Box Pose and Shape Estimation and Domain Adaptation for Large-Scale Warehouse Automation

Xihang Yu[1], Rajat Talak[1], Jingnan Shi[1], Ulrich Viereck[2], Igor Gilitschenski[2,3], and Luca Carlone[1]

[1] Laboratory for Information & Decision Systems (LIDS)
Massachusetts Institute of Technology, Cambridge, USA,
{jimmyyu,talak,jnshi,lcarlone}@mit.edu
[2] Symbotic, Wilmington, Massachusetts,
uviereck@symbotic.com
[3] Vector Institute
University of Toronto, Ontario, Canada,
gilitschenski@cs.toronto.edu

Abstract. Modern warehouse automation systems rely on fleets of intelligent robots that generate vast amounts of data — most of which remains unannotated. This paper develops a self-supervised domain adaptation pipeline that leverages real-world, unlabeled data to improve perception models without requiring manual annotations. Our work focuses specifically on estimating the pose and shape of boxes and presents a correct-and-certify pipeline for self-supervised box pose and shape estimation. We extensively evaluate our approach across a range of simulated and real industrial settings, including adaptation to a large-scale real-world dataset of 50,000 images. The self-supervised model significantly outperforms models trained solely in simulation and shows substantial improvements over a zero-shot 3D bounding box estimation baseline.

Keywords: Certifiable models, computer vision, 3D robot vision, object pose estimation, safe perception, self-supervised learning.

1 Introduction and Problem Statement

Warehouse automation has the potential to increase operational efficiency and accuracy while reducing labor costs and human errors. A key task in this process involves robots picking, transporting, and placing boxes between buffer and storage shelves (see Figure 1). Executing such tasks reliably over long durations without failure requires accurate perception in the operating domain.

In this work, we consider the problem of estimating the pose and shape of boxes encountered in warehouse automation applications. We parameterize the box as a cuboid and aim to simultaneously estimate its pose $\boldsymbol{T} \in \mathrm{SE}(3)$ and shape $\boldsymbol{S}$ (*i.e.,* width, height, and depth). Automated warehouses are a source of large amounts of unannotated data, collected by the robots during operation. *Our aim is to use the large-scale unannotated data collected by robots and enable self-supervised domain adaptation to improve the perception results.*

Fig. 1: (left) Robot in a Symbotic warehouse picking up a box from a shelf. (right) A real-world pick-up task using a model trained entirely in simulation and adapted with our self-supervised pipeline on unlabeled data.

Contributions. Our contributions are threefold: (1) We propose a pipeline that can accurately estimate the pose and shape of a box from stereo images. (2) We implement a self-training pipeline, leveraging the **correct-and-certify** approach from [1,2,3,4,5]. The approach utilizes corrected and certified estimates to self-train the model and avoids the need for data annotation. (3) We report an industry-scale demonstration of accurate box pose and shape estimation in the desired operating domain. This is made possible by self-training on a dataset of 50,000 images collected from Symbotic warehouses.

2 Related Work

2.1 Category-Level Object Pose and Shape Estimation

Object pose and shape estimation involves recovering the 3D pose and shape of an object. Existing methods can be classified based on whether they assume access to known instance-level shape priors. Approaches that rely on known shape priors typically use predefined CAD models of each object instance [3,2]. In contrast, category-level methods aim to generalize across unseen instances within the same object category, without requiring instance-specific CAD models. These approaches often learn to model shape deformations or normalized coordinate representations to capture intra-class variation [6,7,8,9,4].

In this work, we focus on estimating the pose and shape of box-like objects without relying on instance-level shape priors. Several prior methods have addressed this problem from different perspectives. For example, [10] proposes Front Face Shot (FFS), a method that estimates box pose from front-view RGB images. While FFS generalizes well to unseen pallet appearances, it depends heavily on accurate front-face visibility and bounding box detection, which limits its robustness in the presence of occlusion. Another approach, Cube R-CNN [11], is a zero-shot RGB-only method trained on the large-scale Omni3D benchmark for general 3D bounding box prediction. However, in our experiments, it suffers from substantial performance degradation due to domain shift, making it less effective for our target setting.

