

CNOS: A Strong Baseline for CAD-based Novel Object Segmentation

Van Nguyen Nguyen¹ Thibault Groueix² Georgy Ponomatkin¹ Vincent Lepetit¹ Tomas Hodan³
¹LIGM, École des Ponts ²Adobe ³Reality Labs at Meta

Source code: <https://github.com/nv-nguyen/cnos>

Abstract

We propose a simple yet powerful method to segment novel objects in RGB images from their CAD models. Leveraging recent foundation models, Segment Anything and DINOv2, we generate segmentation proposals in the input image and match them against object templates that are pre-rendered using the CAD models. The matching is realized by comparing DINOv2 `cls` tokens of the proposed regions and the templates. The output of the method is a set of segmentation masks associated with per-object confidences defined by the matching scores. We experimentally demonstrate that the proposed method achieves state-of-the-art results in CAD-based novel object segmentation on the seven core datasets of the BOP challenge, surpassing the recent method of Chen et al. by absolute 19.8% AP.

1. Introduction

Object pose estimation plays a critical role in robotics and augmented reality applications. While supervised deep learning methods have achieved remarkable performance, they rely on extensive training data specific to each target object [16, 23, 24]. Introducing objects unseen during training therefore requires a significant effort to synthesize or annotate data and retrain the model. This restricts the application of the supervised methods in industry. For instance, in a logistic warehouse, it appears impractical to retrain the pose estimation method for every new product.

Performing object pose estimation typically involves two main steps: (1) the target objects are detected/segmented in the input image, and (2) the 6D object poses are then estimated from the detected regions [23]. Recent works such as template-pose [19] and MegaPose [17] introduced effective CAD-based object pose estimation methods. However, these methods mainly focus on the second step and require input 2D bounding boxes, which restricts their applicability to scenarios where precise 2D bounding boxes are available.

To address the gap, we propose a simple method for object detection and segmentation that only requires CAD

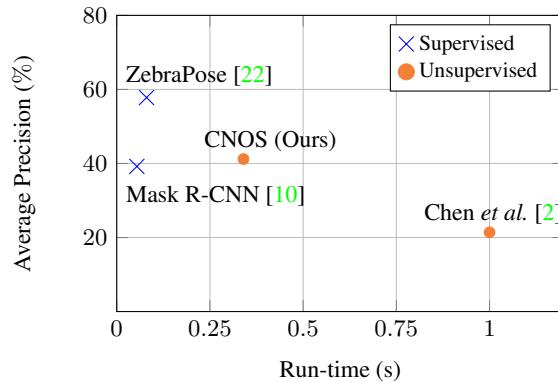


Figure 1. Performance on the seven core BOP datasets [23]. Our method, CNOS, relies on FastSAM [28] for generation of segmentation proposals and on DINOv2 [20] for visual description. CNOS outperforms the unsupervised method of Chen et al. [2] and even the supervised method Mask R-CNN [10], which was trained on tens of thousands of images per BOP dataset and used in CosyPose [16]. Similarly to Mask R-CNN [10], the runtime of CNOS is dominated by the proposal stage.

models of the target objects. The method is dubbed **CNOS** for CAD-based Novel Object Segmentation.

In CNOS, new objects are onboarded by rendering their CAD models and describing each rendered template by the DINOv2 `cls` token [20]. Given an RGB input image, segmentation proposals are extracted from the image by Segment Anything (SAM) [15] or Fast Segment Anything (FastSAM) [28] and matched against the templates based on the similarity between their DINOv2 `cls` tokens. Rendering the templates takes less than 2 seconds per CAD model which is much faster than retraining of supervised methods, which typically requires several hours. The choice of DINOv2 for measuring the similarity between templates and proposals is mainly motivated by its ability to effectively address the domain gap between real and synthetic images. We also demonstrate that photo-realistic rendering techniques of BlenderProc [3], which require approximately 1 second to render an image, can be leveraged to further mitigate this domain gap and enhance accuracy. Experiments on the seven core datasets of the BOP challenge [23]

demonstrate the state-of-the-art performance of CNOS.

As shown in Figure 1, CNOS outperforms the recent unsupervised method for CAD-based segmentation by Chen *et al.* [2] and even Mask R-CNN [10], a supervised method that was trained on tens of thousands of images per BOP dataset and used in CosyPose [16].

2. Related work

This section provides a brief overview of existing methods for object detection and segmentation that are commonly used in 6D object pose estimation pipelines.

