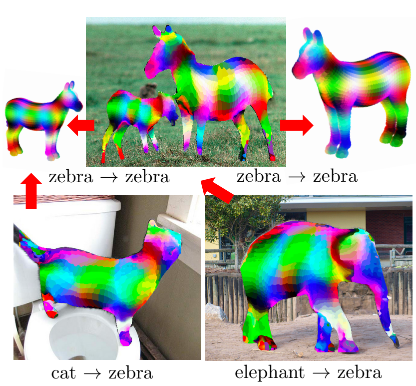
Reviewer 1 **[does not] justify the choice of the addition** There are two motivations for our continuous embeddings (see also lines 22 to 48). The first is to **simplify Dense Pose**. Charts in Dense Pose are a legacy from prior models such as DenseReg, which used them to improve inference accuracy *despite* the added complexity and issues such as the arbitrary seams between the charts. Our continuos embeddings remove charts with the same or even slightly better accuracy. The resulting model is simpler (fewer lines of code). Furthermore, the mesh pre-processing is not manual as in Dense Pose, but automatic, and can be obtained by using an off-the-shelf LBO implementation. The second motivation is to make it much easier to build **unified models of multiple objects** (lines 31 to 48). The original Dense Pose considered a single class (humans), and recent papers such as [44] had to do quite a bit of manual work just to tackle two classes (humans and chimps). Our method requires much less manual work because the continuous embeddings support functional maps, which are one of the most straightforward ways of modelling correspondences between 3D shapes (as they reduce the problem to simple linear algebra operations). **Not-so-easy to implement, automated co-segmentation** These are valid alternatives, but, respectfully, we do not see how these would be simpler than our approach. Our method makes implementing Dense Pose simpler and removes the need for segmentation charts, manually or automatically. Furthermore, note that most of the conceptual complexity is limited to relating different shapes via functional maps, which is not needed for modelling a single category. **Unsupervised learning of dense correspondences [22], other works.** These methods tackle the problem of 3D mesh alignment, not of detecting pose in 2D images as we do. In our setting, they could be used as a replacement component for ZoomOut (line 184) to establish the initial 3D mesh alignment (as a matter of fact, we did test [22] as a ZoomOut replacement, but found results to be unsatisfactory when aligning only a handful of meshes).

Reviewer 3. **Runtime analysis.** At test time our model is typically slightly faster than the standard Dense Pose network because the overall output dimensionality of the network is usually lower (D vs number of charts by 3 in Dense Pose).

Reviewer 4. **(1) Only a single mesh per category** Dense Pose uses a single mesh for all humans, so one mesh per animal class is likely to be sufficient in most cases. Note that each mesh provides a canonical representation of dense correspondences, and as such it does not need to be geometrically faithful to any particular object instance. Instead, the mesh should be close enough to the object to allow human annotators to mark correspondences and to provide a weak geometric prior on the possible correspondences between different animals. **(1) LB doesn't stay static.** True, the LBO is invariant only to isometric deformations. Nevertheless, it is still heavily used in computational geometry even for meshes that are non-isometrically related, even in presence of major topological changes (see for example Functional map networks for analyzing and exploring large shape collections. Huang, Wang, Guibas. ACM Trans. Graph., 33(4):36:1–36:11, 2014.) This is because: (1) *part* of the LBO structure is approximately preserved even in these cases and (2) LBO is still useful as a generic 'smoothness' prior on the functional/correspondence maps even when it is not invariant. See also lines 181 to 188. **(2) transfer function…linear…sufficient?** Yes, they are sufficient. Note that these are *functional maps*, namely linear maps acting on the space of *functions* $S \to \mathbb{R}^d$, not merely on their *range* $\mathbb{R}^d$. In particular, functional maps include all warps of functions defined on a shape S (in particular the embedding function e_X) to functions defined on a shape S' (in our case $e'_{X'}$). Since we assume that the embeddings of two shapes are related by a warp, this relation is expressible as a (linear) functional map. Intuitively, in the discrete case, functional maps include all *permutations* matrices. Naturally, in a practical implementation one only considers a finite-dimensional subspace of all possible functional maps; this means that only sufficiently smooth warps can be represented, which, rather than being a problem, is good for regularization. If anything, we require additional constraints for the functional map to *only* express a warp (see also lines 483 – 494 in the appendix). **(3) extreme deformations.** We do agree that this aspect can be improved in the future. Nevertheless, 'Functional map networks for analyzing and exploring large shape collections' (see above) did show that even large topological changes can be handled by functional maps as these can collapse parts that are not in correspondence.



(4) baseline improvement marginal…mismatches… This is the first paper able to extend Dense Pose to several classes using a single canonical space. Even for our baseline model, the new basic machinery (continuos embeddings, LBO) is still required. On top of this contribution, the prior arising form 3D mesh alignment improves result further. Finally, it is true that there are some mismatches, but we believe our results constitute overall a significant positive delta in this space. **(5) Visualization, template mesh** Thank you for the suggestion. Some preliminary examples using a higher-frequency procedural texture are given in the figure (we are working on implementing the use of arbitrary handcrafted textures). **(6) 'Learning to Segment Every Thing, CVPR 18': MPL vs linear function** See point (2): using a linear function*al* instead of an MLP is not really a limitation as these already include all useful warps (permutations, correspondences) of the embedding vectors between meshes.