# Epochs	Con.	Fwd	Bwd
1	15.09	9.68	33.72
5	14.32	7.76	32.86
10	5.30	6.93	13.58
110 (paper)	2.52	6.69	9.19

# imgs	Con.	Fwd	Bwd
10	3.11	12.01	7.31
100	3.44	7.25	10.77
1000	4.01	6.86	11.30
All (paper)	2.52	6.69	9.19

Table 1: *Con.*: Avg. Consistency error, *Fwd*: Forward error, *Bwd*: Backward error. **Left**: results using different core networks (GC1). **Right**: results by varying the # of training images (GC2).

• **All Rs:** Thank you for your insightful comments. There were some concerns raised regarding conducting additional experiments which we hope are addressed in General Comments (GC) 1 & 2 and individual responses.

• **GC1: Accuracy vs quality of core network** While our core (fully supervised) network in the paper was trained for 110 epochs (PCKh=79%), we also used the weights after the 1st (PCKh=22.95%), 5th (PCKh=57.67%), and 10th (PCKh=57.67%) epochs of the training process. With these networks as cores, the achieved accuracies are shown in Table 1 (**Left**). Due to limited space, we report only on AFLW dataset using the full dataset to train the regressor (see also Sec 4.2 of the main paper). These results clearly illustrate the importance of training a powerful core network.

• **GC2: Accuracy vs # training images** To this end, we chose a random subset of 10, 100, and 1000 images. The achieved accuracies are shown in Table 1 (**Right**). Visually, we observed that using only 10 images makes the network converge to a singe point, hence the poor forward and good backward and consistency errors. Using a subset of 1000 images, results are getting close to using the whole CelebA dataset (200k images). Fine-tuning the core network using 1000 images, yields consistency, forward and backward errors of 5.38, 7.52, and 13.36 which are much worse than our approach. This shows that our method is more effective with limited training data. We will include an in-depth study.

• **R1**: • **R1.1**: *1. Finetune*: "Finetuning" is the same as training from scratch but with the weights initialized from the human pose estimation network. Thank you, we will clarify this. • **R1.2**: *2. Performance on MPII*: On MPII, one can just use the core network so no drop in performance occurs. This is also equivalent to setting the weight transform kernels to identity. If we understood correctly, you also requested to see what happens if the target domain is also set to MPII. In this case, the discovered landmarks are different from the ones that the core network learns to predict in a supervised manner. This is not unreasonable as the objective functions for the supervised and the unsupervised cases are completely different. Thank you for this we will include it. • **R1.3**: *3. Fine-tuning only the last layers*: We did try this: the results on AFLW for the consistency, forward, and backward errors are 4.10, 7.6, and 12.0 showing that this is actually worse than fine-tuning the whole network. • **R1.4**: *4. Hyperparams*: All networks are trained in the same way (with augmentation) until the predicted points on the training set do not change w.r.t a threshold.

• **R2**: • **R2.1**: *Strong assumption + category dependency*: Indeed, the target domain learned filters are a linear combination of the core ones; however this doesn't seem to be so restrictive as long as the core network is trained on a difficult task (this is why we chose the one of human pose estimation). Thank you for this, we will also show the inverse experiment (face->human) in the revised version. So far we have tried face->MPII with little success for all methods probably because MPII is too hard (e.g. multiple objects per image). • **R2.2**: *Evaluation for cat head, shoes.*: We follow prior works (e.g. [13]) and evaluate shoes and cats qualitatively. On top of that we report consistency measure. Please see supplementary. We believe this is sufficient. • **R2.3**: *Taxonomy.*: This is interesting but given the lack of annotated data it is hard to conduct. In any case, the most critical is to have a core network trained on a difficult task, see also R2.1. • **R2.4**: *Interpretation of 3D data*: Our method does work for LS3D as both qualitative results and the consistency measure show (see Table 3, paper). However, as the forward error shows, it is hard to learn the mapping between 2D landmarks (as learned by our method) and the 3D landmark annotations. Unfortunately, we do not have visibility labels so we cannot measure accuracy on visible landmarks only. Thank you for this we will clarify it. • **R2.5**: *Landmark consistency*: We agree that the consistency alone is not sufficient. However, as we show in our work, forward error is not sufficient either. This is why we use ALL 3 errors (forward, backward, and consistency) always in combination with visual quality. We believe that such evaluations are SOTA. • **R2.6**: *Performance according to quality of core*: Please see GC1. • **R2.7**: *Performance on small no. of training images*: Please see GC2. • **R2.8**: *Categorical dependency relationships*: We will also include face->human. See R2.1 and R2.3.

• **R3**: • **R3.1**: *Technical contribution*: We are the first to propose the mixed training strategy of Fig 1c (core–supervised, target domain–unsupervised) and show that this is beneficial for constraining the optimization problems encountered in unsupervised landmark discovery. • **R3.2**: *Table 1*: In all tables including Table 1, our method always outperforms both trained-from-scratch and fine-tune networks when these are implemented in-house hence ensuring a fair comparison. The implementations of [13] and [45] report better numbers but they use different ways to process and crop the images which can significantly impact the results. Also please notice that these numbers are the forward errors which as we show in our paper can be biased. Adapting an image classification model could be possible, but we opted for using a human pose one since this is readily available and we are interested in landmarks. • **R3.3**: *Comparisons with only two baselines.*: One of our baselines is the trained-from-scratch network from [13] which is SOTA. The other baseline improves upon that. Hence, we have ensured sufficient comparison with SOTA. • **R3.4**: *Fine-tuning after matrix projection*: We tried this and observed no further improvement: we got 2.51, 6.46, and 9.52 for the consistency, forward, and backward errors, respectively. We will add this to the paper. • **R3.5**: *Try other core nets*: Please see GC1.