Segmentation of seen object. Many object pose estimation methods [16, 24] employ object segmentation methods such as Mask R-CNN [10], typically fine-tuned on extensive training data specific to each target object [23]. Such supervised methods have been demonstrated robust in challenging scenarios with heavy occlusions and lighting variations. However, these methods cannot deal with new objects without retraining, which is a deal-breaker for many applications. In this work, we address this limitation by focusing on the segmentation of previously-unseen objects from their CAD models without retraining.

Segmentation of unseen objects. Object segmentation methods traditionally focus on scenarios known as “closed-world” settings, where the training and test sets share the same object classes. Nevertheless, recent observations by Du *et al.* [6, 7] suggest that class-agnostic instance segmentation networks can effectively generalize to previously unseen object classes. Building upon this insight, Zhao *et al.* [29] leverage saliency detection models to solve the novel class discovery task in 2D segmentation. Nguyen *et al.* [18] propose a two-stage 6D tracking approach based on these observations. Their approach assumes the availability of an initial bounding box to segment object using [6] and then propagates the box to next frames using optical flow. Subsequently, 6D tracking of novel objects is performed based on predicted object masks. In contrast, our objective solely focuses on segmenting objects in images derived from CAD models without initial boxes.

Commonly used in robotics, UOIS-Net [27] employs a two-stage approach to segment novel objects. It operates on the depth channel of captured RGB-D images to generate object instance center votes and assembles them into rough initial masks. These masks are subsequently refined using the RGB channels. Xiang *et al.* [26] also propose an RGB-D based method that uses learned feature embeddings and applies a mean shift clustering algorithm to discover and segment unseen objects. To avoid using depths, Durner *et al.* [8] use horizontal correlation to extract disparity RGB-based features and segment novel objects from stereo RGB images. It is worth noting that UOIS-Net [27], Xiang *et al.* [26], and Durner *et al.* [8] are RGB-D or stereo RGB

approaches, while our method targets the segmentation of unseen objects from only a single RGB image and CAD models, which is more applicable.

Recently, Segment Anything (SAM) [15] has introduced a powerful foundation model for image segmentation capable of segmenting all objects in a given RGB image. Chen *et al.* [2] utilize SAM to extract object proposals, which are then combined with visual clues extracted by ImageBlind [9]. Feature matching is subsequently applied for CAD-based novel object segmentation. Experiments presented in this paper show that CNOS surpasses the method of Chen *et al.* by absolute 19.8% AP.

3. Method

In this section, we provide a detailed description of our three-stage approach for CAD-based novel object segmentation. We first describe the onboarding stage in Section 3.1, where we extract visual descriptors from renderings of the CAD models. In Section 3.2, we explain the proposal stage, which involves obtaining all possible masks and their descriptors. Finally, in Section 3.3, we discuss the matching stage, where object masks are retrieved and labeled based on visual descriptors of their CAD models.

3.1. Onboarding stage

In the onboarding stage, we render a set of RGB synthetic templates and extract their visual descriptors using DINoV2 [20]. To ensure robust object segmentation under different orientations, we render CAD models under 42 viewpoints as shown in Figure 3. These 42 viewpoints are defined by the icosphere primitive of Blender¹ which has been shown in [19] to provide well-distributed view coverage of CAD models for robust template matching. Additionally, we experiment with denser viewpoints by dividing each triangle of the icosphere into four smaller triangles. The rendering process results in a total of $N_o V$ templates, where N_o is the number of CAD models and V is the number of viewpoints. We then crop the templates with the ground-truth bounding boxes and use the DINoV2 `cls` tokens as their visual descriptors \mathbf{D}_r of size $N_o \times V \times C$. By default, we use $V = 42$ and $C = 1024$.

3.2. Proposal stage

For each testing RGB image, we use SAM [15] or Fast-SAM [28] with a default configuration to generate a set of N_p unlabeled proposals, where each proposal i is defined by a mask M_i . N_p is not fixed and varies depending on the content of the input RGB image.

