Swap Path Network for Robust Person Search Pre-training

Lucas Jaffe[1,2]
[1]Lawrence Livermore National Laboratory
jaffe5@llnl.gov

Avideh Zakhor[2]
[2]University of California, Berkeley
avz@berkeley.edu

Abstract

In person search, we detect and rank matches to a query person image within a set of gallery scenes. Most person search models make use of a feature extraction backbone, followed by separate heads for detection and re-identification. While pre-training methods for vision backbones are well-established, pre-training additional modules for the person search task has not been previously examined. In this work, we present the first framework for end-to-end person search pre-training. Our framework splits person search into object-centric and query-centric methodologies, and we show that the query-centric framing is robust to label noise, and trainable using only weakly-labeled person bounding boxes. Further, we provide a novel model dubbed Swap Path Net (SPNet) which implements both query-centric and object-centric training objectives, and can swap between the two while using the same weights. Using SPNet, we show that query-centric pre-training, followed by object-centric fine-tuning, achieves state-of-the-art results on the standard PRW and CUHK-SYSU person search benchmarks, with 96.4% mAP on CUHK-SYSU and 61.2% mAP on PRW. In addition, we show that our method is more effective, efficient, and robust for person search pre-training than recent backbone-only pre-training alternatives.

1. Introduction

Person search is the combined formulation of two subproblems: *detection* of all person bounding boxes in a set of gallery scenes, and *re-identification* (*re-id*) of all detected boxes with respect to a query person box. Recent person search models are mainly *end-to-end*, and have a feature extraction backbone, followed by separate heads for detection and re-identification. It is typical to initialize this feature backbone using weights pre-trained for classification on the *ImageNet-1k* dataset [38]. Only recently have approaches emerged considering pre-training backbones for person search on annotated person data [8, 24], even then only considering cropped bounding boxes from the *LUPerson* [13] or *Market* [51] datasets.

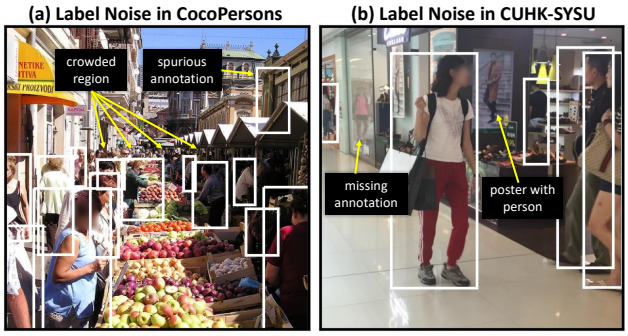

Figure 1. Examples of label noise and annotation challenges in real annotated scenes, with white boxes used for annotations.

For the detection portion of person search, annotations and predictions typically take the form of rectangular bounding boxes. In datasets annotated for person detection, label noise is unavoidable, shown in Fig. 1. For example, how small does a person have to be before they become part of a crowd, shown in Fig. 1a? How does a model handle the presence of missing annotations or images of people in the scene, shown in Fig. 1b? What if a person is behind a window in a building, visible in a monitor screen, or reflected in a mirror? If a detector is used to automatically label people in a scene, how does a model handle spurious extra annotations? These problems are compounded by pre-training, because annotation biases in the pre-training dataset may not reflect those in the target fine-tuning dataset.

This leaves three key issues unresolved in the pursuit of effective pre-training for person search: 1) current backbone pre-training is done using a pretext task unrelated to the person search task, 2) all parameters beyond the backbone are randomly initialized, and 3) current pre-training approaches do not consider robustness to label noise and annotation biases present in person data. Therefore, a model is needed which can simultaneously pre-train for detection and re-id, while achieving domain transfer from a pre-training dataset of heterogeneous imagery that contains label noise and bias.

In this paper, we develop a novel model which addresses these challenges by pre-training using the *query-centric de-*

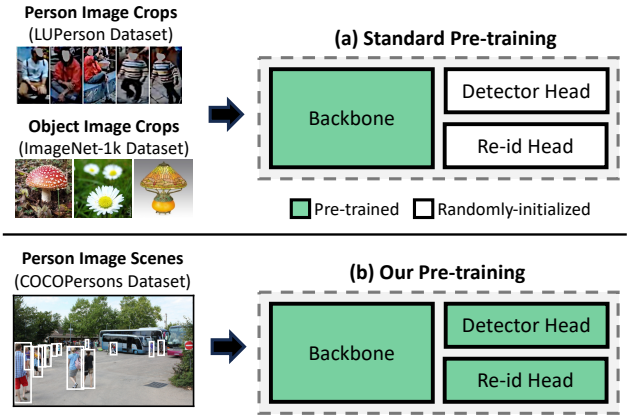

Figure 2. The standard person search model pre-training approach (shown top) pre-trains only backbone weights using either ImageNet-1k classification or person image crops from *e.g.*, LUPerson. Our approach (shown bottom) initializes all model weights using full person scenes with multiple annotated persons. (LUPerson images from paper [13])

tection pretext task. In query-centric detection, we detect matches to a query object in a gallery scene, vs. standard *object-centric detection*, where we detect a pre-determined set of objects in a scene. The model is named **S**wap **P**ath **Net** (**SPNet**) because it can swap between query-centric and object-centric pathways, while using the same weights. The two pathways are needed because the query-centric mode is robust to label noise and learns more generalizable features, making it ideal for pre-training, while the object-centric mode performs better at person search and is much more efficient during inference, making it preferable for fine-tuning on person search. In addition, SPNet is capable of pre-training using *weakly-labeled* person bounding boxes, *i.e.* identity correspondence between scenes is not known. We visualize our pre-training approach vs. typical backbone-only pre-training in Fig. 2.

We show that when SPNet is pre-trained in the query-centric mode, and fine-tuned in the object-centric mode, it achieves state-of-the-art performance on the benchmark *CUHK-SYSU* [46] and *Person Re-identification in the Wild* (*PRW*) [52] person search datasets. We demonstrate that weakly-supervised pre-training on the *COCOPersons* dataset [29,39] using our method is more effective, efficient, and robust than recent backbone-only pre-training alternatives for unsupervised person re-id [8, 15, 53].

Our contributions are as follows:

- The *Swap Path Network*: an efficient end-to-end model of person search which can operate in query-centric or object-centric modes.
- A query-centric pre-training algorithm unique to the Swap Path Network that results in SOTA performance and is robust to label noise.

We support these claims with extensive experiments demonstrating the efficiency of the SPNet model and the efficacy of the pre-training approach. Further, we ensure reproducibility by providing the code and installation instructions required to repeat all experiments, which are included in the corresponding GitHub repository[1].

2. Related Work

2.1. Person Search

Weakly-Supervised. In the context of person search, weak supervision (WS) refers to training on person bounding boxes without identity-level labels. Several methods [16, 17, 24, 42, 44, 47] have emerged in recent years to tackle this problem by using weakly-supervised objectives directly on the target dataset. These methods focus on clustering visual features and applying contextual information to determine pseudo-labels used for a contrastive re-id loss. In contrast, our method is orthogonal to the problem of determining pseudo-labels, focusing on the query-centric vs. object-centric training distinction. In addition, we apply weak supervision only during pre-training on a source dataset, then perform fully-supervised fine-tuning on the target dataset. We note that other methods assume the underlying data has multiple images per identity, and exploit this to form pseudo-labels based on common features, while we do not. Therefore, other methods may perform better for the WS scenario on PRW, CUHK-SYSU, but are not suitable for pre-training on large unlabeled person datasets where there are few or only one image per person like COCOPersons.

Query-Centric vs. Object-Centric. Most person search models implement an *object-centric* (OC) approach [3, 12, 25, 27, 48, 50], in which we detect persons independent of any query, then afterwards compare queries to detected persons. By contrast, *query-centric* (QC) person search models [11, 34, 35, 43] use query person images during the detection process, producing proposals tailored to each query, but significantly increasing time complexity of inference computation.

