $\hat{y}_u$ will be updated with the growth of prototype $P$, so its correctness continues to improve. And other unlabeled pixels will still contribute to the MDP framework, thanks to the consistency regularization strategy.

## 4 Experiments

In this section, we evaluate MDP on several widely used benchmarks, as well as detailed ablation studies to reveal how each module affects the final performance.

### 4.1 Experimental Setups

**Datasets.** Our experiments are conducted on four datasets, namely:

- **Cityscapes** (Cordts et al., 2016) has 5,000 finely annotated urban scene images, with 2,975/500/1,524 for train/val/test, respectively. The segmentation performance is reported on 19 challenging categories, such as person, sky, car, and building *etc*.

Table 1: Overall results evaluated on four different datasets: Pascal VOC, ADE20K, COCO-Stuff and Cityscapes, using different number of datasets for pretraining.

| Method | Pretrained Dataset | | | | Epoch | mIoU | | | |
|---|---|---|---|---|---|---|---|---|---|
| | ImageNet | VOC | ADE20K | COCO | | VOC | ADE20K | COCO | Cityscpes |
| Scratch | | | | | - | 44.78 | 28.67 | 25.02 | 54.27 |
| MoCo-v2 | ✓ | | | | 800 | 71.59 | 38.29 | 33.64 | 77.52 |
| Supervised | ✓ | | | | - | 75.63 | 39.36 | 35.25 | 77.60 |
| pixel-to-pixel | | ✓ | ✓ | | 100 | 71.19 | 38.81 | 32.44 | 77.17 |
| MDP | | ✓ | ✓ | | 100 | 73.79 | 40.17 | 34.26 | 77.75 |
| MDP | | ✓ | ✓ | | 200 | 74.65 | 40.83 | 35.04 | 78.59 |
| MDP | | ✓ | ✓ | ✓ | 200 | **78.25** | **42.69** | **38.47** | **80.64** |

- **Pascal VOC 2012** (Everingham et al., 2015) consists of 10,582 training (including the annotations provided by (Hariharan et al., 2011)), 1,449 validation, and 456 test images with pixel level annotations for 20 foreground object classes and one background class.
- **ADE20K** (Zhou et al., 2019) contains around 25K images spanning 150 semantic categories, of which 20K for training, 2K for validation, and another 3K for testing.
- **COCO-Stuff** (Caesar et al., 2018) is a large scale dataset, which includes 164K images from COCO 2017 (Lin et al., 2014). Among them, 118k images are used for training and 5k images for validation. It provides rich annotations for 80 object classes and 91 stuff classes.

**Evaluation.** Unless specified, we choose the training split of Pascal VOC and ADE20K for pretraining since they are comparable at scale, and evaluate the performance on all datasets to validate its generalization ability. For the semi-supervised setting, we choose COCO-Stuff as unlabeled data. Following the standard, we use mean Intersection-over-Union (mIoU) for performance evaluation.

**Implementation details.** We choose DeepLab-v3+ (Chen et al., 2018) based on a standard ResNet-50 as backbone (He et al., 2016). We employ a momentum key encoder and adds two 3-layer projection heads after the ASPP layer of the DeepLab-v3+ model, which finally results in two 256-d embedding vectors for each pixel for contrastive learning and consistency regularization, respectively. The model is pretrained using an SGD optimizer with momentum $0.9$ and weight decay $4e-5$. The batch size and the initial learning rate are set to 128 and $0.8$ respectively, over 8 NVIDIA Tesla V100 GPUs. The learning rate is decayed to 0 by cosine scheduler (Loshchilov & Hutter., 2016). The input size is set to $224 \times 224$ for efficiency. We use the same set of augmentations as in MoCo-v2 (Chen et al., 2020b). The temperature parameter $\tau$ of contrastive loss is set to $0.07$, and the size of the memory bank is set to about half of the dataset size. For consistency regularization, inspired by Caron et al. (2021), we set $\tau_q$ to 0.07 and $\tau_k$ to 0.04 respectively for the query encoder and the momentum key encoder to avoid collapse. The balancing factor for the total loss is empirically set to $0.2$. For the semi-supervised extension, the confidence threshold $T$ is set to $0.8$.