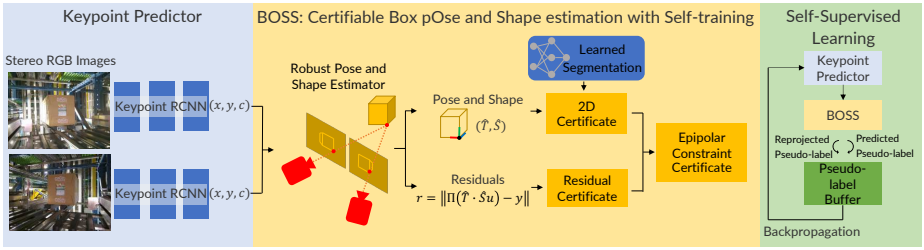


Fig. 2: Illustration of the proposed pipeline. We take stereo images as input and use two keypoint prediction networks — one for each view — to predict the box corners. Only high-confidence confidence keypoints are used for pose and shape estimation. The box pose and shape estimation problem is formulated as a two-view Perspective-n-Point optimization. Then, pose and shape estimates that pass certain checks are used to generate pseudo-labels for self-supervised learning. In particular, a predicted keypoint is considered a valid pseudo-label for self-supervised learning if it passes a number of image-level and keypoint-level checks (certificates). To ensure robustness against outliers, we apply Geman-McClure [21] as a robust loss in the pose and shape estimator.

2.2 Test-Time Adaptation

Test-time adaptation has been explored through various strategies. Related works [12,13] leverage auxiliary tasks, *e.g.,* image rotation prediction, to guide feature learning during test time. [14] generalizes this idea to reinforcement learning, where action-observation pairs naturally serve as feedback signals. Another line of work focuses on domain-level consistency across a mini-batch of test inputs by minimizing softmax-entropy loss at test time [15,16]. To handle the more challenging scenario of having only a single test sample, [17] uses data augmentation to synthesize a mini-batch. Temporal consistency has also been leveraged as a source of self-supervision [18,19]. These methods maintain a coherent 3D scene over time and render it into 2D views to provide consistent supervisory signals for 2D vision tasks.

Another stream of methods follows a correct-and-certify paradigm [1,2,3,4,5], where model outputs are first corrected, and only those that pass certain certification criteria are used as pseudo-labels for self-training. These methods often rely on auxiliary sensor inputs such as CAD models [5], depth maps [4], or segmentation masks [2]. In contrast, our approach does not assume additional sensor modalities. Instead, it relies on the SAM2 model [20], making the framework simple and easily adaptable to a variety of warehouse automation tasks.

3 Technical Approach

We consider a robot operating in a warehouse environment, equipped with two calibrated RGB stereo-cameras. These cameras capture RGB images of 3D scenes that contain an object of interest. We assume the object to be a storage box parametrized by its width, height, and depth (a, b, c). Let $\boldsymbol{S} = \mathrm{diag}(a, b, c)$ represent the anisotropic scaling factors to a unit cube (*i.e.,* a cuboid with all edges

of length 1). The goal is to compute the pose and shape of the box. Figure 2 shows our pipeline. It consists of a pre-trained stereo keypoint detection model trained on the small labeled dataset or in simulation, an estimator to compute the pose $\boldsymbol{T}$ and the shape $\boldsymbol{S}$ of the box in the 3D scene, and a self-training procedure for the keypoint detection model to improve pose estimation on unlabelled data. We call the resulting approach `BOSS` (Box pOse and Shape estimation with Self-training).

Keypoint Predictor. We use Keypoint-RCNN as our keypoint predictor network [22], with one network for each view. The network outputs the corners of the boxes as keypoints with confidence scores. We only keep keypoints with a confidence score greater than a specified threshold ϵ_{conf}.

Stereo Box Pose and Shape Estimator. We estimate the box pose and shape through a two-view Perspective-n-Point (PnP) optimization problem. The objective is to estimate the object pose and scaling factors that align the reprojected keypoints with their predicted keypoints in the two camera views. The optimization problem is formulated as:

$$\min_{\boldsymbol{T},\boldsymbol{S}} \left\{ \sum_{i=1}^{N^l} \left\| \boldsymbol{\delta}_i^l \right\|_2^2 + \sum_{i=1}^{N^r} \left\| \boldsymbol{\delta}_i^r \right\|_2^2 \right\}, \tag{1}$$

