Relative Pose Consistency for Semi-Supervised Head Pose Estimation

Felix Kuhnke[1], Sontje Ihler[2], Jörn Ostermann[1]
[1] Institut für Informationsverarbeitung, Leibniz University Hannover, Germany
[2] Institut für Mechatronische Systeme, Leibniz University Hannover, Germany
kuhnke@tnt.uni-hannover.de

***Abstract*— Human head pose estimation from images plays a vital role in applications like driver assistance systems and human behavior analysis. Head pose estimation networks are typically trained in a supervised manner. Unfortunately, manual/sensor-based annotations of head poses are prone to errors. A solution is supervised training on synthetic training data generated from 3D face models which can provide an infinite amount of perfect labels. However, computer generated face images only provide an approximation of real-world images which results in a domain gap between training and application domain. To date, domain adaptation is rarely addressed in current work on head pose estimation. In this work we propose relative pose consistency, a semi-supervised learning strategy for head pose estimation based on consistency regularization. It allows simultaneous learning on labeled synthetic data and unlabeled real-world data to overcome the domain gap, while keeping the advantages of synthetic data. Consistency regularization enforces consistent network predictions under random image augmentations. We address pose-preserving and pose-altering augmentations. Naturally, pose-altering augmentations cannot be used on unlabeled data. We therefore propose a strategy to exploit the relative pose introduced by pose-altering augmentations between augmented image pairs. This allows the network to benefit from relative pose labels during training on the unlabeled, real-world images. We evaluate our approach on a widely used benchmark (Biwi Kinect Head Pose) and outperform domain-adaptation SOTA. We are the first to present a consistency regularization framework for head pose estimation. Our experiments show that our approach improves head pose estimation accuracy for real-world images despite using only labels from synthetic images.**

I. INTRODUCTION

Head pose estimation (HPE) describes the problem of predicting the orientation of the human head. It is a vital part of many vision algorithms for facial analysis. HPE can be used for automatic assessment of the focus of attention, e.g. in driver assistance systems [4] or for human behavior analysis. It is also the starting point for many gaze estimation methods [42]. Furthermore, HPE is closely related to face alignment and is part of many systems for face recognition.

Due to the many applications, there has been a lot of progress in this field of research, especially through deep learning methods. Nevertheless, collecting the required training data is still a challenging task for several reasons. Manual annotation is a problem, because humans cannot accurately annotate a 3D head pose from a 2D image. This has led to the creation of head pose datasets using devices like depth sensors and 3D head scans [8], [4], or special tracking equipment attached to the head [24], [32], [33]. However,

978-1-6654-3176-7/21/$31.00 ©2021 IEEE

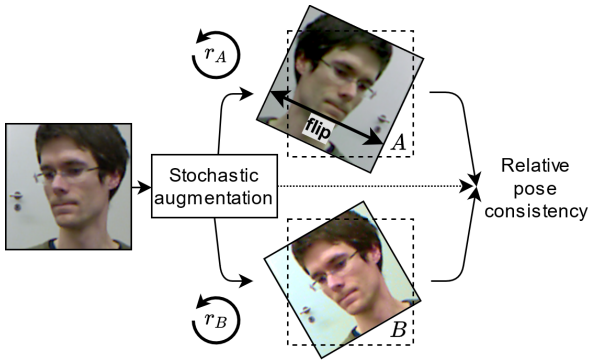

Fig. 1. From an image with unknown head pose (picture from [8]) two different augmented versions A and B are created. Augmentations can be color distortions, blurring, etc., but also pose-altering transforms like rotations (r_A, r_B) and flipping. In addition to a supervised loss, our method allows to train a network on unlabeled data with relative pose consistency between A and B. Relative pose consistency provides an unsupervised loss to make consistent predictions under different augmentations, but also to make predictions that comply with a relative pose.

with these recording setups, it is cumbersome and costly to reach a high diversity in subjects, environments and poses.

A solution is to use synthetic (rendered, computer generated) face images to provide inexpensive and virtually unlimited quantities of perfectly labeled data. Several methods train on synthetic [23], [20], [19], [12], [39], [16] or synthetically extended (warped) images (e.g. [31], [41] on 300W-LP dataset [44]). Unfortunately, learning-based approaches trained only on synthetic data (source domain) tend to perform poorly on real-world data (target domain) compared to methods trained on real-world data. This can be explained by the difference between domains (domain gap). However, only [16] addressed this issue for HPE by using a method for domain adaptation (DA).

Similar to [16], our goal is to improve the performance of HPE on real-world data using only labels from a synthetic dataset in combination with an unlabeled real-world dataset. In [16] an adversarial training approach based on domain adversarial neural networks [10] is used to force the extraction of domain-invariant features. In contrast, we propose to tackle the problem using consistency regularization, which has been successfully used for domain adaptation [9].

Consistency regularization is a semi-supervised learning (SSL) technique. Semi-supervised learning utilizes labeled and unlabeled data simultaneously during training. Consistency regularization forces network outputs for the same

input under different perturbations to be consistent. For visual tasks, these perturbations are typically implemented as various image augmentations, e.g. spatial transforms. However, head poses are not invariant to spatial image transforms, like flipping and rotation. If the ground truth pose is known, the pose label can be adjusted, however, the ground truth pose is unknown for our target-domain data. In this work, we therefore propose to take advantage of relative pose.

