Improving Zero-Shot Abstraction of Unknown Spacecraft 3D Shape as Primitive Assembly

Tae Ha Park*, Emily Bates† and Simone D'Amico‡

This paper investigates the problem of zero-shot abstraction of 3D structures of unknown targets in space from single images captured on-board the satellites. Such capability is crucial to rapidly gathering coarse knowledge of the environment around a spaceborne agent, facilitating downstream guidance, navigation and control algorithms for rendezvous and proximity operations with applications including on-orbit inspection and debris removal. The previous work by the authors [19] developed a Convolutional Neural Network (CNN) model and a supervised training pipeline to reconstruct the target's 3D model as an assembly of a fixed number of superquadric primitives. This work proposes several improvements in an effort to augment the CNN model so that it generalizes better to previously unseen spacecraft. The preliminary results indicate that the proposed methods allow reconstruction using a variable number of primitives depending on the structural complexity of the target. However, they also reveal that it is extremely difficult to bridge the performance gap against previously unseen targets without significant expansion of the training dataset in terms of the diversity of the spacecraft models.

keywords: Rendezvous, Deep Learning, Shape Recovery, Pose Estimation, Space Situational Awareness

1. Introduction

Autonomous rendezvous with non-cooperative Resident Space Objects (RSO) is a critical technology that enables numerous missions for sustainable space development such as In-Space Assembly and Manufacturing (ISAM) and active debris removal. Many recent works have focused on utilizing a monocular camera and Deep Neural Networks (DNN) to process an image input and predict the 6D pose (i.e., position and orientation) of the target with respect to the servicer spacecraft [5, 8, 10, 22–24]. While these works contributed to advancing the state-of-the-art for using DNNs in spaceborne computer vision applications, one major shortcoming is that they assume prior knowledge of the 3D structure of the target RSO. In order to support missions in which such an assumption cannot be made (e.g., debris removal), DNNs and the overarching Guidance, Navigation and Control (GN&C) algorithms must be capable of simultaneously recovering the unknown 3D geometry of the target and performing pose estimation from a sequence of image data. The problem is akin to the well-known Simultaneous Localization and Mapping (SLAM) problem in robotics [18, 27], though tracking the rigid-body motion of an unknown RSO in space further requires the knowledge of the target's mass distribution.

A number of recent works have explored Deep Learning (DL) methods for vision-based spaceborne SLAM applications. For example, Mergy et al. [15] and Caruso et al. [3] apply various Neural Radiance Field (NeRF) models [16] to implicitly reconstruct the scene of an unknown RSO from unlabeled images, while Legrand et al. [11] first trains a NeRF and uses it to generate images to train a separate pose estimation NN. Similarly, Barad et al. [1] uses 3D Gaussian splatting [9] to reconstruct the target model as a cloud of 3D Gaussian blobs while simultaneously optimizing for poses associated with each unlabeled images.

As opposed to the aforementioned *online* methods, the authors' previous work [19] investigated an *offline* method which seeks to rapidly abstract a coarse, normalized 3D model of an unknown RSO from a single image of the target captured during Rendezvous and Proximity Operations (RPO). This is achieved by a novel Convolutional Neural Network (CNN) model whose training is supervised using the SPE3R dataset [20] consisting of high-fidelity Unreal Engine synthetic images (see Fig. 1) and 3D spacecraft mesh model labels. The CNN is trained to predict an assembly of superquadric primitives [2] which each describe a wide range of simple geometric shapes using only two parameters. The motivation was that this coarse, compact and normalized model could be used to initialize and kickstart any downstream SLAM algorithm. However, this research also revealed a number of limitations. One is that the CNN is designed to predict a fixed number of primitives regardless of the perceived structural complexity of the target RSO. The other is that using 3D mesh models as ground-truth

*Nara Space Technology, Seoul, Republic of Korea, thpark@naraspace.com

†Department of Aeronautics & Astronautics, Stanford University, Stanford, CA 94305, USA, ebates2@stanford.edu

‡Department of Aeronautics & Astronautics, Stanford University, Stanford, CA 94305, USA, damicos@stanford.edu

