Supplementary Material
Learning 3D Dense Correspondence via Canonical Point Autoencoder

Anonymous Author(s)
Affiliation
Address
email

A Computational Time and Model Size

A.1 Training Details

We use the chair category as an example to illustrate the training pipeline. With 500 training samples, the training takes about 24 hours to converge, (4 hours for stage one ($\alpha = 1$) and 20 hours for stage two ($\alpha = 0$)), on a single Tesla V100 GPU. Please see Algorithm 1 and Section 4.1 in the main paper for more details on the two stage training paradigm. *The source code will be released to public upon publication.*

Algorithm 1 : The training phase of our approach consists of two stages: (1) A pre-training stage trained with L_{ACD} and L_{rec} (2) A fine-tuning stage trained with L_{ACD}, L_{rec}, and L_{cross}

(A) STAGE-1: PRE-TRAINING ▷ *4 hours on Chair category*

Randomly sub-sample k points from the input point cloud S_A;
Initialize weight of the global feature encoder $E(\cdot)$, canonical mapping encoder $\Phi(\cdot)$ and inverse mapping decoder $\Psi(\cdot)$;
for epoch **in range** [0,T) **do**
  foreach iteration **do**
    $\boldsymbol{z_A} \leftarrow E(S_A)$;
    $\boldsymbol{U_A} \leftarrow \Phi([p, z_A])$, where $p \in S_A$;
    $\hat{\boldsymbol{S}}_{\boldsymbol{A \rightarrow A}} \leftarrow \Psi([q, z_A])$, where $q \in U_A$;
    Obtain loss L_{ACD} ($\alpha = 1$) and L_{rec};
    Update weight;

(B) STAGE-2: FINE-TUNING ▷ *20 hours on Chair category*

Generate randomly paired samples S_A and S_B;
while not converged **do**
  foreach iteration **do**
    $\boldsymbol{z_A} \leftarrow E(S_A)$;
    $\boldsymbol{U_A} \leftarrow \Phi([p, z_A])$, where $p \in S_A$;
    $\hat{\boldsymbol{S}}_{\boldsymbol{A \rightarrow A}} \leftarrow \Psi([q, z_A])$, where $q \in U_A$;
    Obtain loss L_{ACD} ($\alpha = 0$) and L_{rec};
    $\boldsymbol{z_B} \leftarrow E(S_B)$;
    $\boldsymbol{U_B} \leftarrow \Phi([p, z_B])$, where $p \in S_B$;
    $\hat{\boldsymbol{S}}_{\boldsymbol{A \rightarrow B}} \leftarrow \Psi([q, z_B])$, where $q \in U_A$;
    $\hat{\boldsymbol{S}}_{\boldsymbol{B \rightarrow A}} \leftarrow \Psi([q, z_A])$, where $q \in U_B$;
    Obtain loss L_{cross};
    Update weight;

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

A.2 Inference Details

Please see Fig 1 and Section 3.2 in the main paper for more details on the inference pipeline. Our proposed CPAE model contains 2.07M parameters which is 2.5× less than the 5.22M parameters in [1]. At inference time, the computational time for label transfer between a pair of shapes (each with 2048 points) is 0.03 second including runtimes of the nearest neighbor search for both shapes.

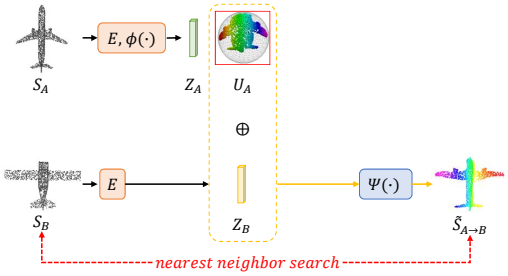

Figure 1: The inference flow of our approach.

B Application on Texture Transfer

Given a source 3D shape, we transfer texture from the source shape to multiple target shapes using computed correspondences. Our method is able to detect points that do not have correspondents in the source 3D shape (e.g., airplane without tail wings or stabilizers).

(a) Source

(b) Targets

Figure 2: Applications on texture transfer.

C Qualitative Results of Instance-aware Primitives

In Figure 3, we demonstrate more examples of the instance-aware primitives as discussed in Section 3.1 in the main paper. Two observations can be made from this figure: a) The instance-aware primitives produced by our canonical mapping are closely adhered to a canonical sphere. b) points of the same semantic parts are mapped to nearby locations on the primitives, as shown by the colored keypoints in Figure 3. These observations demonstrate that our model is able to learn correspondence across different shapes in the same category.