To compute the visual descriptor for each proposal i , we first remove the background of the input image using the corresponding mask M_i . Subsequently, we crop the image

¹`bpy.ops.mesh.primitive_ico_sphere_add()`

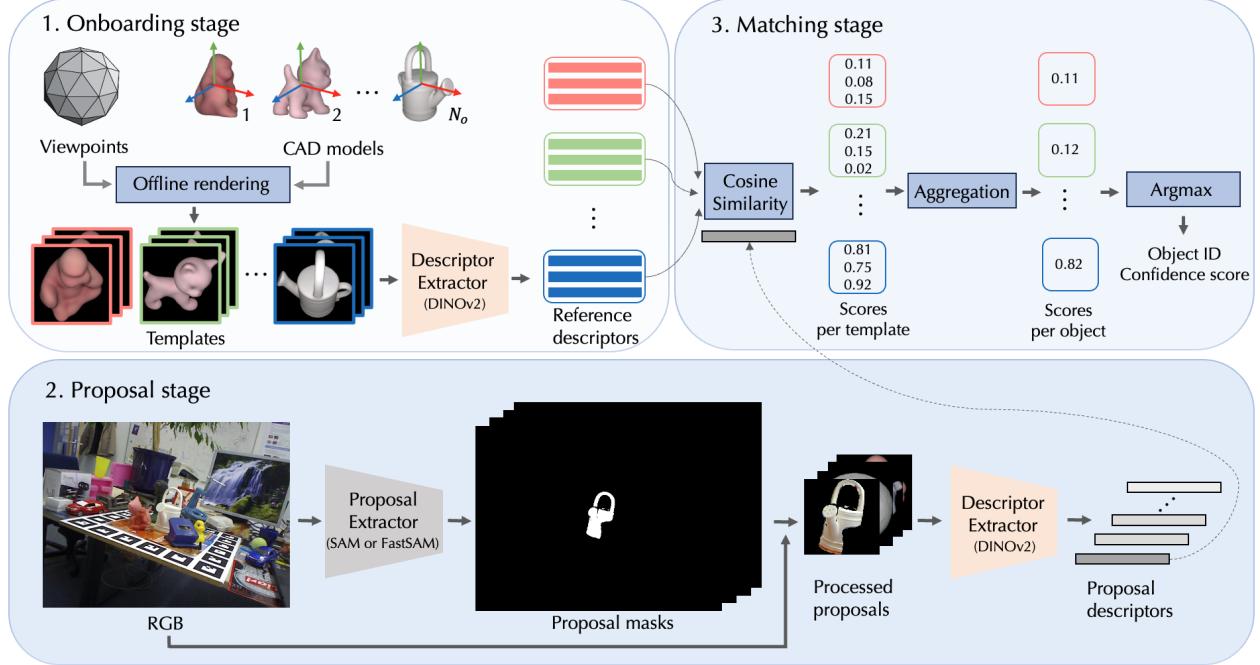


Figure 2. **CNOS overview.** Given CAD models of the target objects, the objects are onboarded by (i) rendering a set of templates showing the models from different viewpoints, and (ii) describing the templates by the DINOv2 `cls` token (Section 3.1). At inference time, segmentation proposals are generated from the input RGB image using SAM or FastSAM (Section 3.2), and the proposals are matched against the templates by comparing their DINOv2 `cls` tokens (Section 3.3).

using the model bounding box derived from M_i . Since each proposal mask M_i has a different bounding box size, parallel processing becomes unfeasible. To overcome this, we add a simple image processing step including scaling and padding in order to resize all proposals to a consistent size of 224×224 . This standardization enables efficient parallel processing of proposals in a single batch. Then, we extract the DINOv2 `cls` tokens from the processed proposals and use them as their visual descriptors \mathbf{D}_p of size $N_p \times C$.

3.3. Matching stage

The goal of the matching stage is to assign each proposal i an object ID o_i and a confidence score s_i . To this end, we compare each proposal descriptor in \mathbf{D}_p with each template descriptor in \mathbf{D}_r using the cosine similarity. This comparison step produces a similarity matrix of size $N_p \times N_o$.

View aggregation. By aggregating the similarity scores over all V templates for each CAD model, we obtain a matrix of size $N_p \times N_o$. This matrix represents the similarity between each proposal p_i and each CAD model. We experiment with different aggregation functions, such as Mean, Max, Median, and Mean of top k highest, noted Mean_k , and find that Mean_k yields the best results.

Object ID assignment. To assign the object ID o_i and confidence score s_i to each proposal, we simply apply the argmax and max functions on the similarity matrix $N_p \times N_o$

over the N_o objects. This yields a similarity matrix of size N_p defining the confidence score for N_p proposals.