During the fine-tuning stage, we follow the basic configuration of MMSegmentation[1] except using a standard ResNet-50 backbone and removing the auxiliary head. For Pascal VOC, we fine-tune the pretrained model for $40k$ iterations using $513 \times 513$ input size, while for ADE20K and COCO-Stuff, the iterations are set to $80k$ with $512 \times 512$ input size, and for Cityscapes, the iterations are set as $40k$ with $512 \times 1024$ input size. *Note that our implementation aims to show the effectiveness of multi-dataset pretraining and use some basic settings and structure for evaluation.* While there are several tricks that can definitively further improve the performance, such as larger resolution for pretraining and more advanced structure with auxiliary head, this is out of the scope of this paper.

### 4.2 Results

In this section, we report the overall results when fine-tuning on four datasets. As shown in Table 1, for completeness, we also list the results of using other pretrained models, which includes: a) training

[1] `https://github.com/open-mmlab/mmsegmentation`

Table 2: Ablation studies on hyper-parameters.

| Size $N_b$ | Temperatue $\tau$ | mIoU |
|---|---|---|
| 10240 | 0.07 | 71.98 |
| 2560 | 0.07 | 71.30 |
| 30720 | 0.07 | 70.96 |
| 10240 | 0.3 | 69.87 |
| 10240 | 0.5 | 68.23 |

Table 3: Comparisons with more baselines.

| Method | mIoU |
|---|---|
| pixel-to-pixel | 71.17 |
| Cross Entropy (single-head) | 70.75 |
| Cross Entropy (multi-head) | 71.09 |
| pixel-to-region (Wang et al., 2021) | 71.44 |
| pixel-to-prototype (ours) | 71.98 |

Table 4: Comparisons of different cross-dataset mixing strategies.

| Augmentation | | mIoU |
|---|---|---|
| Region | Pixel | |
| | | 71.98 |
| ✓ | | 72.66 |
| | ✓ | 73.00 |
| ✓ | ✓ | **73.32** |

Table 5: The effects of consistent learning.

| Fully Sup. | | No Sup. | $\mathcal{L}_{con}$ | mIoU |
|---|---|---|---|---|
| VOC | ADE20K | COCO | | |
| ✓ | ✓ | | | 73.32 |
| ✓ | ✓ | | ✓ | 73.79 |
| ✓ | ✓ | ✓ | | 74.74 |
| ✓ | ✓ | ✓ | ✓ | 76.57 |

from scratch with random initialization; b) self-supervised model based on MoCo-v2 using ImageNet as pretraining data; c) fully supervised model on ImageNet. d) the baseline in Sect. 3.1 that simply conducts pixel-to-pixel contrastive learning over multi-dataset. From this table we find that:

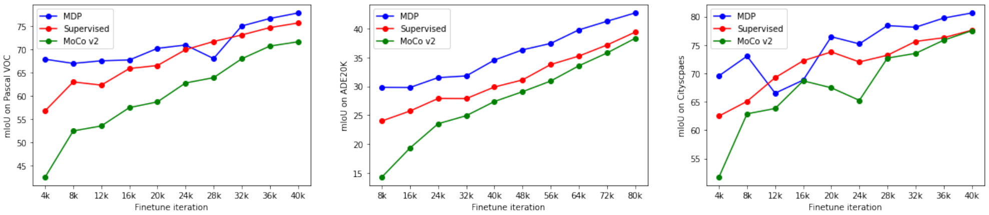
**Comparing with other pretrained models.** All pretraining models surpass training from scratch by a large margin, and MDP beats both supervised and self-supervised ImageNet pretrained models. Specifically, when using only Pascal VOC and ADE20K (around 30K images) for pretraining, we achieve 74.65% and 40.83% accuracies on Pascal VOC and ADE20K, respectively. The performance can be further improved by adding a large scale COCO-Stuff dataset, which surpasses the fully supervised model by a noticeable margin, that is 2.62% performance gain ($75.63\% \rightarrow 78.25\%$) on Pascal VOC, 3.33% gain ($39.36\% \rightarrow 42.69\%$) on ADE20K. It should be noted that compared to ImageNet pretraining, MDP only uses less than 10% samples for pretraining, which is more efficient.