Despite the greater time complexity of query-centric person search, multiple successful approaches have emerged. QEEPS [34] and QGN [35] use query information both during detection and computation of re-id embeddings by combining query features with backbone features using squeeze-and-excitation layers [21]. IGPN [11] and TCTS [43] are two-step models which extract person embeddings from a separate network, and combine these embeddings with features from the detector network. In our approach, we consider query-centric (QC) learning as a pre-training objective, while fine-tuning in the object-centric (OC) mode, though our framework supports query-centric evaluation as well. This allows us to learn more robust features in the QC mode, while retaining efficient evaluation in

[1]Project repository: https://github.com/LLNL/spnet

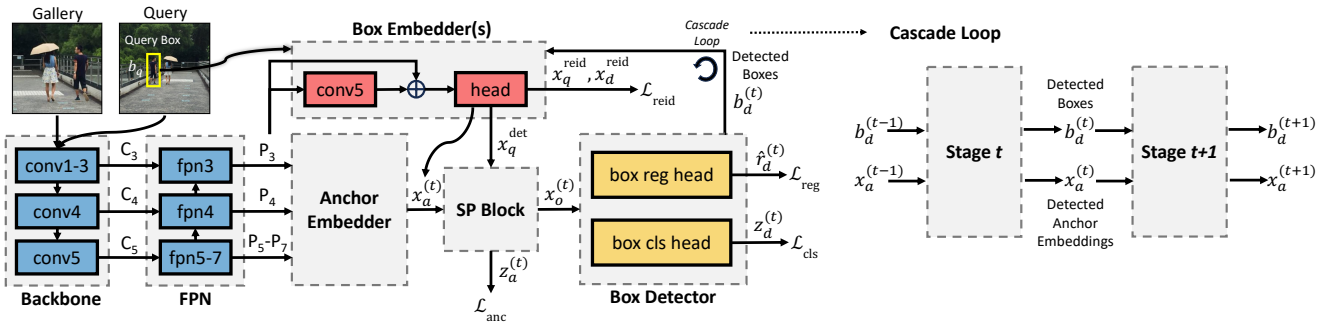

Figure 3. Full SPNet architecture is shown on the left, with details about the Cascade Loop shown on the right. The subscripts q, d, a stand for "query", "detection", and "anchor" respectively. The superscript "reid" means the embedding or loss is used for re-id, and the superscript "det" means the embedding is used for detection.

the OC mode.

Pre-training. Person search models typically use backbone weights initialized from standard ImageNet-1k classifier training, while other model weights are initialized randomly. To our knowledge, the only other work to date to explore another backbone initialization strategy is SOLIDER [8]. The SOLIDER framework trains on unlabeled person crops from the LUPerson dataset [13] by optimizing backbone features using cluster pseudo-labels based on tokens learned with DINO [5]. Unlike our proposed method, SOLIDER initializes only the backbone, and not the rest of the model. In addition, SOLIDER trains only on person crops, while we train on full person scenes, containing potentially many persons. While SOLIDER is the only prior pre-training method which measures fine-tuning for the person search task, there is a rich body of work on similar unsupervised person re-id methods [9,14,15,33,49,53], which also train only on cropped person images.

2.2. Unsupervised Detector Pre-training

In UP-DETR [10], a DETR detector model [4] is pre-trained with the *random query patch detection* pretext task, and fine-tuned for object detection. This model is the closest current model to ours in function, as it supports query-centric detection pre-training with random patches, and object-centric fine-tuning without layer modifications. By comparison, our model implements search instead of only detection, and optimizes much faster due to the explicit spatial inductive bias of anchor-based models vs. DETR. In DETReg [1], a DETR model is trained with pseudo-label boxes generated by selective search [41]. While this approach has the potential to transfer well if the selective search object distribution resembles that of the downstream task, this is unlikely in practice, and the model will have to "unlearn" bad pseudo-boxes during fine-tuning, an issue which is avoided with our query-centric approach.

3. Methods

3.1. SPNet Description

A diagram detailing components of SPNet and the Cascade Loop subcomponent is shown in Fig. 3. SPNet takes Gallery images as input, and outputs detected boxes b_d with corresponding class logits z_d and re-id embeddings x_d^{reid}. SPNet also takes a Query scene as input with a corresponding ground truth person box b_q, and outputs the re-id embedding for that box x_q^{reid}. The query re-id embedding x_q^{reid} is compared to gallery re-id embeddings x_d^{reid} via cosine similarity to produce re-id scores, which are used to rank predictions for retrieval. Image features (C_3-C_5) are learned by the backbone and refined by the FPN (P_3-C_7), and then fed to the Box Embedder, with either query boxes b_q or detected boxes b_d, to learn re-id embeddings x^{reid}, x^{det}. FPN features are also fed to the Anchor Embedder to learn initial anchor embeddings x_a^0.

The Box Embedder can be optionally duplicated to produce separate query embeddings for re-id x_q^{reid} and detection x_q^{det}, or shared *i.e.* $x_q^{\text{reid}} = x_q^{\text{det}}$, shorthanded as x_q. Using a shared Box Embedder has a regularizing effect by pushing x_q to comply with both re-id and detection losses, while duplicating the Box Embedder gives more capacity for each task, which can lead to overfitting. Embeddings x_a, and x_q for QC, then enter the SP Block, where they may pass through either the QC or OC pathway, depicted in Fig. 4.

QC Pathway. In the QC pathway, shown in Fig. 4a, we compute offset embeddings x_o from the *Offset Layer*, by simply subtracting each anchor embedding from the query embedding: $x_o = x_q - x_a$. Offset embeddings x_o are passed to a *Logits Layer*, explained in Sec. 3.3, to produce classifier predictions z_a for each anchor box, which predict whether a given anchor matches the query q. These z_a are used similarly to classifier logits in a standard Faster R-CNN Regional Proposal Network (RPN) [36]: they filter a large number of anchors, typically more than $100,000$ down to less than $1,000$, which are then refined to produce

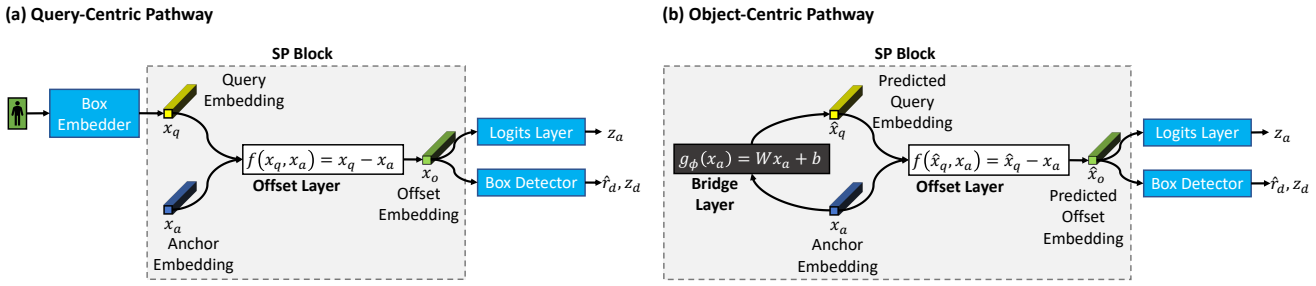

Figure 4. Query-centric (a) and object-centric (b) pathways of the SP Block. Note that the query-centric pathway takes as input a query embedding x_q extracted from a person image, while the object-centric pathway predicts the query embedding $\hat{x}_q$ directly from a matching anchor embedding x_a using the Bridge Layer g_ϕ.

more accurate boxes.

OC Pathway. In the OC pathway, shown in Fig. 4b, we do not have knowledge of any query embedding x_q, so we instead compute a pseudo-query embedding $\hat{x}_q$ using the *Bridge Layer*, by default a single affine layer $g_\phi(x_a) = Wx_a + b$ with learnable parameters $\phi = \{W, b\}$. By mimicking the QC pathway instead of predicting localization offsets and logits directly from x_a, we improve transfer performance between the two pathways. Crucially, the layer g_ϕ is the only difference between the QC and OC models, meaning that all SPNet weights, aside from ϕ, can be trained in one mode, and ported to the other, or vice versa.

Box Prediction. Both pathways output offset embeddings x_o that are passed to a separate Box Detector module, shown in Fig. 3, which performs the box refinement. The `box reg head` is a 4-layer MLP with input x_o which predicts offsets $\hat{r}_d$ between anchor boxes and matching ground truth boxes. The `box cls head` uses the same Logits Layer as the SP Block to produce class logits z_d. Predicted boxes b_d are fed back to the Box Embedder, the same as that used to produce x_q^{reid}, to produce predicted re-id embeddings x_d^{reid}. The x_q^{reid} from queries are compared to x_d^{reid} from gallery images for ranking.