where $\boldsymbol{\delta}_i^l = \Pi^l(\boldsymbol{T} \cdot \boldsymbol{S}\boldsymbol{u}_i) - \tilde{\boldsymbol{y}}_i^l$ and $\boldsymbol{\delta}_i^r = \Pi^r(\boldsymbol{T}_l^r \cdot \boldsymbol{T} \cdot \boldsymbol{S}\boldsymbol{u}_i) - \tilde{\boldsymbol{y}}_i^r$ are the distances between the i^{th} reprojected and predicted keypoints. N^l and N^r are the numbers of keypoints in the left and right images, respectively, $\boldsymbol{u}_i$ are the 3D keypoints on the unit cube centered at the origin of the object frame, $\tilde{\boldsymbol{y}}_i^l$ and $\tilde{\boldsymbol{y}}_i^r$ are the observed 2D keypoint positions in the left and right images, $\boldsymbol{T}_l^r$ is the known transformation from the left to the right camera frame, Π^l and Π^r represent the projection models of the left and right cameras, respectively. In words, Equation (1) minimizes the mismatch between the projections of the estimated box corners (parametrized by the pose $\boldsymbol{T}$ and shape $\boldsymbol{S}$) and the keypoint measurements. The optimization is solved via gradient descent in PyTorch [23]. We relax the rotation matrix constraint and, in each iteration, project the optimized rotation back onto SO(3) using SVD.

Self-Training. To self-train, we use certificates to select pseudo-labels. We use a 2D certificate, a residual certificate, and an epipolar constraint certificate. We admit keypoints as pseudo-labels only if they pass all three certificates.

Let $\hat{\boldsymbol{T}}$ and $\hat{\boldsymbol{S}}$ be the estimated pose and shape from Equation (1) respectively. Our 2D certificate is based on intersection over union (IoU), given by

$$\mathcal{OC}_{2D}(\hat{\boldsymbol{S}}, \hat{\boldsymbol{T}}) = \mathbb{I}\left\{ \frac{\text{ar}(\boldsymbol{M} \cap \hat{\boldsymbol{M}})}{\text{ar}(\boldsymbol{M} \cup \hat{\boldsymbol{M}})} > 1 - \epsilon_{2D} \right\}, \tag{2}$$

where $\text{ar}(\boldsymbol{M})$ denotes the pixel area of all pixels (i, j) in the mask $\boldsymbol{M}$ with $\boldsymbol{M}(i, j) = 1$, and ϵ_{2D} is a given threshold. IoU is computed using a reprojected 3D model $\hat{\boldsymbol{M}}$ and ground truth (GT) or detected segmentation mask $\boldsymbol{M}$.

Our residual certificate filters keypoints based on residuals in Equation (1). Let $\boldsymbol{\delta}$ represent the residuals in the pose and shape estimator, indexed as $\boldsymbol{\delta}^l$ and $\boldsymbol{\delta}^r$ for the left and right views, respectively. The residual-based certificate is defined as

$$\mathcal{OC}_{res}(\tilde{\boldsymbol{y}}, \bar{\boldsymbol{T}}) = \mathbb{I}\left\{\|\boldsymbol{\delta}\|_2 < \epsilon_{res}\right\}. \tag{3}$$

where $\|\cdot\|_2$ is l^2-norm and ϵ_{res} is a tunable threshold. If $\mathcal{OC}_{res}(\tilde{\boldsymbol{y}}, \bar{\boldsymbol{T}}) = 1$, we use $\tilde{\boldsymbol{y}}$ as a pseudo-label. Otherwise, if $\mathcal{OC}_{res}(\tilde{\boldsymbol{y}}, \bar{\boldsymbol{T}}) = 0$, we use the reprojected keypoint $\Pi(\bar{\boldsymbol{T}} \cdot \hat{\boldsymbol{S}}\boldsymbol{u}_i)$ as a pseudo-label, where $\bar{\boldsymbol{T}} = \hat{\boldsymbol{T}}$ for the left view and $\bar{\boldsymbol{T}} = \boldsymbol{T}_l^r \cdot \hat{\boldsymbol{T}}$ for the right view.

We use an epipolar constraint certificate as a final check. Given the known intrinsics and extrinsics of both cameras, we rectify the keypoints so that epipolar lines align with the x-axis, ensuring corresponding points share the same y-coordinates. We then verify the consistency of these y-coordinates in the rectified frames. Denote the rectified keypoints $\tilde{\boldsymbol{y}}_l$ and $\tilde{\boldsymbol{y}}_r$ be $\tilde{\boldsymbol{y}}'_l$ and $\tilde{\boldsymbol{y}}'_r$, respectively. Then epipolar constraint certificate is defined as:

$$\mathcal{OC}_{epi}(\tilde{\boldsymbol{y}}_l, \tilde{\boldsymbol{y}}_r) = \mathbb{I}\left\{(\tilde{\boldsymbol{y}}'_l - \tilde{\boldsymbol{y}}'_r)[2] < \epsilon_{epi}\right\}. \tag{4}$$

where $(\cdot)[2]$ denotes the y-coordinate and ϵ_{epi} is a given threshold.

