2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10222300

CAN HUMAN ATTRIBUTE SEGMENTATION BE MORE ROBUST TO OPERATIONAL CONTEXTS WITHOUT NEW LABELS?

Hejer Ammar[*], *Angelique Loesch*[*,+], *Corentin Vannier*[*,+], *Romaric Audigier*[*,+]

[*]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
[+]Vision Lab, ThereSIS, Thales SIX GTS, Campus Polytechnique, Palaiseau, France
{*firstname.lastname*}@*cea.fr*

ABSTRACT

Human Attribute Segmentation (HAS) describes, pixelwise, the different semantic parts of people in an image. This fine-grained description is useful for several applications (e.g. security, fashion). However, despite the good performance reached by supervised Semantic Segmentation (SS) approaches, they are usually biased by the source training dataset and suffer from a performance drop when applied on new domains. Pixelwise image annotation for each new encountered context is tedious and expensive. So how can HAS become more robust to new contexts without new annotations? In this first study of Unsupervised Domain Adaptation (UDA) for HAS, we present UDA-HPTR, a new method based on HPTR [1] (Human Parsing with TRansformers) combined with self-supervised and semi-supervised learning paradigms to deal with UDA. UDA-HPTR improves performance on both source (labeled) and target (unlabeled) datasets compared to the fully supervised version (HPTR). It also outperforms HRDA, a state-of-the-art UDA method in autonomous driving benchmarks, by $+6.7$ p.p. on the source and $+8.8$ p.p. on the target, when applied to HAS while using only half the number of parameters.

Index Terms— Human attribute semantic segmentation, human parsing, unsupervised domain adaptation, semi-supervised learning, self-supervised learning

1. INTRODUCTION

Human Attribute Segmentation (HAS) is a specific semantic segmentation task that localizes pixelwise the human attributes (visible body parts, clothes and accessories) in an image (cf. Fig. 1). This fine-grained description of people is generally handled together with the person detection and segmentation tasks, in the so-called Human Parsing (HP). It is particularly useful for retrieval or augmented reality applications. But, large amounts of precisely annotated data are needed to achieve adequate performance on different operational contexts. While acquiring images containing people can be easy, annotating them pixelwise is tedious and expensive. So, how to avoid annotating new data?

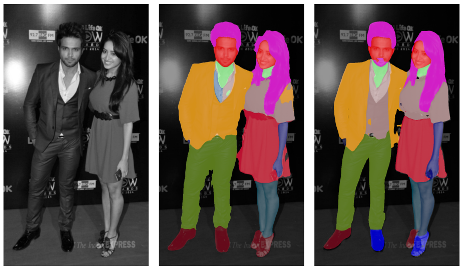

Fig. 1. Human Attribute Segmentation (HAS) of a sample (left) from unlabeled target dataset MHP-GB [2] by HRDA [3] (middle) and UDA-HPTR (right).

Supervised Human Parsing. Several methods tackle supervised HP. *Single-person* HP approaches [4, 5] make the assumption that only one person is present in the image, without dealing with the attribute-to-person assignment problem when an image contains many people. On the contrary, *multi-person* HP approaches distinguish instances. *Top-down* approaches can be either *two-stage* [6] if they require human instance segmentation as an additional input (thus, they are similar to *single-person* methods), or *one-stage* [7, 8, 9] if they jointly provide human instances and attributes. Nevertheless, in both cases, the inference time highly depends on the number of humans per image. In contrast, *bottom-up* methods [1, 10, 2] reach lower performance but are more scalable, as their inference time is constant regardless of the number of people per image, which is especially useful for real-time applications. In addition, for such applications, inference time should not only be constant, but also reduced. Yet, some bottom-up methods are not fast enough due to their expensive post-processing [10] or heavy GAN architectures [2]. Currently, HPTR [1] (Human Parsing with TRansformers) is the fastest bottom-up approach while having comparable performance with other state-of-the-art (SOTA) methods. It is an

Co-funded by the European Union under Grant Agreement Nb. 101073951. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.
This publication was made possible by the use of the Factory-AI supercomputer, financially supported by the Ile-De-France Regional Council.

end-to-end multi-task approach based on the object detector DETR [11], jointly providing human detection and instance segmentation, in addition to predicting human attributes and their characteristics (size, pattern, color). Note that all these methods are fully supervised, using manually annotated training dataset, and testing on images from the same distribution.