Output. At the end of the matching stage, we obtain a set of labeled proposals, where each proposal is defined as $\{M_i, o_i, s_i\}$, where M_i is the modal mask (*i.e.*, a mask covering the visible part of the object surface [23]), o_i is the object ID, and s_i is the confidence score. Some of these proposals may still be incorrectly labeled. To address this, it is possible to apply a threshold δ on the confidence score threshold. The figures in Section 4.2 show CNOS’s segmentation results with $\delta = 0.5$.

4. Experiments

In this section, we describe the experimental setup (Section 4.1), compare our method with previous works [2, 10, 22] on the seven core datasets of the BOP challenge [23] (Section 4.2), and conduct an ablation study focused on the accuracy under different aggregating functions and different numbers of rendering viewpoints, and on the run-time (Section 4.3). Finally, we discuss the use of CNOS in a pipeline for 6D pose estimation of novel objects (Section 4.4) or in CAD-free novel object segmentation.

4.1. Experimental setup

Datasets. We evaluate our method on the test set of seven core datasets of the BOP challenge [23]: LineMod Occlu-

| Method | Rendering | BOP Datasets | | | | | | | | |
|--------------|---------------------------|-----------------|--------|-------|--------|-------|------|-------|------|------|
| | | LM-O | T-LESS | TUD-L | IC-BIN | ITODD | HB | YCB-V | | |
| Supervised | 1 Mask R-CNN [10] (Synth) | - | 37.5 | 51.7 | 30.6 | 31.6 | 12.2 | 47.1 | 42.9 | 36.2 |
| | 2 Mask R-CNN [10] (Real) | - | 37.5 | 54.4 | 48.9 | 31.6 | 12.2 | 47.1 | 42.9 | 39.2 |
| | 3 ZebraPose [22] (Synth) | - | 50.6 | 62.9 | 51.4 | 37.9 | 36.1 | 64.6 | 62.2 | 52.2 |
| | 4 ZebraPose [22] (Real) | - | 50.6 | 70.9 | 70.7 | 37.9 | 36.1 | 64.6 | 74.0 | 57.8 |
| Unsupervised | 5 Chen <i>et al.</i> [2] | - | 17.6 | 9.6 | 24.1 | 18.7 | 6.3 | 31.4 | 41.9 | 21.4 |
| | 6 CNOS (SAM) | Pyrender [21] | 33.3 | 38.3 | 35.8 | 27.2 | 14.8 | 45.9 | 57.6 | 36.1 |
| | 7 CNOS (SAM) | BlenderProc [3] | 39.6 | 39.7 | 39.1 | 28.4 | 28.2 | 48.0 | 59.5 | 40.4 |
| | 8 CNOS (FastSAM) | BlenderProc [3] | 39.7 | 37.4 | 48.0 | 27.0 | 25.4 | 51.1 | 59.9 | 41.2 |

Table 1. **Comparison of CNOS with [2, 10, 22] on the seven core datasets of the BOP challenge [23].** Mask R-CNN and ZebraPose are retrained specifically on the target objects with renderings of the CAD models (noted as “Synth”) or real images of the object (noted as “Real”). We classify these methods as “supervised”. CNOS and Chen *et. al.* [2] are classified as “unsupervised” as these methods require no retraining for novel objects. We report the AP metric (higher is better) using the protocol from [23]. The best supervised results are highlighted in blue and the best unsupervised results in yellow. CNOS not only significantly outperforms [2] under the same settings but also surpasses the supervised method Mask R-CNN, highlighting its ability to generalize.

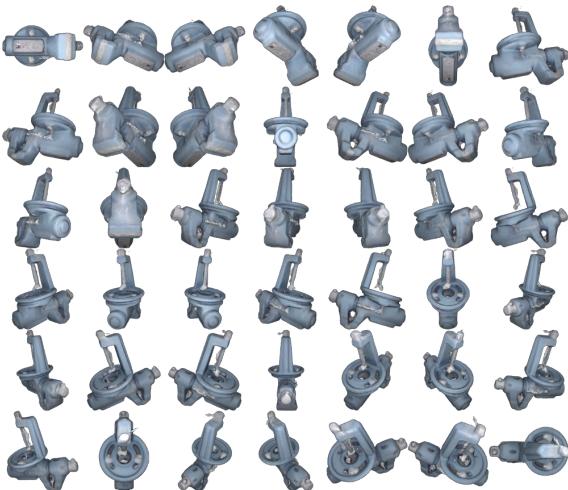


Figure 3. **Visualization of templates for the “benchwise” object from LM-O [11] rendered with Pyrender [21].** 42 templates were rendered from viewpoints defined by the icosphere [19].

sion (LM-O) [1], T-LESS [12], TUD-L [13], IC-BIN [4], ITODD [5], HomebrewedDB (HB) [14] and YCB-Video (YCB-V) [25]. In total, the datasets include 132 different objects shown in cluttered scenes with occlusions. The objects are of various types: textured or untextured, symmetric or asymmetric, household or industrial.