The relative pose, which we store in a relative pose label, is the pose difference between two realizations of the same input (see Fig. 1 for an example). Recalling that training with consistency regularization requires different realizations of the same input, we implement relative pose label in a consistency training framework (see Figure. 3 for an overview of our method). This extends the consistency supervision from static augmentations to relative pose labels. As a consequence, the network is trained not only to make consistent predictions, but also predictions that comply with the relative poses. Our consistency-enforcing method does not require absolute pose information and can therefore be used with unlabeled data samples in semi-supervised or domain-adaption scenarios. We show the effectiveness of our approach on the popular BIWI Kinect Head Pose estimation benchmark [8]. Our approach can also be adapted to other pose estimation problems.

Our main contributions are as follows:

- We show, for the first time that consistency regularization can be used for pose regression problems.
- We propose relative pose consistency, a novel extension to consistency regularization.
- We achieve state-of-the-art results for a challenging domain-adaptation problem on a head pose benchmark.

II. Related Work

A. Head Pose Estimation

In recent years, traditional approaches based on facial landmarks and 3D face models are mostly superseded by deep learning methods [31]. In addition to images, different modalities such as depth images [8] or temporal information [12] can be used. In this review we will focus only on deep-learning methods for HPE from a single RGB image.

The first convolutional neural networks (CNN) to directly regress the head pose from an image are presented by Anh et al. [1] and Patacchiola and Cangelosia [28]. Recent works propose variations of loss functions and network architectures. Ruiz et al. [31] combine a regression loss with binned pose classification, by assigning continuous pose to discrete pose categories (bins). Shao et al. [34] use a similar combined loss, but also evaluate the effect of adjusting the margin around the face image that is fed into a CNN. Similarly, Lathuilière et al. [21] evaluate various factors of deep regression, like hyperparameter selection or image preprocessing, in the context of head pose estimation. Wang et al. [39] present a coarse-to-fine approach, where head pose is coarsely classified in bins, and later refined by regression. An attention based network structure for HPE is proposed by Yang et al. [41]. Their goal is to extract a set of representative features by learning a fine-grained structure mapping before a feature aggregation step. Zhou et al. [43] extend the work of [31] to full-range HPE by proposing a wrapped loss that allows training with the full range of yaw angles (-180°,180°]. They further show that a small model, EfficientNet-B0 [36], can reach SOTA HPE performance. In contrast Gu et al. [12] present an approach for temporal prediction of facial features. They propose to use a recurrent neural network (RNN) on top of a VGG16 network [35] for joint estimation and tracking of head pose in videos. The above methods can be seen as orthogonal to our approach, because we are not trying to improve supervised performance with new losses or network architectures for HPE. For simplicity and comparability we focus on Mean Squared Error (MSE) loss and ResNet [14] network architecture. Nevertheless, our method can be applied to other loss functions or network architectures as well.

Another approach to HPE is multi-task learning [29], [17], [5], [30], [38]. In this setting, multiple tasks like HPE, landmark detection, age estimation, visibility, etc., are solved simultaneously. A benefit of multiple tasks is that multiple data sources can be used for training, which considerably increases the amount of training data. In contrast, our method does not focus on sharing knowledge between related tasks, but transferring knowledge between domains.

An interesting unsupervised approach is presented by Mustikovela et al. [25]. In their work, a viewpoint estimation network is trained purely via self-supervision with an analysis-by-synthesis framework using a network similar to HoloGAN [26]. Similar to our work, they enforce flip consistency by applying a flip consistency loss. In contrast, their loss forces synthesized images from a flipped latent code to be consistent.

Lastly, it is a common approach to use synthetic face datasets from 3D models for HPE [23], [19], [39], [20], [12], [16]. This has the advantage of learning from a high amount of diverse images with perfect labels. To date, [12] and [19] are publicly available datasets. Except for Kuhnke and Ostermann [16], the related works do not explore any domain adaptation or semi-supervised techniques.

[16] improve HPE for an unlabeled target dataset by enforcing a network to extract domain-invariant features. They use synthetic face images from [12] as labeled source domain and real-world images as unlabeled target domain. To account for an only partially-shared label space, they apply a weighted resampling of the source domain during training to filter out dissimilar samples. In this work, we tackle the same problem but choose a completely different approach. Our approach does not need an additional discriminator network with adversarial training. Furthermore, our approach does not require to resample the source data.

B. Consistency Regularization

Consistency-enforcing methods provide state-of-the-art performance for semi-supervised learning. During training, consistent network predictions for unlabeled data under input

and network perturbations are enforced. Although one can find many terms and variants like self-ensembling, consistency regularization, self-training, temporal ensembling, or pseudo-labels, the core principle of enforcing consistent outputs is similar. Consistency-enforcing methods have also been successfully applied to domain adaption scenarios, where the unlabeled data is from another domain. While first used as semi-supervised methods, these principles are now popular for unsupervised pre-training of neural networks and paved the way for modern contrastive (self-supervised) methods like SimCLR [6], MoCo [13] and BYOL [11].

Laine and Aila [18] proposed two self-ensembling methods, Π-Model and temporal ensembling. Both methods enforce consistent network predictions for the same input under different stochastic input augmentations and network perturbations. In this case, dropout was used to provide network perturbations. The Π-Model randomly augments the same input twice during an iteration and forces consistent predictions. In contrast to the Π-Model, temporal ensembling forces network predictions over multiple previous training epochs to be consistent to the current prediction. Self-training and training with pseudo-labels, e.g. [40], [22], can be seen as a variant of temporal-ensembling. The Mean Teacher method by Tervainen et al. [37] adapted this idea but instead of reusing previous predictions, they added a teacher network that is an average of previous network weights. The teacher network predictions and the current model (named student) predictions are forced to be consistent. French et al. [9] applies the Mean Teacher method to domain adaptation and proposes modifications to improve DA performance.