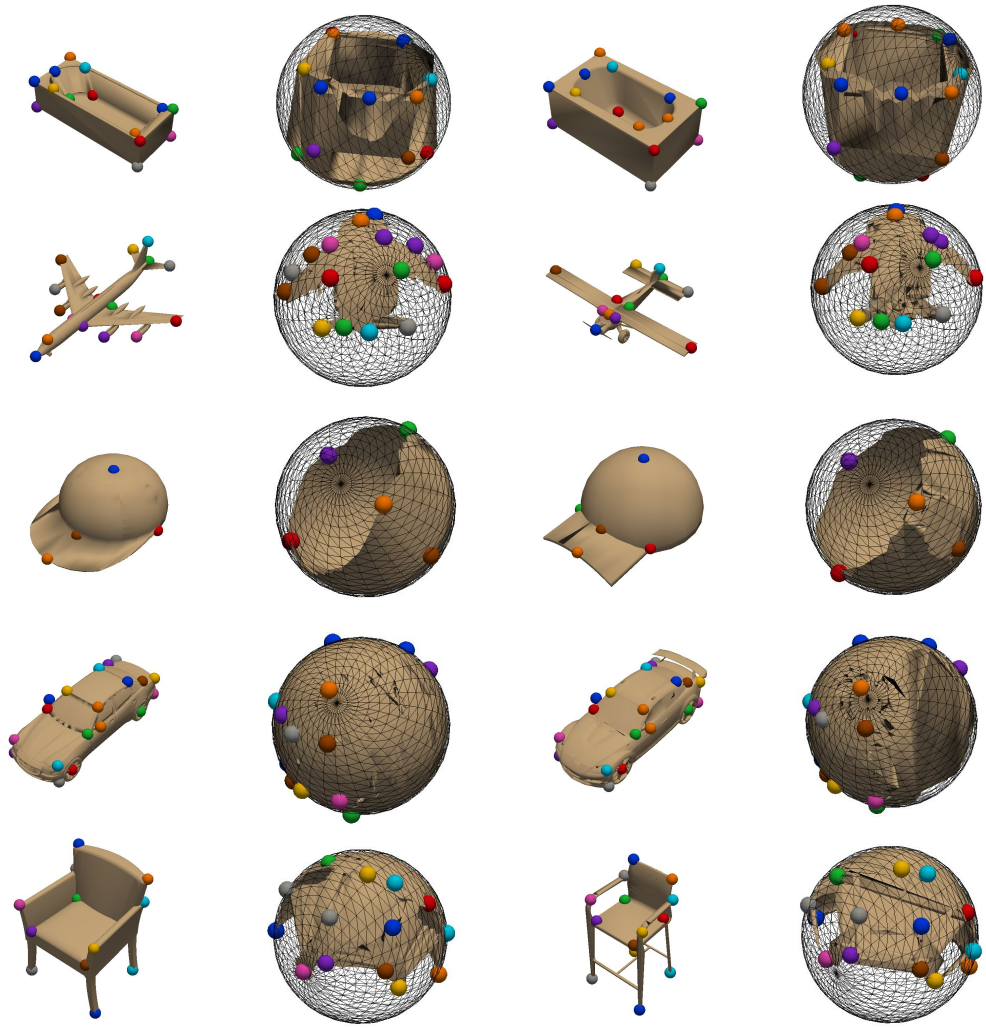

Figure 3: Iinstance-aware primitives on five different categories from the KeypointNet dataset [2]

D Qualitative Results of Part Label Transfer

In Figure 4, we show more qualitative results of the part label transfer task. Thanks to the dense correspondences learned by our model, we can transfer part labels for small parts (e.g. the handle of mugs or the tail wings of airplanes) and handle large intra-class variations (e.g. different legs of chairs in row three in Figure 4).

Figure 4: Part label transfer results. B→A refers to transferring shape B's label to shape A.

E Correspondence Confidence Heatmap Visualization

We use the confidence score (mentioned in main paper Section 3.4) to draw heatmaps for multiple target shapes in the same categories. As shown in Fig. 5, the predicted confidence heat maps successfully indicate the intra-class variations and capture uncertainty in correspondence prediction. For instance, low confidences can be found in mugs with different handles, knives with different blades, guitars with different bodies, etc.

