One-shot 3D Object Canonicalization based on Geometric and Semantic Consistency

Supplementary Material

A. Zero-Shot 3D Semantic Perception

This section provides a comprehensive overview of the zero-shot 3D semantic perception module. We will detail the rendering, semantic label generation, and the 2D to 3D label fusion used for 3D semantic calculations.

Multi-view Rendering. Our approach begins with rendering ten multi-view images for each object. The camera is positioned on a sphere with a radius of $2.25m$. Elevation angles are set at $\{-20°, 10°, 40°\}$, while azimuth angles are uniformly sampled along circles parallel to the x-y plane for each elevation angle. This setup ensures comprehensive coverage of the object's geometry from multiple perspectives.

Semantic Label Generation. GLIP requires both images and their corresponding semantic labels as inputs. Manually annotating semantic labels for each category is labor-intensive and time-consuming. To address this, we leverage Large Language Models (LLMs) like ChatGPT, which can generate semantic labels for any category in a zero-shot manner. Specifically, we prompt ChatGPT to identify a set of semantic region names useful for determining object orientation and to generate semantic labels for arbitrary categories.

2D to 3D label fusion. In the 2D to 3D processes, we aim to project detections onto 3D vertices efficiently. Therefore, constraints are incorporated into the formula to ensure that the 2D bounding box of a part's semantic instance lies within the object's overall 2D bounding box. With the parts detection results in muti-view images generated by GLIP, we calculate the proportion of each vertex belonging to a specific semantic. For each vertex $\mathbf{x}_l$ on face f, we compute the proportion P_l^s of vertex $\mathbf{x}_l$ belonging to semantic s as follows:

$$P_l^s = \frac{\sum_k [\text{VIS}_k(f(\mathbf{x}_l))][\text{INM}_k(f(\mathbf{x}_l))][\text{INB}_k]}{\sum_k [\text{VIS}_k(f(\mathbf{x}_l))]}, \quad (1)$$

where $\text{VIS}_k(f(\mathbf{x}_l))$ denotes whether the face $f(\mathbf{x}_l)$ is visible in view k, $[\cdot]$ indicates the Iverson bracket, $\text{INM}_k(f(\mathbf{x}_l))$ is whether the pixel p corresponding to f in view k is inside the semantic mask M_k^s. INB_k denotes whether the bounding box of the mask M_k^s is entirely enclosed within the shape in view k.

B. Support-Plane-based Object Initialization

To obtain the right semantic perception by accommodating the influence of the initial pose, we implement the Support-plane-based Object Initialization to generate the initial poses. For a test object in arbitrary pose, we first compute the 3D convex hull and refine it to polygonal faces [5]. Then we project the centroid of mesh, which is the weighted average of object vertexes, onto each polygonal face. If the projected point is within the polygon's boundaries, the polygon is the support polygon, and its plane is the support plane.

For each support plane, we can initialize the object by aligning the plane parallel to the ground. The initial rotation $\mathbf{R}_i$ for the i-th support plane is derived from the axis-angle representation, defined by the rotation axis $\mathbf{u}_i$ and angle θ_i. Specifically, the normal vector of the supporting plane is denoted as $\mathbf{n}_s^i$, and the ground's normal vector is $\mathbf{n}_g^i$.

The axis of rotation $\mathbf{u}_i$ is determined by the cross product of the normal vectors $\mathbf{n}_s^i$ and $\mathbf{n}_g^i$:

$$\mathbf{u}_i = \mathbf{n}_s^i \times \mathbf{n}_g^i \quad (2)$$

The angle θ_i between the two normal vectors $\mathbf{n}_s^i$ and $\mathbf{n}_g^i$ is given by:

$$\theta_i = \arccos\left(\frac{\mathbf{n}_s^i \cdot \mathbf{n}_g^i}{\|\mathbf{n}_s^i\|\|\mathbf{n}_g^i\|}\right) \quad (3)$$

The rotation matrix $\mathbf{R}_i$ is then calculated using Rodrigues' rotation formula:

$$\mathbf{R}_i = \mathbf{I} + \sin(\theta_i)\mathbf{K}_i + (1 - \cos(\theta_i))\mathbf{K}_i^2 \quad (4)$$

where $\mathbf{K}_i$ is the skew-symmetric matrix derived from the axis of rotation $\mathbf{u}_i = [u_x, u_y, u_z]^T$.