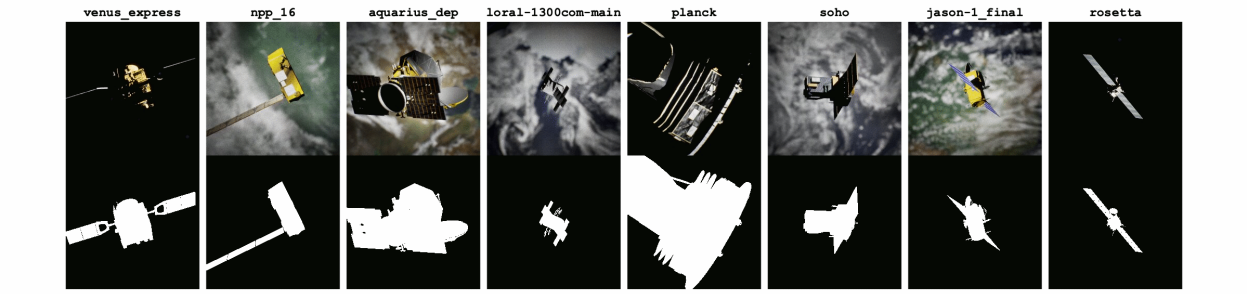


Fig. 1: Visualization of select RGB images and binary masks of the SPE3R dataset.

labels inevitably introduces pose biases, especially since spacecraft have no canonical "up" and "forward" directions to allow for consistent mesh alignment and prediction. Coupled with a lack of diversity of spacecraft models in the SPE3R dataset, it forces the CNN to essentially "memorize" the spacecraft, unable to generalize to images of a new RSO previously unseen during training.

The obvious remedy to memorization is to augment the SPE3R dataset to include even more spacecraft models. For example, ShapeNet [4] boasts over 50,000 unique 3D models across 55 common object categories, while recent Objaverse [7] and ObjaverseXL [6] respectively contain 800K and 10M+ annotated 3D models. Such quantity allows learning rich representations particular to various 3D-related tasks and object classes, showing remarkable zero-shot performances. Recently, Mathihalli et al. [14] took a Zero123XL model [12] pre-trained on ObjaverseXL and fine-tuned it on a set of 190 spacecraft models which subsumes the SPE3R dataset. Their experiments showed improved performance on the task of novel view synthesis, demonstrating that the size and diversity of the dataset matter significantly. However, it still remains a challenging task to collect and assemble such a quantity of spacecraft 3D models due to their scarcity and the security-driven nature of the aerospace industry. Therefore, the natural follow-up question is how much the reconstruction performance can be maximized given a dataset with the aforementioned restrictions.

In response, this work investigates how various elements of the CNN architecture and supervised training pipeline of Park and D'Amico [19] affect the overall reconstruction quality and generalization to unseen models while the training is subject to the same SPE3R dataset. To that end, three areas of improvement are explored at the superquadric primitive, CNN architecture and training pipeline levels. The preliminary results indicate that the proposed methods enable reconstruction using a variable number of primitives depending on the size and complexity of the target spacecraft, allowing even more compact representations without loss of reconstruction quality. However, the results also show that the proposed methods are simply insufficient to ever bridge the performance gap on previous unseen targets without significantly augmenting the training dataset with more diverse spacecraft shapes. This work reaffirms the importance of constructing large-scale datasets for spacecraft 3D reconstruction and provides insight into how one might resolve such a problem.

This paper is organized as follows. Section 2 provides details on the proposed improvements, followed by the experimental setup, results and discussions in Sec. 2. The paper ends with conclusions in Sec. 4.

2. Proposed Methods

2.1 *Superquadric Sampling Strategy*

The main loss function that drives the training of zero-shot 3D abstraction is the bi-directional Chamfer distance ($\mathcal{L}_{\text{chamfer}}$) defined for two sets of point clouds $(\boldsymbol{\mathcal{X}}_c, \boldsymbol{\mathcal{Y}}_c)$ as

$$\begin{aligned}\mathcal{L}_{\text{chamfer}}(\boldsymbol{\mathcal{X}}_c, \boldsymbol{\mathcal{Y}}_c) = \frac{1}{|\boldsymbol{\mathcal{X}}_c|} \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_c} \min_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}_c} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \\ + \frac{1}{|\boldsymbol{\mathcal{Y}}_c|} \sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}_c} \min_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}_c} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2\end{aligned} \quad [1]$$

The loss encourages both point clouds to subsume one another by minimizing the distance metric between the closest pairs of points. Successful implementation of the Chamfer distance loss depends on how uniformly one can sample surface points from each superquadric primitive across the entire assembly, where any point on the superquadric with the size parameters $\boldsymbol{\alpha} \in \mathbb{R}^3$ and the shape parameters $\boldsymbol{\epsilon} \in \mathbb{R}^2$ can be retrieved via

$$\boldsymbol{r}(\eta, \omega; \boldsymbol{\alpha}, \boldsymbol{\epsilon}) = \begin{bmatrix} \alpha_1 \cos^{\epsilon_1} \eta \cos^{\epsilon_2} \omega \\ \alpha_2 \cos^{\epsilon_1} \eta \sin^{\epsilon_2} \omega \\ \alpha_3 \sin^{\epsilon_1} \eta \end{bmatrix} \quad [2]$$

In general, the above formulation of superquadrics suffers from numerical instability as $\epsilon_i \to 0$, resulting in samples concentrated around the edges and corners of the primitives when uniformly sampling the elevation and azimuth angles (η, ω). Park and D'Amico [19] resolved the numerical stability issue by leveraging the dual superquadrics, allowing efficient and uniform sampling from the primitives of arbitrary shape parameters. Then, any points internal to other primitives are ignored so that the points are sampled from the surfaces of an overall assembly.