Cascade Loop. We can iteratively refine box accuracy and scores by repeating the same basic prediction structure in a loop, referred to as the *Cascade Loop* in Fig. 3, originating from the Cascade R-CNN detector [2], and also done in the SeqNet [27] and SeqNeXt [23] person search models. In the Cascade Loop, predicted boxes $b_d^{(t)}$ from stage t are fed back into the Box Embedder to produce anchor embeddings $x_a^{(t)}$ for the next step $t+1$. The outputs $z_a^{(t)}$, $x_o^{(t)}$, $z_d^{(t)}$, and $\hat{r}_d^{(t)}$ are also updated during each Cascade Loop step. The detector re-id embeddings x_d^{reid} are computed from the detected boxes coming out of the final Cascade Loop stage. The SP Block, Box Detector, and Box Embedder are all duplicated for each round, meaning that we use modules with the same architecture, but do not share weights of the same module between rounds.

3.2. Pretext Task

The goal of our pre-training pretext task is to initialize nearly all model weights to optimize for transfer to the downstream person search task. In Fig. 5, we show the QC detection pretext task compared to standard OC detection. In OC detection, the goal is to regress each anchor class probability p to 1 if the anchor matches a ground truth box b_g *e.g.*, $b_a^{(1)}$, or 0 if it does not *e.g.*, $b_a^{(2)}$. The overlap threshold for matching boxes is 0.5 IoU. For matching anchors, we also regress box offsets $\hat{r}^{(1)}$ to targets $r^{(1)}$. In QC detection, the regression targets are the same, but predictions for each anchor are computed only *relative* to a query, explained by the QC vs. OC pathways in the previous section. QC detection results in more robust optimization, because it can better handle label noise like the missing annotation in Fig. 5 or examples in Fig. 1. Intuitively, QC detection allows us to learn salient features for transfer to person search without rigidly defining positive and negative detections.

Our pre-training experiments utilize weakly-labeled person boxes. Since identities are not tracked between images for weakly-labeled data, we generate correspondence between images using data augmentation: we know that a box in an image corresponds to the same box in an augmented version of that image, with potentially different coordinates, visualized in Fig. 5. Augmentations consist of scaling, cropping, and horizontal flipping.

For QC detection on weakly-labeled data, each image in a batch is augmented k times, where $k = 2$ for all experiments, with each augmented image taking a turn as both query and gallery. Annotated boxes in the query image are compared against the corresponding box in the gallery image, and assigned to anchors with overlap ≥ 0.5 IoU. Self comparisons are allowed as well, in which the query and gallery are the same image.

For all cases, OC optimization treats annotated boxes as true objects with the same anchor matching criterion as QC, and pushes anchor scores to 0 or 1 as shown in Fig. 5. We also augment batch images k times for the OC case to fairly compare with QC, though it is not necessary.

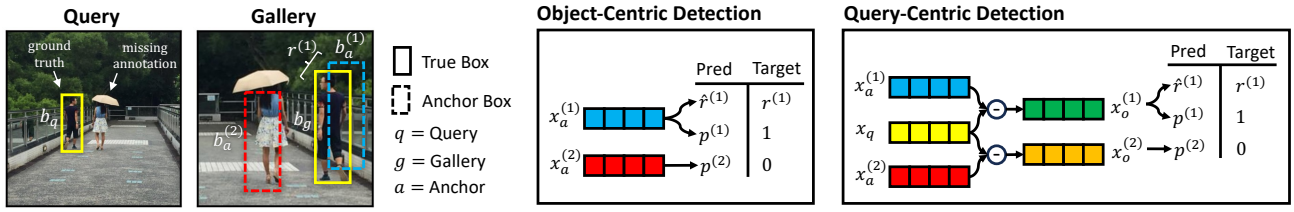

Figure 5. Visual comparison of object-centric (OC) and query-centric (QC) detection tasks between two augmentations of the same base image (Query, Gallery). One person box is annotated (ground truth b_q, b_g), while the other is not (missing annotation). Note that anchor box$_1$ $b_a^{(1)}$ overlaps ground truth box b_g while anchor box$_2$ $b_a^{(2)}$ does not. Anchor embeddings x_a are used to compute box offsets $\hat{r}$ and anchor probabilities p either directly (OC) or relatively using $x_o = x_q - x_a$ (QC). We do not compute box offsets $\hat{r}$ for $b_a^{(2)}$ because it does not match any ground truth.

3.3. Losses

Classifier Losses. For the offset embeddings x_o shown in Fig. 3, it is critical to define a loss function which enforces a consistent relationship between query embeddings x_q and anchor embeddings x_a, while performing the classification regression. The goal is to regress query-anchor matches to 1 and non-matches to 0, shown in Fig. 5. For this task, we model the anchor loss after the Norm-Aware Embedding concept from [6], and the Focal Loss from [28].

Recall the offset embedding is defined as the difference between query and anchor embeddings: $x_o = x_q - x_a$ with $x_o \in \mathbb{R}^d$. Assign $w = \|x_o\|$. In the Norm-Aware Embedding work [6], classification logits are computed from embedding norms using a Batch Norm layer [22].To avoid use of unstable batch statistics, we instead compute a fixed rescaling by modeling $x_o \sim \mathcal{N}(0, I_d)$, which implies $w \sim \text{Chi}(d)$. Then we can standardize w using the mean and standard deviation of the Chi distribution, with class logits given by $z = -(w - \mu_{\text{Chi}})/s_{\text{Chi}}$. Finally, we compute class probabilities with the logistic sigmoid $p = \sigma(z)$.

For anchor logits z_a, we compute a loss averaged across all anchors $\mathcal{L}_{\text{anc}}$, applying the Focal Loss defined in [28], with hyperparameters $\alpha_{\text{FL}} = 0.5, \gamma_{\text{FL}} = 1$. For cascade layer class logits z_d, we compute a loss $\mathcal{L}_{\text{cls}}$ averaged across predicted boxes using the unweighted Cross Entropy Loss.

Additional justification is given in Supplementary Sec. 9.2, where we compare our *FixedNorm* logits formulation vs. the standard BatchNorm, and describe other intuitive benefits of our formulation.

Box Regression Loss. For bounding box offset regression, we use the generalized IoU (GIoU) loss [37], shown as $\mathcal{L}_{\text{reg}}$. This loss has a beneficial interaction with the classifier loss: for matching query and anchor boxes (high IoU/GIoU), the target box regression offsets should be small, corresponding to $\|x_o\|$ small. This allows the model to learn that the magnitude of $\|x_o\|$ correlates directly to the size of predicted box offsets.

Re-id Losses. For our re-id losses, we use two variations of the normalized temperature-scaled cross-entropy loss [7, 40]. We use the terms *positive pair* and *negative pair* to refer to two embeddings with the same label or different labels respectively.

We define the probability for positive pair (x, x^+) under the contrastive objective as

$$p(x, x^+) = \frac{s_\tau(x, x^+)}{s_\tau(x, x^+) + \sum_{x^- \in \mathcal{X}_x^-} s_\tau(x, x^-)} \tag{1}$$

With the full contrastive loss expressed as

$$\mathcal{L}_{\text{reid}} = -\log \sum_{x \in \mathcal{X}} \sum_{x^+ \in \mathcal{X}_x^+} p(x, x^+) \tag{2}$$

using $s_\tau(u, v) \triangleq \exp\left(\text{sim}(u, v)/\tau\right)$ and $\text{sim}(u, v) \triangleq u \cdot v/\|u\|\|v\|, u, v \in \mathbb{R}^d$. $\mathcal{X}$ denotes the set of all x, $\mathcal{X}_x^+$ the set of all positive samples for x, and $\mathcal{X}_x^-$ the set of all negative samples for x. To produce variations of this loss, we vary compositions of the sets $\mathcal{X}_x^+$ and $\mathcal{X}_x^-$.

Fine-Tuning Re-id Loss. During fine-tuning, we use the standard online instance matching (OIM) loss from [46].

Pre-Training Re-id Loss. During pre-training, we use a variation of the momentum contrast loss (MoCo) from [18]. Like in MoCo, we define an encoder network which is updated via gradient descent, and a momentum network, which is updated as a moving average of the encoder. We call embeddings from the encoder network x_e and embeddings from the momentum network x_m. Define $\bar{x}_m$ as the mean of all predicted box embeddings with corresponding box having IoU ≥ 0.7 with a given ground truth box. We store embeddings $\bar{x}_m$ in a queue during training. Then, we form positive pairs $\mathcal{X}_{x_e}^+$ with all embeddings $(x_e, \bar{x}_m^+)$ and negative pairs $\mathcal{X}_{x_e}^-$ with all embeddings $(x_e, \bar{x}_m^-)$.