To our knowledge consistency regularization has not been applied to head pose estimation, or any pose estimation task before.

III. Method

Semi-supervised learning is typically used to learn from a large dataset which is only partially labeled. We take up this idea for domain adaptation to learn from labeled synthetic images (source domain) and unlabeled real-world images (target domain). On the one hand, synthetic data provides perfect labels for a wide variety of poses. On the other, it only provides an approximation of real-world image features. Real data provides real-world features but lacks annotation quantity and quality. Combining them in a training scheme, where both datasets can be used simultaneously, is a promising way to improve performance on real-world images.

We will first introduce the required notations and baseline supervised learning. Then, we will describe the consistency regularization framework. Subsequently, we will describe the concept of relative pose labels and how these are embedded into the training framework. Finally, we discuss how we avoid degenerate solutions with consistency regularization.

In a semi-supervised or domain-adaptation scenario, data is available from the labeled source domain $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where n_s is the number of data samples $x_i^s \in X_s$ and associated labels $y_i^s \in Y_s$. For head pose estimation, x is an image of a head and y is a vector of the three corresponding Euler angles of the head. We are interested in utilizing the unlabeled target data $\mathcal{D}_t = \{(x_i^t)\}_{i=1}^{n_t}$, which includes only samples but no labels.

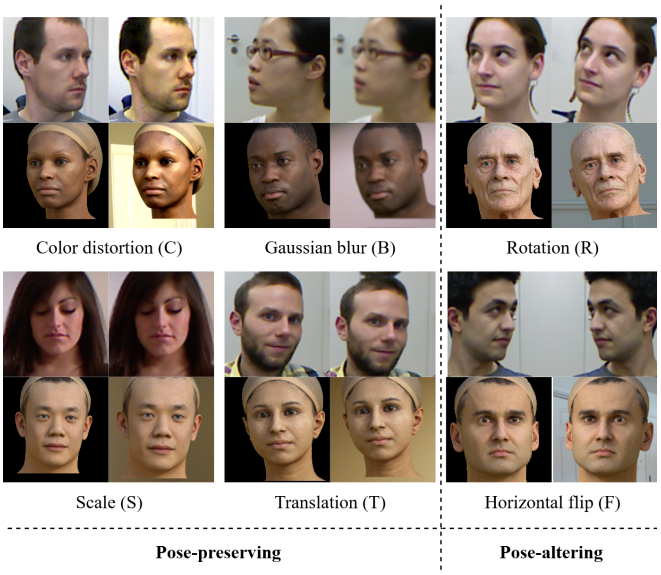

Fig. 2. Illustrations of the studied image augmentations. Each augmentation transforms the input image with random transformation parameters. The left images in each square show the inputs and the right images randomly transformed outputs. For synthetic images, a random background is added. Top row images from [8] and bottom row images from [12].

A network f can be trained using the source data (X_s and Y_s) and a supervised loss. For head pose estimation the supervised loss is typically the Mean Squared Error

$$\ell_{\text{mse}}(\hat{y}, y) = \|\hat{y} - y\|^2, \tag{1}$$

between the predicted Euler angles $\hat{y} = f(x)$, and the ground truth angles.

A. Consistency Regularization Framework

Stochastic input perturbations are a central aspect of consistency-based models. In practice standard image augmentations like blurring, translation and scaling (usually implemented as random cropping), horizontal flipping, rotation, and color distortions provide appropriate image perturbations (see Fig. 2).

Given a sample x we create two randomly perturbed (augmented) inputs x' and x'' which are fed into the network f to produce predictions $f(x')$ and $f(x'')$.

A consistency loss $\mathcal{L}_{\text{cons}}$ enforces that both predictions are similar. This consistency loss is typically the Mean Squared Error or KL divergence [3]. We formulate our total loss

$$\mathcal{L}_{\text{total}} = \underbrace{\sum_{(x,y)\in\mathcal{D}_s} \ell_{\text{mse}}(f(x'), y)}_{L_{\text{super}}} + \lambda \underbrace{\sum_{x\in\mathcal{D}_t} \ell_{\text{mse}}(f(x'), f(x''))}_{L_{\text{cons}}}, \tag{2}$$

with λ controlling the relative effect of the consistency term in the overall loss.

The same stochastic perturbations are applied to both source and target images. Note that in SSL the consistency