C. Canonicalization Hypothesis Generation

To mitigate the noise in 3D semantic perception, we represent the semantic points using a 3D Gaussian distribution. For the prior semantic points $(\mathbf{X}_r, \mathbf{C}_r^s)$, the semantic distribution is represented as a normal distribution $\mathcal{G}_r^s$:

$$\mathcal{G}_r^s \sim \mathcal{N}(\boldsymbol{\mu}_r^s, \boldsymbol{\Sigma}_r^s) \quad (5)$$

where $\boldsymbol{\mu}_r^s$ and $\boldsymbol{\Sigma}_r^s$ denote the mean and covariance of the semantic distribution, respectively:

$$\boldsymbol{\mu}_r^s = \frac{\sum_{k=1}^L c_r^{s,k} \mathbf{x}_r^k}{\sum_{k=1}^L c_r^{s,k}} \quad (6)$$

Table 1. **3D object canonicalization on the ShapeNet dataset.** Lower scores indicate better performance.

Method	Prior num.	*Car*		*Table*		*Chair*		*Plane*		*Couch*		*Lamp*		*Water.*	
		IC	GEC	IC	GEC	IC	GEC	IC	GEC	IC	GEC	IC	GEC	IC	GEC
CaCa [7]	2442	**0.059**	0.102	0.659	1.334	**0.120**	**0.300**	**0.115**	**0.230**	1.030	1.662	**0.089**	**1.979**	**0.083**	**0.156**
ConDor [6]	2442	0.260	0.313	**0.495**	0.891	0.386	0.668	0.228	0.290	**0.387**	0.598	0.964	3.085	0.268	0.395
Ours	1	0.077	**0.087**	0.702	**0.783**	0.558	0.656	0.224	0.238	0.479	**0.544**	2.651	2.874	0.141	0.157
Method	Prior num.	*Bench*		*Speaker*		*Cabinet*		*Firearm*		*Monitor*		*Cell.*		**Avg.**	
		IC	GEC	IC	GEC	IC	GEC	IC	GEC	IC	GEC	IC	GEC	IC	GEC
CaCa [7]	2442	0.496	0.776	1.844	**2.044**	1.544	1.699	**0.051**	**0.187**	0.500	0.625	**0.300**	0.548	**0.570**	0.942
ConDor [6]	2442	0.646	1.004	**1.312**	2.061	**0.726**	1.199	1.216	1.467	0.632	0.886	0.487	0.782	0.591	1.028
Ours	1	**0.355**	**0.411**	1.945	2.134	0.987	**1.160**	0.261	0.275	**0.324**	**0.364**	0.430	**0.504**	0.703	**0.784**

Table 2. **Comparison of Metrics for Evaluating 3D Object Canonicalization.**

Metric	*Aeroplane*	*Car*	*Bowl*	*Bottle*	*Camera*	*Can*	*Mug*	**Avg.**
IC	2.911	3.260	3.500	3.863	1.726	2.384	0.920	2.652
CC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
GEC	2.891	3.473	3.902	3.743	1.777	2.598	0.897	2.754

$$\mathbf{\Sigma}_r^s = \frac{\sum_{k=1}^{L} c_r^{s,k}(\mathbf{x}_r^k - \boldsymbol{\mu}_r^s)(\mathbf{x}_r^k - \boldsymbol{\mu}_r^s)^T}{\sum_{k=1}^{L} C_r^{s,k}} \tag{7}$$

Similarly, for the initialized object $(\mathbf{X}_i, \mathbf{C}_i^s)$, the semantic distribution is modeled as $\mathcal{G}_i^s$:

$$\mathcal{G}_i^s \sim \mathcal{N}(\boldsymbol{\mu}_i^s(\boldsymbol{\omega}_i^s), \mathbf{\Sigma}_i^s(\boldsymbol{\omega}_i^s)) \tag{8}$$

where $\boldsymbol{\mu}_i^s(\boldsymbol{\omega}_i^s)$ and $\mathbf{\Sigma}_i^s(\boldsymbol{\omega}_i^s)$ represent the mean and covariance of the semantic distribution, respectively:

$$\boldsymbol{\mu}_i^s(\boldsymbol{\omega}_i^s) = \frac{\sum_{k=1}^{I} c_i^{s,k} Exp(\boldsymbol{\omega}_i^s)\mathbf{x}_i^k}{\sum_{k=1}^{I} c_i^{s,k}} \tag{9}$$

$$\mathbf{\Sigma}_i^s(\boldsymbol{\omega}_i^s) = \frac{\sum_{k=1}^{I} c_i^{s,k} \mathbf{y}_k \mathbf{y}_k^\top}{\sum_{k=1}^{N} C_i^{s,k}} \tag{10}$$

where $\mathbf{y}_k = Exp(\boldsymbol{\omega}_i^s)\mathbf{x}_i^k - \boldsymbol{\mu}_i^s(\boldsymbol{\omega}_i^s)$.