Evaluation metric. We evaluate our method using the Average Precision (AP) metrics, following the COCO metric and the BOP challenge evaluation protocol [23]. The AP metric is calculated as the mean of AP values at different Intersection over Union (IoU) thresholds ranging from 0.50 to 0.95 with an increment of 0.05.

Baselines. We compare our method with Chen *et al.* [2],

the most relevant work to ours. They use a three-stage CAD-based object segmentation approach, incorporating SAM [15] for image segmentation and ImageBlind [9] for visual descriptor extraction. Their use of 72 templates per CAD model resulted in the best performance according to their paper. Additionally, we compare our method with two relevant supervised methods from the BOP challenge [23]: Mask R-CNN [10], which was trained on real or synthetic training images specific to each dataset and used in CosyPose [16], and ZebraPose [22], which is currently the state-of-the-art for this task in the BOP challenge.

Implementation details. For the proposal stage, we use the default ViT-H SAM [15] or the default FastSAM [28], which has demonstrated promising results in terms of runtime efficiency. For extracting visual descriptors, we use the default ViT-H model of DINOv2 [20].

To further evaluate the performance of our method, we conducted a comparison using two sets of templates. The first set of templates was generated using Pyrender [21] from 42 pre-defined viewpoints. It is worthy to note that Pyrender computes the Direct Illumination and it is extremely fast, takes on average 0.026 second per image. The second set of templates comprised 42 realistic rendering templates selected from the available synthetic images of the PBR-BlenderProc4BOP training set provided in the BOP challenge. These realistic templates were specifically chosen to closely match the orientations of the 42 predefined viewpoints in the first set. Since the PBR-BlenderProc4BOP training images possibly have occlusions, we chose only images where the target objects are fully visible. The templates of target objects are finally obtained by making the background black using the ground-truth mask and cropping regions with the ground-truth bounding box.