**Transfer ability on unseen datasets.** We also evaluate the performance on Cityscapes and COCO-Stuff, using the model that does not see any images during the pretraining stage. When only using Pascal VOC and AED20K for pretraining, we achieve 35.04% accuracy on COCO stuff, which is comparable to ImageNet pretraining (35.25%) while using only around 30K training images. We also achieve 78.59% accuracy on Cityscapes, which is already 1% better than the fully supervised model on ImageNet, and the performance on Cityscapes can be further boosted to 80.64% when adding COCO-Stuff for pretraining. The results indicate that MDP also enjoys better transferability.

### 4.3 Ablation Study and Discussion

In this section, we conduct extensive ablation studies as well as some detailed analysis to better understand MDP. All models are pretrained over Pascal VOC and ADE20K for 100 epochs and evaluated on Pascal VOC for efficiency.

**Hyperparameter analysis.** Table 2 studies the influence of temperature $\tau$ and memory bank size $N_b$. Note that the actual size of the class prototype is $N_b/bs$. We conclude that too large or too small memory bank size will cause performance degradation. This is because too old or too few features stored in the memory bank fail to construct a representative and robust class prototype. We also find that the temperature $\tau = 0.07$ brings better performance under supervised pixel level contrastive learning setting, which is different from self-supervised learning where $\tau$ is usually larger ($\tau = 0.2$ in MoCo-v2). The reason is that $\tau$ controls the strength for contrastive learning, with smaller $\tau$ indicating stronger penalizing for compactness and separability. It makes sense that pixel level contrastive learning gets better results for small $\tau$ since it is guided by ground truth labels.



(a) (b) (c)

Figure 3: Comparisons of MDP, MoCo-V2, and supervised ImageNet pretraining on a) Pascal VOC, b) ADE20K and c) Cityscapes at different fine-tuning iterations.

**Comparing with more baselines.** We now compare pixel-to-prototype loss with some other baselines including 1) cross-entropy loss on a concatenated label space $\mathcal{Y}$ (single-head), 2) cross-entropy loss with different segmentation heads for different datasets (multi-head) and 3) pixel-to-region loss proposed in Wang et al. (2021). The results are shown in Table 3. The single-head and multi-head cross-entropy loss achieve lower performance (70.75% and 71.09%), for they fail to explicitly explore the structural information of pixels and cannot effectively utilize the class relationship among datasets. Pixel-to-region loss is a cross-image extension of pixel-to-pixel loss but also suffers from resource constraints, which will no longer be applicable as the pre-training scale increases. We are surprised to find that even in the small-scale ablation setting, the performance of pixel-to-region loss (71.44%) is still weaker than our method (71.98%). We think this is because our method can better deal with long-tail problems caused by multiple datasets and the appendix includes detailed category results.

**Effects of cross-dataset learning and semi-supervised setting.** Table 4 inspects the influence of cross-dataset mixing. It can be seen that both mixing methods bring performance improvements (0.68% gain for region-level mixing and 1.02% gain for pixel-level mixing), and combining them brings larger improvement (1.34%). Table 5 reveals the effectiveness of pixel-to-prototype consistency regularization as well as semi-supervised setting. Intuitively, adding COCO-stuff as unlabeled data brings performance gain (73.32% to 74.74%). While consistency regularization module consistently improves the results, which brings 0.47% (73.32% to 73.79%) gain under fully supervised setting, and 1.83% (74.74% to 76.57%) gain under the semi-supervised setting. The performance gain is much larger for semi-supervised learning even with a higher baseline. We think this is because both pretraining and fine-tuning make use of Pascal VOC data. In this case, it may be more effective to directly separate the label space of Pascal VOC from the label spaces of other datasets, while consistency regularization benefits more when the labels are not confident.