Unsupervised Domain Adaptation (UDA) has not been studied yet for HP or HAS but has received much attention **for Semantic Segmentation (SS)**. The goal is to transfer the learned knowledge from a *source* annotated dataset to a *target* unlabeled dataset which can have a different distribution. The existing UDA approaches for SS are based on either *adversarial training* [12, 13] or *pseudo-labeling* [14, 3]. The idea behind adversarial training is to align features and/or images of source and target domains using a discriminator network. Adversarial training aims to align features and/or images of source and target domains using a discriminator network. Recently, SOTA methods have focused on pseudo-labeling inherited from the *semi-supervised* learning paradigm, where only a small portion of the dataset is annotated. These methods usually adopt a student-teacher scheme, where the teacher's predictions on the unlabeled part of the dataset are used as pseudo-labels to train the student model jointly with the labeled images. Different techniques are used to regularize the training. In particular, Unbiased Teacher [15, 16] (UT) proposes to mutually train the student and a gradually updated teacher to ensure better quality pseudo-labels, which significantly improves Object Detection (OD) performance. Similarly to semi-supervised learning, several UDA methods use a student-teacher scheme [14, 3], trying to transfer information from a labeled source dataset to an unlabeled target dataset. Particularly, HRDA [3] uses the UT solution to continuously propagate information from student to teacher, along with the architecture and regularization techniques of DAFormer [14] such as rare class sampling, forward distance and learning rate warm-up. It also uses domain mixing by adding source instances to target images and learns a multi-resolution input fusion. Currently, it significantly outperforms SOTA in UDA for SS on autonomous driving datasets by adapting models learnt on synthetic data to real-world unlabeled images.

Self-Supervised Learning is another paradigm that exploits unlabeled data for a better generalization. It learns representations from automatically-labeled *pretext tasks*. These tasks are generally used to pre-train the model followed by a fine-tuning on the *down-stream task*, or added to the training process as an *Auxiliary Task* (AT). The most recent works use contrastive learning [17, 18] to bring closer features of different forms of the same image while keeping those of different images distant. This achieves great results for image classification. To target more localization-dependent tasks such as OD and SS, several other methods deal with local representation learning at pixel level [19, 20, 21]. In particular, ReSim [21] maximizes region similarity of corresponding sliding windows over different views of the same image, which aims to learn a more spatially and semantically consistent feature representation, as required by HP.

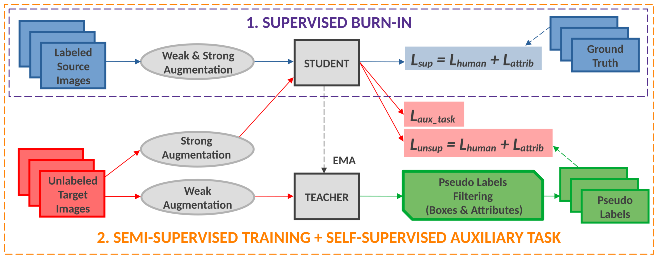
1. SUPERVISED BURN-IN
Labeled Source Images
Weak & Strong Augmentation
STUDENT
$L_{sup} = L_{human} + L_{attrib}$
Ground Truth
L_{aux_task}
$L_{unsup} = L_{human} + L_{attrib}$
Unlabeled Target Images
Strong Augmentation
Weak Augmentation
EMA
TEACHER
Pseudo Labels Filtering (Boxes & Attributes)
Pseudo Labels
2. SEMI-SUPERVISED TRAINING + SELF-SUPERVISED AUXILIARY TASK

Fig. 2. Overview of UDA-HPTR training: after (1) a supervised burn-in on the source dataset for OD and HAS, (2) semi- and self-supervised learning on the target dataset allow UDA.

