HumanBench: Towards General Human-centric Perception with Projector Assisted Pretraining

Shixiang Tang\textsuperscript{1,4*}, Cheng Chen\textsuperscript{4}, Qingsong Xie\textsuperscript{4}, Meilin Chen\textsuperscript{2,4}, Yizhou Wang\textsuperscript{2,4}, Yuanzheng Ci\textsuperscript{1}, Lei Bai\textsuperscript{3†}, Feng Zhu\textsuperscript{4}, Haiyang Yang\textsuperscript{4}, Li Yi\textsuperscript{4}, Rui Zhao\textsuperscript{4,5}, Wanli Ouyang\textsuperscript{4}

\textsuperscript{1}The University of Sydney, \textsuperscript{2}Zhejiang University, \textsuperscript{3}Shanghai AI Laboratory, \textsuperscript{4}SenseTime Research, \textsuperscript{5}Qing Yuan Research Institute, Shanghai Jiao Tong University, China

*Equal contribution. This work was done in SenseTime.
†Corresponding author.

Abstract

Human-centric perceptions include a variety of vision tasks, which have widespread industrial applications, including surveillance, autonomous driving, and the metaverse. It is desirable to have a general pretrain model for versatile human-centric downstream tasks. This paper forges ahead along this path from the aspects of both benchmark and pretraining methods. Specifically, we propose a HumanBench based on existing datasets to comprehensively evaluate on the common ground the generalization abilities of different pretraining methods on 19 datasets from 6 diverse downstream tasks, including person ReID, pose estimation, human parsing, pedestrian attribute recognition, pedestrian detection, and crowd counting. To learn both coarse-grained and fine-grained knowledge in human bodies, we further propose a Projector Assisted Hierarchical pretraining method (PATH) to learn diverse knowledge at different granularity levels. Comprehensive evaluations on HumanBench show that our PATH achieves new state-of-the-art results on 17 downstream datasets and on-par results on the other 2 datasets. The code will be publicly at https://github.com/OpenGVLab/HumanBench.

1. Introduction

Human-centric perception has been a long-standing pursuit for computer vision and machine learning communities. It encompasses massive research tasks and applications including person ReID in surveillance [16, 17, 60, 96, 112], human parsing and pose estimation in the metaverse [47, 48, 61, 79, 90, 92], and pedestrian detection in autonomous driving [8, 51, 87]. Although significant progress has been made, most existing human-centric studies and pipelines are task-specific for better performances, leading to huge costs in representation/network design, pretraining, parameter-tuning, and annotations. To promote real-world deployment, we ask: whether a general human-centric pretraining model can be developed that can benefit diverse human-centric tasks and be efficiently adapted to downstream tasks?

Intuitively, we argue that pretraining such general human-centric models is possible for two reasons. First, there are obvious correlations among different human-centric tasks. For example, both human parsing and pose estimation predict the fine-grained parts of human bodies [29, 49] with differences in annotation granularities. Thus, the annotations in one human-centric task may benefit other human-centric tasks when trained together. Second, recent achievements in foundation models [5, 11, 38, 65, 66, 80] have shown that large-scale deep neural networks (e.g., transformers [13]) have the flexibility to handle diverse input modalities and the capacity to deal with different tasks. For example, Uni-Perceiver [115] and BEITv3 [85] are applicable to multiple vision and language tasks.

Despite the opportunities of processing multiple human-centric tasks with one pretraining model, there are two obstacles for developing general human-centric pretraining models. First, although there are many benchmarks for every single human-centric task, there is still no benchmark to fairly and comprehensively compare various pretraining methods on a common ground for a broad range of human-centric tasks, data distributions, and application scenarios. Second, different from most existing general foundation models trained by unified global vision-language consistencies, pretraining human-centric models are required to learn both global (e.g., person ReID and pedestrian detection) and fine-grained semantic features (e.g., pose estimation and human parsing) of human bodies from diverse annotation granularity simultaneously.