The issue with this sampling strategy is that there is a discrepancy in the total number of sampled points depending on how much the predicted primitives overlap. Moreover, a primitive that is completely subsumed by another no longer plays a role in the learning process. In response, this work adopts a proportional sampling strategy [25] in which a total of N points are sampled from both the predicted superquadric assembly and the ground-truth 3D mesh model. Here, the points are sampled from each primitive proportional to their surface areas regardless of whether they are contained within other primitives or not. This results in a sample density of N points that is more uniformly distributed across the entire assembly.

2.2 *Autoregressive Inference via Transformer*

As shown in Fig. 2a, the previous CNN pipeline [19] consists of an encoder that produces a common feature vector $\boldsymbol{z}$ that is fed into multiple independent Multi-Layer Perceptron (MLP) branches that each predict the size ($\boldsymbol{\alpha}$), shape ($\boldsymbol{\epsilon}$), and pose ($\boldsymbol{t}, R(\boldsymbol{r}) \in SO(3)$) of the superquadrics with respect to the overall assembly. Note that apart from the common feature vector $\boldsymbol{z}$, there is no exchange of information between different MLP branches that predict the parameters of shared superquadric primitives. Moreover, the network is trained to predict a fixed number (M) of primitives regardless of the size and complexity of the spacecraft in question.

In order to improve the NN pipeline and allow for predictions of variable size assemblies, this work investigates the applicability of a transformer architecture [29] commonly adopted for Large Language Models (LLM). As LLMs excel at predicting the most plausible words given the previous sentences generated so far, the motivation is to apply the same mechanism to autoregressively predict the superquadric primitives. As visualized in Fig. 2b, the feature vector $\boldsymbol{z}$ from the encoder is considered the first "token" input to the transformer, which is used to predict the output feature $\boldsymbol{y}_1$ that is passed through the header layer to become the parameters associated with the first superquadric $\boldsymbol{p}_1$. The feature $\boldsymbol{y}_1$ is then appended to the input sequence, progressively predicting the next output feature $\boldsymbol{y}_2, \boldsymbol{y}_3, \ldots$ in an autoregressive manner.

While sentences have end-of-sentence tokens that can be learned to identify when it has terminated, spacecraft 3D models do not have such hints without part-wise annotations. Therefore, in order to allow the prediction of variable size assemblies, the superquadrics are predicted in decreasing order of size so that $\|\boldsymbol{\alpha}_{i+1}\|_\infty \leq \|\boldsymbol{\alpha}_i\|_\infty$ is enforced. Once the predicted primitive size is below a user-set threshold, the assembly is considered complete and the inference ends.

2.3 *Additional Supervision with Part-Wise Labels*

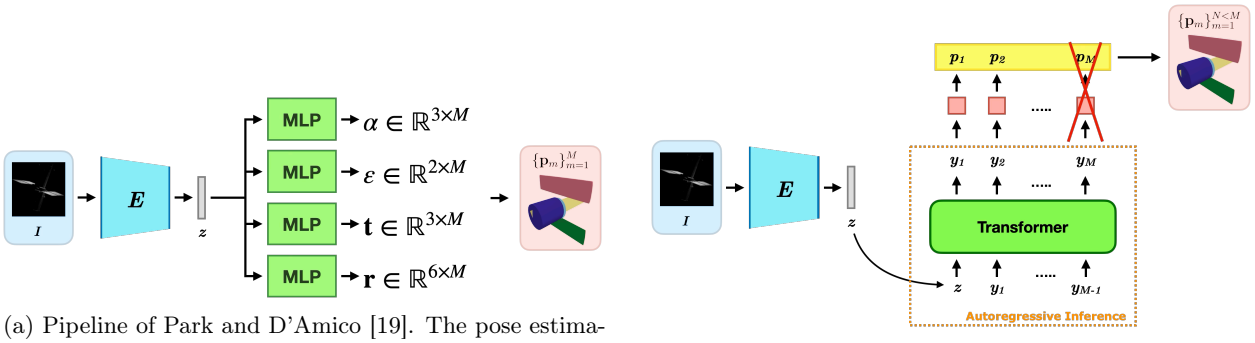
The qualitative results from the previous work [19] indicated that enforcing losses on the entire superquadric assembly without part-wise annotation leads to the prediction of primitives that may not correspond to specific parts (e.g., solar panels) of the spacecraft. Therefore, in order to alleviate the difficulty of learning, the part-wise annotations are manually prepared and additionally provided to NN during training as shown in Fig. 3. The Chamfer loss is then applied to each primitive and its corresponding subset of the ground-truth point clouds, while all other losses (e.g., reprojection loss via differentiable rendering) are optimized for the complete assembly. Moreover, in order to prevent the NN from learning specific orders of primitives, each predicted primitive is associated with the part annotation whose centroid is located at the closest distance.

