GCE-Pose: Global Context Enhancement for Category-level Object Pose Estimation

Supplementary Material

1. More Implementation Details

Robust Partial Feature Extraction. We document more implementation details about GCE-Pose. For feature extraction, we employ DINOv2 [48] to process images cropped to 224×224 resolution using the $dinov2\_vits14$ model variant. Point cloud features are extracted via PointNet++ [51] with multi-scale grouping, generating per-point features for partial observation and shape reconstruction tasks. Following AG-Pose [41], we utilize 96 key points. For the object-aware chamfer distance, we set the outlier threshold $\tau_1 = 0.1$ in Eq. (1) and the keypoint diversity regularization threshold $\tau_2 = 0.2$ in Eq. (3).

Semantic Shape Reconstruction. For each model k, we initialize prototypes $c^k \in \mathbb{R}^{N\times 3}$ using the K-Means++ algorithm [1]. The deformation field v^k applied to prototype c^k uses point-wise parameterization with vectors of dimension $D \times (N \times 3)$. To balance complexity and efficiency [43], we set the number of basis vectors D to 5. The deformation network $\mathcal{D}^k$ processes centered partial point clouds to produce shape parameters $a \in \mathbb{R}^5$, corresponding to coordinates in the linear space defined in Eq. (4). Similarly, the scale network $\mathcal{S}^k$ outputs scaling parameters $s \in \mathbb{R}^3$ for anisometric scaling along all axes. Both networks share a PointNet++-based [51] encoder for feature extraction. Training proceeds in two stages:

- First, we train the deep linear shape model using ground truth point clouds sampled from object meshes via farthest point sampling (FPS). Following [43], we employ curriculum learning by optimizing the prototype c^k first, then gradually increasing the deformation field v^k basis vector dimensions. We train for 1000 epochs using the Adam optimizer with a $1e^{-3}$ learning rate.
- Second, we augment centered partial observation inputs with random rotations (0° to 20° on all axes with 0.5 probability). To balance the parameter loss $\mathcal{L}_{\text{para}}$ defined in Eq. (6), we set $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$. For the reconstruction loss in Eq. (7), we use $\lambda_{\text{CD}} = 1.0$ and $\lambda_{\text{para}} = 0.1$. Training runs for 30 epochs using Adam with learning rate $1e^{-3}$.

We use Pytorch3D to position 8 virtual perspective cameras in a cube configuration around the normalized target object for semantic prototype construction. We also position a point light to enhance the surface detail and generate realistic rendering. We employ the DINOv2 pre-trained model for feature extraction to generate pixel-aligned feature descriptors. We gathered 200 nearest neighbors for each sampled point using the KNN algorithm described in Eq. (9) to aggregate semantic features from the dense semantic point to our deep linear shape reconstruction.

Pose Size Estimator. To train the pose estimation network, we balance the loss function defined in Eq. (18) with hyperparameters: $\lambda_1 = 2.0$, $\lambda_2 = 2.0$, $\lambda_3 = 15.0$, $\lambda_4 = 0.3$, $\lambda_5 = 0.3$. We train the network using ADAM optimizer with CyclicLR scheduler in triangular2 mode with base learning rate $lr = 2e^{-5}$ and max learning rate $lr = 5e^{-4}$. To deal with the symmetry issue, we follow [61] to transform the rotation to canonical.

Instance-Segmentation. We follow the previous literature [8, 38, 39, 41, 72] of category-level pose estimation for fair comparisons, using the same segmentation mask as the baseline methods. Specifically, the provided Mask-RCNN segmentation results are used in REAL275, and the GT segmentation mask is used for testing HouseCat6D.

Training process details. The training process of SSR module is in two stages, where the first stage is used to generate the shape parameters as supervision signal for the second stage, in the second stage, the network is trained with noisy sensor point cloud to make shape reconstruction network robust against noise. The pose estimation part is trained independently after both of these stages.