In this paper, we first build a benchmark, called HumanBench, based on existing datasets to enable pretrain-
ing and evaluating human-centric representations that can be generalized to various downstream tasks. HumanBench has two appealing properties. (1) **Diversity.** The images in our HumanBench include diverse image properties, ranging from person-centric cropped images to scene images with crowd pedestrians, ranging from indoor scenes to outdoor scenes (Fig. 1(a)), and from surveillance to metaverse. (2) **Comprehensiveness.** HumanBench covers comprehensive image-based human-centric tasks in both pretraining datasets and downstream tasks (Fig. 1(b)). For pretraining, we include 11 million images from 37 datasets across five representative human-centric tasks, i.e., person ReID, pose estimation, human parsing, pedestrian attribute recognition, and pedestrian detection. For evaluation, HumanBench evaluates the generalization abilities on 12 pretraining datasets, 6 unseen datasets of pretraining tasks, and 2 datasets out of pretraining tasks, ranging from global prediction, i.e., ReID, to local prediction, i.e., human parsing and pose estimation. Results on our HumanBench (Fig. 1(c)) lead to two interesting findings. First, compared with datasets with natural images for general pretrained models, HumanBench is more effective for human-centric perception tasks. Second, as human-centric pretraining requires to learn features of diverse granularity, supervised pretraining methods with proper designs can learn from diverse annotations in HumanBench and perform better than the existing unsupervised pretraining methods, for which details will be shown in Sec. 5.3.

Based on HumanBench, we further investigate how to learn a better human-centric supervised pretraining model from diverse datasets with various annotations. However, naive multitask pretraining may easily suffer from the task conflicts [53, 97] or overfitting to pretrained annotations [67, 107], losing the desirable generalization ability of pretraining. Inspired by [86], which suggests adding an MLP projector before the task head can significantly enhance the generalization ability of supervised pretraining, we propose **Projector Assisted Hierarchical Pre-training (PATH)**, a projector assisted pretraining method with hierarchical weight sharing to tackle the task conflicts of supervised pretraining from diverse annotations. Specifically, the weights of backbones are shared among all datasets, and the weights of projectors are shared only for datasets of the same tasks, while the weights of the heads are shared only for a single dataset – forming a hierarchical weight-sharing structure. During the pretraining stage, we insert the task-specific projectors before dataset heads but discard them when evaluating models on downstream tasks. With the hierarchical weight-sharing strategy, our pretraining method enforces the backbone to learn the shared knowledge pool, the projector to attend to the task-specific knowledge, and the head to focus on the dataset with specific annotation and data distribution.

In summary, our contributions are two folds: (1) we build HumanBench, a large-scale dataset for human-centric pretraining including diverse images and comprehensive evaluations. (2) To tackle the diversity of input images and annotations of various human-centric datasets, we propose PATH, a projector-assisted hierarchical weight-sharing method for pretraining the general human-centric representations. We achieve state-of-the-art results by PATH on 15 datasets throughout 6 downstream tasks (Fig. 1(c)), on-par results on 2 datasets, and slightly lower results on 2 datasets on HumanBench when using ViT-Base. Experiments with ViT-Large backbone show that our method can further achieve considerable gains over ViT-Base, achieving another 2 new state-of-the-art results and showing the promising scalability of our method. We hope our work can shed light on future research on pretraining human-centric representations, such as unified structures.
2. Related Work

General Vision Models. The vision community has witnessed the emergence of general vision models [69, 85, 98, 115], which could learn complex patterns from large-scale data and are powerful when adapted for downstream tasks. Although they achieve promising results, they pretrain on natural images, which are sub-optimal for human-centric perception tasks. Furthermore, most of them leverage contrastive learning or masked reconstruction on global images, but do not learn multi-scale representations of human bodies with a diverse granularity that is important for diverse human-centric perceptions. In this paper, we tackle this problem by supervised pretraining from large-scale publicly available datasets with released annotations. The massive human body annotations with diverse granularity enable the model to learn multi-scale features, which is preferred in various human-centric tasks. SL-MLP [86] is an improved supervised pretraining method that proposes to insert an MLP projector to increase the generalization ability of pretraining models. However, it does not tackle the task conflict problem in a multitask pretraining as our method.