Providing additional supervision may seem contradictory to the goal of improving the NN's generalizability on unseen test spacecraft images, since the selection of the part labels by a human operator adds to more per-model bias in the training set. The motivation instead derives from the observation made in literature which finds that better in-domain (i.e., validation) performances generally lead to better out-of-domain (i.e., test) performances [17, 21]. Moreover, intuitively, by explicitly providing part-wise annotations, this approach could facilitate the NN in associating primitives to macroscopic features of the spacecraft.

3. Experiments

3.1 *SPE3R Dataset*

The neural networks are trained and tested on the SPE3R dataset. Recall that out of 64 spacecraft models in SPE3R, 57 are designated for the training and validation while the remaining 7 are reserved as the test models. Each model is accompanied by



(a) Pipeline of Park and D'Amico [19]. The pose estimation branch is omitted.

(b) Proposed pipeline with a transformer architecture.

Fig. 2: Visualization of different NN pipelines.

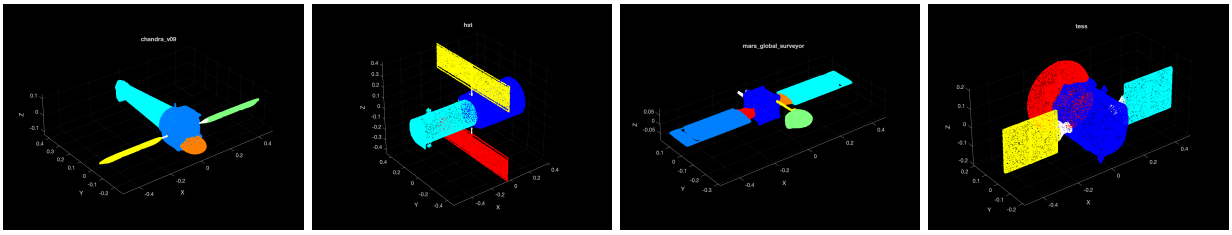


Fig. 3: Manually selected part-wise labels on 3D model point clouds.

1,000 synthetic images which are split 8:2 for the training and validation sets for the corresponding 57 models. Therefore, the validation set represents unseen images of known spacecraft, whereas the test set consist of unseen images of unknown spacecraft previously unobserved during the offline training.

3.2 *Implementations*

The training procedure largely follows that of Park and D'Amico [19]. The input RGB image is resized to 128×128 unless noted otherwise, and the networks are trained with the AdamW optimizer [13] with initial learning rate at 1×10^{-4}. The loss function consists of $\mathcal{L}_{\text{chamfer}}$ (Eq. 1) and the reprojection loss ($\mathcal{L}_{\text{repr}}$) defined as

$$\mathcal{L}_{\text{repr}}(\boldsymbol{\mathcal{X}}_m, \boldsymbol{\mathcal{Y}}_m) = \|\boldsymbol{\mathcal{X}}_m - \boldsymbol{\mathcal{Y}}_m\|_F^2 \quad \text{where} \quad \boldsymbol{\mathcal{X}}_m = \mathcal{R}(\hat{\boldsymbol{p}}, \boldsymbol{t}, \boldsymbol{R}) \tag{3}$$

where $\hat{\boldsymbol{p}}$ is the predicted superquadric primitives, $(\boldsymbol{R}, \boldsymbol{t})$ are the ground-truth pose labels for the input image, $\mathcal{R}(\cdot)$ denotes a differentiable renderer, and $(\boldsymbol{\mathcal{X}}_m, \boldsymbol{\mathcal{Y}}_m)$ are respectively the predicted and ground-truth binary masks. Unless the part-wise annotations are used, the final loss values also include the regularization on the overlap of each primitives.

For the transformer-based generator, two models are considered. The first is GPT-mini[§] which is small (7.25M parameters) and trained from scratch. The second is GPT2 [26] (92.6M parameters) with pre-trained weights from OpenAI. The latter model is studied specifically to investigate whether a foundation model such as GPT2 is capable of leveraging its learned prior and bridging the validation-test gap in SPE3R.