In this work, we propose UDA-HPTR (see Fig. 2), a first solution studying UDA for HAS. It is based on HPTR [1], the fastest and scalable HP method with competitive performance. We study how to improve its robustness to contexts by combining two label-free paradigms. (i) We train our model in a semi-supervised manner by adapting our UDA problem to the UT [15] scheme. (ii) We exploit the self-supervised approach ReSim [21] by adding it to the training, both as a pretrain and an AT. The effectiveness of the proposed approach is demonstrated through extensive ablation experiments on a new proposed benchmark for UDA-HAS. Our contributions can be formulated as follows: (1) We propose UDA-HPTR, an UDA method for HAS by jointly adopting self-supervised and semi-supervised approaches to the specific task of HAS. To the best of our knowledge, this is the first attempt that combines these two paradigms for a better UDA, and that studies UDA for HAS. (2) We create and share [22] a new suitable benchmark for UDA-HAS. (3) We show that our UDA-HPTR highly outperforms the SOTA method for UDA-SS. Suprisingly, despite its good performance on autonomous driving datasets, HRDA is not necessarily the most effective for the particular segmentation case of HAS (cf. illustration in Fig. 1).

2. METHOD

2.1. Overview

UDA-HPTR, deals with UDA for HAS (see Fig. 2) using source annotated and target unlabeled datasets. The goal is to improve the model robustness by adapting it to the target domain without any annotation, despite its discrepancy from the source dataset. Therefore, we use HPTR [1] as our base architecture. The original method performs human detection and instance segmentation, as well as attribute and characteristic segmentation. UDA-HPTR is lightened (cf. Fig. 3) to focus on human detection and attribute segmentation.

To perform UDA, we combine two main paradigms. We adapt the UT [15] scheme to HPTR method. We further in-

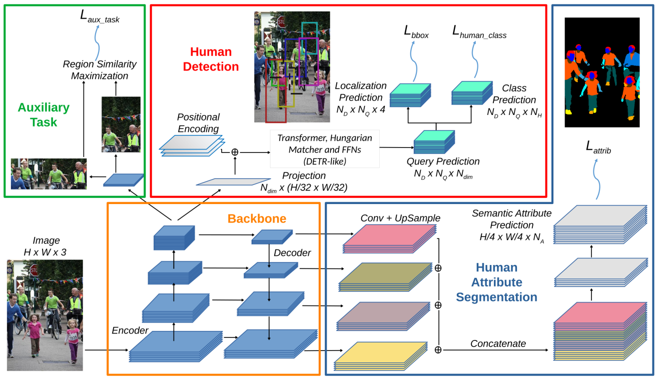
L_{aux_task}
Region Similarity Maximization
Auxiliary Task
Human Detection
L_{bbox}
L_{human_class}
Localization Prediction N_D x N_Q x 4
Class Prediction N_D x N_Q x N_H
Positional Encoding
Transformer, Hungarian Matcher and FFNs (DETR-like)
Query Prediction N_D x N_Q x N_{dim}
Projection N_{dim} x (H/32 x W/32)
L_{attrib}
Image H x W x 3
Backbone
Decoder
Encoder
Conv + UpSample
Semantic Attribute Prediction H/4 x W/4 x N_A
Human Attribute Segmentation
Concatenate

Fig. 3. Overview of UDA-HPTR architecture: *Backbone*, *Human Detection* and *Human Attribute Segmentation* blocks are the same as in HPTR [1]. A self-supervised *Auxiliary Task* on target is plugged into the backbone.

tegrate within the semi-supervised training step, the ReSim [21] AT for its capacity to better learn local features representation. This helps the model have better pseudo-labels for HAS, and learn features related to the target as well.

2.2. Supervised Burn-in and Semi-Supervised Training

Following UT, our training is composed of two stages. First, a *Supervised Burn-in (SBI)* training uses weak and strong augmentations (cf. Sec. 3.1) of the source images. This step serves as initialization of both student and teacher models. Here, we remove the instance segmentation and the characteristic segmentation branches to alleviate the base HPTR architecture, since we focus on UDA-HAS. The new architecture (Fig. 3) performs human detection and HAS. At SBI step, the training objective is $L_{sup} = L_{human} + L_{attrib}$, where $L_{human} = L_{Hungarian} + L_{bbox} + L_{human\_class}$. The different terms are such as defined in HPTR [1].

