First, we thank the reviewers for their detailed, constructive, and positive feedback on the paper. We are happy to see the they appreciate our contributions in proposing a method that is “mathematically sound” [R1], “simple and elegant” [R3], and “enjoyable to read” [R4], while addressing “an important topic in machine learning, computer vision and computer graphics” [R1]. Below we address specific feedback from the reviewers.

Geometry alignment in the normalized frame [R1, R3]. In the ShapeNet deformation space experiment, learning happens in the normalized shape frame (as provided by ShapeNet), which is similar to most works in the literature concerning 3D geometry generation. The inherent alignment in this normalized space can facilitate a more meaningful geometric loss via the symmetric Chamfer distance when considering deformation between objects. However, as we illustrate in experiment 4.1.2: Canonicalization of Shapes, by deforming through a common “hub” shape, the deformation can further regularize the shapes in this space to achieve much better alignment in the deformed canonical shape space – thus providing more semantically meaningful correspondences between geometries. For extending this method in real-world settings, one can envision a two-step approach that combines an initial pose estimation algorithm with the “reconstruction via deformation” approach proposed in this work.

Disentangling deformation and global transformation [R3]. The deformation methodology proposed in this work uses a single flow field to serve as the deformation function, which includes all transformations, global or local. It works well when dense correspondences are provided (e.g., Figure 5 for the animation experiment). To explicitly decouple global and local transforms, one can learn composed flow fields of pure rotational flow (parameterized with 3 DOF), pure translational flow (parameterized with 3 DOF), and local deformation flow (parameterized with a neural network). We leave more detailed investigation of this approach to future studies.

Related work in Graphics [R4]. We thank the reviewer for pointing out the work of Von Funck et al., we will cite that in the camera-ready version of this work.

Accuracy of statements [R3, R4]. (1) Regarding the discussion about morphable models, we note in our paper that morphable models are commonly used for shapes with small intra-class variations, and generally assume a shared topology (at inference time, not training), which we believe is an accurate statement. We would kindly ask the reviewer to provide a reference to a paper where this statement does not hold. (2) Regarding the claim about detail preservation, there is only one instance where we mentioned that, in L232, in the context that “details are preserved by the deformation.” We are not claiming that detail is preserved from the input point cloud to the output geometry. (3) *Marching Cubes is fast* – Yes, other aspects of volumetric methods are less efficient (sampling for OccNet/DeepSDF, high memory usage for voxel methods, etc.). We will rephrase L50 to be more accurate. (4) *You only implicitly regulate towards a subset, while using a loss on divergence change would explicitly regulate all cases for volume conservation.* – We agree with the first statement, but even using a loss on divergence, the flow itself is still just a parametric flow.

Miscellaneous questions [R4]. (1) *Why not use an encoder?* – an encoderless scheme is more flexible and forgiving regarding partial and incomplete inputs, whereas a scheme that uses an encoder will require a substantial amount of data augmentation to train – it is just a design choice. (2) *Comparison to direct nearest neighbor retrieval* – an example of the directly retrieved geometry is given in Figure 2, rightmost column. (3) *To apply a deformation, costly integration has to be performed* – the integration step executed on a GPU is fast, «1sec. (4) *How are the k meshes deformed merged into one* – we pick the best one; we will add a more detailed description in L210. (5) *Why introduce meshes and edges in L120* – mesh structure of the source is constant in the deformation process, with the manifold property preserved in the process. (6) *Why do you need invertibility* – invertibility of the space facilitates training. For instance, training $A \to B$ automatically trains $B \to A$. Moreover, training $A \to H$ (hub) facilitates the training of $* \to H \to A$. (7) *Has the sketched idea (enforcing vector field props via the loss) been tried?* – No, but this will require computing loss on random volumetric samples, which quickly becomes expensive, and does not guarantee constraint satisfaction. (8) *Do any of the results use L172?* – Yes, in the ShapeNet experiment. (9) *In L211 it was not clear why meshes are subsampled* – for training a deformation space, we need to batch multiple samples together, and subsampling makes training more efficient; we do not subsample at test time.

Typos, math notation issues, LaTeXglitch [R3, R4]. Thanks, we will address them in the camera ready version!