[§] `https://github.com/karpathy/minGPT`

3.3 *Evaluation Metrics*

This work employs the same two performance metrics as in Park and D'Amico [19]: Chamfer-ℓ_1 distance, which is equivalent to $\mathcal{L}_{\text{chamfer}}$ but computed with ℓ_1-norm, and (b) 2D Intersection-over-Union (IoU) between $\boldsymbol{\mathcal{X}}_m$ and $\boldsymbol{\mathcal{Y}}_m$. There are two key differences in implementation of the Chamfer-ℓ_1 distance metric compared to Park and D'Amico [19]. First, the number of surface samples to compute the Chamfer-ℓ_1 distance is increased to 100,000 in order to obtain the unbiased estimate of the metric. The points are sampled from each primitive in numbers proportional to their surface area much akin to Sec. 2.1 and in alignment with previous literature (e.g., [25]). Second, when computing the Chamfer-ℓ_1 distance for the test set, the two point clouds (predicted assembly vs. ground-truth) are first aligned by minimizing the Chamfer distance in Eq. 1. This is so that any pose bias in the ground-truth 3D mesh models in the test set do not affect the final measurement of the reconstruction quality. Once both prediction and ground-truth

Table 1: Quantitative evaluation of performance metrics on different configurations. Arrows indicate the direction towards better performance. Bold faces indicate the best performance. Checkmarks indicate the configuration that is adopted in the following experiments.

Config.	validation		test	
	Chamfer-ℓ_1 (↓)	IoU (2D) (↑)	Chamfer-ℓ_1 (↓)	IoU (2D) (↑)
(a) Park and D'Amico [19]	0.0355 ± 0.0192	0.850 ± 0.112	0.1040 ± 0.0589	**0.533 ± 0.165**
(b) ✓ Proportional Sampling	0.0274 ± 0.0126	0.884 ± 0.089	**0.0986 ± 0.0473**	0.532 ± 0.164
Generator Arch. (Fixed Num. SQ)				
(c) ✓ GPT-mini	0.0282 ± 0.0129	0.870 ± 0.081	0.1096 ± 0.0589	0.522 ± 0.166
(d) GPT2	0.0292 ± 0.0195	0.868 ± 0.101	0.0991 ± 0.0519	0.529 ± 0.168
Variable Num. SQ				
(e) Decreasing size	0.0646 ± 0.0336	0.638 ± 0.142	0.1052 ± 0.0477	0.454 ± 0.153
(f) ✓ Part-wise labels	0.0270 ± 0.0155	0.900 ± 0.065	0.1151 ± 0.0653	0.516 ± 0.163
Higher Resolution Input				
(b) with 256 × 256 input	0.0262 ± 0.0105	0.894 ± 0.086	0.1028 ± 0.0522	0.514 ± 0.167
(f) with 256 × 256 input	**0.0253 ± 0.0119**	**0.914 ± 0.052**	0.1140 ± 0.0665	0.523 ± 0.158