Pretraining on Human-centric Tasks. Human-centric perception has been studied for decades. However, there are only a few works on pretraining diverse human-centric tasks. HCMoCo [29] pretrains human representations for only pose estimation and human parsing with human-centric images. Our work pretrains a model for more diverse human-centric tasks, ranging from global identification to local prediction, from tasks on cropped human-centric images to scene images. Furthermore, different from HCMoCo which mainly learns human-centric representation from the multi-modal (RGB-D) representation of the same image, we learn human representations from multi-datasets where RGB images with one or two kinds of annotation, which is a more practical and data-scalable setting for pretraining. For example, publicly available datasets for ReID, attribute, and counting do not have RGB-D data.

Multi-task Multi-dataset Pretraining. Multi-task and Multi-dataset Pretraining is a popular framework for pretraining models [27, 84, 115]. However, they may easily suffer from task conflicts. There are generally two routes to reduce task conflicts, including network designs [25, 35, 55, 59, 63, 78] and optimization strategies [7, 20, 36, 55, 68]. Different from existing methods to tackle task conflicts from the balance of losses, Our PATH explores a new network optimization strategy from framework design at the training stage. Specifically, we insert a task-specific projector into the backbone and the dataset head, and design the parameters in the backbone, the parameters in the task-specific projector and the head are shared among all datasets, datasets of the same task and not shared. With this strategy, the dataset-specific heads and the task-specific projectors are the regularizations for the backbone of learning shared knowledge across various human-centric tasks. Our method does not design new networks because we only evaluate the generalization of the backbone.

3. HumanBench

3.1. Pretraining Datasets

According to biologists [9], nonverbal communication in daily life includes identity, visual appearance, and posture information. Following this domain knowledge, we select person ReID as the identification task, pedestrian attribute recognition, pedestrian detection, human parsing as the visual appearance task, and pose estimation as the posture task in HumanBench. 37 datasets containing 11,019,187 images \(^1\) are collected for pretraining. Tab. 1 presents the number of datasets and images in each task. For the selected datasets, we leverage their original annotations except for the noisy labeled person ReID dataset, i.e., LUPerson-NL. In LUPerson-NL, we observe that identities with relatively few images are accurate. Therefore, we only select the identities that contain 15 to 200 images in LUPerson-NL, corresponding to 151,595 identities and 5,178,420 images.

To ensure no data leakage and small information redundancy, we further de-duplicate the pretraining dataset from two aspects. First, we remove all potential duplicates from pretraining datasets that may appear in the evaluation datasets (detailed in Sec. 3.2) to enable a meaningful evaluation of generalization. Specifically, we first utilize the Difference Hash [77] to calculate the hash code of images in the evaluation datasets and pretraining datasets. Then, we delete the images in the pretraining datasets that have the same hash code as any image in the evaluation datasets. Second, some images come from some video-based datasets, e.g., AIST++ [41] and UppenAction [102], which contain large information redundancy between consecutive frames. In this case, we select only one image from every 8 consecutive frames to reduce redundancy.

3.2. Evaluation Scenarios and Protocols

Evaluation Scenarios. Our benchmark comprehensively quantifies the generalization ability of human-centric representation on 6 human-centric tasks from 19 datasets.

---

\(^1\) Full list of these 37 datasets are given in the supplementary.
Table 2. Summary of datasets for in-dataset evaluations, out-of-dataset evaluations, and unseen-task evaluations.