We are effectively comparing current embeddings to moving average cluster centroids [24], in the limit where the cluster consists of only one person image. In absence of cluster pseudo-labels similar to those used in [24] and other weakly-supervised methods, it is critical to have a representative embedding to compare against aside from the image itself. As in [12], we found it useful to utilize proposals with IoU ≥ 0.7, but did not find it beneficial to weight them by IoU in the re-id loss, as in that work.

Final Loss. The final loss for both pre-training and fine-tuning is simply the unweighted sum of all described losses: $\mathcal{L} = \mathcal{L}_{\text{reid}} + \mathcal{L}_{\text{anc}} + \sum_{t=1}^{T+1} \mathcal{L}_{\text{reg}}^{(t)} + \mathcal{L}_{\text{cls}}^{(t)}$, where T is the number of cascade stages.

4. Experiments

4.1. Datasets and Evaluation

Datasets. We perform weakly-supervised pre-training on the train partition of the COCOPersons dataset [39], which contains 64k scenes with 257k person box annotations. COCOPersons is the subset of MS-COCO2017 [29] images containing at least one person annotation; non-person annotations are ignored. We fine-tune models on the two standard person search benchmark datasets: CUHK-SYSU [46] and PRW [52]. Other metadata about the datasets are presented in Supplementary Sec. 7.

Evaluation. We evaluate all models by measuring fine-tuning performance once on the standard retrieval test scenario for CUHK-SYSU or PRW, described in Supplementary Sec. 7. While we compare QC vs. OC pre-training, fine-tuning is always done in the OC mode. In Supplementary Sec. 9.1, we show that QC fine-tuning is less effective than OC fine-tuning.

To measure performance, we use standard detection performance metrics of recall@0.5 IoU and average precision@0.5 (AP@0.5) IoU, indicating predictions with an overlap of > 0.5 IoU with a ground truth box are considered matches. For person search evaluation, we use the standard metrics of mean average precision (mAP) and top-1 accuracy (top-1).

4.2. Implementation Details

We describe the most important implementation details, with additional information given in Supplementary Sec. 8.

Model Configurations. We perform experiments with two variants of the SPNet model: SPNet-S(mall) and SPNet-L(arge), with differences shown in Tab. 2. Unless otherwise stated, SPNet-S uses a ConvNeXt-T [30] backbone, and SPNet-L uses a ConvNeXt-B backbone.

Optimization. For all experiments, we pre-train and fine-tune for 30 epochs each, using the AdamW optimizer [32], a cosine-annealed schedule, and linear warmup following [31, 32].

Pre-training Optimization. For experiments pre-training SPNet with our method, the backbone is initialized using ImageNet-1k classifier weights unless otherwise stated. For the pre-training re-id loss, we use the MoCo objective described in Sec. 3.3 with a queue length of $65,536$, momentum of 0.9999, and temperature $\tau = 0.1$. We use the Pre-train config from Tab. 2 for learning rate and weight decay settings. We also experiment with four variations of layer freezing and learning rates for the backbone and post-backbone modules, with results shown in Supplementary Sec. 9.1 Fig. 7. We note that QC outperforms OC pre-training for nearly all configurations.

Fine-tuning Optimization. For the fine-tuning re-id loss, we use the OIM objective described in Sec. 3.3 with temperature $1/30$, momentum 0.5, and queue length $5,000$ for CUHK and 500 for PRW, which are the standard settings. We use the Fine-tune config from Tab. 2, with image size 512×512 used for SPNet-S, and image size 1024×1024 used for SPNet-L.

Training and Inference Speed. All models were trained using a single A100 GPU with 82GB VRAM. Pre-training and fine-tuning times are shown in Supplementary Sec. 8, with the QC pre-training taking 30 hours for SPNet-S and 46.5 hours for SPNet-L. While the SOLIDER [8] authors do not give precise training times, they train for 110 epochs on 8 V100 GPUs, and reported the training took several days. A related approach from the LUPerson paper [15] trains for 200 epochs on 8 V100 GPUs. Another crop-only approach callled PASS [53] pre-trains for 100 epochs on 8 A100 GPUs, which takes 60-120 hours depending on the backbone. In addition, LUPerson, PASS, and SOLIDER each pre-train on the much larger LUPerson dataset with 4.18M images vs. our 64k images in the COCOPersons dataset. Given the results comparison in Tab. 3, this shows our method achieves much greater pre-training efficiency for person search. Further, we show in Tab. 1 that SPNet-L achieves comparable metrics to other top models when using a ResNet50 backbone [19], with inference speed more than 5 FPS greater than the next fastest SeqNeXt model at 27.6 vs. 22.3 FPS.

		CUHK-SYSU		PRW		
Model	**Backbone**	mAP	top-1	mAP	top-1	FPS
SeqNet [27]	ResNet50	93.8	94.6	46.7	83.4	12.2
SeqNeXt [23]	ResNet50	93.8	94.3	51.1	85.8	22.3
SPNet-L	ResNet50	93.8	94.5	51.2	86.9	27.6
COAT [50]	ResNet50	94.2	94.7	53.3	87.4	11.1

Table 1. Person search metrics and inference speed for models with ResNet50 backbone with ImageNet-1k classifier initialization. Inference speed in frames per second (FPS) measured on PRW with a single A100 GPU.

4.3. Comparison with State-of-the-art

In Tab. 3 (top), we compare performance of SPNet with recent state-of-the-art person search models. To show cases where more than one pre-training step is used, we indicate the first step with Pre-training (1) and the second with Pre-training (2). If no pre-training is used, indicated by "-", all weights are randomly initialized. We show that SPNet-L, with QC pre-training on COCOPersons, matches or exceeds other models in most metrics for the CUHK-SYSU and PRW datasets *e.g.*, improving mAP by +1.7% on PRW over previous SOTA LEAPS [12].

Config. Name	Cascade Steps	Shared Heads	Re-id Dim
SPNet-S	0	✓	128
SPNet-L	2		2048

	Backbone		Post-Backbone		
Config. Name	LR	WD	LR	WD	Image Size
Pre-train	1e−5	0	1e−4	1e−3	512
Fine-tune	1e−4	5e−4	1e−4	5e−4	512 or 1024

Table 2. SPNet architecture (top) and layer optimization (bottom) hyperparameter configurations. LR = Learning Rate, WD = Weight Decay.

In addition, we show that the QC pre-training itself on SPNet-L adds 0.5% mAP for CUHK-SYSU and 2.3% mAP for PRW. This shows that the benefit from query-centric pre-training extends to the high-end of the model size / performance spectrum, even when performance statistics are nearly saturated, as on CUHK-SYSU and PRW. In contrast, we note that OC pre-training slightly degraded performance on CUHK, demonstrating the need for SPNet and the QC pre-training approach, *i.e*. that performance cannot be trivally improved for any person search model by simply incorporating OC pre-training.

4.4. Comparison with Pre-training Alternatives

In Tab. 3 (bottom), we compare our pre-training approach to the backbone-only SOLIDER pre-training approach [8] vs. random initialization or standard ImageNet-1k classifier initialization. To do a fair experimental comparison, we use models with the Swin-B backbone variant from the SOLIDER codebase for all trials. In addition, we isolate for the effect of the pre-training dataset by re-running SOLIDER on person crops from the COCOPersons dataset. We report results from our SPNet-L model and the original SeqNet model used by SOLIDER.

For SPNet-L, we find that our QC pre-training approach outperforms all alternative initialization strategies, in some cases by wide margins *e.g*., +4.5% on PRW over ImageNet-1k initialization alone. While SOLIDER pre-training on LUPerson is significantly better than random initialization, shown in the first two rows, it is less effective than simple ImageNet-1k classifier pre-training, and far less effective than our QC pre-training approach. In addition, when we apply additional pre-training to SOLIDER on the COCOPersons dataset, there is no improvement to fine-tuning performance, and even degradation in the case of ImageNet-1k classifier initialization. This shows that the COCOPersons crops alone are not nearly as effective for crop-only pre-training as the LUPerson dataset, which is over 16× larger. Further, it shows these crops add no additional information beyond either LUPerson or ImageNet-1k. This means that any benefits provided by our pre-training approach, which utilizes full scenes and not just crops, come from scene context and the pre-training method itself.