**Does MDP obtain more discriminative features?** Fig. 3 compares the results of MDP and ImageNet pretraining at different fine-tuning iterations over three datasets. It can be seen that MDP is substantially higher than ImageNet pretrained model. It should be noted that at the beginning of fine-tuning, MDP has achieved far better performance, *even for Cityscapes that the model does not see during the pretraining stage*, which proves that MDP can obtain more discriminative features due to pixel level learning. MoCo-v2 suffers the worst initial performance, which also indicates that the instance level contrast learning cannot handle pixel level semantic segmentation tasks well. To further validate the discriminative power learned by MDP, we also evaluate the performance when the backbone is fixed during fine-tune, and the results are shown in Table 6 in the appendix.

## 5 Conclusion

This paper proposed a multi-dataset pretraining framework for semantic segmentation, which can efficiently make use of the off-the-shelf annotations to construct a better and more general pretrained model. The main contributions are three folds. First, we propose a pixel-to-class prototype contrastive loss to effectively model intra-class compactness and inter-class separability of multiple datasets regardless of their taxonomy labels. Second, we extend the pixel level embeddings via cross-dataset mixing and pixel-to-prototype consistency regularization for better transferability. Third, we extend MDP to semi-supervised setting, which is able to effectively make use of the vast unlabeled data for better feature representation. Our method consistently outperforms the pretrained models over ImageNet on several widely used benchmarks, while using much fewer samples for pretraining.

## REFERENCES

Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C. Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. *arXiv preprint arXiv:2104.13415*, 2021.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 833–851, 2018.

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

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

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1734–1747, 2015.

Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. *arXiv preprint arXiv:2102.06191*, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 991–998, 2011.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2876–2885, 2020.

Xiaodan Liang, Eric Xing, and Hongfei Zhou. Dynamic-structured semantic propagation network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 752–761, 2018.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 1209–1218, 2014.

Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J. Davison. Bootstrapping semantic segmentation with regional contrast. *arXiv preprint arXiv:2104.04465*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115(3):211–252, 2015.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. *arXiv preprint arXiv:2101.11939*, 2021.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, pp. 3733–3742, 2018.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pp. 6023–6032, 2019.

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

Xiangyun Zhao, Samuel Schulter, Gaurav Sharma, Yi-Hsuan Tsai, Manmohan Chandraker, and Ying Wu. Object detection with a unified label space from multiple datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 178–193, 2020a.

Xiangyun Zhao, Raviteja Vemulapalli, Philip Andrew Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label-efficient semantic segmentation. *arXiv preprint arXiv:2012.06985*, 2020b.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision (IJCV)*, 127:302–321, 2019.

Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. *arXiv preprint arXiv:2102.13086*, 2021.

# A Appendix

Table 6: Results over Pascal VOC with fixed backbone.

| Method | Epoch | mIoU |
|---|---|---|
| MoCo-v2 | 800 | 63.51 |
| Supervised | - | 61.97 |
| pixel-to-pixel | 100 | 57.31 |
| MDP | 100 | 63.74 |
| MDP | 200 | 65.26 |
| MDP (with COCO) | 200 | 74.04 |

Table 7: Results of pixel-to region loss and our pixel-to prototype loss on head classes and tail classes of ADE20k.