2. Evaluation on Instance Reconstruction

We evaluate our instance reconstruction with chamfer distance (CD) on the HouseCat6D Dataset. We measure the Chamfer distance between our reconstructed pointclouds and the ground-truth pointclouds sampled from the CAD model in NOCS space. We represent the CD metric with

$$d_{\text{CD}}(M, \hat{M}) = \sum_{x\in M} \min_{y\in \hat{M}} \|x - y\|^2 + \sum_{y\in \hat{M}} \min_{x\in M} \|x - y\|^2. \quad (21)$$

where M and $\hat{M}$ denote the reconstructed point cloud and the ground-truth point cloud sampled from the CAD model, respectively. The term $\|\cdot\|^2$ represents the squared Euclidean distance, and min computes the nearest neighbor distance for each point in one point cloud to the other.

As shown in Tab. 4, we achieve 2.39×10^{-3} mean chamfer distance of our reconstruction.

To evaluate the reconstruction performance of our method, we compare the reconstructed shape with the groundtruth shape using the Chamfer Distance. We report the per-category shape reconstruction result in Tab. 4.

Category	Bottle	Box	Can	Cup	Remote	Teapot	Cutlery	Glass	Tube	Shoe	**Average**
Chamfer Distance	1.59	7.79	3.45	1.77	1.18	2.79	0.46	1.93	1.26	1.70	2.39

Table 4. Reconstruction performance for the categories in HouseCat6D dataset [29]. Evaluated with Chamfer Distance metric $\left(10^{-3}\right)$.

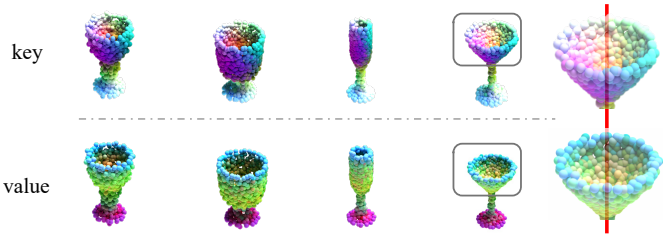

Figure 6. Visualization of feature point cloud using PCA. Upper row: Key feature. Bottom row: Value feature. The zoom-in visualizations indicate embedding changes in the key feature around symmetric areas, which are negligible for the value feature.

Keypoint	λ_2	λ_3	**5°2cm**	**10°2cm**	IoU75
96	**2.0**	**15.0**	**24.8**	**55.4**	**60.6**
128	2.0	15.0	24.2	52.9	57.8
96	0.5	15.0	23.5	54.6	58.9
96	2.0	3.0	23.8	53.4	59.7

Table 5. Hyperparameter comparison, the default setting is in bold.

3. More Ablation Studies

In our Global Context Enhancement Feature Fusion module, we demonstrate the best result using the DINO value feature from partial observation and the DINO key feature from semantic global reconstruction. For symmetric objects, e.g., "glass", as shown in Figure 10, the value features are symmetric and ambiguous around the rotational axis, while the key features are embedded with positional code and thus distinctive, which is helpful when handling symmetric cases. We also report the quantitative results for "glass" in Tab. 6, showing higher performance when using key features.

Experimental results of the full benchmark on the efficacy of global context feature fusion in Tab. 7 show that our feature fusion strategy can enhance pose estimation performance effectively.

We additionally conduct experiments on the robustness to hyper-parameter variation. Tab. 5 shows the results of hyper-parameter testing, including the number of key points and the hyper-parameters of the loss function defined in Eq. (18). The results are slightly different but still stable.

4. More Results and Visualization

Intra-class semantic variation. Our method extracts semantic features from the powerful pre-trained large foundational model DINOv2, which is capable of handling intra-class semantic variation effectively as shown in Fig. 7. On the other hand, our SSR module is designed to aggregate the categorical semantics into the reconstructed instances effectively through the learned deep linear shapes. As shown in Fig 5 (B) of the main text, with our SSR module, the semantics are consistent across the instances with shape variance demonstrated in Fig. 7.

Metrics	Value feature	Key feature (Ours)	Difference
5°2cm	59.16	**64.86**	5.70
5°5cm	62.50	**66.11**	3.61
10°2cm	86.29	**92.30**	6.01
10°5cm	91.72	**94.49**	2.77

Table 6. Quantitative comparison for symmetric category "glass".

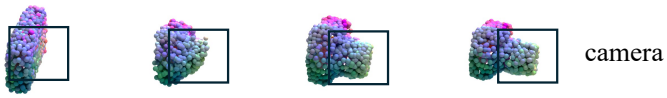

Figure 7. Semantics consistency for the "camera" class despite geometry changes in the lens area.