Further, we show that our QC approach exceeds our OC approach when both are initialized from ImageNet-1k classifier pre-training, with both the ConvNeXt-B backbone shown in Tab. 3 (top), and the Swin-B backbone shown in Tab. 3 (bottom). This validates our reasoning that QC pre-training learns more robust features than OC pre-training.

Finally, we show that although ImageNet-1k classifier pre-training outperforms LUPerson SOLIDER pre-training for SPNet-L, SOLIDER is significantly better for the SeqNet model, as reported in the SOLIDER paper. Differing performance for SPNet is likely due to a mismatch of the SOLIDER pre-training objective in creating effective features for the feature pyramid network in SPNet, caused by exacerbating scale misalignment of features [48]. In contrast, ImageNet-1k features are more general, and pair better with the feature pyramid network out-of-the-box.

4.5. Pre-training with Noisy Labels

In this section, we show that QC pre-training results in better fine-tuning performance than OC pre-training in the weakly-labeled scenario with noisy labels. We model two noisy labels use cases: 1) ground truth annotations which should be present are missing and 2) there are additional spurious annotations. This has applications both for manual labeling, in which persons may be missed, not all persons are annotated by design, or there is inherent labeling ambiguity as shown in Fig. 1. It is also relevant for auto-labeling, in which a detector is used to label imagery, but may have low recall, creating missing annotations, or high recall but also high false positive rate, creating spurious annotations.

To model missing labels, we create successive partitions of COCOPersons with increments of 40% of annotations removed, with each smaller partition being a subset of all larger partitions. To model spurious labels, we add increments of 40% of the total labels in the original set, drawing from the existing distribution of bounding box shapes in the dataset. The results are shown in Fig. 6, where we compare how QC and OC pre-training are affected by quantity of missing or spurious labels, as measured by fine-tuning performance on CUHK-SYSU. Results are compared to the baseline ImageNet-1k classifier backbone initialization, shown by the black dashed line. For all plots, we show that QC pre-training exceeds OC pre-training in all sample regimes for full fine-tuning. Even when only 20% of samples are used, QC pre-training offers a small benefit to fine-tuning for person search, shown by mAP in Fig. 6a, where OC pre-training actually harms fine-tuning performance by forcing the model to learn that most ground truth person boxes are background. This trend is reflected for detection performance as well, shown in plot Fig. 6b.

4.6. Ablation Studies

Query-Centric vs. Object-Centric. Additional ablations

Person Search Model	Backbone	Pre-training (1) Method / Dataset	Pre-training (2) Method / Dataset	CUHK-SYSU mAP	CUHK-SYSU top-1	PRW mAP	PRW top-1
SOTA Model Comparison							
SeqNet [27]	ResNet50	Classifier / IN1k	-	93.8	94.6	46.7	83.4
COAT [50]	ResNet50	Classifier / IN1k	-	94.2	94.7	53.3	87.4
PSTR [3]	PVTv2-B2 [45]	Classifier / IN1k	-	95.2	96.2	56.5	89.7
SeqNeXt [23]	ConvNeXt-B	Classifier / IN1k	-	96.1	96.5	57.6	89.5
LEAPS [12]	PVTv2-B2	Classifier / IN1k	-	**96.4**	96.9	59.5	89.7
SPNet-L	ConvNeXt-B	Classifier / IN1k	-	95.9	96.6	58.9	89.7
SPNet-L	ConvNeXt-B	Classifier / IN1k	Ours-OC / COCO	95.8	96.4	60.7	90.2
SPNet-L	ConvNeXt-B	Classifier / IN1k	Ours-QC / COCO	**96.4**	**97.0**	**61.2**	**90.9**
Pre-training Comparison							
SPNet-L	Swin-B	-	-	69.8	71.8	20.3	68.7
SPNet-L	Swin-B	SOLIDER / LUP	-	88.0	89.4	38.1	81.3
SPNet-L	Swin-B	SOLIDER / LUP	SOLIDER / COCO	88.0	89.4	38.1	81.3
SPNet-L	Swin-B	Classifier / IN1k	-	94.2	95.0	49.7	85.8
SPNet-L	Swin-B	Classifier / IN1k	SOLIDER / COCO	94.0	94.8	49.7	86.1
SPNet-L	Swin-B	Classifier / IN1k	Ours-OC / COCO	94.4	95.2	52.6	88.7
SPNet-L	Swin-B	SOLIDER / LUP	Ours-QC / COCO	95.1	95.8	53.0	88.3
SPNet-L	Swin-B	Classifier / IN1k	Ours-QC / COCO	**95.8**	**96.3**	54.2	**89.0**
SeqNet	Swin-B	-	-	58.5	59.1	13.8	55.9
SeqNet	Swin-B	Classifier / IN1k	-	88.8	89.6	45.1	82.5
SeqNet	Swin-B	SOLIDER / LUP	-	94.9	95.5	**59.7**	86.8

Table 3. (Top) Comparison of SOTA models. (Bottom) Comparison of performance gained from pre-training SPNet using our method vs. SOLIDER and other initialization strategies. ImageNet-1k is abbreviated as IN1k, LUPerson as LUP, and COCOPersons as COCO. When both Pre-training columns have "-", no pre-training is used and all weights are randomly initialized.

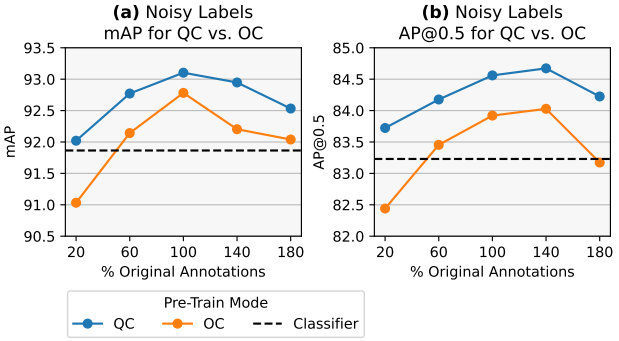

Figure 6. Plots of retrieval (a) and detector (b) stats for fine-tuning on the CUHK-SYSU dataset for QC and OC models pre-trained on COCOPersons with noisy labels vs. the classifier baseline.

comparing QC vs. OC pre-training and fine-tuning are given in Supplementary Sec. 9.1. These results show that QC pre-training outperforms OC pre-training across a range of settings, including variations in the pre-training loss and variations in learning rates and weight freezing. We also break down performance for the detection and re-id sub-tasks, showing that QC pre-training helps both tasks, with differences in magnitude depending on the dataset.

Architecture Ablations. In Supplementary Sec. 9.2, we show that the baseline model with FixedNorm logits achieves greater performance for all metrics over the model with BatchNorm logits. We also consider the impact of model hyperparameters, including embedding dimension, number of cascade stages, shared vs. separate embedding heads, and backbone choice. Notably, we find that all model variants benefit from QC pre-training.

5. Conclusion

We propose and validate Swap Path Net (SPNet), an end-to-end model for person search, which supports query-centric (QC) and object-centric (OC) modes of operation. We show that the model benefits from QC pre-training and OC fine-tuning, and that pre-training can be done using only weakly-labeled person bounding boxes. We show that 1) pre-training provides a significant boost to performance for all model variants, 2) QC pre-training benefits fine-tuning more than OC pre-training and is more robust to label noise, and 3) the model with QC pre-training achieves SOTA fine-tuning performance on CUHK-SYSU and PRW. Finally, we show that our end-to-end person search pre-training method is more effective and efficient than backbone-only pre-training alternatives.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, with release number LLNL-CONF-2001396.