D. Additional Experimental Results

Evaluation Metric Choice. We observe that the Category-Level Consistency (CC) metric tends to degrade when the canonicalizing transformation is set to identity. As shown in Table 2 , for objects in arbitrary poses, setting the canonical pose to identity results in high IC and GEC metric values, indicating poor consistency across different object poses. In contrast, the CC metric misleadingly suggests that objects in arbitrary poses, without any canonicalization, are well-aligned. Consequently, GEC proves to be more reliable in reflecting the consistency of different objects after canonicalization. Based on this observation, we use the IC and GEC metrics to evaluate the performance of 3D object canonicalization, as they provide a more robust and accurate assessment compared to the CC metric.

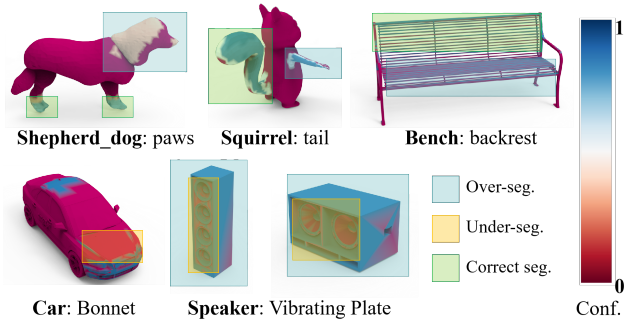

Figure 1. **Zero-shot semantic segmentation examples.**

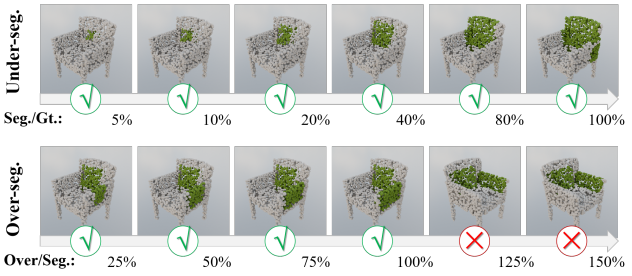

Figure 2. **Robustness of our alignment method under varying segmentation results.**

Characteristics of 3D Segmentation. The failure cases of 3D segmentation can be categorized into two main types:

1. **Under Segmentation**: As shown in figure 1, certain objects (e.g., the car in the figure) were not fully segmented.
2. **Over-segmentation**: Parts that do not belong to the target label were incorrectly assigned the same label, such as the Shepherd dog in figure 1. Additionally, for the "Speaker" category, segmentation completely failed, with significant over-segmentation observed.

This represents a limitation of our method, which we hypothesize is due to the insufficient training data for the "Vibrating Plate" label in the VLM model.

Characteristics of Alignment. To evaluate the robustness of our alignment method under segmentation noise, we introduced perturbations to manually segmented labels and tested the performance under **under-segmentation** and **over-**

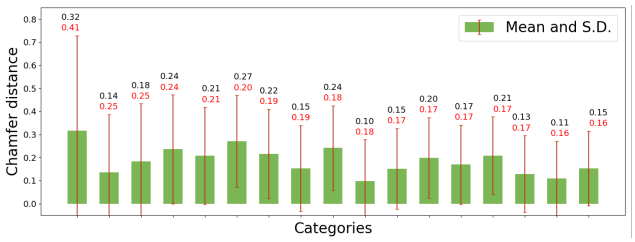

Figure 3. **Statistics on handling shape diversity.**

segmentation conditions. As shown in figure 2, our method demonstrates strong robustness to noise.

Handling Shape Diversity. To demonstrate the ability of our method to handle shape diversity, we used the Chamfer Distance metric to measure shape variations. As shown in figure 3, we visualized the 17 categories with the highest standard deviation in the COD dataset. The results show significant shape differences in our aligned data, further validating the robustness and effectiveness of our method.

Additional Results. We provide additional comparative results against state-of-the-art methods [6, 7], evaluated under the condition that the methods [6, 7] are trained on full training datasets.

As Table 1 shows, on the simulated dataset ShapeNet, where extension training data is provided, the methods [6, 7] achieve better performance using the full training set. In stark contrast, our method relies on only a single prior model, requiring less than $\frac{1}{2000}$ of the training data used by the other two methods, achieving comparable performance.