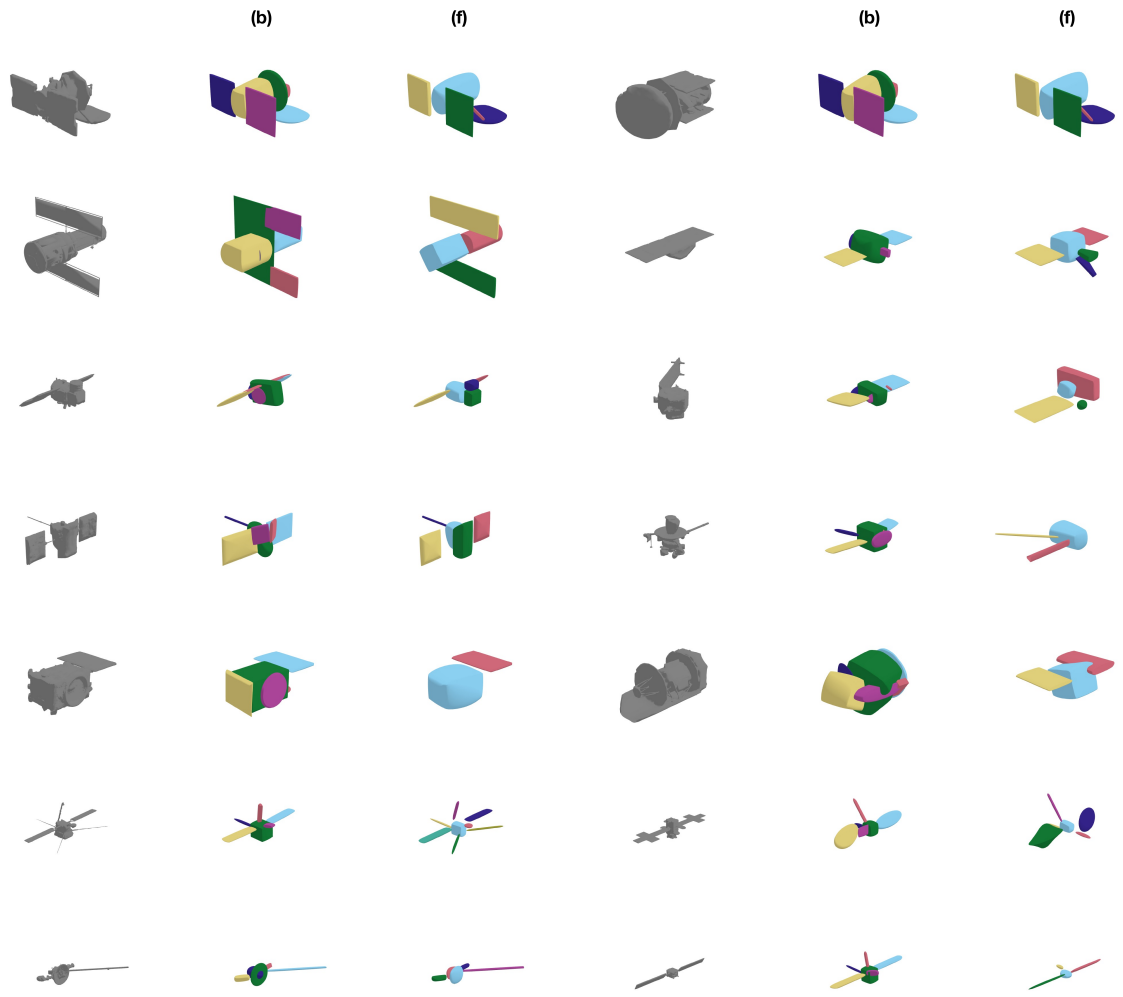
are aligned, the Chamfer-ℓ_1 distance is computed. Finally, unlike Park and D'Amico [19] which projects the predicted assembly using the predicted poses, the projection is performed using the ground-truth pose labels since this work does not concern with the problem of simultaneous pose estimation.

3.4 *Results*

Table 1 summarizes the quantitative performances for various configurations on both validation and test sets of the SPE3R dataset. Here, unless noted otherwise, the CNN is trained to predict $M = 6$ primitives. First, simply improving the surface sampling strategy on top of the vanilla architecture of Park and D'Amico [19] results in a significant performance boost on both validation and test sets. Unfortunately, replacing the generator with either transformer architecture does not result in immediate further improvement of both performance metrics when the network always predicts M primitives. Allowing predictions of variable length primitives in a decreasing order of size significantly backfires across all metrics except the Chamfer-ℓ_1 distance on the test set. This hints at how disconnected the predictions on the test models are from their respective ground-truths across all configurations. Training instead with part-wise labels improves the validation performance with marginally degraded performance compared to configuration (b). Finally, using image inputs with increased resolution also contributes to improved performance on the validation models; however, there is no such improvement for the test models.

Figure 4 visualizes various reconstructed primitive assemblies for configurations (b) and (f). It is interesting to see that with a proper sampling strategy, the CNN learns to predict primitives for the miscellaneous parts that the human operator has overlooked during the manual labeling process. For example, see the prediction on the 5th row of Fig. 4a where the CNN captures additional parts that correspond to distinct yet macroscopically irrelevant structures. However, learning with part-wise labels makes it easy for the network to predict only the assigned number of primitives, allowing an even more compact representation when the spacecraft structure is relatively simple. Again, see the prediction on the 5th row of Fig. 4a. However, on the test set, there is virtually no way to visually tell whether configuration (b) or (f) is better. In the end, the test set predictions resemble those of the validation set models that look the closest to the given inputs from the test set.

Finally, the shared feature output vectors $\boldsymbol{z}$ of the encoder for all validation and test images are visualized in 2D via t-SNE [28] in Fig. 5. As expected, the encoded feature vectors are fairly well clustered for each spacecraft model. Intuitively, if the network is truly learning the rich representations pertaining to the underlying macroscopic structure of the satellites, then, for example, all images of the spacecraft with two extended solar panels would be expected to entangle with each other instead of showing distinctive clusters in t-SNE visualization. Figure 5 is another qualitative example that the network ends up memorizing all spacecraft models when trained with SPE3R.