<table>
<thead>
<tr>
<th>Task</th>
<th>Datasets</th>
<th>in-dataset evaluations</th>
<th>out-of-dataset evaluations</th>
<th>unseen-task evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>RelD</td>
<td>Market1501 [109]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSMT [88]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CUHK03 [45]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SenseReID</td>
<td>[106]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pose</td>
<td>COCO [52]</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Human3.6M [33]</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIC [89]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parsing</td>
<td>MPII [2]</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Human3.6M [33]</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LIP [19]</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CIHP [18]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATR [50]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>PA-100K [56]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAPs2 [39]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PETA [10]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detection</td>
<td>CrowdHuman [70]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Caltech [12]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Counting</td>
<td>ShTech PartA [109]</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ShTech PartB [104]</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifically, we establish three evaluation scenarios for HumanBench: (1) in-dataset evaluation: we select 12 representative datasets whose training subsets are allocated to the pretrained dataset and evaluation subsets assigned to the evaluation dataset to evaluate the performance of a general pretrained model on diverse seen datasets (meaning similar data distribution for training and evaluation). (2) out-of-dataset evaluation: we select 5 datasets that do not appear in pretraining but belong to the seen task for evaluating the ability of the pretrained model on unseen datasets (meaning potentially different data distribution for training and evaluation). (3) unseen-task evaluation: we further add 2 datasets for crowd counting to evaluate the generalization ability to unseen tasks. More detailed distributions of these evaluation datasets are presented in Tab. 2.

**Evaluation Protocols.** For each evaluation scenario, we expect a good representation can generalize to specific human-centric tasks without updating the feature extractor or being a good starting point when adapted to any specific human-centric tasks by finetuning. Therefore, we present three evaluation protocols for the experiments presented in Sec. 5.

**Full Finetuning.** Full Finetuning evaluates the generalization ability when pretraining models serve as training starting points. In this case, we load the pretrained backbone and finetune all layers by supervision from downstream tasks.

**Head Finetuning.** Head Finetuning is very similar to linear probing [24] in self-supervised learning on natural image classification. It evaluates the generalization ability of pretrained models without updating. Therefore, we keep the pretrained backbone frozen and learn simple task heads for downstream datasets.

**Partial Finetuning.** Partial Finetuning is a setting between head finetuning and full finetuning [23], which finetunes the last several layers while freezing the others. This evaluation protocol can take advantage of both full finetuning and head finetuning, i.e., it can efficiently evaluate the opportunity of pursuing strong but non-linear features.

4. Methodology

We now introduce our proposed projector-assisted pretraining method (PATH) with hierarchical weight sharing. Our method is motivated by [86], which reveals that inserting an MLP projector before the objective function can significantly increase the generalization ability of supervised pretraining. To avoid task conflicts among various tasks, we improve this method by inserting task-specific projectors between the backbone and the head of every dataset and designing a new hierarchical weight-sharing strategy. Concretely, the projectors are very lightweight modules composed of an attention module and a gating module (Sec. 4.2). The hierarchical weight-sharing strategy enforces that parameters of backbone, projectors, and heads are shared among all datasets across different tasks, shared among datasets in the same task, and not shared, respectively (Sec. 4.1). As such, we expect the backbone to learn general representations of all human-centric tasks, the projector to attend to the task-specific features from the general representations, and the head to supervise the network optimization by the annotations of every dataset. To evaluate the generalization ability of the pretrained backbone, we discard the projectors and heads, using the backbone only.

4.1. Hierarchical Weight Sharing

We design a hierarchical weight-sharing strategy to reduce task conflicts among various annotations. Specifically, our model consists of three components: a single backbone shared by all datasets, $T$ task-specific projectors shared by all datasets in the same task, and $N$ dataset-specific heads that are not shared, where $N = N_1 + N_2 + ... + N_T$ is the number of datasets in the pretraining dataset and $N_t$ is the number of datasets in the $t$-th task.

**Backbone.** The backbone $F$ is implemented by a plain vision transformer [13] in the experiments. The parameters of the backbone are shared by all datasets regardless of tasks.

**Task-specific projector.** Each task-specific projector $P^t$ consists of sets of attention modules and gating modules, which link with the backbone $F$, where $t \leq T$ and $T$ is the number of tasks. Since the parameters of the task-specific projector are shared among the datasets with the same task, the attention modules in the task-specific network can be considered as selecting features from the shared backbone network, whilst the shared backbone network learns a compact global feature pool across all datasets.
Dataset-specific head. To tackle the possible data distribution shift in different datasets, we still preserve the dataset-specific head \( H^i_j \), whose parameters are not shared. Here, \( H^i_j \) is the \( j \)-th dataset in the \( i \)-th task.