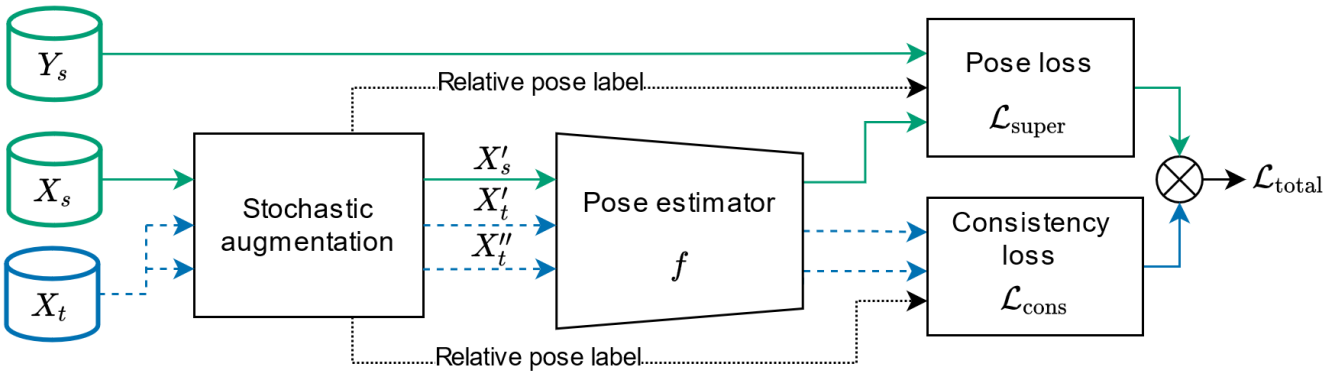

Fig. 3. Proposed framework for relative pose consistency regularized head pose estimation. Labeled data (in green) from the source domain and unlabeled data (in blue) from the target domain can both be used in a semi-supervised fashion. Input images X_s and X_t are perturbed by stochastic augmentations. The stochastic augmentation module can also change the pose of the input images by rotation and flipping. This information is stored in the relative pose label. Source data follows the supervised path (green) to train the pose estimator f. Target data is copied before the stochastic augmentation module which creates two different augmented versions of the target input. Note that even though the ground truth pose is unknown for X_t the relative pose between the augmented versions $X_t{}'$ and $X_t{}''$ can differ and is stored in a relative pose label. The relative pose label and predictions are fed into the consistency loss. The consistency loss provides supervision from consistency and relative pose labels. f is trained jointly on both losses.

loss is typically applied to samples from both $\mathcal{D}_s$ and $\mathcal{D}_t$ [18], [37], [3]. Following [9], who use consistency regularization for domain adaptation, we apply the consistency loss only to samples from the target domain $\mathcal{D}_t$. Our framework is shown in Figure 3.

Unfortunately, *flipping and rotation* will *change the ground truth label* of a source-domain sample *and* produce target-domain inputs that *break the consistency assumption* that x' and x'' share the same label. We therefore need to distinguish between pose-preserving and pose-altering augmentations and need to redefine our loss functions for pose-altering augmentations. As shown in Fig. 2, **pose-preserving augmentations** are random color distortion, blurring, translation, and scaling and **pose-altering augmentations** are flipping and rotation. The required changes for pose-altering augmentations will be described in the next section.

B. Relative Pose Consistency

Pose-altering augmentations change the head pose. Knowing the spatial transformation and the true pose, an augmented image can be relabeled. However, this is not possible if the true pose is unknown. We create a new consistency loss based on the relative pose between augmented samples to benefit from pose-altering augmentations on our real-world target data.

We will first give a short recap on pose representation and then provide the interdependence of image rotation and flipping to the orientation change of the head pose and required adaptions to the loss functions. Both augmentations require[1] that the pose is stored in Euler angles (Tait–Bryan angles) that describe intrinsic rotations around Z-Y'-X". These are known as: roll, yaw, and pitch. This means that the rotation is performed by three successive rotations around the Z, Y' and X" axis. Recall that for intrinsic rotations the first rotation around Z will create a new coordinate system from which Y' will be used for the second rotation and so on. For this representation a rotation around Z can be carried out independently from Y' and X" rotations. That means that any image rotation will result in an additive rotation term to the roll label.

Augmenting an image with (unknown) **rotation** r with two random rotations r_A and r_B would result in images A and B with rotations $r + r_A$ and $r + r_B$, respectively. One can easily see that the difference in rotation between the two augmented images is $r_B - r_A$ which is the relative pose difference between the images. To account for this difference we can change the consistency loss for the roll angle to:

$$\ell_{\text{mse}}(f(A)_{\text{roll}} + (r_B - r_A), f(B)_{\text{roll}}), \tag{3}$$

where $f(A)_{\text{roll}}$ and $f(B)_{\text{roll}}$ describe the predicted rotations. To use rotation augmentations for the source domain, one can simply replace r_B with 0 and $f(B)$ with the true rotation label r.

Flipping is performed by negating the yaw and roll angles of the flipped image. For the consistency loss, we can negate the yaw and roll angles of the predictions. A full example, showing all the angles, with A being flipped and random rotations would result in

$$\ell_{\text{mse}}\left(\begin{bmatrix} f(A)_{\text{pitch}} \\ f(A)_{\text{yaw}} \\ f(A)_{\text{roll}} \end{bmatrix} \odot \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r_B - r_A \end{bmatrix}, f(B)\right), \tag{4}$$

where $\odot$ is the element-wise product. This complete example is also illustrated in Figure 1. Cases where B, or A and B are flipped are handled with negating B's, or A's and B's yaw and roll angles, respectively. Note that the information provided by a relative pose only influences yaw and roll angles, as the pitch angle is untouched by rotation and flipping.

[1]For simplicity we describe our method for Z-Y'-X" rotations, but other representations such as extrinsic rotations around X-Y-Z will also work.

C. Avoiding degenerate solutions

Several works report difficulties when training with consistency regularization. In contrast to previous works, we apply consistency regularization to a regression problem and therefore use a different loss combination. For this reason, instead of class logits, we regularize the predicted pose angles. In the following we will address these difficulties and how we dealt with them.

