Self-supervised Learning Of Visual Pose Estimation Without Pose Labels By Classifying LED States

Nicholas Carlotti[1], **Mirko Nava**[1], **and Alessandro Giusti**[1*]

Abstract: We introduce a model for monocular RGB relative pose estimation of a ground robot that trains from scratch without pose labels nor prior knowledge about the robot's shape or appearance. At training time, we assume: (i) a robot fitted with multiple LEDs, whose states are independent and known at each frame; (ii) knowledge of the approximate viewing direction of each LED; and (iii) availability of a calibration image with a known target distance, to address the ambiguity of monocular depth estimation. Training data is collected by a pair of robots moving randomly without needing external infrastructure or human supervision. Our model trains on the task of predicting from an image the state of each LED on the robot. In doing so, it learns to predict the position of the robot in the image, its distance, and its relative bearing. At inference time, the state of the LEDs is unknown, can be arbitrary, and does not affect the pose estimation performance. Quantitative experiments indicate that our approach: is competitive with SoA approaches that require supervision from pose labels or a CAD model of the robot; generalizes to different domains; and handles multi-robot pose estimation.

Keywords: Self-supervised Learning, Pretext Task, Visual Pose Estimation

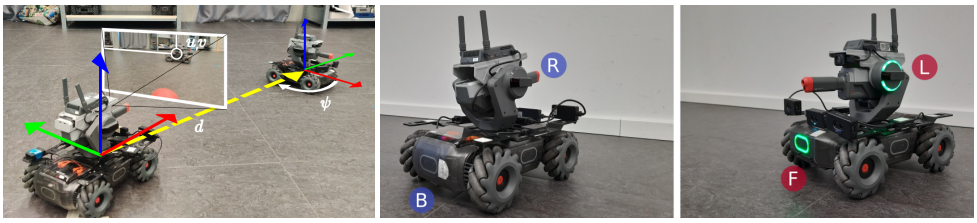


Figure 1: By solving the multi-LED state classification task (blue for off LEDs, red for on; F for front, B for back, L for left, R for right), our model learns from scratch to estimate the location of the robot (u, v) in the image, its relative distance d, and relative bearing angle ψ w.r.t. the camera. These variables are used in combination with the camera intrinsics to recover the pose of the robot.

1 Introduction

Relative localization of mobile robots is fundamental for applications involving multiple robots that must coordinate with each other [1]. SoA pose estimation approaches solve the problem by training a deep neural network in a supervised way, assuming access to a large dataset of images representing the robot, each labeled with its true relative pose [2, 3]. Acquiring such a dataset in real environments is expensive and requires external infrastructure for generating ground truth labels [4]. Acquiring it in simulation requires a realistic and textured CAD model of the robot, while trained predictors suffer from the sim-to-real gap [5]. Novel object pose estimation approaches also assume access to CAD models, used to generate templates with known robot poses and matched with patches of the input

*[1] All authors are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, 6962, Switzerland nicholas.carlotti@idsia.ch. This work is supported by the Swiss National Science Foundation, grant number 213074.

image using learning-based descriptors [6, 7, 8]. However, the availability of visually-accurate and up-to-date CAD models of the robot is a strong requirement, especially for custom-built platforms.

In this work, we assume no access to data labeled with robot poses nor prior knowledge of the robot's 3D shape or texture. In contrast, we assume each robot to be equipped with K LEDs that can be independently turned on and off, and to know the approximate direction from which each LED is visible relative to the robot's heading. In this challenging scenario, we propose a novel approach for training a robot pose estimation model from scratch: training data is collected fully autonomously, without any external supervision, by a pair of robots that move in an arbitrary, possibly unknown way in the environment, without a shared frame of reference. Each robot broadcasts (e.g., via a radio link) the true state (on or off) of all of its LEDs, which are toggled multiple times during data collection, independently from each other, and in a random way. Our approach handles data that features a visible robot infrequently: considering that robot poses are unavailable during data collection, each robot will be visible in the other's Field of View (FOV) only in a small, unknown subset of the frames acquired. Given a frame of the camera taken from a robot, called observer, our model predicts the state of each of the K LEDs of the other robot, called target, i.e., a classification problem on K independent binary labels. This is a *pretext task* as solving it is not our ultimate goal; indeed, our *end task* is to predict the full pose (position and orientation) of the target.

During inference, the LEDs are no longer needed: our model estimates, directly from a camera frame, the robot's position in the image, its bearing relative to the camera, and its apparent image size (see Figure 1). We compute the metric distance of the robot from its apparent image size with a calibration based on a single image depicting the robot and annotated with its distance from the camera, a similar assumption to Depth Anything [9, 10].