(a) Validation (b) Test

Fig. 4: Visualization of reconstructed assemblies using different configurations.

3.5 *Discussions*

The experimental results suggest that various strategies studied in this work—improving superquadric sampling, employing the transformer architecture and adding part-wise supervision labels—do produce some tangible improvement in the validation performance, but they do not seem to contribute to bridging the generalization gap towards previously unseen models. Nevertheless, the significance of the results should be interpreted with a grain of salt, as the experiments are performed only within the context of training with the SPE3R dataset. The results indicate that augmenting the training pipeline and the network architecture is simply insufficient to overcome the sheer lack of diversity in the training dataset. However, it is also reasonable that improved architecture (e.g., autoregressive inference of transformer architecture) and the learned priors of foundation models such as GPT2 could contribute to improving the generalization capability as the training is subject to orders of magnitude more 3D models.

The lessons learned in this work highlight the dire need for an extensive dataset comprising a large number of spacecraft 3D models which is necessary to train any DNN model for image-based spaceborne 3D reconstruction. If RPO is going to depend on well-learned priors on a general 3D structure of manmade objects in space, such a dataset would become a core requirement. However, given the lack of diversity in already existing spacecraft on Earth orbits and beyond, it is infeasible to construct a dataset of spacecraft 3D models at a scale even matching Objaverse [7] which

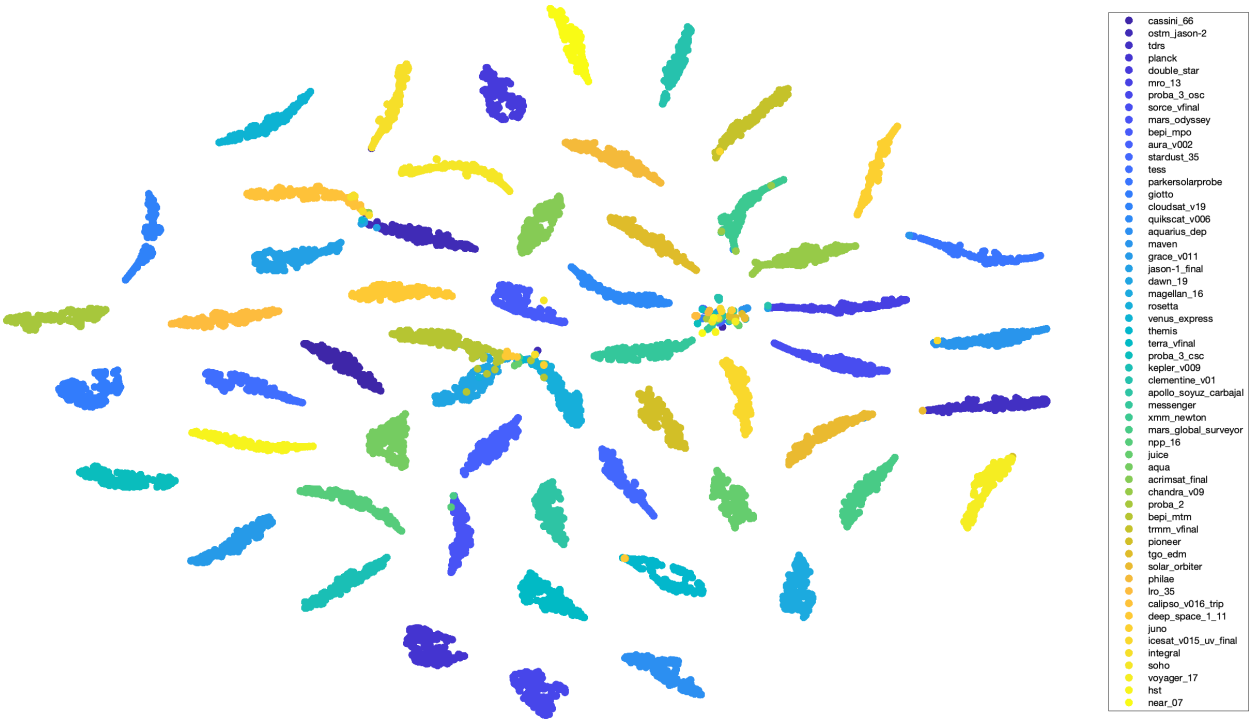


Fig. 5: t-SNE visualization of the encoded feature vector $\boldsymbol{z}$ on the validation set.

contains 800K models. Instead, the question would be to develop a sizable dataset that would facilitate the fine-tuning of a foundation model pre-trained on datasets such as ObjaverseXL [6], which is aligned with the conclusion of Mathihalli et al. [14]. Therefore, future efforts should focus on artificially inflating the datasets such as SPE3R using techniques such as generative models.