References

[1] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J. Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. DETReg: Unsupervised Pretraining With Region Priors for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14605–14615, 2022. 3

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving Into High Quality Object Detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, June 2018. ISSN: 2575-7075. 4

[3] Jiale Cao, Yanwei Pang, Rao Muhammad Anwer, Hisham Cholakkal, Jin Xie, Mubarak Shah, and Fahad Shahbaz Khan. PSTR: End-to-End One-Step Person Search With Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9458–9467, 2022. 2, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 213–229, Cham, 2020. Springer International Publishing. 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, Oct. 2021. ISSN: 2380-7504. 3

[6] Di Chen, Shanshan Zhang, Jian Yang, and Bernt Schiele. Norm-Aware Embedding for Efficient Person Search. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12612–12621, June 2020. ISSN: 2575-7075. 5

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607. PMLR, Nov. 2020. ISSN: 2640-3498. 5

[8] Weihua Chen, Xianzhe Xu, Jian Jia, Hao Luo, Yaohua Wang, Fan Wang, Rong Jin, and Xiuyu Sun. Beyond Appearance: A Semantic Controllable Self-Supervised Learning Framework for Human-Centric Visual Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15050–15061, 2023. 1, 2, 3, 6, 7

[9] Yoonki Cho, Woo Jae Kim, Seunghoon Hong, and Sung-Eui Yoon. Part-Based Pseudo Label Refinement for Unsupervised Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7308–7318, 2022. 3

[10] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised Pre-training for Object Detection with Transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1601–1610, Nashville, TN, USA, June 2021. IEEE. 3

[11] Wenkai Dong, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Instance Guided Proposal Network for Person Search. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2582–2591, June 2020. ISSN: 2575-7075. 2

[12] Zhiqiang Dong, Jiale Cao, Rao Muhammad Anwer, Jin Xie, Fahad Khan, and Yanwei Pang. LEAPS: End-to-End One-Step Person Search With Learnable Proposals, Mar. 2023. arXiv:2303.11859 [cs]. 2, 5, 6, 8

[13] Dengpan Fu, Dongdong Chen, Jianmin Bao, Hao Yang, Lu Yuan, Lei Zhang, Houqiang Li, and Dong Chen. Unsupervised Pre-Training for Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14750–14759, 2021. 1, 2, 3

[14] Dengpan Fu, Dongdong Chen, Jianmin Bao, Hao Yang, Lu Yuan, Lei Zhang, Houqiang Li, and Dong Chen. Unsupervised Pre-Training for Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14750–14759, 2021. 3

[15] Dengpan Fu, Dongdong Chen, Hao Yang, Jianmin Bao, Lu Yuan, Lei Zhang, Houqiang Li, Fang Wen, and Dong Chen. Large-Scale Pre-Training for Person Re-Identification With Noisy Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2476–2486, 2022. 2, 3, 6

[16] Byeong-Ju Han. Context-Aware Unsupervised Clustering for Person Search. *Proceedings of BMVC*, 2021. 2

[17] Chuchu Han, Kai Su, Dongdong Yu, Zehuan Yuan, Changxin Gao, Nong Sang, Yi Yang, and Changhu Wang. Weakly Supervised Person Search with Region Siamese Networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11986–11995, Montreal, QC, Canada, Oct. 2021. IEEE. 2

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, Seattle, WA, USA, June 2020. IEEE. 5

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. 6

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, Mar. 2015. arXiv:1503.02531 [cs, stat]. 2

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, June 2018. ISSN: 2575-7075. 2

[22] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. PMLR, June 2015. ISSN: 1938-7228. 5

[23] Lucas Jaffe and Avideh Zakhor. Gallery Filter Network for Person Search. In *2023 IEEE/CVF Winter Conference on*

Applications of Computer Vision (WACV), pages 1684–1693, Jan. 2023. ISSN: 2642-9381. 4, 6, 8, 1

[24] Chengyou Jia, Minnan Luo, Caixia Yan, Linchao Zhu, Xiaojun Chang, and Qinghua Zheng. Collaborative Contrastive Refining for Weakly Supervised Person Search. *IEEE Transactions on Image Processing*, 32:4951–4963, 2023. Conference Name: IEEE Transactions on Image Processing. 1, 2, 5

[25] Yang Li, Huahu Xu, Minjie Bian, and Junsheng Xiao. Cross-scale global attention feature pyramid network for person search. *Image and Vision Computing*, 116:104332, Dec. 2021. 2

[26] Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, Dec. 2018. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. 2, 3

[27] Zhengjia Li and Duoqian Miao. Sequential End-to-end Network for Efficient Person Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2011–2019, May 2021. Number: 3. 2, 4, 6, 8

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 5

[29] 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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 740–755, Cham, 2014. Springer International Publishing. 2, 6

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 6

[31] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*, Nov. 2016. 6

[32] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, Feb. 2022. 6

[33] Hao Luo, Pichao Wang, Yi Xu, Feng Ding, Yanxin Zhou, Fan Wang, Hao Li, and Rong Jin. Self-Supervised Pre-Training for Transformer-Based Person Re-Identification, Nov. 2021. arXiv:2111.12084 [cs]. 3

[34] Bharti Munjal, Sikandar Amin, Federico Tombari, and Fabio Galasso. Query-Guided End-To-End Person Search. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 811–820, Long Beach, CA, USA, June 2019. IEEE. 2

[35] Bharti Munjal, Alessandro Flaborea, Sikandar Amin, Federico Tombari, and Fabio Galasso. Query-guided networks for few-shot fine-grained classification and person search. *Pattern Recognition*, 133:109049, Jan. 2023. 2

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 28, 2015. 3

[37] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 5

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec. 2015. 1

[39] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. CrowdHuman: A Benchmark for Detecting Human in a Crowd, Apr. 2018. arXiv:1805.00123 [cs]. 2, 6

[40] Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 5

[41] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision*, 104(2):154–171, Sept. 2013. 3

[42] Benzhi Wang, Yang Yang, Jinlin Wu, Guo-jun Qi, and Zhen Lei. Self-similarity Driven Scale-invariant Learning for Weakly Supervised Person Search, Feb. 2023. arXiv:2302.12986 [cs]. 2

[43] Cheng Wang, Bingpeng Ma, Hong Chang, Shiguang Shan, and Xilin Chen. TCTS: A Task-Consistent Two-Stage Framework for Person Search. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11949–11958, Seattle, WA, USA, June 2020. IEEE. 2

[44] Jiabei Wang, Yanwei Pang, Jiale Cao, Hanqing Sun, Zhuang Shao, and Xuelong Li. Deep Intra-Image Contrastive Learning for Weakly Supervised One-Step Person Search, Feb. 2023. arXiv:2302.04607 [cs]. 2

[45] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with Pyramid Vision Transformer. *Computational Visual Media*, 8(3):415–424, Sept. 2022. 8

[46] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint Detection and Identification Feature Learning for Person Search. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3376–3385, Honolulu, HI, July 2017. IEEE. 2, 5, 6, 1

[47] Yichao Yan, Jinpeng Li, Shengcai Liao, Jie Qin, Bingbing Ni, Ke Lu, and Xiaokang Yang. Exploring Visual Context for Weakly Supervised Person Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):3027–3035, June 2022. Number: 3. 2

[48] Yichao Yan, Jinpeng Li, Jie Qin, Song Bai, Shengcai Liao, Li Liu, Fan Zhu, and Ling Shao. Anchor-Free Person Search. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7686–7695, June 2021. ISSN: 2575-7075. 2, 7

[49] Zizheng Yang, Xin Jin, Kecheng Zheng, and Feng Zhao. Unleashing Potential of Unsupervised Pre-Training With Intra-Identity Regularization for Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14298–14307, 2022. 3

[50] Rui Yu, Dawei Du, Rodney LaLonde, Daniel Davila, Christopher Funk, Anthony Hoogs, and Brian Clipp. Cascade Transformers for End-to-End Person Search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7267–7276, 2022. 2, 6, 8

[51] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable Person Re-identification: A Benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, Dec. 2015. ISSN: 2380-7504. 1

[52] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person Re-identification in the Wild. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3346–3355, Honolulu, HI, July 2017. IEEE. 2, 6, 1

[53] Kuan Zhu, Haiyun Guo, Tianyi Yan, Yousong Zhu, Jinqiao Wang, and Ming Tang. PASS: Part-Aware Self-Supervised Pre-Training for Person Re-Identification. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 198–214, Cham, 2022. Springer Nature Switzerland. 2, 3, 6

Swap Path Network for Robust Person Search Pre-training

Supplementary Material

This supplementary material contains content beneficial to but not required for understanding of the main paper. It includes information about code used to produce results from the paper, additional dataset metadata, implementation details, and results of additional ablations which explore the hyperparameter design space.

6. Code and Reproduction

We include all code, configs, and instructions required to reproduce results from the paper in the corresponding GitHub repository[2].

7. Dataset Metadata

The two standard person search benchmark datasets we use for model fine-tuning are the CUHK-SYSU dataset [46] and the PRW dataset [52].

The CUHK-SYSU dataset has 18k scenes with 95k annotations, split among 23k boxes for 8.5k known identities, and 72k boxes for unknown identities. The scenes in CUHK-SYSU come from a mixture of handheld device photos taken on city streets, and scenes from TV shows and films. The standard test retrieval scenario for CUHK-SYSU uses 2,900 queries with 100 scenes in each gallery.