4 Certificate Validation

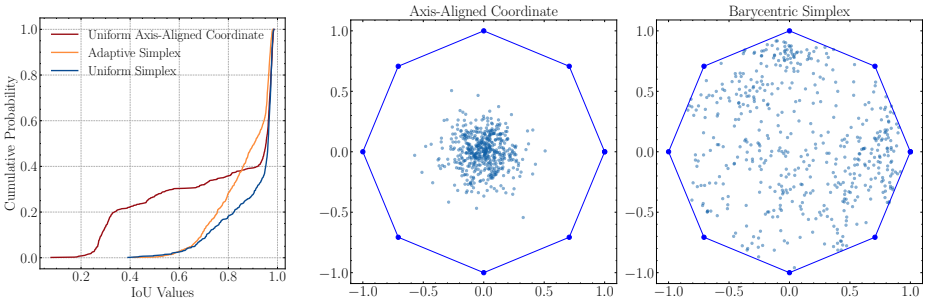
In this section, we empirically validate the three certificates. Over an annotated dataset we show that the certificate scores correlate highly with the ground-truth metrics such as the keypoint root mean square error (RMSE). We compare the three certificate values: (i) IoU (see Equation (2)), (ii) $\|\boldsymbol{\delta}\|_2$ (see Equation (3)), and (iii) $(\tilde{\boldsymbol{y}}'_l - \tilde{\boldsymbol{y}}'_r)[2]$ (see Equation (4)), with the keypoint RMSE.

4.1 Validation of 2D Certificates

Figure 4a validates the effectiveness of the 2D certificate $\mathcal{OC}_{2D}$. This plot helps us compare how the IoU score, which can be computed at test time, correlates with the RMSE with the ground-truth keypoints. A clear trend is observed: higher IoU scores correspond to lower RMSE values. Notably, pseudo-labels with IoU values exceeding 0.95 yield average keypoint errors below 10 pixels, which is small relative to the image resolution (1640×1232). This empirical relationship supports the use of $\mathcal{OC}_{2D}$ as a reliable proxy for keypoint accuracy during pseudo-label validation.

4.2 Validation of Residual Certificates

In Figure 4b we validate the residual certificate $\mathcal{OC}_{res}$. We plot the residual certificate value $\|\boldsymbol{\delta}\|_2$ (see Equation (3)) on the x-axis. On the y-axis, we plot the count of instances where the predicted RMSE is either greater than (blue curve) or less than (red curve) the reprojected RMSE, across varying residual certificate values. The predicted RMSE is the RMSE of the predicted keypoints that are output directly from the keypoint network, and the reprojected RMSE is



(a) (b)

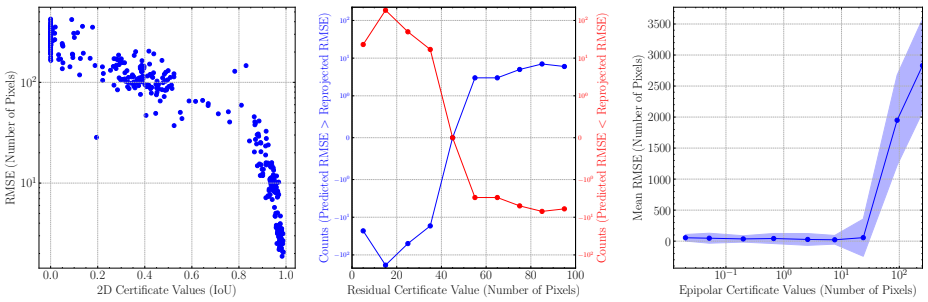
Fig. 3: Analysis of SAM2 sampling strategies. (a) Cumulative distribution of IoU values for three SAM2 sampling strategies. The x-axis shows the IoU between predicted and ground-truth segmentations, and the y-axis indicates cumulative probability. Uniform Simplex method outperforms both Adaptive Simplex and Uniform Axis-Aligned Coordinate. (b) Visual comparison of two sampling strategies within a regular octagon. With the same number of samples, the Axis-Aligned Coordinate sampling is densely concentrated near the center, while the Barycentric Simplex sampling provides more uniform coverage of the polygon.

the RMSE of the optimized keypoints reprojected from the PnP pose and shape estimator. Note that the x-axis is the certificate value that can be computed at test time, whereas the y-axis (*i.e.,* predicted and reprojected RMSE) requires knowledge of the ground truth.