Figure 2 shows a detailed visualization of our PATH. The detailed pipeline is described as follows.

**Step1:** Extract the general features of images in the pretraining dataset. Given an image \( x \) sampled from \( D^i_j \) which is the \( j \)-th dataset in the \( i \)-th task in the pretraining dataset, extract the intermediate and final feature maps \( F \) by the backbone, which will be fed into the projectors.

**Step2:** Attend the task-specific features by the task-specific projector (Sec. 4.2). Given the feature maps \( F \) from the backbone, we attend the task-specific features \( p = \mathcal{P}^i_l(F) \) by the \( t \)-th layer-specific projector.

**Step3:** Calculate the activations by dataset-specific heads, losses by the activations, and optimize the parameters of the backbone, the projector, and the head simultaneously by backward propagation (Sec. 4.3).

During the evaluation stage, we discard the projectors and evaluate the generalization ability of the backbone \( \mathcal{F} \) using the protocols in Sec. 3.2.

### 4.2. Design of Task-specific Projector

The task-specific projector is designed to attend to task-specific features from backbone outputs, by applying an alternating chain of the attention module and gating module to the features in the shared backbone. Given an image \( x \) sampled from the \( j \)-th dataset in the \( t \)-th task, \( i.e., D^i_j \) and its intermediate feature maps \( f_l \) in the \( l \)-th transformer block, we leverage a squeeze-and-excitation layer [31] to implement the channel attention and a self-attention module [80] to implement spatial attention to generate the attended feature maps \( z_l \). Mathematically, \( z_l = \mathcal{A}^l(E^l(f_l)) \), where \( \mathcal{A}^l \) and \( E^l \) respectively denote standard self-attention blocks [80] and squeeze-and-excitation blocks [31] for the \( t \)-th task. We will detail the structure of the Squeeze-and-excitation module and self-attention module in the supplementary materials.

In order to effectively aggregate features from different layers of the backbone, we design a gating module to dynamically aggregate features from different layers. Specifically, given the feature map \( z_l \) after the attention module and the gated feature maps \( p_{l-1} \) in the \((l-1)\)-th layer, the gating function aggregates features as follows:

\[
p_l = \mu_l z_l + (1 - \mu_l) p_{l-1},
\]

where \( p_1 = z_1 \), \( \mu_l = \sigma(\alpha_l / T) \) is a gate parameterized with a learnable zero-initialized scalar \( \alpha_l \) and temperature \( T(=0.1) \), and \( \sigma \) is the sigmoid function. By iteratively computing Eq. 1 from \( l = 1 \) to \( L \), we generate the final feature maps \( i.e., p = p_L \) for the dataset head.

### 4.3. Dataset-specific Head and Objective Functions

Dataset-specific heads aim at transforming task-specific features into activations for computing losses of every dataset. In general multi-dataset learning with \( N \) datasets, the features \( P_i \) after the projector of all images \( X_i \) and labels \( Y_i, i = 1, 2, ..., N \) in \( i \)-th dataset, the objective function is defined as \( \mathcal{L} = \sum_{i=1}^{N} \lambda_i L_i(Z_i, Y_i) \), where \( Z_i \) is
the activation generated by the dataset-specific head. This is the linear combination of dataset-specific losses \( \mathcal{L}_i \) with task weightings \( \lambda_i \). In this paper, we follow some basic head and loss function designs of all pretraining tasks we include. Specifically, we follow the head and loss function designs in ViT-Pose [92] for pose estimation, in TransReID [26] for person ReID, in Segformer [91] for human parsing, in Anchor Detr [87] for pedestrian detection, in Label2Label [44] for pedestrian attribute recognition, and in DR-VIC [21] for crowd counting. More details of these head and loss designs will be elaborated in the supplementary materials.