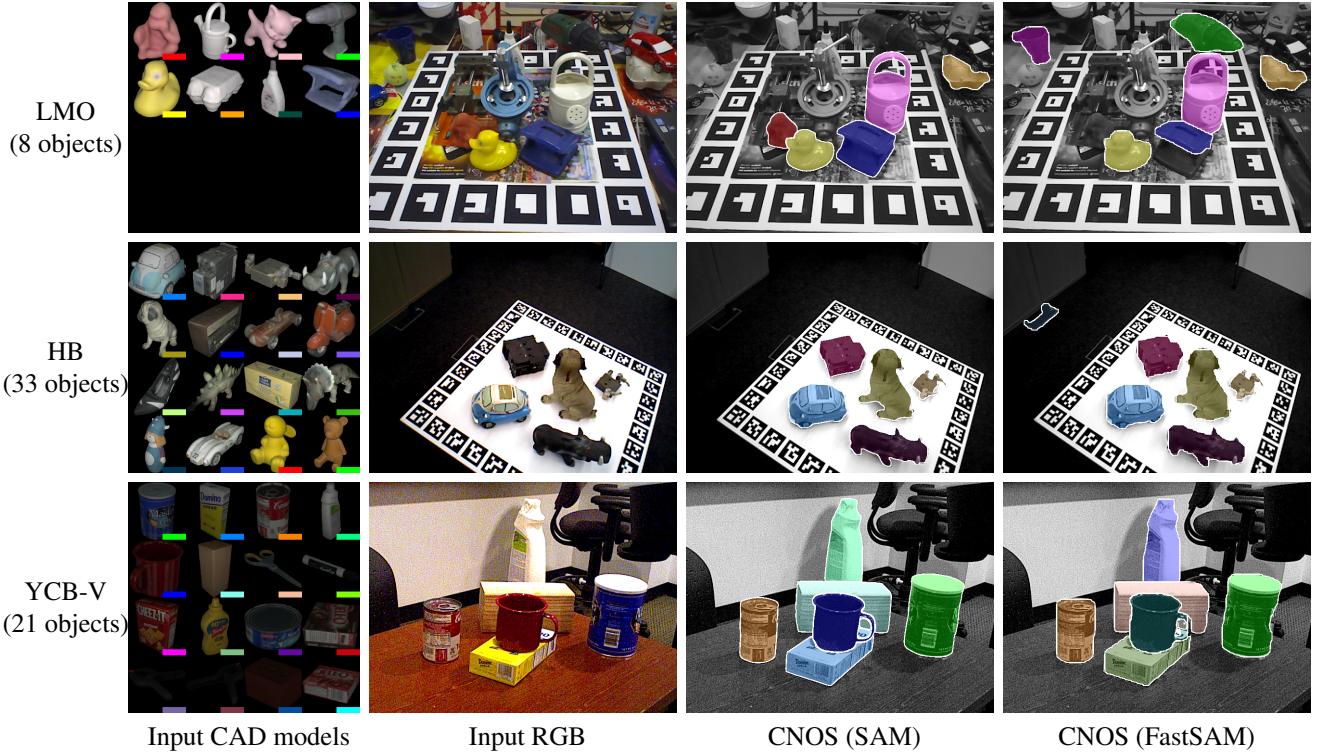


Figure 4. **Qualitative results on LM-O [1], HB [14] and YCB-V [25]**. The first column shows the input CAD models. In cases where there are more than 16 models, we only show the first 16 to ensure better visibility. The second column show the input RGB image and the last two columns depict the detections produced by CNOS (SAM) and CNOS (FastSAM) with confidence scores greater than 0.5. Interestingly, in the last row, even though the segmentation proposals in CNOS (SAM) and CNOS (FastSAM) are very similar, their final labels differ for a few objects. This inconsistency arises from DINOv2-based classification of the proposals as discussed in Section 4.2.

In order to maintain a consistent run-time across all datasets, we resize the images while preserving their aspect ratio. Specifically, we ensure that the width of each input RGB testing image is fixed at 640 pixels. All our experiments were conducted on a single V100 GPU.

4.2. Comparison with the state of the art

In Table 1, we show that CNOS outperforms Chen *et al.* [2] by a significant margin of absolute 19.8% AP. Furthermore, despite not being trained on the testing objects of the BOP datasets, our method surpasses the performance of Mask R-CNN [10] used in CosyPose [16], which was specifically trained on these objects. This highlights the generalization capability of our method.

We qualitatively found that the generated segmentation proposals usually include ones that are very well aligned with the target object instances, and that most mistakes are due to erroneous DINOv2-based classification of the proposals. Improving the proposal classification would be crucial to close the gap between CNOS and supervised state-of-the-art approaches such as ZebraPose.

In Figure 4, we show qualitative results of our method on LM-O [1], HB [14] and YCB-V [25] datasets.

4.3. Ablation study

Model size vs. run-time. We present the results for FastSAM and DINOv2 using various base models in Table 2, highlighting the trade-off between accuracy and run-time.

Rendering. Table 1 demonstrates the performance of our method using two types of rendering: Pyrender [21] in row 6 and BlenderProc [3] in row 7. The results indicate that incorporating realistic rendering significantly reduces the domain gap between synthetic and real images, yielding a 4.3% improvement in the AP metric.

Number of viewpoints. As shown in Table 3, using more viewpoints does not bring any improvement compared to the coarse viewpoints. This can be explained by the fact that the current set of 42 coarse viewpoints already provides sufficient coverage of the 3D objects.

Aggregating function. In Table 4, we explore different types of the function for aggregating the similarities between descriptors of templates and proposals. Among the tested functions, Mean_k ($k=5$), which is the average of the k highest similarity, achieves the best performance.

| Segmentation model | Descriptor model | AP | Run-time (s) |
|---------------------|------------------|------|--------------|
| FastSAM-s | ViT-s | 32.1 | 0.18 |
| FastSAM-s | ViT-l (default) | 33.8 | 0.25 |
| FastSAM-x (default) | ViT-s | 38.0 | 0.27 |
| FastSAM-x (default) | ViT-l (default) | 39.7 | 0.33 |

Table 2. **Ablation study of different FastSAM segmentation models [28] and DINOv2 descriptor models [20] on LM-O.**

| Method | Viewpoint density | |
|----------------|-------------------|-------------|
| | Coarse (42) | Dense (162) |
| CNOS (SAM) | 39.6 | 39.5 |
| CNOS (FastSAM) | 39.7 | 39.7 |

Table 3. **Ablation study of the number of viewpoints on LM-O.** The denser viewpoints are created by subdividing each triangle of the icosphere (used to create coarse viewpoints) into four triangles.