We again observe a clear trend. For low residual values (left side of the x-axis), the majority of instances fall under the red curve, indicating that predicted keypoints are more accurate than reprojected ones. As the residual certificate increases, the trend reverses—beyond a residual value of approximately 42 pixels, the reprojected keypoints tend to outperform the predicted ones, as indicated by the rising blue curve. This transition point around 42 suggests an empirical threshold at which the residual certificate reliably filters high-quality predictions.

4.3 Validation of Epipolar Constraint Certificates

In Figure 4c, we validate the epipolar constraint certificates. We plot the epipolar certificate value (*i.e.,* Equation (4)) on the x-axis. The corresponding mean RMSE of the selected pseudo-label keypoints and ground-truth keypoints (in pixels) are plotted on the y-axis. Note that while the epipolar certificate value can be computed at test time without the knowledge of the ground truth, the mean RMSE requires ground truth. We again observe a clear trend. As shown in Figure 4c, we report the mean RMSE (in pixels) across test samples as a function of the epipolar certificate values. At low thresholds (*e.g.,* $<$20 pixels), the RMSE remains consistently low. However, as the value increases beyond 20 pixels, the RMSE grows rapidly, along with its variance. This behavior highlights the importance of enforcing epipolar certificates.



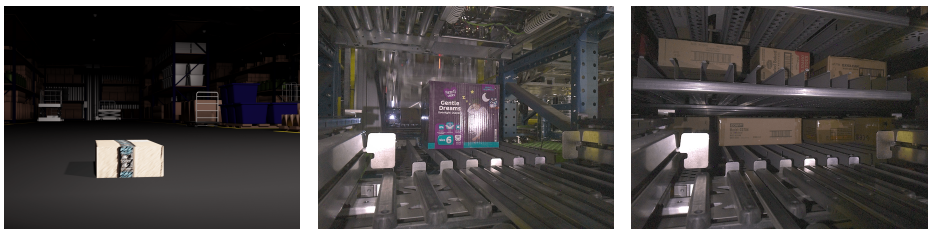
(a) (b) (c)

Fig. 4: (a)Validation of the 2D Certificate (*i.e.,* Equation (2)). The x-axis is the IoU score in the $\mathcal{OC}_{2D}$ certificate. The y-axis is the RMSE error of the pseudo-labels averaged across one image sample. We use ground-truth segmentation for IoU calculation. Pseudo-labels with IoU values larger than 0.95 have average keypoint errors of fewer than 10 pixels which is small relative to the image size (1640×1232). (b) Validation of the Residual Certificate (*i.e.,* Equation (3)). The x-axis is the residual value and the y-axis is the counts of keypoints that either predicted ones are more accurate (red) or reprojected ones are more accurate (blue). We found that there is a heuristic threshold that enables hybrid keypoint selection. (c) Validation of Epipolar Constraint Certificate (*i.e.,* Equation (4)). The x-axis is the discrepancy of the y coordinates between rectified selected pseudo-label keypoints and ground truth keypoints in the $\mathcal{OC}_{epi}$ certificate. Y-axis is the RMSE error of the pseudo-labels averaged across one bin batch. This figure highlights the importance of enforcing epipolar constraint certificates.

4.4 Impact of Sampling Strategies

In scenarios where the ground truth mask $\boldsymbol{M}$ is unavailable—commonly the case in industrial settings—we leverage the SAM2 model [20] to generate pseudo-ground truth masks for object boxes. To produce masks, SAM2 requires samples in the pixel space. In this section, we examine how various sampling strategies influence the quality of the resulting pseudo-ground truth masks.