4.4. Technical Details

Replacing all Batchnorm with Layernorm in pose and parsing decoders. Generally, the original feature normalization method in pose estimation and human parsing tasks is batch normalization with CNN backbone, which renders the model to learn powerful feature distribution from the statistics of batch inputs when trained on a single domain. However, in HumanBench, each task has different datasets, which may have domain gaps, resulting in inaccurate normalization statistics when the dataset-specific head is fed with features from the task-share projector. To reduce the inaccurate statistics, we replace the Normalization method from BatchNorm [32] to LayerNorm [3] and experimentally find that it can improve feature representation.

Sharing Positional Embedding among All Datasets. In HumanBench, the input image size of different tasks varies largely, resulting in different numbers of patch embeddings and positional embeddings after projecting an image to patch embedding. As a result, different tasks cannot share positional embeddings when the model is trained in a distributed manner. To tackle this problem, we parameterize positional embeddings as \( 224 \times 224 \) in all tasks and interpolate its size according to the actual input image size of each dataset during the pretraining stage.

5. Experiment

5.1. Experimental Setup

The backbone used for experiments is the plain ViT-base. It has 12 transformer blocks with the dimension of patch embedding 768 and 12 attention heads. In the pre-train stage, each GPU is responsible for one dataset independently for training in a distributed manner. We use Adafactor [71] optimizer with base learning rate of \( 5 \times 10^{-4} \) and weight decay of 0.05. We linearly warmup the learning rate from \( 1 \times 10^{-7} \) to \( 5 \times 10^{-4} \) for the first 1500 iteration steps. Step learning rate decay of 0.5 is used in 50%, 75%, 95% iterations. For the ViT-Base encoder, we set a layer-wise learning rate decay of 0.75 for 12 transformer blocks and the model is trained for 80k iterations.

5.2. Experimental Results

As detailed in Sec. 3.2, we implement in-dataset evaluation, out-of-dataset evaluation, and unseen-task evaluation on HumanBench. Both in-dataset evaluation, out-of-dataset evaluation include 5 human-centric tasks, i.e., person ReID, pose estimation, human parsing, pedestrian attribute recognition, and pedestrian detection. The unseen downstream task which is not in the pretraining tasks, i.e., crowd counting, evaluates the generalization ability to unseen tasks. The compared methods are the state-of-the-art methods of each task and two popular pretraining models, i.e., MAE [23] and CLIP [64]. MAE is a newly proposed vision self-supervised pretraining method. Pretrained on ImageNet-1K, MAE achieves excellent results for many visual tasks. CLIP learns generic and transferable representations from a dataset of 400 million (image, text) pairs. We summarize our experimental results with 3 evaluation scenarios and 3 evaluation protocols in Tab. 3.

In-dataset Evaluation. In-dataset evaluation quantifies the ability of the pretraining method when it is evaluated on the data with similar data distribution and pretrained tasks. As shown in Tab. 3, compared with SoTA methods used in different papers for their specific tasks, our HumanBench with full finetuning achieves better performance on 8 datasets. Specifically, for human parsing, we improve the current state-of-the-art results by +2.5% mIOU, +1.1% mIOU and +1.2% mIOU on Human3.6M, LIP and CIHP, respectively. We also improve the person ReID by +4.9% mAP on CUHK03 datasets. We notice our results are lower than PASS [116] on Market1501 and MSMT, probably because PASS uses techniques, i.e., part models [76, 83], that are time-consuming (120 hours using 8 A100 GPUs) but specifically effective for ReID. Besides, we improve pose estimation by +3.0% AP, -1.2% MR \(-2(\downarrow)\) on AIC and Human3.6m, respectively. Furthermore, we improve pedestrian attribute recognition +1.5% mA and +0.2% mA on PA-100K and RAPv2 datasets, respectively.