| Method | Aggregating function | | | |
|----------------|----------------------|--------|-------|-------------------|
| | Mean | Median | Max | Mean _k |
| CNOS (SAM) | 36.6 | 34.9 | 39.1 | 39.6 |
| CNOS (FastSAM) | 36.2 | 33.8 | 39.7 | 39.7 |
| Average | 36.40 | 34.40 | 39.40 | 39.65 |

Table 4. **Ablation study of aggregating functions on LM-O.** Using the Mean_k function, which calculates the average of the top k ($k = 5$) highest values, yields the best accuracy.

| Method | Run-time (second) | | |
|--------------------------|-------------------|----------|----------|
| | Onboarding | Proposal | Matching |
| CNOS (SAM, Pyrender) | 1.22 | 1.58 | 0.13 |
| CNOS (FastSAM, Pyrender) | 1.22 | 0.22 | 0.12 |
| CNOS (FastSAM, PBR) | 42.1 | 0.22 | 0.12 |

Table 5. **Run-time.** We report the run-time of each stage of CNOS on a single V100 GPU. The run-time of the onboarding stage includes both the rendering time and the visual descriptor extraction time for each CAD model.

Run-time. In Table 5, we present the average run-time of each stage in our method for a given CAD model. In the onboarding stage, the average rendering time for one image with Pyrender [21] is 0.026 second while with Blender-Proc [3] is around 1 second per image on a single V100 GPU. It is important to note that the onboarding stage is performed once for each CAD model. In terms of run-time, the onboarding stage is clearly bottlenecked by the generation of templates, while the proposal stage is currently bottlenecked by the segmentation algorithm.

4.4. Discussion

Pose initialization. Our intention was originally to use the DINOv2 cls token not only to recognize the object but also to estimate its initial pose that could be refined in a subsequent step. However, as illustrated in Figure 5, this

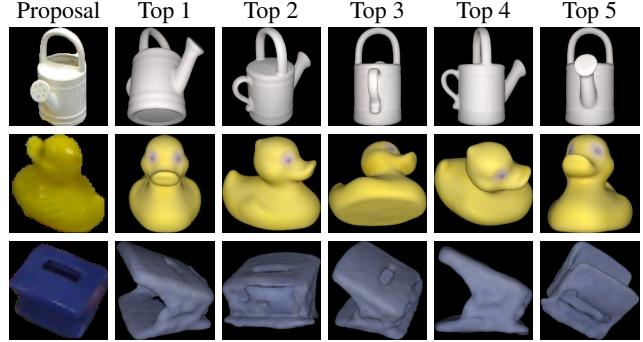


Figure 5. **Visualization of the nearest neighbors.** We show proposals along with five retrieved templates with the most similar DINOv2 cls tokens. The retrieved templates correspond to the same object but to poses that do not match the pose in the proposals – this suggests that the DINOv2 cls token can be effectively used to recognize the objects, but not to estimate the pose.

approach did not yield successful results, as the DINOv2 cls token seems to carry sufficient information about the object identity but not about the object pose.

CAD-free novel object segmentation. In this work, we focus on CAD-based novel object segmentation. However, the proposed CNOS method could be seamlessly adapted to address one-shot or few-shot novel object segmentation settings, where only one or a few reference images are available and CAD models are unknown. Specifically, the reference descriptors could be extracted directly from the available reference image(s), while the rest of the pipeline could be kept untouched.

5. Conclusion

We presented a simple yet powerful method for novel object segmentation solely based on their CAD models, without the need of any training. The method achieves a surprisingly high accuracy, comparable to previous supervised methods trained on large-scale annotated datasets. We hope that CNOS will serve as a standard baseline for CAD-based novel object segmentation and will be employed as the initial stage of novel object pose estimation pipelines.

Acknowledgments. We thank Nermin Samet for helpful discussions. This research was produced within the framework of Energy4Climate Interdisciplinary Center (E4C) of IP Paris and Ecole des Ponts ParisTech, and was supported by 3rd *Programme d’Investissements d’Avenir* [ANR-18-EURE-0006-02] and by the Foundation of Ecole polytechnique (Chaire “Défis Technologiques pour une Énergie Responsable” financed by TotalEnergies). This work was performed using HPC resources from GENCI-IDRIS 2022-AD011012294R2.