Figure 3 analyzes the effect of different sampling strategies on SAM2 segmentations for a 2D convex polygon. We only discuss 2D convex polygon because the 2D projection of a box is a polygon. We consider three strategies: **(1) Uniform Axis-Aligned Coordinate:** Candidates are generated by taking convex combinations of the polygon's vertices. Specifically, we sample a non-negative weight for each vertex from a uniform distribution over $[0, 1]$, then normalize the weights so that they sum to 1. **(2) Uniform Simplex:** Candidates are sampled uniformly from the convex hull of the polygon's vertices using a triangulation-based approach. The polygon is first decomposed into simplices (i.e., triangles in 2D), and a simplex is selected via importance sampling, with the selection probability proportional to its area. A point is then sampled uniformly within the chosen simplex using barycentric coordinates, ensuring uniform coverage across the entire polygon [24]. **(3) Adaptive Simplex:** Similar to **Uniform Simplex**, but with a key difference: while **Uniform Simplex** uses a constant number of



(a) Simulation (b) inbound buffer (c) storage aisles

Fig. 5: Sample images from the simulated and real datasets used in the experiments.

samples regardless of the area of the triangle, **Adaptive Simplex** scales the number of samples proportionally to the triangle's area.

In Figure 3a, the cumulative IoU distribution shows that Uniform Simplex sampling significantly outperforms both Uniform Axis-Aligned Coordinate and Adaptive Simplex, achieving a higher proportion of accurate segmentation masks. Figure 3b visualizes the core difference by simulating sampling in a regular octagon. In Axis-Aligned Coordinate sampling, points tend to cluster densely near the center of the feasible region and are sparsely distributed near its boundaries while Simplex sampling generates points uniformly in the polygon.

5 Experiments

We conducted three sets of experiments to evaluate `BOSS`. First, we validated the effectiveness of our pipeline on a synthetic dataset (Section 5.1). Next, we demonstrated its ability to bridge the sim-to-real gap (Section 5.2). Finally, we will demonstrate its ability to perform self-supervised learning using a large-scale unlabeled dataset (Section 5.3).

	Pose and Shape Estimation Comparison					
	Sim2Sim			**Sim2Real**		
Approach	APE [m]	ARE [rad]	ASE [m]	APE [m]	ARE [rad]	ASE [m]
`Model w/o SSL`	0.584	0.219	0.369	2.080	0.554	1.589
`BOSS-SAM2`	0.038	0.069	0.084	0.134	0.223	0.238
`BOSS-GT`	0.041	0.063	0.078	0.148	0.239	0.259
`BOSS-SAM2 (50k)`	-	-	-	0.135	0.217	0.247
`Model Supervised`	0.024	0.053	0.045	0.111	0.212	0.208

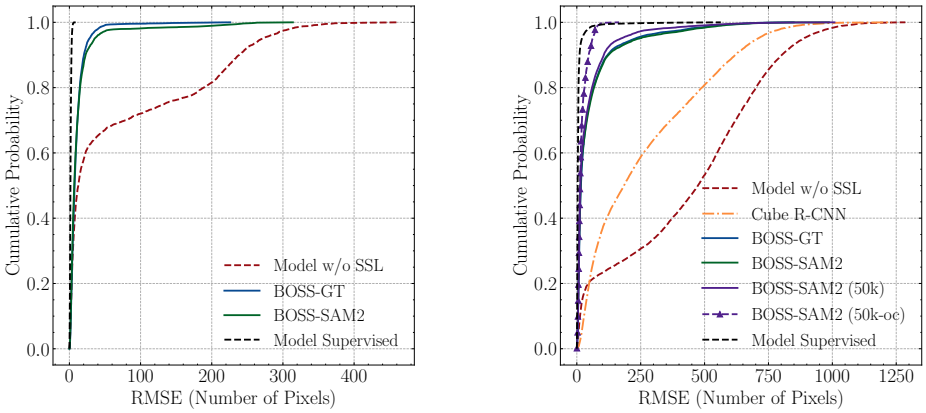
Table 1: Pose and shape estimation for self-supervised pipeline and other baselines. APE denotes the average position error; ARE denotes the average rotation error; ASE denotes the average shape error.