The PRW dataset has 12k scenes with 43k annotations, split among 34k boxes for 1k known identities, and 9k boxes for unknown identities. The scenes from PRW come from six fixed cameras installed at the Tsinghua University campus in Beijing. The standard test retrieval scenario for PRW uses 2,057 queries with all 6,112 test scenes in the gallery.

8. Additional Implementation Details

Model Configurations. In all model versions, a 4-layer MLP is applied to offset embeddings x_o to produce box regression offsets, while classification logits are computed directly from $\|x_o\|$. The training batch size is 8, with $k = 2$ augmentations per image during pre-training.

Anchor Sampling. To reduce differences between QC and OC pre-training from anchor sampling during detection, we ensure that the same number of anchors are optimized in each batch between QC and OC trials (2,048 by default). For the OC case, this is calculated as number of images per batch $\times$ number of anchors per image. For the QC case, this is calculated as number of images per batch $\times$ number of query-image pairs $\times$ number of anchors per query-image pair.

Query-image pairs constitute the pairing of a given query embedding x_q with anchor embeddings from a given image x_a. With the total number of query-image pairs equal to number of queries $\times$ number of images, this can quickly exceed memory limitations if all pairs are used, so some subsampling procedure is required. We select pairs in the following order up to a fixed number (32 by default): 1) queries are selected for images which they are not present in, but share a label with a box in the image, 2) queries are selected for images which they are present in, and 3) queries are selected for images which they are neither present in, nor share a label with any boxes in.

Both QC and OC sampling methods attempt to balance the number of positive and negative samples, but there are usually more negative samples in practice due to sparsity of positives and typical hyperparameter selection.

Image Augmentation. We adopt three methods of image augmentation, which all consist of combinations of scaling, cropping, and horizontal flipping. For consistency, we label these here using the configuration name used in the code. We call the standard augmentation for person search *window resize* as in [23], abbreviated `wrs`. This consists of scaling the image to fit in a window with shortest side length 900 and longest side length $1,500$. `wrs` augmentation is used only for evaluation. In `rrc2` augmentation, we first perform `wrs` scaling, then randomly select from two types of crops (followed by random horizontal flip): 1) fixed size square random crop containing at least one bounding box in the image, chosen at random, and 2) random sized crop containing all bounding boxes in the image, that is then resized to a fixed square size. These square crop sizes are 512×512 during pre-training and SPNet-S fine-tuning, and 1024×1024 during SPNet-L fine-tuning. Crop size is one of the most important parameters for controlling memory usage. `rrc2` augmentation is used for all fine-tuning runs. `rrc_scale` augmentation is the same as `rrc2` augmentation, except we scale randomly in the range $0.5\times$ to $2\times$ instead of `wrs` scaling. `rrc_scale` augmentation is used for all pre-training runs.

[2]Project repository: https://github.com/LLNL/spnet

Model	Backbone	Pre-train (QC) COCOPersons	Pre-train (OC) COCOPersons	Fine-tune (OC) CUHK-SYSU	Fine-tune (OC) PRW
SPNet-S	ConvNeXt-T	30.0h	24.5h	2.5h	1.5h
SPNet-L	ConvNeXt-B	46.5h	40.5h	9.0h	4.5h

Table 4. Training times for person search model variants on a single A100 GPU. Pre-training and fine-tuning are each done for 30 epochs on the respective datasets.

	CUHK-SYSU			PRW		
Pre-train Method	mAP	GT mAP	AP@0.5	mAP	GT mAP	AP@0.5
Random Init.	69.8	85.8	47.8	22.3	28.7	66.8
Classifier	91.9	95.8	83.2	52.7	56.6	88.8
Ours-OC	92.8	**96.3**	83.9	54.5	57.6	**89.1**
Ours-QC	**93.3**	95.9	**84.9**	**55.6**	**58.7**	88.9

Table 5. Person search (mAP), re-id (GT mAP), and detection metrics (AP@0.5) for pre-training method comparison on SPNet-S with ConvNeXt-Tiny backbone.

Training Times. All models were trained using a single A100 GPU with 82GB VRAM. Pre-training and fine-tuning times are shown in Tab. 4, with the QC pre-training taking 30 hours for SPNet-S and 46.5 hours for SPNet-L. We note that OC pre-training is more efficient, taking about 6 hours less for either model variant.

9. Supporting Ablations

9.1. QC vs. OC Ablations

Subtask Performance. In Tab. 5, we compare QC vs. OC pre-training to ImageNet-1k Classifier pre-training, and random backbone initialization. The comparison is done for SPNet-S with a ConvNeXt-Tiny backbone, and we show metrics of person search (mAP) in addition to metrics of re-id (GT mAP) and detection (AP@0.5). This helps us understand the contribution of pre-training to detection, re-id, and the combined person search problem. Measuring ground truth re-id performance (GT mAP) is equivalent to using a perfect object detector, and helps us understand re-id performance independent of detector quality.

We show both QC and OC pre-training improve over ImageNet-1k Classifier pre-training for all metrics, and that QC is superior to OC pre-training for most metrics. Most importantly, we show that QC pre-training exceeds OC pre-training for person search (mAP) on both datasets (+0.5% on CUHK-SYSU, +1.1% on PRW), though whether re-id or detection benefit more depends on the dataset. Finally, we show that all pre-training vastly exceeds random initialization. Additional ablations comparing QC vs. OC pre-training and fine-tuning are shown in the following subsections. These results show that QC pre-training outperforms OC pre-training across a range of different conditions.

Re-id Pre-training. In Tab. 6, we compare different methods of handling the re-id loss during pre-training. The goal was to isolate the effect of the re-id loss on pre-training, for both QC and OC methods. We found that pre-training only the re-id loss, without optimizing the other losses, was not beneficial for all statistics.

We found that QC pre-training using detected boxes with IoU ≥ 0.7 performed best on balance. When only ground truth (GT) boxes were used to compute the re-id loss, and no detected boxes, we found that QC pre-training was impaired more than OC pre-training.

Query-Centric vs. Object-Centric Pre-training. We compare query-centric vs. object-centric pre-training for four configurations, given in Tab. 7 (bottom), with results in Fig. 7 for the PRW dataset. All models with pre-training significantly outperform the baseline in mAP and top-1 accuracy, and nearly all QC models outperform the corresponding OC model. One exception is the Frozen Backbone model for top-1 accuracy, showing that the query-centric pre-training performs better when the backbone is optimized in addition to later layers.

For pre-training, we also explore the effect of the *Learning without Forgetting* (LwF) loss from [26], equivalent to the knowledge distillation loss from [20]. Since we initialize our model backbone from ImageNet-1k classifier weights before pre-training, the idea is that this loss will help preserve useful features learned during the classifier pre-training. This loss

Method	CUHK-SYSU mAP	CUHK-SYSU top-1	PRW mAP	PRW top-1
Baseline	91.9	93.2	52.7	86.6
Re-id Only	91.5	93.0	51.9	87.4
OC Re-id GT	92.4	93.7	**55.7**	89.1
QC Re-id GT	93.2	**94.3**	55.1	88.5
OC Original	92.8	94.1	54.5	88.8
QC Original	**93.3**	94.1	55.6	**89.5**

Table 6. Comparison of different settings of the re-id pre-training loss for fine-tuning performance. For *re-id GT*, only GT box embeddings are used to compute re-id loss, and for *re-id only*, we optimize only the re-id loss (with GT boxes) and no other losses. In original trials, both GT and detected box embeddings are used to compute re-id loss.

Config. Name	Ablation Name	Cascade Steps	Shared Heads	Re-id Dim
SPNet-S	c0-share-d128	0	✓	128
-	c0-sep-d128	0		128
-	c2-share-d128	2	✓	128
-	c2-sep-d128	2		128
SPNet-L	c2-sep-d2048	2		2048

Config. Name	Ablation Name	Backbone LR	Backbone WD	Post-Backbone LR	Post-Backbone WD	LwF Loss
-	Uniform LR	1e−4	1e−3	1e−4	1e−3	
Pre-train	Mixed LR	1e−5	0	1e−4	1e−3	
-	Mixed LR + LwF	1e−5	0	1e−4	1e−3	✓
-	Frozen-BB	Frozen	Frozen	1e−4	1e−3	
Fine-tune	Fine-Tune	1e−4	5e−4	1e−4	5e−4	

Table 7. SPNet architecture (top) and layer optimization (bottom) hyperparameter configurations. LR = Learning Rate, WD = Weight Decay, LwF = Learning without Forgetting [26].