In Table 3 and Table 4, we provide results evaluated on two real datasets [3, 8]. These two datasets just provide a limited training set for each category. Moreover, the training data provided in these two datasets is synthetic, while the provided test data is real scanned. As shown, our method achieves state-of-the-art, even existing domain gaps between the simulated prior and the real data, which demonstrates the robustness of our method.

Furthermore, more qualitative results are shown in Figure 4 5 6. More results can be found at Project homepage: COD.com.

On Downstream Tasks. We tested our dataset's effectiveness via ablation studies on category-level rotation estimation tasks. Results on the ShapeNet sub-test set (from [1]) (Table 5) show significant accuracy gains with our curated dataset over raw data.

E. Limitations

Our approach uses zero-shot 3D semantic perception, rendering 2D semantic detection accuracy crucial for reliable canonicalization. We utilize a multi-hypothesis initialization and selection strategy to better accommodate semantic errors. However, enhancing zero-shot 3D semantic perception precision remains a promising research direction.

Table 3. **3D object canonicalization on the DREDS dataset.**

Method	Prior num.	*Aeroplane*		*Car*		*Bowl*		*Bottle*	
		IC	GEC	IC	GEC	IC	GEC	IC	GEC
CaCa [7]	135	0.421	0.548	0.901	0.756	1.543	1.738	**0.029**	0.222
ConDor [6]	135	0.235	0.381	**0.045**	**0.068**	0.266	0.306	0.078	0.083
Ours	1	**0.051**	**0.058**	0.103	0.118	**0.011**	**0.012**	0.031	**0.034**
Method	Prior num.	*Camera*		*Can*		*Mug*		**Avg.**	
		IC	GEC	IC	GEC	IC	GEC	IC	GEC
CaCa [7]	135	1.177	1.442	1.787	2.042	0.622	0.677	0.926	1.061
ConDor [6]	135	**0.283**	1.296	0.231	0.492	0.067	0.084	**0.184**	0.447
Ours	1	1.116	**1.177**	**0.030**	**0.037**	**0.018**	**0.019**	0.194	**0.208**

Table 4. **3D object canonicalization on the NOCS dataset.**

Method	**Prior num.**	*Laptop*		*Mug*		*Bowl*		*Bot-*
		IC	GEC	IC	GEC	IC	GEC	IC
CaCa [7]	135	0.596	0.755	0.733	0.753	1.386	1.635	**0.053**
ConDor [6]	135	**0.163**	0.315	**0.061**	**0.076**	0.157	0.185	0.068
Ours	1	0.187	**0.222**	0.091	0.082	**0.033**	**0.036**	0.144
Method	**Prior num.**	*-tle*	*Camera*		*Can*		**Avg.**	
		GEC	IC	GEC	IC	GEC	IC	GEC
CaCa [7]	135	0.157	1.291	1.502	1.522	1.796	0.930	1.100
ConDor [6]	135	**0.079**	**0.454**	**1.046**	0.239	0.472	0.193	0.359
Ours	1	0.149	0.874	1.067	**0.099**	**0.099**	**0.145**	**0.129**

Table 5. **Downstream Task.** Accuracy evaluated within 30 degrees.

Train data	*Car*	*Chair*	*Plane*	*Couch*	*Lamp*	*Water.*
Objaverse-lvis	16.7	26.3	15.8	31.6	5.3	10.5
Shapenet	16.7	31.6	26.3	31.6	5.3	21.1
COD	**22.2**	**36.8**	**31.6**	**31.6**	**5.3**	**26.3**
Train data	*Bench*	*Speaker*	*Cabinet*	*Firearm*	*Cell.*	***Avg.***
Objaverse-lvis	21.1	5.3	0.0	15.8	10.5	14.4
Shapenet	15.8	5.3	0.0	31.6	15.8	18.3
COD	**21.1**	**10.5**	**5.3**	**31.6**	**15.8**	**21.6**

References

[1] Rohith Agaram, Shaurya Dewan, Rahul Sajnani, Adrien Poulenard, Madhava Krishna, and Srinath Sridhar. Canonical fields: Self-supervised learning of pose-canonicalized neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4500–4510, 2023. 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[3] Qiyu Dai, Jiyao Zhang, Qiwei Li, Tianhao Wu, Hao Dong, Ziyuan Liu, Ping Tan, and He Wang. Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects. In *European Conference on Computer Vision*, pages 374–391. Springer, 2022. 3, 4