4. Conclusion

In summary, this work proposed several improvements to the model and training pipeline of Park and D'Amico [19] to perform better zero-shot image-based abstraction of 3D shapes of unknown spacecraft. Experimental results reveal that the proposed methods do allow more compact and parsimonious 3D representations of spacecraft with less structural complexity, but they fail to improve the model's generalizability to previously unseen target spacecraft due to sheer lack of training data. The paper provides careful analyses and proposes a future direction in current research.

References

[1] K. R. Barad, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez. Object-centric reconstruction and tracking of dynamic unknown objects using 3D gaussian splatting, 2024. URL `https://arxiv.org/abs/2405.20104`.

[2] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981. doi: 10.1109/MCG.1981.1673799.

[3] B. Caruso, T. Mahendrakar, V. M. Nguyen, R. T. White, and T. Steffen. 3D reconstruction of non-cooperative resident space objects using instant NGP-accelerated NeRF and D-NeRF. In *AAS/AIAA Spaceflight Mechanics Conference*, 2023.

[4] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[5] B. Chen, J. Cao, A. Parra, and T.-J. Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2816–2824, 2019. doi: 10.1109/ICCVW.2019.00343.

[6] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-XL: A universe of 10M+ 3D objects. *arXiv preprint arXiv:2307.05663*, 2023.

[7] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. 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.

[8] A. Garcia, M. A. Musallam, V. Gaudilliere, E. Ghorbel, K. Al Ismaeil, M. Perez, and D. Aouada. LSPnet: A 2D localization-oriented spacecraft pose estimation neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2048–2056, June 2021.

[9] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), July 2023. ISSN 0730-0301. doi: 10.1145/3592433.

[10] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and S. D'Amico. Satellite pose estimation challenge: Dataset, competition design and results. *IEEE Transactions on Aerospace and Electronic Systems*, 56(5):4083–4098, 2020. doi: 10.1109/TAES.2020.2989063.

[11] A. Legrand, R. Detry, and C. D. Vleeschouwer. Leveraging neural radiance fields for pose estimation of an unknown space object during proximity operations, 2024. URL `https://arxiv.org/abs/2405.12728`.

[12] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9264–9275, 2023. doi: 10.1109/ICCV51070.2023.00853.

[13] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[14] N. Mathihalli, A. Wei, G. Lavezzi, P. M. Siew, V. Rodriguez-Fernandez, H. Urrutxua, and R. Linares. DreamSat: Towards a general 3D model for novel view synthesis of space objects. In *75th International Astronautical Congress (IAC), Milan, Italy*, 2024.

[15] A. Mergy, G. Lecuyer, D. Derksen, and D. Izzo. Vision-based neural scene representations for spacecraft. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2002–2011, June 2021.

[16] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 405–421, Cham, 2020. Springer International Publishing. doi: 10.1007/978-3-030-58452-8_24.

[17] J. P. Miller, R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon, and L. Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7721–7735. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/miller21b.html`.

[18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. doi: 10.1109/TRO.2015.2463671.

[19] T. H. Park and S. D'Amico. Rapid abstraction of spacecraft 3D structure from single 2D image. In *AIAA SCITECH 2024 Forum*, 2024. doi: 10.2514/6.2024-2768.

[20] T. H. Park and S. D'Amico. SPE3R: Synthetic dataset for satellite pose estimation and 3D reconstruction. Stanford Digital Repository, 2024. Available at `https://purl.stanford.edu/pk719hm4806`.

[21] T. H. Park and S. D'Amico. Bridging domain gap for flight-ready spaceborne vision, 2024. URL `https://arxiv.org/abs/2409.11661`.

[22] T. H. Park and S. D'Amico. Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap. *Advances in Space Research*, 73(11):5726–5740, 2024. doi: 10.1016/j.asr.2023.03.036.

[23] T. H. Park, S. Sharma, and S. D'Amico. Towards robust learning-based pose estimation of noncooperative spacecraft. In *2019 AAS/AIAA Astrodynamics Specialist Conference, Portland, Maine*, August 11-15 2019.

[24] T. H. Park, M. Märtens, M. Jawaid, Z. Wang, B. Chen, T.-J. Chin, D. Izzo, and S. D'Amico. Satellite pose estimation competition 2021: Results and analyses. *Acta Astronautica*, 204:640–665, 2023. doi: 10.1016/j.actaastro.2023.01.002.

[25] D. Paschalidou, L. Van Gool, and A. Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1057–1067, 2020. doi: 10.1109/CVPR42600.2020.00114.

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2018. URL `https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf`.

[27] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. doi: 10.5555/1121596.

[28] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.