5.1 Validation on Synthetic Dataset

Setup. We use Blender to generate a dataset comprising a training dataset of 75 images and a test dataset of 375 images featuring five types of boxes. A typical example from the synthetic dataset is shown in Figure 5a. The training dataset includes images captured from a fixed viewpoint of a single object type with varying lighting conditions and randomized object poses, while the test dataset features both novel views of known objects and entirely new objects. We test `BOSS`' ability to perform self-supervised learning on the test dataset.
Results and Insights. Keypoint detection results are shown in Figure 6a. The baseline model without self-supervised learning `Model w/o SSL` is trained solely on the simulation training dataset. In contrast, the self-supervised models are trained on the same dataset but also perform self-supervised learning on the test dataset without annotations. This model has two variations: one using ground truth segmentation for the 2D certificate `BOSS-GT` and another using SAM2 masks [20] `BOSS-SAM2`. Finally, the supervised model `Model Supervised` is trained directly on the simulation test dataset (*i.e.,* this is the best achievable performance with the architecture). Our goal is to fill the area between curves of `Model w/o SSL` and `Model Supervised`, commonly known as the domain gap. Notably, SSL effectively bridges this gap, with up to 90% of keypoints exhibiting errors below 20 pixels—remarkably small relative to the image resolution of 1640×1232. Table 1 presents the pose and shape statistics. The model with SSL significantly enhances pose and shape estimation, achieving accuracy more than 10 times higher —nearly matching that of a supervised model— with only around 4cm average error for position estimation; for reference, the average dimension of the simulated boxes is 0.23m. Notably, for both keypoint detection and pose and shape estimation, the SAM2 variant performs comparably to the GT variant.

5.2 Adaptation to Real Dataset

Setup. Symbotic provided a dataset with 9,000 images (Symbotic-9k), including various types of boxes in two industrial environments: buffer shelves at inbound (Figure 5b) and storage aisles (Figure 5c). The dataset provides keypoint annotations for stereo images, with keypoints predefined as the box corners. Symbotic-9k is split into 7k/0.5k/1.5k images for train/val/test respectively.
Results and Insights. We evaluate all models on the test dataset split and show results in Figure 6b. `Model w/o SSL`, trained solely on synthetic data, serves as a lower bound. The upper bound model `Model Supervised` is trained and validated on the train/val dataset. The area between `Model w/o SSL` and `Model Supervised` is referred to as a sim-to-real gap. `BOSS` has two variations using GT segmentation `BOSS-GT` or SAM2 `BOSS-SAM2`. Both models are first trained on synthetic data and then refined through self-supervised learning on the train and validation datasets. For comparison, we include `Cube R-CNN`, an RGB-only zero-shot bounding box prediction model trained on the large-scale Omni3D benchmark [11] (234k images) using 48 V100 GPUs, covering both indoor and outdoor environments. The results clearly show that the SSL models, initially trained on synthetic data and adapted using GT or SAM2-learned segmentation, successfully bridge the sim-to-real gap. It also outperforms `Cube R-CNN` by a large margin. Table 1 presents detailed results on the pose and shape estimation. Since ground truth pose and shape are unavailable for the real dataset, we generate



(a) Sim2Sim keypoint detection comparisons for the proposed self-supervised architecture with upper bound and other baselines.

(b) Sim2Real keypoint detection comparisons for the proposed self-supervised architecture with zero-shot large model and other baselines.

Fig. 6: Comparison of keypoint detection performance in Sim2Sim (left) and Sim2Real (right) scenarios.

pseudo ground truth by running our pose and shape estimator on ground truth keypoints. Notably, the self-supervised model consistently improves both pose and shape estimation with significantly lower errors for all position, rotation, and shape estimation. Self-supervised baseline also approaches the performance of the supervised upper bound. We also observe that, for both keypoint detection and the pose and shape estimation, the SAM2 variation has a very similar performance to the GT variation.

5.3 Adaptation to Large-scale Dataset

Setup. We are interested in how performance scales with the size of the dataset. Symbotic provides an additional dataset of about 50,000 images, referred to as Symbotic-50k, which however has no ground-truth keypoint annotations. `BOSS-SAM2 (50k)` is first pre-trained on synthetic data and then refined via self-supervised learning using a combination of the train and validation datasets, along with Symbotic-50k. Note that for all `BOSS-GT`, `BOSS-SAM2`, and `BOSS-SAM2 (50k)`, we use the same certificate thresholds to have a fair comparison. We additionally report the performance of the model when evaluated only on outputs that pass all certificate checks, denoted as `BOSS-SAM2 (50k-oc)`. We present the keypoint detection results on the test split in Figure 6b. Pose and shape estimation results are presented in Table 1.

Results and Insights. Interestingly, `BOSS-SAM2 (50k)` outperforms `BOSS-SAM2` and `BOSS-GT` by a small margin. This suggests that keypoint detection performance scales with dataset size. We can gain further performance improvement by filtering out bad labels during inference as shown `BOSS-SAM2 (50k-oc)`, whose performance is quite close to that of the supervised baseline. However, the improvement of `BOSS-SAM2 (50k)` compared with `BOSS-SAM2` is limited. We believe

this can be improved in the future by an automatic certificate threshold update scheme during training. The current training uses a fixed threshold profile.