We visualize more results of 3D bounding box prediction of our GCE-Pose for the HouseCat6d dataset in Fig. 9 and NOCS dataset in Fig. 8. We choose four images per scene in the test set and indicate the groundtruth results in green and the prediction in red.

To demonstrate the effectiveness of our method in improving pose estimation and NOCS coordinates prediction, we provide further visualization of 3D bounding box prediction and evaluate NOCS errors on key points. Fig. 10 presents a qualitative comparison of AG-Pose [41] (DINO) and our proposed method on the Housecat6D dataset. The visualizations display the NOCS error map overlaid on the images, where red dots indicate higher errors and green dots indicate lower errors. Our qualitative results show that our method achieves high precision in predicting NOCS coordinates, which is essential for accurate pose estimation. Additionally, we have included videos in the attached file that showcase the complete results across the full sequences of the HouseCat6D dataset.

Furthermore, we conducted experiments on HouseCat6D using the state-of-the-art shape-prior-based method, Seld-DPDN [38]. Appendix 3 highlights the quantitative results, demonstrating the advantages of incorporating shape and

Method	Ins. recon.	Mean shape	Geo.	Sem.	5°2cm	5°5cm	10°2cm	10°5cm	IoU50	IoU75
(0) AG-Pose (DINO)	×	×	×	×	21.34	22.27	52.00	55.12	76.79	56.07
(1) Ours (Only Geo.)	✓	×	✓	×	22.73	24.28	52.83	56.51	78.59	58.17
(2) Ours (Only Sem.)	×	✓	×	✓	22.16	23.65	52.44	57.31	78.10	55.07
(3) Ours (Mean shape, Geo. & Sem.)	×	✓	✓	✓	23.37	24.24	53.85	56.83	79.27	60.46
(4) GCE-Pose (full pipeline)	✓	✓	✓	✓	24.85	25.73	55.44	58.43	79.15	60.61

Table 7. Ablation study on different global priors. Ins. recon: instance shape reconstruction as the prior; Mean shape: mean shape of the categories as the prior; Geo.: geometric features from the global prior; Sem.: semantic features for the global prior.

Dataset	Method	Shape Prior	Semantic Prior	5°2cm	5°5cm	10°2cm	10°5cm	IoU50	IoU75
HouseCat6D	Self-DPDN [38]	✓		6.4	6.9	22.2	25.8	56.2	26.0
	GCE-Pose (Ours)	✓	✓	24.8	25.7	55.4	58.4	79.2	60.6

Table 8. Quantitative comparison of category-level object pose estimation with shape-prior on the HouseCat6D dataset [29].

semantic priors for category-level object pose estimation.

We showcase semantic prototype and their corresponding semantic transfers with more classes in the HouseCat6D dataset, as shown in Fig. 11. The classes, listed from top to bottom, include box, bottle, glass, teapot, cup, shoe, can, tube, cutlery, and remote.