Second, a *Semi-Supervised Training (SST)* step is conducted. At each iteration, we use the teacher model to generate predictions on weakly augmented target images, which are then filtered. For object detection, we follow UT by using Non-maximum Suppression (NMS) and confidence thresholding keeping non-overlapped bounding boxes having high confidence $> \delta$. For attribute segmentation, for each pixel, we apply a $softmax$ function on the model's output and choose the class attribute having the maximum probability. These filtered predictions are then used as pseudo-labels to train the student on a strongly augmented version of the same target images. The supervised training on weakly and strongly augmented source images continues alongside. The different augmentation techniques help the model regularization. This approach ensures a gradual improvement of the student generalization. On the other hand, the knowledge learned by the student is transferred to the teacher using an Exponential Moving Average (EMA) on the network weights to gradually improve the quality of the pseudo-labels, as in UT. At SST step, the training objective is the weighted sum of L_{sup} and L_{unsup}, which have the same definition applied to predictions on resp. labeled/unlabeled images and their related ground truth/pseudo-labels: $L_{semi} = L_{sup} + \lambda_u L_{unsup}$.

2.3. Auxiliary Task

To further improve UDA, SST is assisted by a self-supervised AT (cf. Fig. 3). We choose to integrate the knowledge of the target domain to our backbone using ReSim [21] by adding a specialized head. It maximizes region similarity of sliding windows over different views of the same image. We apply this method on the target dataset to learn more adapted feature representation. It also allows an implicit feature alignment between the source and the target datasets as this task is performed alongside the supervised learning on the source dataset. To this end, we optimize the model by adding L_{aux_task} (such as defined in [21]) to the semi-supervised loss. At SST step, $L_{final} = L_{sup} + \lambda_u L_{unsup} + L_{aux\_task}$.

3. EXPERIMENTS AND RESULTS

3.1. Implementation Details

For UDA-HPTR, we use the HPTR [1] base code. The encoder is a ResNet-50 pre-trained by ReSim [21] on ImageNet for 400 epochs. Our training is done on 8 NVIDIA A100 80GB, with 64 images per batch, and a learning rate $\gamma = 0.0001$. We train the SBI stage during 100 epochs, and the SST for another 100 epochs. Each batch in SST is equally divided into 32 source and 32 target images. The EMA rate and the pseudo-labels filtering for OD are the same as in UT [15]. To balance the supervised and unsupervised losses during SST, we use $\lambda_u = 1$. Lower values of λ_u and γ are chosen compared to [15] to deal with the smaller datasets, in order to prevent over-fitting. We use the same augmentations as UT, however, since CCIHP [1] labels include left and right variants of attributes arms, legs, and shoes, we have disabled horizontal flips. Please refer to [15] for more details.

For HRDA, we use the same code and configuration available at [23], while disabling the flip augmentations for the same reason as above. We have also used the forward distance on all attribute classes since all of them represent *things*. For a fair comparison, we conducted an extensive hyper-parameter tuning on image sizes, LR and HR crop sizes, image normalization, learning rate warm-up, rare class sampling, EMA and confidence estimate threshold, for our specific task and datasets. However, the default parameters gave always better performance, hence they were kept.

3.2. Datasets and Evaluation Protocols

CIHP [10] is the largest existing multi-person HP dataset, with 28,280 images for train and 5,000 images for validation. It contains 110,700 segmented people, with an average of 3

				Source (CCIHP)			Target (MHP-GB)				
Method	SST	AT	#P	Src-Only	UDA	Rel. Src	Src-Only	UDA	Rel. Src	Oracle	Rel. Tgt
HRDA [3]	✓	✗	85.2	47.1	47.7	101.3%	41.7	42.1	101.0%	**58.0**	72.6%
UDA-HPTR	✓	✓	**41.5**	**53.5**	**54.4**	**101.7%**	**49.2**	**50.9**	**103.5%**	53.2	**95.7%**
UDA-HPTR-v1	✓	✗	41.5	53.5	54.0	100.9%	49.2	50.5	102.6%	53.2	94.9%
UDA-HPTR-v2	✗	✓	41.5	53.5	53.2	99.4%	49.2	48.8	99.2%	53.2	91.7%
HPTR	✗	✗	41.5	53.5	-	100.0%	49.2	-	100.0%	53.2	92.5%

Table 1. Human Attribute Segmentation performance (mIoU in %, relative gain of UDA in %, and performance wrt supervised Oracle in %) of SOTA method HRDA compared to our UDA-HPTR, and its variants with ablation of Semi-Supervised Training (SST) and Auxiliary Task (AT). #P is the number of parameters of the model (in million).

people per image. It segments humans into 19 possible semantic attribute classes. **Characterized CIHP (CCIHP)** [1] adds characteristics annotations of color, size, and pattern, for each attribute of CIHP and improves HAS annotation. We define CCIHP as our source dataset.