| Method | Head classes (top 5%) | | | | | | | mIoU |
|---|---|---|---|---|---|---|---|---|
| | wall | building | sky | floor | tree | ceiling | road | |
| pixel-to-region | 73.21 | 79.48 | 93.29 | 76.77 | 70.96 | 79.83 | 80.16 | 39.22 |
| pixel-to-prototype | 73.12 | 79.93 | 93.62 | 77.11 | 71.58 | 80.75 | 79.74 | 39.60 |
| Method | Tail classes (last 5%) | | | | | | | mIoU |
| | monitor | bulletin board | shower | radiator | glass | clock | flag | |
| pixel-to-region | 37.69 | 30.64 | 0.0 | 29.19 | 1.34 | 15.23 | 1.9 | 39.22 |
| pixel-to-prototype | 45.25 | 35.31 | 0.0 | 38.45 | 1.06 | 21.35 | 12.01 | 39.60 |

## A.1 Evaluation with fixed backbone

We fix the backbone and fine-tune only the segmentation head to verify the discriminability of the features obtained by MDP. The results are shown in Table 6. MDP achieves 6.43% performance gain comparing with the pixel-to-pixel baseline (57.31% to 63.74%), and also outperforms supervised or self-supervised ImageNet pretraining (61.97% and 63.51%). The performance can be further improved by a large margin by adding the COCO-Stuff dataset and prolong the pre-training epochs. In this case, our model achieves 74.04% mIoU, which is comparable to the supervised ImageNet pretraining that fine-tuning all layers (75.63%, shown in Table 1).

## A.2 Category results

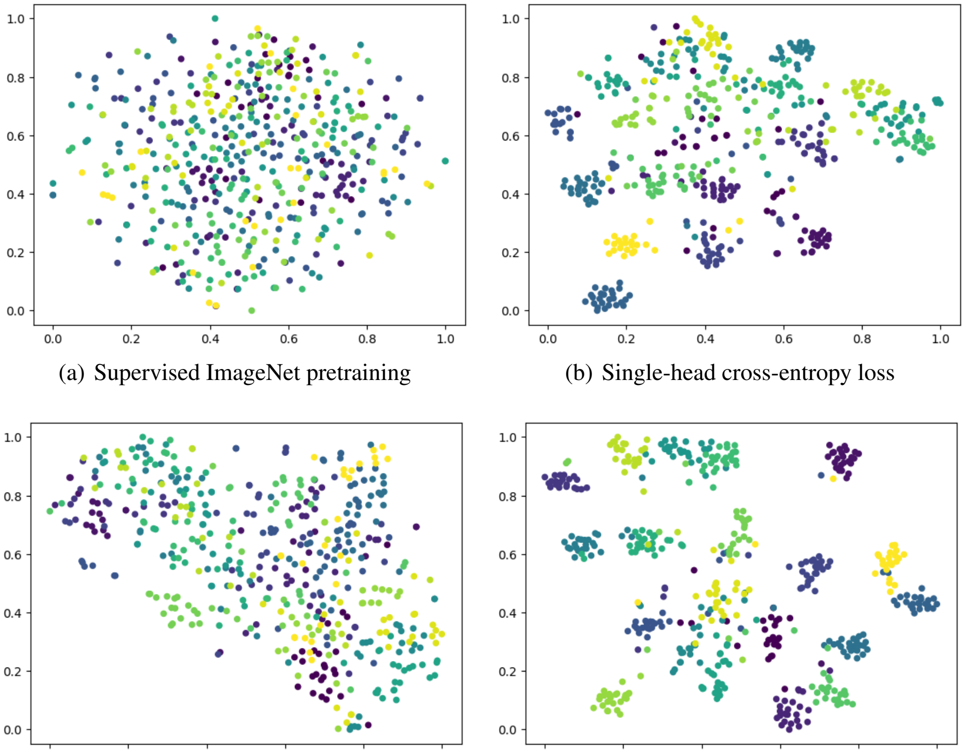
We further compare the performance of pixel-to-region loss (Wang et al., 2021) and our pixel-to-prototype loss on ADE20K, which has more categories and more serious long-tail problems, to verify the effectiveness of our method in dealing with rare categories. In Table 7, we respectively show the overall performance, the head classes performance, and the tail classes performance of the two methods. Results confirm that the performance improvement of our method mainly comes from tail classes, with an average of 5.36% gain compared to pixel-to-region loss.