References

- [1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In *ECCV*, 2014. 4, 5
- [2] Jianqiu Chen, Mingshan Sun, Tianpeng Bao, Rui Zhao, Li-wei Wu, and Zhenyu He. 3D Model-Based Zero-Shot Pose Estimation Pipeline. In *arXiv Preprint*, 2023. 1, 2, 3, 4, 5
- [3] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blender-Proc. In *arXiv Preprint*, 2019. 1, 4, 5, 6
- [4] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd. In *CVPR*, 2016. 4
- [5] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing Mvtec Itodd-A Dataset for 3D Object Recognition in Industry. In *ICCV Workshops*, 2017. 4
- [6] Yuming Du, Wen Guo, Yang Xiao, and Vincent Lepetit. 1st Place Solution for the UVO Challenge on Image-Based Open-World Segmentation 2021. In *ICCV Workshops*, 2021. 2
- [7] Yuming Du, Yang Xiao, and Vincent Lepetit. Learning to Better Segment Objects from Unseen Classes with Unlabeled Videos. In *ICCV*, 2021. 2
- [8] Maximilian Durner, Wout Boerdijk, Martin Sundermeyer, Werner Friedl, Zoltan-Csaba Marton, and Rudolph Triebel. Unknown Object Segmentation from Stereo Images. In *IROS*, 2021. 2
- [9] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One Embedding Space to Bind Them All. In *CVPR*, 2023. 2, 4
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 3, 4, 5
- [11] Stefan Hinterstoßer, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary R. Bradski, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In *ACCV*, 2012. 4
- [12] Tomas Hodan, Pavel Haluza, Stepan Obdrzalek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-Less Objects. In *WACV*, 2017. 4
- [13] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glentbuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, and Others. Bop: Benchmark for 6D Object Pose Estimation. In *ECCV*, 2018. 4
- [14] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: RGB-D Dataset for 6D Pose Estimation of 3D Objects. In *ICCV Workshops*, 2019. 4, 5
- [15] Alexander Kirillov, Eric Mintun, Nikhil Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, and Others. Segment Anything. In *arXiv Preprint*, 2023. 1, 2, 4
- [16] Yann Labb  , Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-View Multi-Object 6D Pose Estimation. In *ECCV*, 2020. 1, 2, 4, 5
- [17] Yann Labb  , Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. In *CoRL*, 2022. 1
- [18] Van Nguyen Nguyen, Yuming Du, Yang Xiao, Michael Ramamonjisoa, and Vincent Lepetit. PIZZA: A Powerful Image-Only Zero-Shot Zero-CAD Approach to 6 DoF Tracking. In *International Conference on 3D Vision*, 2022. 2
- [19] Van Nguyen Nguyen, Yinlin Hu, Yang Xiao, Mathieu Salzmann, and Vincent Lepetit. Templates for 3D Object Pose Estimation Revisited: Generalization to New Objects and Robustness to Occlusions. In *CVPR*, 2022. 1, 2, 4
- [20] Maxime Oquab, Timoth  e Darct, Th  o Moutakanni, Huy Vo, Marc Szafraaniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, and Others. Dinov2: Learning Robust Visual Features Without Supervision. In *arXiv Preprint*, 2023. 1, 2, 4, 6
- [21] Pyrender. <https://github.com/mmatl/pyrender>. 4, 5, 6
- [22] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to Fine Surface Encoding for 6 DoF Object Pose Estimation. In *CVPR*, 2022. 1, 3, 4
- [23] Martin Sundermeyer, Tom  s Hod  , Yann Labbe, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Ji   Matas. Bop Challenge 2022 on Detection, Segmentation and Pose Estimation of Specific Rigid Objects. In *CVPR*, 2023. 1, 2, 3, 4
- [24] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estimation. In *CVPR*, 2021. 1, 2
- [25] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems Conference*, 2018. 4, 5
- [26] Yu Xiang, Christopher Xie, Arsalan Mousavian, and Dieter Fox. Learning RGB-D Feature EMbeddings for Unseen Object Instance Segmentation. In *Conference on Robot Learning*, 2021. 2
- [27] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen Object Instance Segmentation for Robotic Environments. *IEEE Transactions on Robotics and Automation*, 37(5), 2021. 2
- [28] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast Segment Anything. In *arXiv Preprint*, 2023. 1, 2, 4, 6
- [29] Yuyang Zhao, Zhun Zhong, Nicu Sebe, and Gim Hee Lee. Novel Class Discovery in Semantic Segmentation. In *CVPR*, 2022. 2