6 Conclusions

A self-supervised approach can train a box pose and shape estimation model using large-scale, unannotated data collated by a robot fleet in a warehouse. Implementing a simple pipeline to estimate the pose and shape of a box, we show that it can be self-trained leveraging our correct-and-certify approach. The correct-and-certify approach implements certificates to pseudo-label instances during training but requires hard thresholds to be set apriori for training. We devise an empirical way to choose these thresholds and demonstrate that our training can bridge a large domain gap. Several avenues remain open for future research. First, rather than applying hard thresholds to model outputs, can we use soft pseudo-labels to retain more information? This idea is motivated by the observation that certificate values naturally reflect the confidence level of each pseudo-label. Second, we are interested in extending pose and shape estimation to irregularly shaped objects, which would significantly improve generalization across diverse warehouse tasks. Potential solutions include incorporating shape parametrization [25] or learning a latent shape representation [26,27].

References

1. H. Yang, W. Dong, L. Carlone, and V. Koltun, "Self-supervised geometric perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 350–14 361.
2. J. Shi, R. Talak, D. Maggio, and L. Carlone, "A correct-and-certify approach to self-supervise object pose estimators via ensemble self-training," *arXiv preprint arXiv:2302.06019*, 2023.
3. R. Talak, L. R. Peng, and L. Carlone, "Certifiable object pose estimation: Foundations, learning models, and self-training," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2805–2824, 2023.
4. J. Shi, R. Talak, H. Zhang, D. Jin, and L. Carlone, "Crisp: Object pose and shape estimation with test-time adaptation," *arXiv preprint arXiv:2412.01052*, 2024.
5. M. Jawaid, R. Talak, Y. Latif, L. Carlone, and T.-J. Chin, "Test-time certifiable self-supervision to bridge the sim2real gap in event-based satellite pose estimation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 4534–4541.
6. G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2011–2018.
7. H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2642–2651.
8. M. Tian, M. H. Ang Jr, and G. H. Lee, "Shape prior deformation for categorical 6d object pose and size estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 530–546.
9. Y. Fu and X. Wang, "Category-level 6d object pose estimation in the wild: A semi-supervised learning approach and a new dataset," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 469–27 483, 2022.

10. N. Kai, H. Yoshida, and T. Shibata, "Pallet pose estimation based on front face shot," *IEEE Access*, 2025.
11. G. Brazil, A. Kumar, J. Straub, N. Ravi, J. Johnson, and G. Gkioxari, "Omni3D: A large benchmark and model for 3D object detection in the wild," in *CVPR*. Vancouver, Canada: IEEE, June 2023.
12. Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.
13. Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt, "Test-time training with self-supervision for generalization under distribution shifts," in *International conference on machine learning*. PMLR, 2020, pp. 9229–9248.
14. N. Hansen, R. Jangir, Y. Sun, G. Alenyà, P. Abbeel, A. A. Efros, L. Pinto, and X. Wang, "Self-supervised policy adaptation during deployment," *arXiv preprint arXiv:2007.04309*, 2020.
15. D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020.
16. S. Goyal, M. Sun, A. Raghunathan, and J. Z. Kolter, "Test time adaptation via conjugate pseudo-labels," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6204–6218, 2022.
17. M. Zhang, S. Levine, and C. Finn, "Memo: Test time robustness via adaptation and augmentation," *Advances in neural information processing systems*, vol. 35, pp. 38 629–38 642, 2022.
18. R. Zurbrügg, H. Blum, C. Cadena, R. Siegwart, and L. Schmid, "Embodied active domain adaptation for semantic segmentation via informative path planning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8691–8698, 2022.
19. N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 901–14 910.
20. N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714
21. J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4331–4339.
22. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
23. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
24. M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. MIT Press, 2023.
25. M. Shan, Q. Feng, Y.-Y. Jau, and N. Atanasov, "Ellipsdf: Joint object pose and shape optimization with a bi-level ellipsoid and signed distance function description," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5946–5955.
26. P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani, "Autosdf: Shape priors for 3d completion, reconstruction and generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 306–315.
27. M. Liu, R. Shi, K. Kuang, Y. Zhu, X. Li, S. Han, H. Cai, F. Porikli, and H. Su, "Openshape: Scaling up 3d shape representation towards open-world understanding," *Advances in neural information processing systems*, vol. 36, pp. 44 860–44 879, 2023.