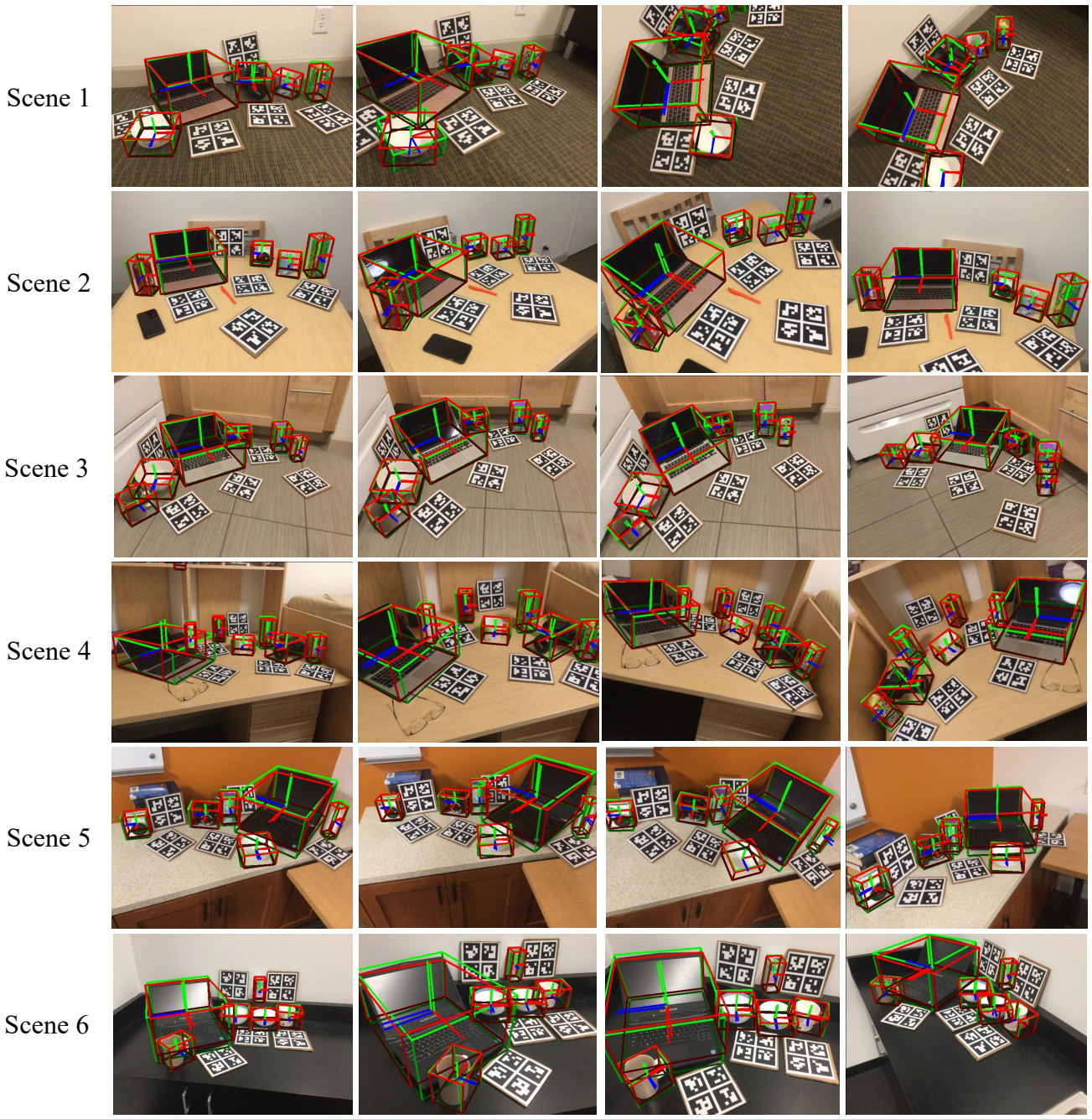

Figure 8. NOCS dataset bounding box visualization. Green indicates GT, and red indicates prediction results.

