R1: We agree that defining a canonical orientation for local patches is mainly aimed at descriptor matching. However, we propose a more general framework that can also be adopted to orient whole objects and perform rotation-invariant shape classification. Moreover, as recently shown in Bai *et al.* in "D3Feat: Joint Learning of Dense Detection and Description of 3D Local Features." (CVPR 2020), while FCGF performs very well when trained and tested on the 3DMatch dataset, it suffers from a large drop in performance in transfer learning towards the ETH dataset. This weakness in transfer learning is particularly critical for supervised methods, such as FCGF, because it limits applicability only to datasets for which the ground truth is available. One reason for this performance drop is likely to be the handling of rotation invariance by data augmentation, which may hardly generalize to unseen datasets. Indeed, 3DSN [10], which achieves rotation invariance by a LRF, outperforms the competitors by a considerable margin in transfer learning on ETH. Based on these considerations, we believe that it is not yet been established that rotation-invariant descriptor matching can be solved without orienting surfaces. Thus, as vouched by its large performance gain with respect to existing LRFs across different datasets, we believe that Compass is a principled and useful contribution that can improve the performance of learned feature descriptors relying on LRFs.

Compass extracts the canonical orientation for a patch in 17.85ms. We will add this information to the revised version.

In the evaluation in Table 2, we follow the standard protocol used in [39] to perform a fair comparison. We believe that this kind of evaluation provides important insights as it highlights whether learned methods can generalize or not to unseen rotations. We agree with the reviewer that it would also be interesting to investigate on the behavior of the competitors when trained on the full spectrum of SO(3) rotations, but we could not run such experiment in the limited time available to complete the rebuttal, as it requires re-training all competitors listed in Table 2. We will mention it as future work and highlight the importance of this complementary assessment in the conclusions.

R2: We used the more general term "pose" to refer to canonical orientation as achieving translation invariance is usually trivial, but we agree that this use may be misleading. As suggested, we will use only the term *orientation*.

(Pointed out also by **R3**) We agree that the notation in (4) should be changed from $g^{-1}(\mathcal{V})$ to $g(\mathcal{V})^{-1}$, since inversion is applied to the output of function g, *i.e.* the learned rotation matrix. We will modify it in the final version of the paper.

We agree that the domain of Spherical CNNs feature maps is key and we will better highlight it in the final version.

Since we seek for one rotation, the loss function in (6) is applied once, and only to the last layer of the network.

The input of our network is a spherical signal that is invariant to permutations of the input data. More details about it can be found in [39]. As for PointNet, we use the original implementation provided by the authors.

We experimentally verified that a larger overlap between fragments improves the repeatability of Compass on local patches. We will add this insight to the discussion of the experimental results.

R3: The output of a spherical correlation is a signal living in SO(3). In particular, each feature map is a cube where each cell, indexed by i, j, k, represents an element of SO(3), *i.e.* a rotation.

As suggested, we trained a version of PointNet without the T-Nets. The performance in test is: NR: 88.49; AR: 8.35.

The use of Blendsor would have required generation of a cumbersome off-line training dataset. We instead used an on-the fly data augmentation where occlusions can be randomly generated across epochs: since it proved effective, as demonstrated by the ablation study, we consider its simplicity a positive aspect, as suggested also by the reviewer.

We rely on Spherical-CNNs because equivariance to rotations is crucial to satisfy (5) and they are equivariant to SO(3) by construction. Conventional neural networks do not possess this property.

We will update Figure 2 including all the symbols adopted in the definition of the methodology. The loss compares the max entries of the final features maps, which corresponds to rotations, as explained above.

R4: We present the first machine learning approach to orient point clouds. Differently from previous handcrafted solutions, no geometrical cues are adopted to design a repeatable canonical orientation, while we leverage the equivariance of Spherical CNNs and show that a fully data-driven approach is feasible and, indeed, more effective than SOTA solutions. It is worth pointing out that this problem can not be tackled by supervised learning as there is no unique manner to define the ground-truth (i.e. a repeatable canonical orientation). Thus, we propose a self-supervised formulation where the network is able to discover the best suited canonical orientation based on training data.

Unlike T-Net, Compass is not trained end-to-end with PointNet. In the simplified scenario where the input data is always under the same pose (NR), a canonicalization step is useless and can only lower performance by injecting noise due to its errors. T-Nets trained jointly with PointNet can instead learn that the best orientation in this scenario is the identity matrix. Yet, Compass could be easily modified to be trained end-to-end with the down stream task: this is an interesting future work we would like to explore.