MHP v2.0 [2] is another major dataset for multi-person HP with 20,403 images and an average of 3 people per image. It provides 58 semantic attribute classes. **MHP Gray Blurred (MHP-GB)** is a version of MHP v2.0 we generated to widen the gap between source and target, and make UDA more challenging. We transform the images to grayscale, and downscale them by a factor 2 before upscaling them back to their original size to add blur. We map the 58 classes to the 19 CCIHP classes (cf. script soon available at [22]). Only the 'face mask' class has no MHP equivalent. So, it is not evaluated on MHP-GB. We define MHP-GB as our target dataset.

Metrics. We use *mIoU* for HAS to evaluate different results. For each method and ablation, we provide the source-only performance (trained only on supervised source data: *Src-Only*), the UDA performance (trained on both supervised source and unsupervised target data: *UDA*), as well as the oracle performance for an upper-bound on target dataset (trained on supervised target data: *Oracle*). We also add metrics to illustrate the relative performance for a fair comparison of different methods: *Rel. Src* is the relative performance gain of UDA over Src-Only, and shows the impact of UDA over the model robustness; *Rel. Tgt* is the relative performance of UDA over Oracle and represents the quality of UDA compared to the supervised upper-bound.

3.3. Comparison of UDA-HPTR with SOTA method for semantic segmentation

In the first two lines of Table 1, we compare UDA-HPTR, against HRDA [3]. On the source dataset, we notice that UDA-HPTR largely surpasses HRDA on UDA absolute results (54.4% vs. 47.7%) despite using less than half the number of parameters. Moreover, for a fair comparison of the gain of the UDA approaches, we compare *Rel. Src*. We can see that UDA-HPTR was able to learn slightly better by gaining 1.7% instead of 1.3%, on the source dataset. More importantly, this is also the case on the target dataset, where UDA-HPTR largely outperforms HRDA (50.9% vs. 42.1%), gaining up to 3.5% instead of 1.0% relative to the non-adapted models.

Moreover, while HRDA has much better oracle performance on the target dataset, it was only able to reach 72.6% of this capacity after UDA. Thus, there is still room for better adaptation. On the other hand, UDA-HPTR reaches 95.7% of its total capacity on the target dataset without using any annotation. In fact, switching from oracle to UDA, HRDA loses 15.9 p.p. (42.1% vs. 58.0%), while UDA-HPTR degrades only by 2.3 p.p. (50.9% vs. 53.2%). This proves the effectiveness of our UDA approach for HAS, resulting in a model better adapted to target and improved on source. Surprisingly, despite its good performance on synthetic to real UDA for autonomous driving scenes, HRDA has lower performance when applied to HAS.

3.4. Ablation of the SST and AT

To study the impact of SST (UT) on UDA-HPTR, we consider the fully supervised version (HPTR) trained for 200 epochs (equivalent in time to 100 for SBI + 100 for SST). This version surpasses the original HPTR (53.5% vs. 52.1% [1]) on CCIHP, thanks to the alleviated tasks learned. Further, comparing the performance of this model, to the one trained in a semi-supervised manner without AT (UDA-HPTR-v1), we notice that SST improves the generalization to both source (54.0% vs. 53.5%) and target datasets (50.5% vs. 49.2%).

Surprisingly, the use of the AT without semi-supervised learning, i.e. as a pre-train task (UDA-HPTR-v2), decreases performance for both source (53.2% vs. 53.5%) and target (48.8% vs. 49.2%) datasets. However, adding the same AT during the SST (UDA-HPTR) shows additional benefit, with 54.4% mIoU on source and 50.9% on target, i.e., a $+0.4$ p.p. improvement on both datasets compared to UDA-HPTR-v1.