The first difficulty is the selection of λ. [18] found that the network can get easily stuck in a degenerate solution if the unsupervised loss component ($\mathcal{L}_{\text{cons}}$) is too high in the beginning of the training. As a solution, they ramp-up λ from 0 to 1 during training. The same procedure was also adopted in [37]. In contrast, French et al. [9] replaced the ramp-up with a confidence threshold. They utilized the predicted class activations as probabilities and the loss of all samples with activations below the threshold is weighted to 0.

In our case we found that high λ values usually yield degenerate solutions, regardless of ramp-up or not. Our explanation is that in order the minimize the consistency loss the network can learn to output only a constant. However, we found a good indication on how to set λ comes from the supervised loss. As a simple rule, the regularization feedback should not be stronger than the supervised loss. Preliminary experiments showed that consistency training is quite robust and λ values in the range [0.1, 0.4] converge to similar performing networks. For $\lambda > 0.5$ the consistency loss became larger than the supervised loss and the overall performance decreased for both, source and target data.

Another issue with consistency regularization arises if the labels of the source and target domain do not come from the same underlying distribution [27]. This is ignored in many works, because it is assumed that the unlabeled data contains the same class distribution as the labeled data. As described by [16], this assumption usually does not hold for regressions tasks like HPE. For classification, [9] introduced a class balance loss term that forces the network's mean class predictions to be uniform. This helped to avoid a degeneration to the most dominant class.

Following this approach, we introduce a weighted relative consistency regularization for HPE. To enforce a more evenly distributed feedback of the consistency loss, we re-weight the consistency loss based on the pose predictions. Poses that are found often in a batch should be weighted down, whereas rarely appearing poses should be weighted up. In most natural image collections of faces, the poses are usually distributed around the pose that is facing the camera. The same holds true for most head pose datasets. Therefore, assuming a normal distribution of poses in a batch, we formulate our weighting:

$$w_p = 1 - e^{\left(\frac{p-\mu_P}{\sigma_P}\right)^2}, \tag{5}$$

where w_p is the weight given to a pose angle p (pitch, yaw, roll) and μ_P and σ_P are the mean and standard deviation of all $p \in P$ in a batch. In particular, we apply this weight to all angles independently. To keep λ constant between experiments, we rescale w_p with

$$\frac{Batchsize}{\sum w_p}, \tag{6}$$

so that the overall weight in a batch sums to one. We compare the results for weighted and unweighted predictions in our experiments IV.

Both λ and re-weighting are associated with the same underlying problem: an effective setting of the regularization strength. Although we have made two proposals, we think that uncertainty or curriculum approaches like [7] are paths worth looking into for future improvements.

IV. Experiments

In the following, we will analyze the performance of our method. We conduct three series of experiments.

Supervised only will serve as baseline. These experiments are trained only with the supervised loss on a synthetic dataset. These experiments only differ in the use of different augmentation combinations. In comparison to former results, these experiments can be seen as inter-domain approaches, as training and testing is done on different datasets. **Consistency regularization** uses our proposed consistency framework. Again different augmentations are evaluated. **Weighted consistency regularization** includes the proposed weighting of angles during training. In comparison to former results, both consistency experiments can be seen as domain-adaption approaches, like PADACO [16].

We measure the mean absolute error for every angle and the overall MAE (mean average error) of them. We report the mean and standard deviation of these values over 10 runs using different random seeds. Please note that this is not the standard deviation of the pose errors, but the standard deviation of the mean errors over all runs.

A. Data

To validate our method we use revised datasets SynHead++ and Biwi+ proposed by [16]. These datasets are extensions of the popular face pose datasets Biwi Kinect Head Pose Database (Biwi) [8] and NVIDIA Synthetic Head Dataset (SynHead) [12]. For both datasets, [16] provide labels in Z-Y'-X"-angle representation and face bounding boxes. SynHead was artificially extended to include more poses, so that SynHead++ is a superset of Biwi+ in regard to pose labels. Here, we give a brief overview of the datasets.

Biwi+ is used as real-world, target-domain dataset. It contains 24 sequences of 20 different subjects recorded with a kinect sensor. **SynHead++** is used as synthetic, source-domain dataset. It contains images of 10 different rendered 3D head models. The total number of images is 15677 for Biwi+ and 653910 for SynHead++. All images are cropped to the given bounding boxes and scaled to 224 x 224 pixels. Exemplary images and illustrative augmentations are shown in Figure 2.

TABLE I

AUGMENTATION PARAMETERS. TRANSLATION PARAMETERS ARE GIVEN RELATIVE TO THE IMAGE SIZE AND APPLIED INDEPENDENTLY FOR X AND Y TRANSLATIONS. VALUES IN RANGES ARE SAMPLED UNIFORMLY.

	Augmentation	Parameter	Probability
C	Color distortion	brightness = 0.4, contrast = 0.4 saturation = 0.4, hue = 0.1	0.8
B	Gaussian blur	$\sigma \in [0.2, 2]$	0.5
S	Scale	[0.9, 1.1]	1.0
T	Translation	[-0.1, 0.1]	1.0
R	Rotation	[-20°, 20°]	1.0
F	Flip		0.5

B. Implementation Details

For all our experiments, the pose estimator f is ResNet18 as provided by PyTorch [24] with last linear layer being replaced by a new linear layer with 512 inputs and 3 outputs for Euler angle estimation. This is consistent to [16], the most relevant work to ours.