[4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 5

[5] Hongbo Fu, Daniel Cohen-Or, Gideon Dror, and Alla Sheffer. Upright orientation of man-made objects. In *ACM SIGGRAPH 2008 papers*, pages 1–7. 2008. 1

Figure 4. **Visual results of 3D object canonicalization on simulated dataset [2]**

Figure 5. **Visual results of 3D object canonicalization on real dataset [3, 8]**

[6] Rahul Sajnani, Adrien Poulenard, Jivitesh Jain, Radhika Dua, Leonidas J Guibas, and Srinath Sridhar. Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16969–16979, 2022. 2, 3

[7] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural information processing systems*, 34:24993–25005, 2021. 2, 3

[8] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 3, 4

[9] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 5

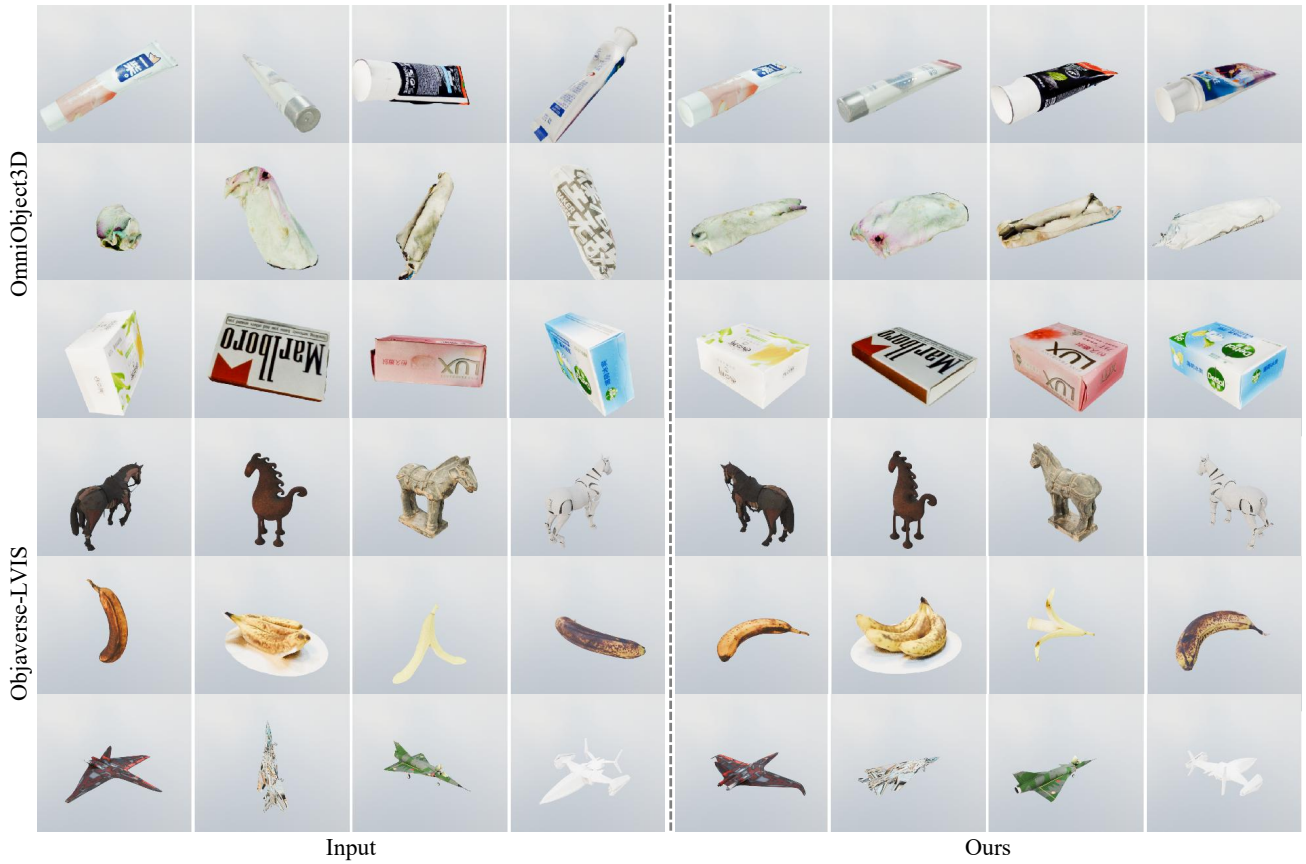

Figure 6. **Visual results of 3D object canonicalization in the wild.** The top three rows are from OmniObject3D [9], while the bottom three rows are from Objaverse-LVIS [4].