4. CONCLUSION

This investigates how to improve HAS robustness to new contexts without extra annotations. We present UDA-HPTR, a first solution to UDA for HAS. It is also a first attempt to combine two well-known paradigms for UDA (semi-supervised and self-supervised learning). We show that, on a novel UDA-HAS benchmark, UDA-HPTR outperforms HRDA, the SOTA method for UDA for SS on autonomous driving datasets, while using less than half the number of parameters.

5. REFERENCES

[1] A. Loesch and R. Audigier, “Describe Me If You Can! Characterized Instance-Level Human Parsing,” in *ICIP*, 2021.

[2] J. Zhao, J. Li, Y. Cheng, T. Sim, S. Yan, and J. Feng, “Understanding Humans in Crowded Scenes: Deep Nested Adversarial Learning and A New Benchmark for Multi-Human Parsing,” in *ACM MM*, 2018.

[3] L. Hoyer, D. Dai, and L. Van Gool, “HRDA: Context-aware high-resolution domain-adaptive semantic segmentation,” in *ECCV*, 2022.

[4] K. Gong, Y. Gao, X. Liang, X. Shen, M. Wang, and L. Lin, “Graphonomy: Universal Human Parsing via Graph Transfer Learning,” in *CVPR*, 2019.

[5] H. He, J. Zhang, Q. Zhang, and D. Tao, “Grapy-ML: Graph Pyramid Mutual Learning for Cross-Dataset Human Parsing,” in *AAAI*, 2020.

[6] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, Y. Zhao, and T. Huang, “Devil in the Details: Towards Accurate Single and Multiple Human Parsing,” in *AAAI*, 2018.

[7] L. Yang, Q. Song, Z. Wang, and M. Jiang, “Parsing R-CNN for Instance-Level Human Analysis,” in *CVPR*, 2019.

[8] L. Yang, Q. Song, Z. Wang, M. Hu, C. Liu, X. Xin, W. Jia, and S. Xu, “Renovating Parsing R-CNN for Accurate Multiple Human Parsing,” in *ECCV*, 2020.

[9] S. Zhang, X. Cao, G.-J. Qi, Z. Song, and J. Zhou, “AIParsing: Anchor-Free Instance-Level Human Parsing,” *IEEE TIP*, 2022.

[10] K. Gong, X. Liang, Y. Li, Y. Chen, M. Yang, and L. Lin, “Instance-level Human Parsing via Part Grouping Network,” in *ECCV*, 2018.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” in *ECCV*, 2020.

[12] F. Pizzati, R. d. Charette, M. Zaccaria, and P. Cerri, “Domain Bridge for Unpaired Image-to-Image Translation and Unsupervised Domain Adaptation,” in *WACV*, 2020.

[13] H. Wang, T. Shen, W. Zhang, L. Duan, and T. Mei, “Classes Matter: A Fine-Grained Adversarial Approach to Cross-Domain Semantic Segmentation,” in *ECCV*, 2020.

[14] L. Hoyer, D. Dai, and L. Van Gool, “DAFormer: Improving Network Architectures and Training Strategies for Domain-Adaptive Semantic Segmentation,” in *CVPR*, 2022.

[15] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, “Unbiased Teacher for Semi-Supervised Object Detection,” in *ICLR*, 2021.

[16] Y.-C. Liu, C.-Y. Ma, and Z. Kira, “Unbiased Teacher v2: Semi-supervised Object Detection for Anchor-free and Anchor-based Detectors,” in *CVPR*, 2022.

[17] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” in *ICML*, 2020.

[18] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments,” in *NeurIPS*, 2020.

[19] M. Minderer, C. Sun, R. Villegas, F. Cole, K. Murphy, and H. Lee, “Unsupervised Learning of Object Structure and Dynamics from Videos,” in *NeurIPS*, 2019.

[20] J. Rabarisoa, V. Belissen, F. Chabot, and Q.-C. Pham, “Self-supervised pre-training of vision transformers for dense prediction tasks,” in *CVPR T4V Workshop*, 2022.

[21] T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell, “Region Similarity Representation Learning,” in *ICCV*, 2021.

[22] https://kalisteo.cea.fr/index.php/free-resources/.

[23] https://github.com/lhoyer/HRDA.