We use different augmentations schemes throughout the experiments with parameters provided in Table I. The parameters of all augmentations are fixed. We used the code[2] and parameters from [13] for color distortions and Gaussian blur. Similar to [9], we process minibatches of source and target data sequentially, to forces batch normalization to use different normalization statistics for each domain during training. For all experiments, we use stochastic gradient descent with momentum 0.9, Nesterov, a batch size of 84, and a learning rate set to 0.01.

For our "supervised only" baselines f is initialized with the default PyTorch pretrained ResNet18. The learning rate is ramped-up to warm start the optimization. During training λ is set to 0. The baselines are trained for 35000 iterations which is equivalent to $\approx$5 epochs of source data.

For all consistency regularization experiments we fine tune a baseline model. To make the comparisons fair for all runs, we select the same supervised only baseline trained with all augmentations (full). The selected model performs similar to the average performance of models in this setting. All models are fine-tuned for 16000 iterations which is equivalent to $\approx$86 epochs of target data. For the consistency regularization experiments, λ is ramped-up to 0.2, to avoid deterioration from too strong regularization. For all experiments the performance at the end of training is reported, i.e. no early stopping is used. It is important to note that performance for Biwi+ is reported without any augmentations.

C. Results and Discussion

Table II shows the results of our experiments and reports results of related work. The experimental settings intra domain, inter domain and self-supervised show related but not straightforward comparable results. These works train with real-world images and typically focus on improving head pose estimation by improved network structures or supervised loss functions. In contrast, our main goal is to learn from synthetic images and improve performance using unlabeled real-world images. This is similar to the partial DA setting of [16]. However, we think it can be valuable to discuss our results in a broader context of related work.

Intra domain. These numbers show the performance for methods trained and evaluated on different splits of the *same dataset*. Compared to our results and other experimental settings, higher performance is likely explained by having no domain gap between train and test set. The performance is also dependent on the train/test split.

Inter domain. Sometimes called cross-domain or *cross-dataset* evaluation, this section shows results where the training set and test set are taken from different datasets. Most commonly, 300W-LP [44], a dataset created from real-world images is used for training. As 300W-LP uses real-images, the domain gap to Biwi is presumably small, compared to using synthetic images. The effect (better performance) is visible if we compare the 300W-LP results to our baselines (supervised only). Only [39] use rendered synthetic images but also a part of the Biwi dataset to train a HPE model. Our proposed method, weighted relative consistency regularization (RCRw), performs similar to most works in this setting, despite using only rendered synthetic images for pose supervision. Only WHENNet-V [43]) considerably outperforms RCRw. Compared to WHENNet, additional data from the (real-world) Panoptic Studio dataset [15] was added to better match the pose distribution of Biwi. Notably, we outperform all works for roll error.

Self-Supervised shows that learning head pose can even be accomplished completely self-supervised. No pose labels are used during training, instead, a linear regressor that maps network outputs to pose labels is trained afterwards on 100 random test set samples. While results are not on par with recent works, it suggests that there is potential in self-supervised pose estimation. Training with our proposed relative pose labels can be seen as a self-supervised approach.

Partial DA shows the results for partial domain adaptation. Like our approach, this setting uses the pose labels of synthetic face images and unlabeled target-domain images (Biwi+) during training. Our proposed method RCRw with full augmentations, slightly outperforms [16]. For roll error, the improvement to [16] is over one degree. Although we get worse pitch performance, overall, we are better on average.

Supervised only reveals the effects of augmentations during "supervised only" training on synthetic data. It is quite notable that augmentations help to improve target performance. Color, scale, translation and blur augmentations create images that might look more similar to the test set. In addition to these augmentations, flipping or using flipping and rotations slightly improves the results. However, augmentations are not sufficient to reach the performance of related work on the Biwi dataset. These results demonstrate two of our key assumptions. First, training on a synthetic image dataset does not provide automatically good results for a real-world image dataset. Secondly, the tested augmentations alone are not sufficient to force the network to learn

[2]https://github.com/facebookresearch/moco

TABLE II

HEAD POSE ESTIMATION RESULTS FOR EXPERIMENTS TESTED ON VARIANTS OF THE BIWI DATASET [8]. VARIANTS: * RANDOM SPLIT (86%/14%[1], 80%/20%[2]), † SEQUENCE SPLIT (16/8[1],21/3[2]), × COMPLETE, PROCESSED BY THE RESPECTIVE AUTHORS, + COMPLETE, PROCESSED BY [16]. EXPERIMENTAL RESULTS ARE GROUPED IN BLOCKS DESCRIBING THE USE OF DATA DURING TRAINING AND TESTING. WE REPORT MEAN AND STANDARD DEVIATION OF THE AVERAGE ABSOLUTE ANGULAR ERRORS IN DEGREE AND MEAN AVERAGE ERROR (MAE) OVER ALL ANGLES FOR 10 TRAINING RUNS. BEST RESULTS IN BOLD.

AUGMENTATIONS FOR EXPERIMENTS: ROTATION (R), FLIP (F), COLOR DISTORTION (C), SCALING (S), TRANSLATION (T), GAUSSIAN BLUR (B).

