We are grateful for all comments and suggestions. Below we address questions raised by individual reviewers.

Suggestions on adding or revising qualitative results and figures. (R1, R2, R4) We are thankful for these suggestions and will add qualitative results on non-human shapes in the next version. This includes adding qualitative results for non-human shapes and cumulative error curves for Table 1 and adjusting color mapping in Figures 2 and 3.

Comparison to LES [9], DeepGeoFunc [10], 3DCODED [11] on Partial-to-full task. (R1) DeepGeoFunc requires a triangular mesh as input, while 3DCODED and LES require a point cloud sampled from a full triangular mesh. In other words, none of these methods ([9], [10], [11]) applies to partial depth scan input. Therefore, we only compared our approach with these methods for the full-to-full task in the appendix (Table 2).

Comparison to the SHOT descriptor. (R1) We used the code from the original SHOT paper to generate SHOT descriptors for the template mesh vertices and used them to train our descriptor module for three days. The resulting average correspondence errors on SURREAL/FAUST/SHREC19 are 5.97cm/6.66cm/11.04cm, respectively. Our results (1.71cm/1.90cm/4.81cm) are significantly better.

Investigating unsupervised setting. (R1) The major contribution of our paper is the transformation propagation module, which learns from the error distribution of point-wise transformations to detect and rectify incorrect correspondences that violate local rigidity. This strategy differentiates our approach from other methods based on hand-crafted heuristics. Similar to other approaches, we can use the distance-preserving loss to train the transformation predictions.

MPI FAUST Results. (R1) Most results that report on FAUST leaderboard use additional training data that significantly impacted the results. In contrast, we compared all the methods under the same setup. We also plan to release the code and data so that other methods can be compared under the same training/testing setup.

Generalization from minimal clothing to clothing. (R2) Our approach is motivated by the piece-wise rigidity assumption. Enforcing such an assumption is learned, meaning it can be adaptive to rigidity at different levels. For clothing, it means more rigidity at the coarse level and less rigidity at the fine level.

End-to-end Training. (R2) The descriptor module is pre-trained in order to generate initial transformations. Then the whole pipeline (including the descriptor module) is trained end-to-end.

Robustness of feature descriptors. (R3) We will add a visualization of the distribution of feature descriptors in the revision. Note that the 10-cm recall of the correspondences derived from feature descriptors are above 95% for all datasets. They provide a good foundation for the transformation synchronization module to rectify the error.

Motivation of iteratively reweighted least squares (IRLS). (R3) IRLS is a popular method for solving regression problems that involve robust objective functions, where the final solution is insensitive to a fraction of noisy measurements. Suppose the noise ratio is below some constant (described in the main theorem). In that case, IRLS can be proven to suppress such errors and converge to the underlying ground-truth. The paper's contribution is a learning approach that leverages side information to reduce the input noise ratio (e.g., by reweighting) so that such a condition holds.

Ablation study on number of transformation propagation layers. (R4) The average correspondence error before transformation propagation and after the first/second/third layer is 2.19cm/1.80cm/1.73cm/1.71cm for SURREAL, 2.59cm/1.97cm/1.89cm/1.90cm for FAUST, and 5.84cm/5.02cm/4.86cm/4.81cm for SHREC19. Most of the improvement comes from the first layer. After the third layer, the effect of transformation propagation becomes marginal.

Learn canonical descriptors freely. (R4) In our approach, canonical descriptors only provide initialization. We first train our descriptor module to match Laplacian embedding descriptors. Then in the next phase of Training, the descriptor module is fine-tuned with the transformation propagation module. Laplacian embedding descriptors worked well. Since they are used only for the initialization, there was not much need to learn alternatives.

What if no down-sampling/up-sampling. (R4) Down-sampling and up-sampling layers enable efficient computation. Empirically, these layers can provide 10x to 20x speedup without noticeable loss of performance.

Transformation becomes invalid after interpolation. (R4) We allow the transformations after propagation to be outside SE(3). This approach adds flexibility to encode local deformation. The same strategy was used in Sumner et al. 07. We will clarify this in the revision.

Questions about equations. (R2, R3) The point distance version of Eq. 3 is much more complicated compared to merely enforcing neighboring transformations to be equal. The weights are generated by a network that takes transformations of neighboring points as input. In Eq. 11, v_{c_i} should refer to a point on the template mesh. In Eq. 12, Φ should be removed.