To evaluate the generalization of different methods when all backbone parameters or most of the backbone parameters are frozen, we further evaluate our HumanBench with head finetuning and partial finetuning with 100% of the downstream data. We observe that our method with only head finetuning can be on par with and even surpasses the SoTAs in 12 seen datasets, such as -1.2% heavy occluded MR \(-2(\downarrow)\) +1.6% mIOU on Human3.6m pose estimation and human parsing tasks. Our HumanBench with partial finetuning performs better than full finetuning in 2 Pedestrian Attribute Recognition datasets (PA-100K and RapV2) of 12 seen datasets, possibly because these two datasets have fewer data.

We also use ViT-Large to verify the model scalability of our method PATH on HumanBench in Tab. 3. Results show that the results with a large backbone under partial
Table 3. Experimental results of our PATH and recent state-of-the-art methods (SoTA in the table) on 6 human-centric tasks. The results include 12 in-dataset evaluations, 5 out-of-dataset evaluations (columns w. gray) and 2 unseen task evaluations on the unseen counting task. Following the most commonly-used metrics, for human parsing tasks, we report pACC for ATR, mIoU for others. † indicates that the results are obtained with additional information, multitask learning, or stronger models. We highlight the best using ViT-Base and ViT-Large backbone, respectively. We also highlight these best results in red if they outperform SoTAs.

<table>
<thead>
<tr>
<th></th>
<th>Human Parsing</th>
<th>Person ReID</th>
<th>Pedestrian Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Human3.6M</td>
<td>LIP</td>
<td>CIHP</td>
</tr>
<tr>
<td>SoTA †</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAE</td>
<td>62.0</td>
<td>57.2</td>
<td>62.9</td>
</tr>
<tr>
<td>CLIP</td>
<td>58.2</td>
<td>53.4</td>
<td>61.7</td>
</tr>
<tr>
<td>ViT-B PATH (w/o FT)</td>
<td>63.9</td>
<td>56.3</td>
<td>63.9</td>
</tr>
<tr>
<td>ViT-B PATH (FT)</td>
<td>65.0</td>
<td>61.4</td>
<td>66.8</td>
</tr>
<tr>
<td>ViT-B PATH (Head FT)</td>
<td>64.1</td>
<td>59.9</td>
<td>63.3</td>
</tr>
<tr>
<td>ViT-B PATH (Partial FT)</td>
<td>63.7</td>
<td>60.0</td>
<td>63.1</td>
</tr>
<tr>
<td>ViT-L PATH (w/o FT)</td>
<td>65.0</td>
<td>62.9</td>
<td>67.1</td>
</tr>
<tr>
<td>ViT-L PATH (Partial FT)</td>
<td>66.2</td>
<td>62.6</td>
<td>67.5</td>
</tr>
</tbody>
</table>

finetuning can further achieve considerable gains over the best ViT-Base results, showing the promising scalability of our proposed pretraining method PATH on HumanBench.

**Out-of-dataset Evaluation.** To quantify the generalization ability of pretrained models on tasks with potentially different data distribution but the same task in the pretraining dataset, we implement out-of-dataset evaluations on 5 datasets, i.e., ATR, SenseReID, Caltech, MPII, PETA, one dataset for each pretraining task. As shown in Tab. 3, our pretraining method PATH performs better than previous methods in 4 of 5 unseen datasets and comparable in the remaining one. To be concrete, our method improves by +0.1% pACC, +4.4% Top1 accuracy, -0.5% heavy occluded MR$^{-2}$ (↓) and +2.7% mA on ATR (human parsing), SenseReID (person ReID), Caltech (pedestrian detection) and PETA datasets (pedestrian attribute recognition), respectively. These significant and consistent performance gains across different datasets verify the generalization ability of our pretrained model to tasks with potentially different data distributions. We also observe the results when we only finetune the last two layers are already on par or even better than the results by full finetuning. Especially, the results of SenseReID, Caltech and PETA by partial finetuning are better than that of full finetuning by +0.5% Top1 accuracy, -1.8% heavy occluded MR$^{-2}$ (↓) and +1.8% mA, showing the good generalization of our pretrained models and its easy deployment in the real world. Similar to the results in the out-of-dataset evaluation, partial finetuning performs better than full finetuning when the dataset is small in Caltech (4250 images) and PETA (9500 images). Therefore, partial finetuning can be a choice when the downstream dataset has few samples.