Experiment	Method	Network	Training set	Test set	MAE	Pitch	Yaw	Roll
Intra domain	Anh [1]	Custom	Biwi*[1]	Biwi*[1]	2.93	3.4	2.8	2.6
	Ruiz (Hopenet) [31]	ResNet50	Biwi†[1]	Biwi†[1]	3.23	3.39	3.29	3.00
	Gu [12]	VGG16	Biwi†[1]	Biwi†[1]	3.66	4.03	3.91	3.03
	Lathuilière [20]	VGG16	Biwi†[2]	Biwi†[2]	3.62	4.68	3.12	3.07
	Yang (FSA) [41]	Custom	Biwi†[1]	Biwi†[1]	3.60	4.29	2.89	3.60
Inter domain	Ruiz (Hopenet) [31]	ResNet50	300W-LP	Biwi×	4.90	6.61	4.81	3.27
	Yang (FSA) [41]	Custom	300W-LP	Biwi×	4.00	4.96	4.27	2.76
	Zhou (WHENet) [43]	Eff.Net-B0	300W-LP	Biwi×	3.99	4.39	3.99	3.06
	Zhou (WHENet-V) [43]	Eff.Net-B0	300W-LP+[15]	Biwi×	3.48	4.10	3.48	2.73
	Wang [39]	Custom	[39]+Biwi*[2]	Biwi*[2]	4.84	5.48	4.76	4.29
Self-Supervised	Mustikovela (SSV) [25]	Custom	300W-LP	Biwi†[1]	6.8	9.4	6.9	4.2
	Mustikovela (SSV) [25]		300W-LP+Biwi†[1]	Biwi†[1]	5.8	8.5	4.9	4.2
Partial DA	Kuhnke (PADACO) [16]	ResNet18	SynHead++	Biwi+	4.13	4.51	4.11	3.78
Supervised only	Baseline (no aug.)	ResNet18	SynHead++	Biwi+	5.36±.28	5.99±.35	5.60±.60	4.42±.35
	Baseline C·S·T·B				4.72±.11	5.71±.18	4.70±.15	3.75±.08
	Baseline C·S·T·B·F				**4.65**±.09	**5.65**±.16	**4.64**±.17	3.65±.09
	Baseline C·S·T·B·R				4.78±.13	5.71±.20	4.81±.24	3.81±.13
	Baseline C·S·T·B·R·F				**4.65**±.08	5.68±.16	4.69±.16	**3.59**±.06
Consistency Regularization (ours)	CR C·S·T·B (no rot/flip)	ResNet18	SynHead++	Biwi+	4.27±.04	5.67±.06	4.00±.05	3.13±.03
	RCR C·S·T·B·R (no flip)				**4.14**±.04	**5.47**±.09	4.13±.05	2.84±.03
	RCR C·S·T·B·R·F (full)				4.15±.03	5.87±.07	**3.80**±.04	**2.78**±.02
Weighted Cons. Reg. (ours)	CRw C·S·T·B (no rot/flip)	ResNet18	SynHead++	Biwi+	4.11±.03	5.24±.07	4.00±.05	3.08±.02
	RCRw C·S·T·B·R (no flip)				4.03±.04	**5.22**±.07	4.06±.04	2.80±.03
	RCRw C·S·T·B·R·F (full)				**4.01**±.03	5.54±.13	**3.78**±.10	**2.71**±.03

features that generalize well to real-world images.

Consistency regularization. Our proposed consistency framework improves the average performance compared to baselines. Using only pose-preserving augmentations for consistency regularization (CR), already improves the results to baselines. Best performance for yaw and roll is gained when using the full augmentation scheme. Surprisingly, there is no gain or even slight deterioration for pitch.

Weighted consistency regularization (RCRw). The proposed weighting scheme further improves the results to the baselines and the non-weighted results. Compared to the unweighted RCR runs, the weighting produces overall higher performance and even improved pitch estimation. The full augmentation setting produced the lowest roll error, even compared to recent works [41], [43] trained on 300W-LP and even Biwi splits.

Looking at Table II, pitch error is higher than roll or yaw for all reported results. For our experiments, pitch estimation has gained the least from our method. Some runs without weighting even show some degeneration. We suspect that pitch estimation seems to be a harder problem. In our experiments, pitch is consistently underpredicted compared to ground truth pitch. An explanation could be, that there is an offset for pitch between the pose origins of Biwi+ and SynHead++. Most importantly, the pitch angle can not benefit from the relative pose labels, as the pitch angle is constant for all our augmentations. In contrast, roll benefits the most from our relative pose labels. Probably for this reason, we outperform almost all other work in terms of roll error.

V. CONCLUSIONS AND FUTURE WORK

We present relative pose consistency, a new approach to improve deep head pose estimation performance in domain-adaptation scenarios. In this scenario labels are only available for synthetic images and testing is performed on real-world images. The method allows pose-altering augmentations, rotation and horizontal flipping, to be incorporated into a consistency regularization framework. In addition, we present a weighting scheme to improve performance. Compared to previous work, our approach performs similar or even better, despite using only labels from synthetic images. However, there is still a gap to methods trained with real-world image datasets that are similar to the target domain.

In future work, our framework could also be combined with other methods, e.g. the domain-adversarial approach of [16]. However, this is non-trivial, as adding a domain discriminator and pose resampling from [16] to our framework would raise additional questions about how augmentations and data streams are handled. Furthermore, it would be interesting to see a deeper analysis why pitch estimation performs comparatively poor. Lastly, the concept of relative pose consistency could be applied to other pose estimation tasks such as hand or body pose estimation, or scale and translation estimation methods like [2].

REFERENCES

[1] B. Ahn, J. Park, and I. S. Kweon. Real-time head orientation from a monocular camera using deep neural network. In *Asian Conf. on Computer Vision*, pages 82–96. Springer, 2014.