## A.3 2-stage pretraining strategy

We additionally validate a 2-stage pretraining strategy, which conducts MDP on the basis of ImageNet pretraining, and the results are shown in Table 8. We conclude that MDP can be easily combined with ImageNet pretrained weights to further improve the results, especially for datasets such as ADE20K that have a larger domain gap with ImageNet. Table 8 shows that 200 epochs 2-stage pretraining outperforms ImageNet pretraining by a large margin, from 39.36% to 41.93%, when using Pascal VOC and ADE20K for pretraining. For Pascal VOC, the 2-stage pre-training also helps MDP (76.24%) surpass the performance of supervised ImageNet pretraining (75.63%).

Table 8: Results of 2-stage pretraining based on ImageNet pretrained weights. * means conducting MDP on the basis of ImageNet pretraining.

| Method | Pretrained Dataset | Epoch | mIoU | |
|---|---|---|---|---|
| | | | VOC | ADE20K |
| Supervised | ImageNet | - | 75.63 | 39.36 |
| MDP | VOC and ADE20K | 100 | 73.79 | 40.17 |
| MDP | VOC and ADE20K | 200 | 74.65 | 40.83 |
| MDP | VOC, ADE20K and COCO | 200 | 78.25 | 42.69 |
| MDP* | VOC and ADE20K | 100 | 75.70 | 41.57 |
| MDP* | VOC and ADE20K | 200 | 76.24 | 41.93 |



(a) Supervised ImageNet pretraining

(b) Single-head cross-entropy loss

(c) Pixel-to-pixel loss

(d) Pixel-to-prototype loss

Figure 4: *t-sne* visualization of the pixel level feature embeddings of the last layer. The model is pretrained by a) supervised ImageNet pretraining, b) single-head cross-entropy loss, c) pixel-to-pixel loss and d) our pixel-to-prototype loss, respectively.

## A.4 Transfer ability on other tasks

MDP is a general framework and can be easily extended to other tasks such as classification and detection, as long as designing a prototype for each class embedding with image-level or bounding box level labels, and the domain gap can be relieved via cross-dataset mixing and consistency regularization equipped in MDP. In addition to the MDP framework, our pre-trained model is also transferable. Notably, We test the transferability of our model on COCO detection and instance segmentation using Mask R-CNN. The results shown in Table 9 are still competitive. Although our pre-trained model does not completely match the downstream model, our method still surpasses the supervised and self-supervised ImageNet pretraining methods that make use of image-level features for model pretraining, which indicates that the task gap matters when considering pretraining strategy.

Table 9: Transferability of MDP using segmentation dataset for pretraining, and testing over COCO detection and instance segmentation.

| Method | Pretrained Dataset | Pretrained Task | Box AP | Mask AP |
|---|---|---|---|---|
| Supervised | ImageNet | Instance level | 38.9 | 35.4 |
| MoCo v2 | ImageNet | Instance level | 39.3 | 35.7 |
| MDP | VOC, ADE20K and COCO | Pixel level | 39.8 | 36.0 |

### A.5 Feature Visualization

We visualize the pixel level feature embeddings of the last layer to understand the aggregating properties of the proposed method. Fig. 4 shows the *t-sne* feature visualization of different pretraining strategies. We can conclude that: 1) Since supervised ImageNet pretraining (Fig. 4(a)) has a task gap towards downstream, its pixel level features are not discriminable. 2) Single-head cross-entropy loss (Fig. 4(b)) can separate some classes, but since it does not model the intrinsic structure of pixels, features of most pixels are mixed together. 3) Pixel-to-pixel baseline (Fig. 4(c)) only distinguishes the pixel level features in a single image, so although the features are separable to some extent, they do not have clustering characteristics. 4) Features obtained by our pixel-to-prototype loss (Fig. 4(d)) enjoy better intra-class compactness and inter-class separability.