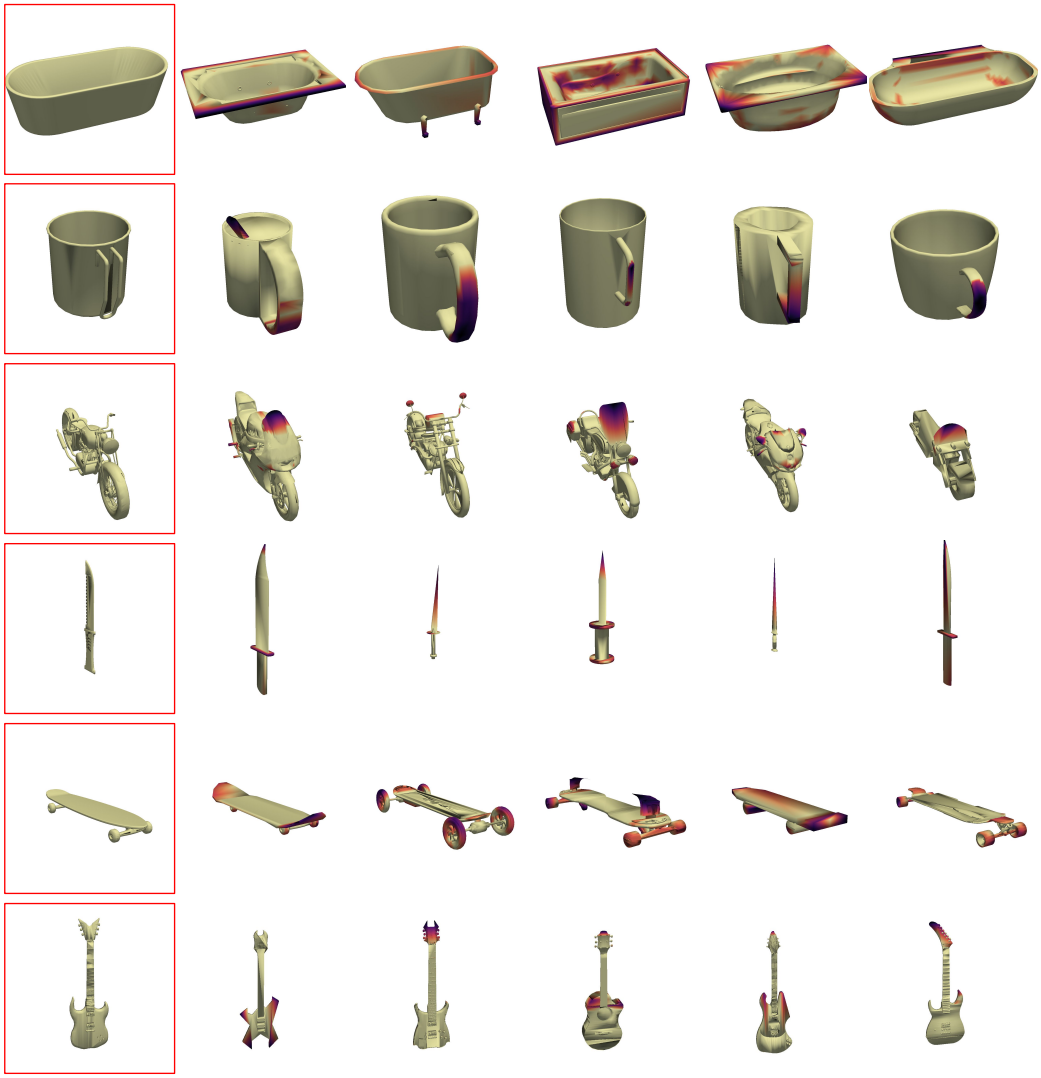

Figure 5: Correspondence confidence heatmaps. Red boxes indicate source shapes. The darker the heatmap, the lower the confidence.

F Limitation

There are two main limitations for our approach: (a) we encode the shape information of a point cloud in a global vector – i.e., fine details like corners and edges may be blurred after reconstruction. (b) We found the correspondences predicted near holes maybe wrong, possibly due to the sparsity of points in the point cloud and the nature of Chamfer and Earth Mover's distance matrices. We leave these limitations for future works.

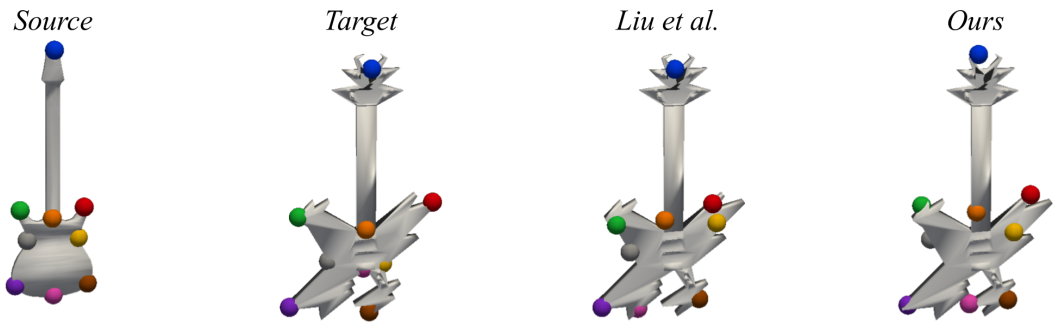

Figure 6: Failure cases.

References

[1] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. In *NeurIPS*, 2020. 2

[2] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *CVPR*, 2020. 3