We are thrilled that all reviewers are supportive of our work. We addressed the valuable comments raised below.

**Q1: Clarification of our contribution. (To Reviewer 1, 2 & 3)**

**1.** Please kindly allow us to clarify our key contribution using reviewers' comments as follows:"*For previous works, it makes perfect sense to apply pre-defined parameterization of known geometric transformation to model deformation, or to apply additional constraint for smoothness, based on traditional computer vision and graphics knowledge. However, the method this paper is proposing gives me an interesting insight that we instead can train a single continuous function to output a smooth and continuous displacement field without any geometric assumption. From my 3D vision background, this lesson aligns with the recent trend of 3D parameterization in a single continuous function such as DeepSDF or OccupancyNet.*".

**2.** Our proposed Arbicon-Net is the first original work for 2D parameterization of arbitrary displacement field using a single continuous function in the form of trainable multilayer perceptron (MLP) for image registration. More specifically, our main contribution is explained in Section 3.3, where we clearly present how the single continuous function is implemented with MLP, which outputs a smooth and continuous displacement field without any geometric assumption and smooth regularization.

**3.** Thanks to reviewers' pointer, we are excited to learn about the alignment with the latest trend of 3D surface parameterization (e.g. DeepSDF, CVPR'19, June, 2019) in a single continuous function, which further verifies the technical soundness of our proposed image registration approach. We will cite the related 3D vision references in the revised version.

**4.** We regret that we overlooked the presentation of our idea in an effective manner, leading to reviewers' confusion in understanding our novel contributions. We will revise our paper according to reviewers' valuable comments to clearly present and highlight the key contributions of our work and enhance the proof and the discussion of continuity and smoothness in its final version.

**Q2: Clarification of the design of the components. (To Reviewer 1 & 3)**

**1.** In section 3.3, we motivate to design a trainable continuous function to output a smooth displacement field. We choose MLP as the function because it in theory can learn a fully continuous function with arbitrary precision according to the universal function approximation by neural networks. We designed the "replicate and concatenate" operator because it can largely ensure the smoothness of the estimated displacement field as the function output (Please kindly refer to **Q3** for the justification of this design of the operator that led to the smoothness). **2.** In section 3.2 of transformation descriptor encoder, we leverage the power of 4D convolutional for image correlation feature learning by integrating additional neighborhood information. Note that the 4D convolutional was originally introduced in [13 of our paper], not in our paper. In revised version, we will place related references properly to clarify what are our original contributions and what are others' works. **3.** Kindly refer to experimental section 4.2 where we verified the effectiveness of the 4D-Conv by comparison between *WeakAlign* and *WeakAlign-4D* .

**Q3: Clarification of Continuity & Smoothness. (To Reviewer 1 & 3)**

**1.** Our MLP based displacement function is a continuous function that, for a given 2D spatial point on image plane, outputs the point's displacement values. However, since a MLP often takes as input high dimensional features, it suffers from the "curse of dimensionality" (i.e. when the dimensionality increases, the volume of the space increases so fast that the available data become sparse), which consequently leads to the over-fitting causing the unstable oscillation of the learned function. Current works usually [1 of our paper] impose an additional smooth regularizer (i.e. a linear operator on spatial gradients of the function) to penalizes local spatial oscillation in the function. In contrast, our proposed "replicate and concatenate" operator is designed to help address this challenging issue as explained below: our MLP takes as input a high dimensional feature which is formed by concatenating a 2D point coordinate and the shared common transformation descriptor. Since the shared common transformation descriptor is constant for all 2D point, our MLP is essentially still defined on 2D even though taking input as the high-dimensional feature, therefore the available data wont become sparse in our case. Our strategy for the avoidance of curse of dimensionality contributes to a continuous and smooth function without imposing any regularizer. In addition, as discussed in section 3.3, we also adopted training strategies (e.g. dropout, weight decay) to prevent our deep neural network from over-fitting, which naturally help our Displacement Field Predictor network reduce the risk of the oscillatory of displacement function. **2.** Please kindly refer our respected reviewers to the latest DeepSDF (Jeong, etc, CVPR'19, June,2019, pointed out by reviewer 2) for more detailed discussions on the continuity and smoothness of the single function used for 3D smooth surface parameterization. **3.** As suggested by reviewer 3, in the revised version we will include more empirical results about the smoothness property as well as additional description on the continuity and smoothness explained above.

**Q4: Other questions. (For Reviewer 1, 2 & 3)**

**To Reviewer 1:** The number of 4D-Conv layers is three, where the kernel size we used is $3 \times 3 \times 3 \times 3$. We define "parametric" as pre-defined parameterization of known geometric transformation to model deformation, such as examples in L219. In contrast, "non-parametric" is referred to a displacement field without any geometric assumption. We will clarify these questions in final revised version. **To Reviewer 2:** We will add the comparison result between our method and CNNGeo-4D, which is the best performing baseline on experiment 4.3, on experiment 4.5. **To Reviewer 3:** We will add the missing citations in section 3.1 and more qualitative results and discussion about failure cases in the revised version.We will correct all typos and minors pointed out from all reviewers in revised version.