[2] V. Albiero, X. Chen, X. Yin, G. Pang, and T. Hassner. img2pose: Face alignment and detection via 6dof, face pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, virtual, June 19-25*, pages 7617–7627. IEEE/CVF, 2021.

[3] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *7th International Conf. on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9*, 2019.

[4] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4661–4670, 2017.

[5] F.-J. Chang, A. Tuan Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni. Faceposenet: Making a case for landmark-free face alignment. In *IEEE Int. Conf. on Computer Vision*, pages 1599–1608, 2017.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conf. on machine learning*, pages 1597–1607. PMLR, 2020.

[7] J. Choi, M. Jeong, T. Kim, and C. Kim. Pseudo-labeling curriculum for unsupervised domain adaptation. In *30th British Machine Vision Conf. 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 67. BMVA Press, 2019.

[8] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. *Int. Journal of Computer Vision*, 101(3):437–458, February 2013.

[9] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for domain adaptation. In *6th International Conf. on Learning Representations, ICLR 2018, Vancouver, BC, Canada*, 2018.

[10] Y. Ganin and V. S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conf. on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37, pages 1180–1189. JMLR.org, 2015.

[11] J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems 33: NeurIPS, December 6-12*, 2020.

[12] J. Gu, X. Yang, S. De Mello, and J. Kautz. Dynamic facial analysis: From bayesian filtering to recurrent neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1531–1540, 2017.

[13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[15] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. C. Nabbe, I. A. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(1):190–204, 2019.

[16] F. Kuhnke and J. Ostermann. Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous label spaces. In *Proceedings of the IEEE/CVF International Conf. on Computer Vision*, pages 10164–10173, 2019.

[17] A. Kumar, A. Alavi, and R. Chellappa. Kepler: keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In *12th IEEE Int. Conf. on Automatic Face & Gesture Recognition*, pages 258–265, 2017.

[18] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[19] A. Larumbe, M. Ariz, J. J. Bengoechea, R. Segura, R. Cabeza, and A. Villanueva. Improved strategies for hpe employing learning-by-synthesis approaches. In *2017 IEEE International Conf. on Computer Vision Workshops (ICCVW)*, pages 1545–1554, 2017.

[20] S. Lathuilière, R. Juge, P. Mesejo, R. Muñoz-Salinas, and R. Horaud. Deep mixture of linear inverse regressions applied to head-pose estimation. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4817–4825, 2017.

[21] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud. A comprehensive analysis of deep regression. *arXiv preprint arXiv:1803.08450*, 2018.

[22] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, (ICML)*, volume 3, 2013.

[23] X. Liu, W. Liang, Y. Wang, S. Li, and M. Pei. 3d head pose estimation with convolutional neural network trained on synthetic images. In *IEEE Int. Conf. on Image Processing*, pages 1289–1293, 2016.

[24] I. Martinikorena, R. Cabeza, A. Villanueva, and S. Porta. Introducing i2head database. In *Proceedings of the 7th Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, pages 1–7, 2018.

[25] S. K. Mustikovela, V. Jampani, S. D. Mello, S. Liu, U. Iqbal, C. Rother, and J. Kautz. Self-supervised viewpoint learning from image collections. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 3971–3981, 2020.

[26] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conf. on Computer Vision*, pages 7588–7597, 2019.

[27] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems 31: NeurIPS 2018, December 3-8, Montréal, Canada*, pages 3239–3250, 2018.

[28] M. Patacchiola and A. Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017.

[29] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *CoRR*, abs/1603.01249, 2016.

[30] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *12th IEEE Int. Conf. on Automatic Face & Gesture Recognition*, pages 17–24, 2017.

[31] N. Ruiz, E. Chong, and J. M. Rehg. Fine-grained head pose estimation without keypoints. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, June 2018.

[32] A. Schwarz, M. Haurilet, M. Martinez, and R. Stiefelhagen. Driveahead-a large-scale driver head pose dataset. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2017.

[33] M. Selim, A. Firintepe, A. Pagani, and D. Stricker. Autopose: Large-scale automotive driver head pose and gaze dataset with deep head orientation baseline. In *International Conf. on Computer Vision Theory and Applications (VISAPP)*, 2020.

[34] M. Shao, Z. Sun, M. Ozay, and T. Okatani. Improving head pose estimation with a combined loss and bounding box margin adjustment. In *The 14th IEEE International Conf. on Automatic Face and Gesture Recognition (FG2019)*, 2019.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[36] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conf. on Machine Learning*, pages 6105–6114. PMLR, 2019.

[37] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017.

[38] R. Valle, J. M. Buenaposada, and L. Baumela. Multi-task head pose estimation in-the-wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[39] Y. Wang, W. Liang, J. Shen, Y. Jia, and L.-F. Yu. A deep coarse-to-fine network for head pose estimation from synthetic data. *Pattern Recognition*, 94:196–206, 2019.

[40] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[41] T.-Y. Yang, Y.-T. Chen, Y.-Y. Lin, and Y.-Y. Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 1087–1096, 2019.

[42] X. Zhang, Y. Sugano, and A. Bulling. Revisiting data normalization for appearance-based gaze estimation. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 1–9, 2018.

[43] Y. Zhou and J. Gregson. Whenet: Real-time fine-grained estimation for wide range head pose. In *BMVC*, 2020.

[44] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 146–155, 2016.