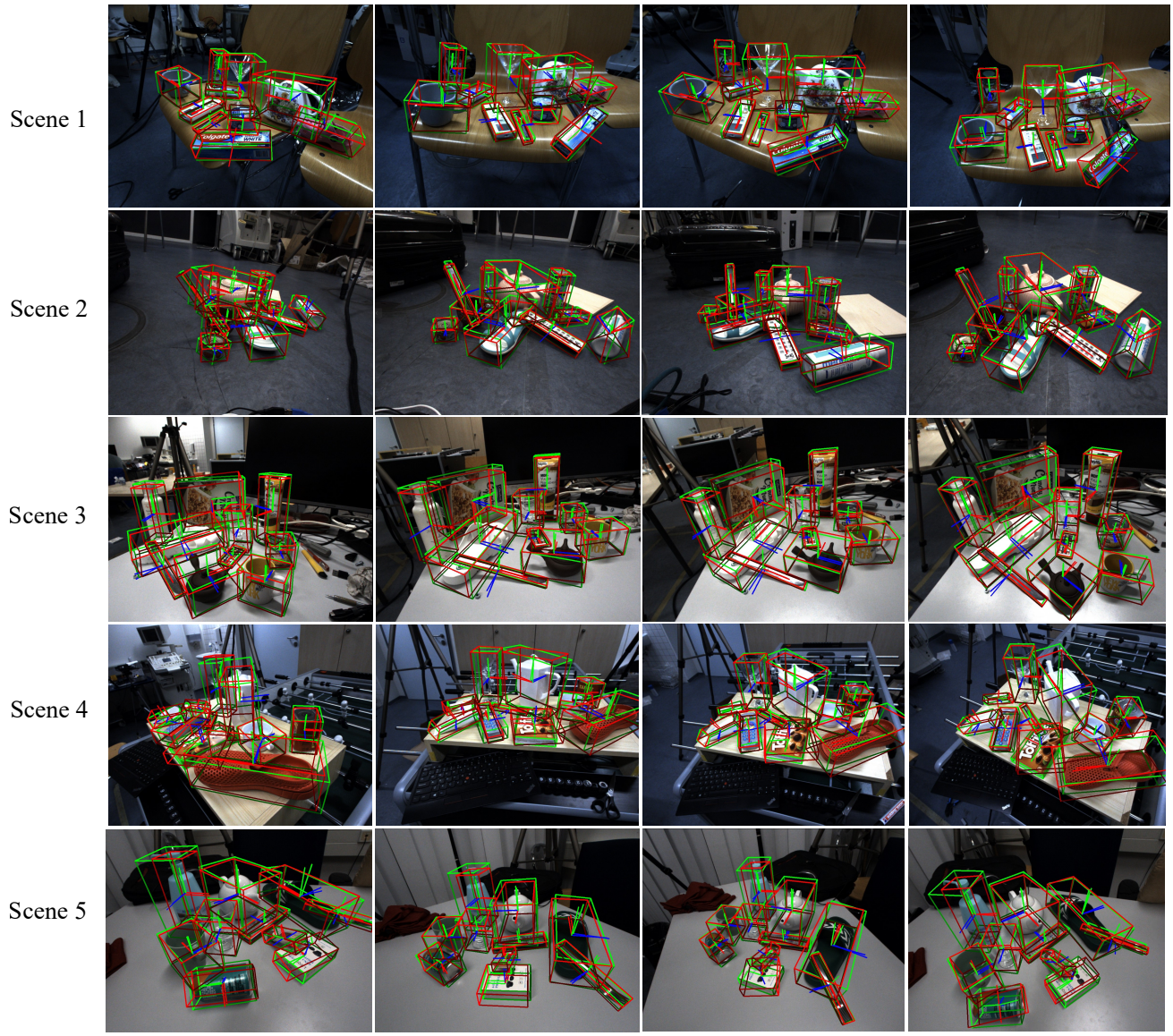

Figure 9. HouseCat6D bounding box visualization. Green indicates GT, and red indicates prediction results.

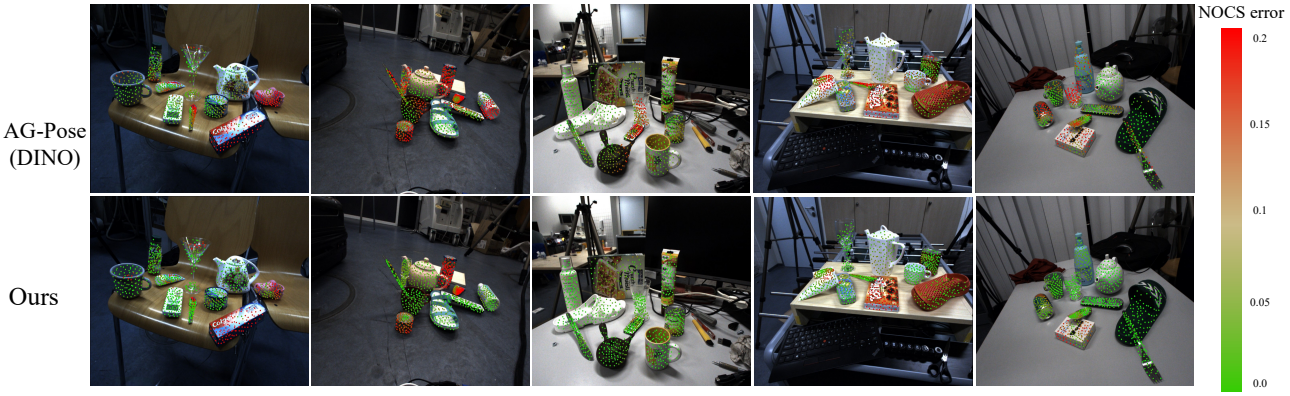

Figure 10. Visualization of HouseCat6D Keypoint NOCS Error. Red indicates a high error; green indicates a low error.