**Unseen-task Evaluations.** To evaluate the generalization ability to unseen tasks, we construct an unseen task evaluation protocol on all tasks. Interestingly, the results in the out-of-dataset evaluation, partial finetuning performs better than full finetuning when the dataset is small in Caltech (4250 images) and PETA (9500 images). Therefore, partial finetuning can be a choice when the downstream dataset has few samples.

**Comparison with MAE and CLIP Models.** We also compare our pretrained method with other popular pretrained models, i.e., MAE and CLIP, on our proposed HumanBench. In Tab. 3, we find our pretraining method performs considerably better than CLIP and MAE under the full finetuning evaluation protocol on all tasks. Interestingly, the
5.3. Ablation Study

Due to the significant computation cost with the large-scale full datasets, as summarized in Table 1 in Supplementary Material, we sample a subset containing a similar number of images as ImageNet-1K (∼1.28 M) from the full training set. We pretrain our models on this subset to verify the effectiveness of our designs by default in this section, and implement 4 in-dataset evaluations (PA-100K, LIP, Market1501, MSMT) and 3 out-of-dataset evaluations (Caltech, PETA, MPII) full dataset fine-tuning.

Effectiveness of hierarchical weight sharing. To verify the effectiveness of our hierarchical weight sharing, we adapt the three projector share strategies: (1) all shared projector (A); sharing the projector parameters across all the tasks and datasets (Table 4 (a)); (2) task-shared projector (T); sharing the projector parameters across all datasets in a single task, while maintaining an independent projector for each task (Table 4 (c)); and (3) specific projector (S); maintaining an independent projector for each dataset (Table 4 (b)). The results show that the task-shared projector is better than the other two. We speculate that the projector is the core component to map the general human-centric features to task-specific features. Therefore, all datasets in the same task are supposed to share the same mapping functionality while different tasks should operate differently due to the existing task gaps.

Effectiveness of shared positional embedding. Experiments (c) and (d) in Tab. 4 ablate whether positional embeddings are shared or not across the different tasks. The performance of CLIP\(^2\) is lower than MAE, which shows that more data on natural images and languages may not naturally benefit a variety of human-centric tasks, which empirically validates the importance of our HumanBench for further research on human-centric pretraining.

Comparison with self-supervised pretraining methods. As shown in Table 5, we first ablate the effectiveness of our dataset on downstream tasks. With almost the same number of images, the MAE pretrained on our subset for 800 epochs surpasses ImageNet pretrained MAE by +4.4%, which shows that by combining the diverse human-centric data across various human-centric tasks, our dataset is more suitable to learn human-centric features. Second, pre-trained on our subset, our supervised pretraining method, i.e. PATH, performs better than both MAE (800 epochs) and MOCOv3 (800 epochs) by +1.8%. Different from MAE and MOCOv3 which ignore the general properties of the human body and the potential association between the data in different tasks, our PATH is designed to capture the potential complementary knowledge between different tasks, leading to learning more general human-centric representations to improve the performance on various human-centric tasks.

6. Conclusion

In this paper, we investigate the opportunities and challenges in pretraining on various human-centric tasks, and propose a new HumanBench with the existing publicly available datasets. Based on HumanBench, we design a projector-assisted pretraining with hierarchical weight sharing (PATH) to learn human-centric information from annotations with different granularities. We hope our HumanBench can facilitate future works such as unified network structure design and multi-task.supervised/self-supervised learning methods on a broad variety of human-centric tasks.
References


[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16000–16009, 2022. 4, 6


[40] Jianshu Li, Jian Zhao, Yunchao Wei, Congyang Lang, Yidong Li, Terence Sim, Shuicheng Yan, and Jiashi Feng. Multiple-human parsing in the wild. arXiv preprint arXiv:1705.07206, 2017. 15


part pooling (and a strong convolutional baseline). In *Proceedings of the European conference on computer vision (ECCV)*, pages 480–496, 2018. 6


