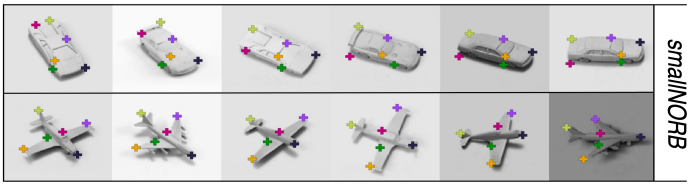
We **thank all Rs** for their comments and for recognizing the novelty of our approach. • **GC1: On SuperPoint initialization** This is part of our framework but we do not claim any methodological contribution for it. It has limitations (e.g. on low-textured surfaces) but also can be seen as an advantage given the tremendous progress on generic landmark detection. SuperPoint is trained in a self-supervised manner. Other works use pre-trained networks too (e.g. [31] uses fully supervised net from other domain, and [13] VGG for perceptual loss). We also provide an ablation study in L.197 where SuperPoint landmarks are replaced by a mixture of noisy g.t. landmarks and random noise which show (see paper's Fig. 2d) that, when the initial keypoints have F-measure of $\sim 0.4$ (on AFLW) our method works well. As the **Table herein** shows, SuperPoint meets this requirement, R2D2 is close to this, but SIFT/SURF/FAST (ran over rebuttal period) provide too poor initialization.

| Method | F-measure |
|---|---|
| SuperPoint | 0.42 |
| R2D2 | 0.33 |
| SIFT | 0.12 |
| SURF | 0.14 |
| FAST | 0.13 |




• **GC2: Performance on MAFL/AFLW** It is true that on "easy" frontal facial datasets our method does not surpass previous methods and this is to be expected as prior works have made tremendous progress on such datasets. Our method outperforms SOTA by large margin on the more difficult large pose (with 3D rotations) LS3D and Human3.6M.

• **R1**• **R1.1**: *Low texture surfaces*: Indeed, the method will not work well where the generic detector fails. Object landmark detectors mostly aim to detect salient points (e.g. mouth/eye corners, knee joints). Low texture points are difficult even for the supervised case. We will discuss this. • **R1.2**: *Why it works for 3D rots*: You're right, actually, it is because of all the points you mention. • **R1.3**: *On MAFL/AFLW*: Please see **GC2**. Note, LS3D contains portion of 300W. • **R1.4**: *On poor accuracy for face contour (LS3D Forward)*: You are right, we will provide the per-landmark errors to make this clear. • **R1.5**: *On labelled data for Superpoint*: Superpoint is trained in a self-supervised manner. So we still believe it qualifies for unsupervised. However, we will make this clear. The edge detector used in Pathak et al CVPR17 learns from manually annotated segmentation masks. • **R1.7**: *Labels for MAFL*: You're right.

• **R2** • **R2.1**: *Some results not being impressive.*: Please see **GC2**. • **R2.2**: *Dependency on SuperPoint*: Please see **GC1**. • **R2.3**: $K : K$ is the underlying number of "discoverable" object landmarks which our method aims to discover, not a hyperparameter. In practice, this number mostly depends on the number of "good initial" landmarks detected by the generic detector. See also GC1. We will make this very clear, thank you. • **R2.4**: *Non-differentiable process*: Clustering followed by the Hungarian algo. are performed at the end of each training round. They are not part of training the network; they just provide the pseudo-labels for the next iteration. We will make this clear, thank you.

• **R3**• **R3.1**: *SuperPoint:*: Please see **GC1**. • **R3.2**: *only 1 3D dataset used:*: Our method outperforms SOTA by large margin on **2 datasets** with 3D rotations, namely **LS3D and Human3.6M**. • **R3.3**: *Guarantee of semantic meaning*: We will rephrase to relax this statement. Our method however achieves a much stronger semantic representation, see L31-L33. • **R3.4**: *Comparison with [22]*: The authors of [22] provide a non-complete Github repo (work-in-progress) at the time of submission. • **R3.5**: *Schematic view*: We will try to fit one. • **R3.6**: *Inconsistent language*: We will improve this. • **R3.7**: *Readability (5, 6 and 11)*: Thank you we will clarify/improve • **R3.8**: *Comparison with SIFT*: Please see **GC1** • **R3.9**: *R2D2 yielding worse results*: We attribute this to the repeatability constraint of R2D2 which leads to sparse keypoints and poorer init. See also **GC1**. • **R3.10**: *Background points for [13]*: We used their version of the code treating each frame separately for the sake of a fair comparison to all other methods. In this case [13] can yield points in background. We will clarify. • **R3.11**: *Comparison on 3D databases*: Please see R3.2. We also conducted the experiment from [13] on smallNORB. Our method works well, some results are shown in **Figure herein**; compare to Fig. 6 from [13]. • **R3.12**: *Comparison with [35]*: [35] represents a simpler setting than ours as it uses 3D objects rendered into 2D images assuming known 3D transformations between them. The objects are pre-segmented. We used significantly more complex real-world images (including human poses) without any knowledge of 3D information.

• **R4**• **R4.1**: *Additional complexity*: It is likely that the pipeline can be simplified. However, our method is the first of its kind to explore ideas like self-training, and correspondence via clustering to solve this problem. We show large improvements on difficult large pose datasets (LS3D, Human3.6M) so we believe that the impact of our results is significant. • **R4.2**: *Performance on MAFL/AFLW*: Please see **GC2**.• **R4.3**: *Comparison with [31]*: [31] uses for initialization a net trained in a fully supervised manner that's why we didn't compare. Such an idea could fit our method too, but this is out of the scope of our paper. • **R4.4**: *Resolution* : Methods are retrained using the provided codes for the same resolution (256px). We used $d = 12$px for resolution 256, equivalent to $d = 6$px for resolution 128 used in [13]. • **R4.5**: *Clustering in 2nd phase*: No clustering is performed in 2nd phase. Clusters are just merged. A detector is learned via self-training with correspondences from 1st step. • **R4.6**: *Comparison with SIFT*: Please see **GC1**. • **R4.7**: *Clarity/reproducibility*: Thank you for such detailed feedback to improve our work! We will clarify ALL points raised and **we will release code**. • **R4.8**: *NMI*: It's not typo, see suppl. L.18-L.29. • **R4.9**: *Wiles et al.*: We will include it. • **R4.10**: *Missing landmarks*:. Yes, we only visualize landmarks with confidence 0.2. • **R4.11**: *Visualization of keypoints/cluster evolution*:. We will include a vis. of detected keypoints over iterative rounds. Vis. of cluster evolution is provided in Fig.2 (a). We will extend this to features learned over iterative rounds.