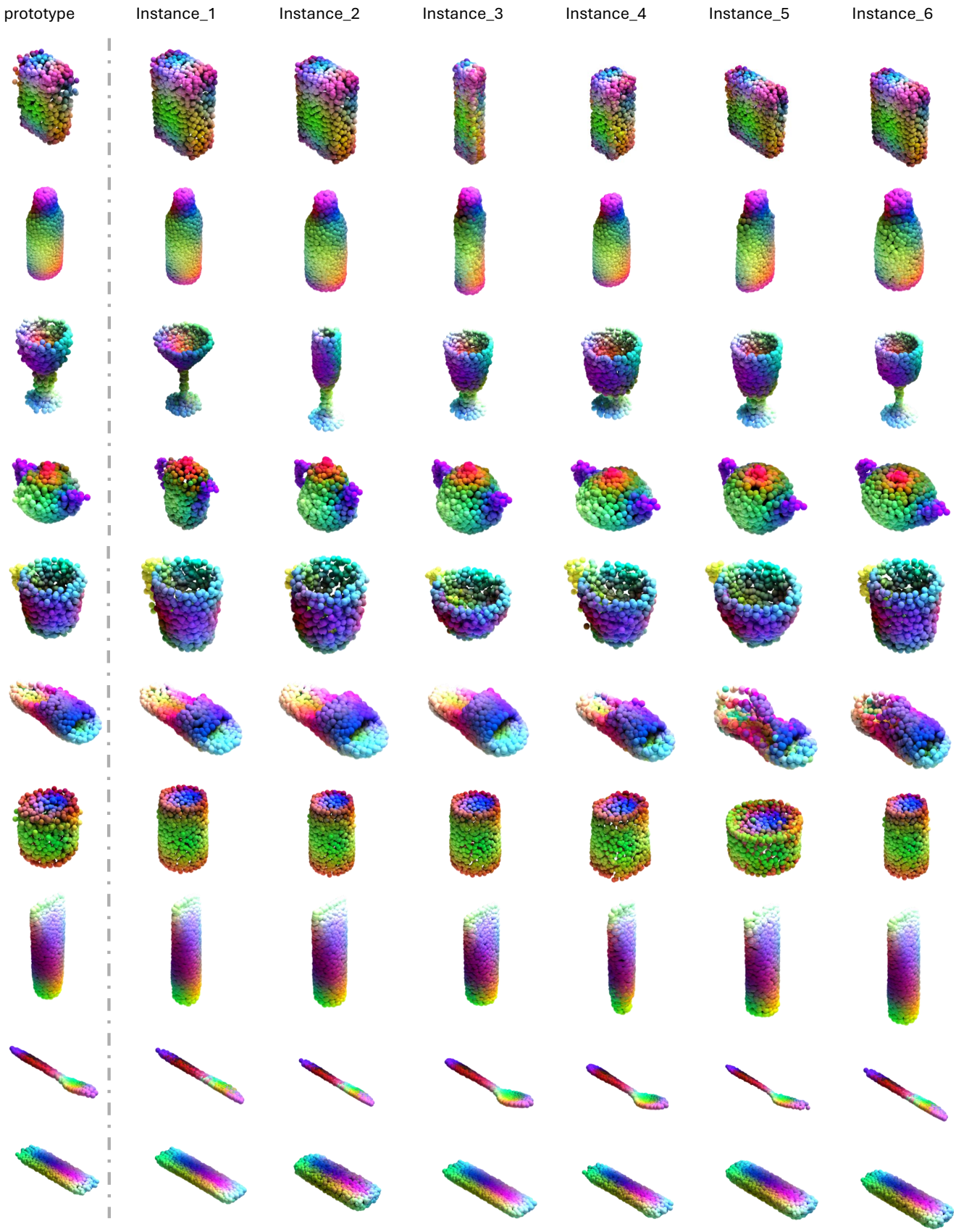

Figure 11. Visualization of Semantic prototypes and in-class semantic transfer results in HouseCat6D dataset.