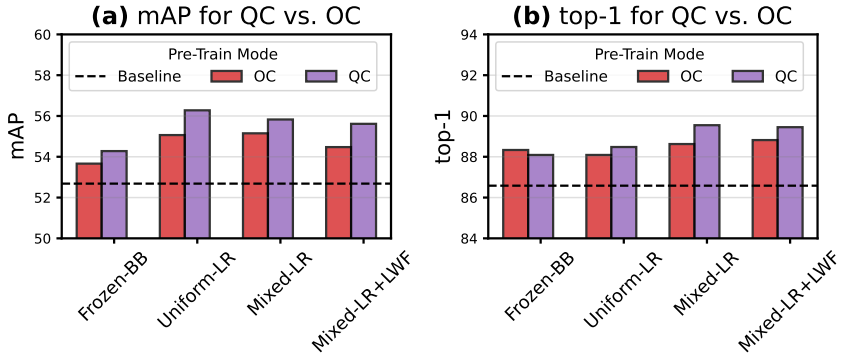

Figure 7. Comparison of SPNet fine-tune performance on PRW with QC vs. OC pre-training for varying layer optimization hyperparameters (configurations described in Tab. 7).

$\mathcal{L}_{\text{LwF}}$ is simply added to the other losses when used, using temperature $T = 2$ as in [26]. As shown in Fig. 7, we did not find use of the LwF loss to result in consistently better performance, so it was not used for other trials.

Query-Centric vs. Object-Centric Fine-tuning. While we primarily explore OC fine-tuning (OC-FT) and evaluation in this work, our framework supports QC fine-tuning (QC-FT) and evaluation as well. QC fine-tuning is done using the same

Method	CUHK-SYSU mAP	CUHK-SYSU top-1	PRW mAP	PRW top-1
OC-FT	91.9	93.2	52.7	86.6
OC-FT+PT	93.3	94.1	55.6	89.5
Gain from PT	+1.4	+0.9	+2.9	+2.9
QC-FT	80.8	87.7	54.0	86.7
QC-FT+PT	77.4	85.4	54.1	87.1
Gain from PT	-3.4	-2.3	+0.1	+0.4

Table 8. Comparison of QC and OC fine-tuning (-FT) for baseline model vs. COCOPersons QC pre-trained model (+PT).

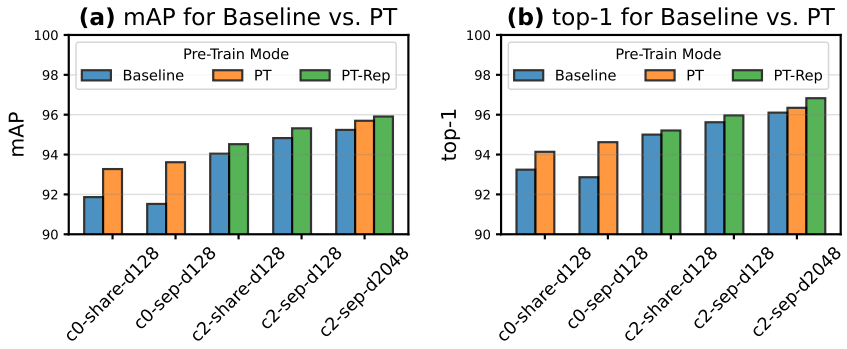

Figure 8. Comparison of SPNet fine-tune performance on CUHK-SYSU with and without QC pre-training for varying architecture hyper-parameters (configurations described in Tab. 7). PT=pre-trained, PT-Rep=pre-trained with replicated weight loading for cascade layers.

procedure as QC pre-training, with $k = 2$ augmentations per image, but using the `rrc2` augmentation method.

In Tab. 8, we compare QC-FT and OC-FT for models with and without COCOPersons QC pre-training. While the QC-FT model outperforms the OC-FT model on PRW top-1 (without PT), OC-FT performance is drastically higher for mAP and top-1 on CUHK-SYSU (>10% mAP). In addition, the OC-FT model benefits significantly from QC pre-training for both datasets and metrics, while the QC-FT model benefits only slightly on PRW and is actually harmed on CUHK-SYSU.

This suggests QC fine-tuning is more impacted by the distribution shift between pre-training and fine-tuning datasets, at least for the shared embedding head model. It also suggests the QC pre-train to OC fine-tune transition has a regularizing effect, limiting the effect of overfitting on the pre-training dataset.

9.2. Model Architecture Ablation

Model Architecture. To understand the model architecture design space, we examine variations in number of cascade steps, shared vs. separate embeddings heads, and size of the re-id embedding dimension. In Fig. 8, we compare the configurations described in Tab. 7 (top). We show that the cascaded model with two steps is better than the base model, using separate embedding and re-id heads results in better fine-tuning performance, and larger re-id embedding dimension is beneficial. In addition, all architecture configurations benefit from pre-training, although the cascaded models benefit less.

Cascaded Weight Loading. When loading weights from pre-trained SPNet into a larger version with multiple cascade stages, we have to make a choice about how to load weights into layers not utilized during pre-training. We explore two simple options: loading weights only into layers that were pre-trained, and duplicating weights into equivalent layers in each cascade stage, shown by *PT* vs. *PT-Rep* in Fig. 8. For the `c2-sep-d2048` configuration, the PT-Rep method of weight loading outperforms PT, showing that the benefits of the pre-trained weight initialization extend to cascade stages, even though the pre-trained model was trained without cascading.

Classifier Logits. Recall from Sec. 3.3 that the offset embedding is defined as the difference between query and anchor embeddings: $x_o = x_q - x_a$ with $x_o \in \mathbb{R}^d$ and $w = \|x_o\|$. Then the class logits z can be calculated as

Method	CUHK-SYSU		PRW	
	mAP	top-1	mAP	top-1
BatchNorm Logits (+)	90.4	91.9	49.8	84.3
BatchNorm Logits (-)	91.2	92.0	42.2	82.2
FixedNorm Logits (+)	91.1	92.2	52.6	86.3
FixedNorm Logits (-)	**91.9**	**93.2**	**52.7**	**86.6**

Table 9. Comparison of BatchNorm Logits from [6] to proposed FixedNorm Logits for CUHK-SYSU and PRW datasets, for the baseline model. (+) vs. (-) indicates whether positive or negative embedding norm was used to compute logits.

$$z = \frac{w - \mu_{\text{Chi}}}{s_{\text{Chi}}} \tag{3}$$

with

$$\mu = \sqrt{2}\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}, \quad s = \sqrt{d - \mu^2} \tag{4}$$

We note for $d \gg 1$, $\mu \approx \sqrt{d - \frac{1}{2}}$ and $s \approx \sqrt{\frac{1}{2}}$.

The *FixedNorm* transformation in Eq. (3) has two key properties: 1) For $x_o \sim \mathcal{N}(0, I_d)$, $\mathbb{E}[z] = 0, \text{Var}[z] = 1$, independent of d. In addition, $\lim_{d\to\infty} z \sim \mathcal{N}(0, 1)$ due to the Central Limit Theorem. Although the distribution of x_o is rarely unit-normal and changes during optimization, this framing gives us a reasonable choice of shifting and scaling parameters which works well in practice, does not require learnable parameters or hyperparameters, and holds independent of d.

2) For x_q similar to x_a *i.e.*, $\|x_o\|$ small, z is larger, and for x_q dissimilar to x_a *i.e.*, $\|x_o\|$ large, z is smaller. Intuitively, when a query embedding matches a given anchor embedding, according to box IoU overlap, we want the two to be more similar, and when they do not match, we want them to be more different. The relationship is also critical to co-optimizing with the re-id loss on x_q and has a nice relationship with the box regression loss.

To validate our FixedNorm logits over the BatchNorm logits from [6], we compare both for the baseline model, with results shown in Tab. 9. The FixedNorm method achieves greater performance for all metrics, especially on the PRW dataset, where mAP exceeds the BatchNorm method by more than 10%. This is likely due in part to unbalanced batch statistics, which are dominated by negative samples, unlike the original Norm-Aware Embedding use case. Tab. 9 also shows that negating the norm after scaling is beneficial, especially for the FixedNorm logits, validating the rationale discussed above.