To the best of our knowledge, no other SoA approach learns pose estimation without pose labels or a CAD model of the robot. Through a careful design of the neural network architecture and loss function, the model is forced to understand the robot's structure, which is crucial for pose estimation. We further remark that the model is given no information on how the LEDs appear visually; our model learns to recognize them as part of its classification task. The only assumption is that the LED state affects the robot's appearance in a way that is observable from an image of the robot acquired from a given, approximately known range of directions.

Our **main contribution** is methodological; we propose an approach for learning visual pose estimation of robots by training on the self-supervised pretext task of multi-LED state classification. Experimental results indicate that: (i) the approach trains pose estimation models that are competitive with SoA approaches requiring supervision from pose labels or a CAD model of the robot; (ii) training is robust to data featuring a visible robot only in a limited amount of camera frames; (iii) the LED state does not impact the pose estimation performance; (iv) models generalize to unseen environments with no fine-tuning and are capable of multi-robot pose estimation; (v) the approach can fine-tune a pre-trained model to a different deployment environment.

2 Related Work

Supervision in Visual Object Pose Estimation. Approaches designed for visual object pose estimation can be directly applied to robots. In this context, different assumptions about the object's appearance are made, serving as supervision during training. Traditional approaches assume access to a large dataset labeled with object poses [11, 2, 3]. A less strict assumption is to have access to a realistic textured CAD model of the object, used to generate inexpensive simulated data [12, 13, 14, 15, 16]. Recent works leverage deep template matching [17] to estimate the pose of novel objects, further leveraging foundation models such as Segment Anything Model (SAM) [18] to segment objects that are matched using DINOv2 [19] features with rendered templates of the object at known poses [6]. The pose of the matched template is refined with a point matching stage [20, 21], with PnP [7, 22], or by iteratively minimizing the optical flow between the template and segmented image [8]. Approaches based on NeRF [23] or Gaussian Splatting [24] assume access to images

of a static scene labeled with the camera pose. They are used to estimate the pose of objects by iteratively minimizing the photometric error with [25] or without an initial pose estimate [26, 27].

All the above methods learn pose estimation on data annotated by strong supervision sources (e.g., tracking system, textured CAD model of objects, calibrated views). By contrast, our proposed approach makes no such assumption: we rely on pose-free, real-world data autonomously generated by two robots and labeled only with the binary state of multiple and independent LEDs.

Weakly Supervised Learning in Computer Vision. In Weakly Supervised Learning (WSL), object detection and image segmentation are learned by training on a classification task with inexpensive image-level labels [28]. Class Activation Map (CAM) [29, 30, 31] approaches are used to find the most discriminative areas for classifying the image. Crucially, discriminative areas for an image classification task depict the object of interest, enabling object detection [32] and segmentation [33]. WSL enables manipulation from image-level labels [34], where the reward signal for an RL agent tasked to pick and place objects is derived from the embeddings of an image depicting an object and another with the object removed. In our work, we take inspiration from WSL in computer vision and go beyond a simple detection task: we introduce a novel approach for full pose estimation of robots, learned by the model as a result of solving a multi-label binary classification task.

Self-supervised Robot Learning. Learning complex robotic tasks from real-world data requires labels as a form of supervision, whose collection is time-consuming and expensive. To reduce reliance on labeled data, approaches pre-train models on pretext tasks to learn valuable pattern recognition skills [35, 36, 37, 38, 39], and later fine-tune them with a limited amount of labels on the end task of interest: as an example, approaches mask image patches and task a model to fill out the missing areas, given as input the masked image [35] and additional views of the scene [36, 37]. Recently, works showed that LED state classification pretext task is conducive to fine-tuning with labeled data on the task of visual robot detection [38], and pose estimation [39]. By contrast, our approach does not require fine-tuning or strongly annotated data to learn robot pose estimation.

3 Method

Given an RGB image $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ of width W and height H collected by the observer robot, our model estimates the 2D pose $\mathbf{P} = \langle x, y, \psi \rangle$ of the target robot relative to observer's camera. Additionally, the model classifies the state (on or off) of K independent LEDs mounted on the target's body and visible from a known range of directions. Formally, we define the deep learning model $m_{\boldsymbol{\theta}}(\boldsymbol{I}) = \langle \hat{u}, \hat{v}, \hat{d}, \hat{\psi}, \hat{\boldsymbol{l}} \rangle$ where $\boldsymbol{\theta}$ are the model parameters; $\hat{u}$ and $\hat{v}$ are the image-space coordinates of the robot; $\hat{d}$ is the distance of the robot from the camera in the scene; $\hat{\psi}$ is the robot's orientation relative to the camera; and $\hat{\boldsymbol{l}} = \langle \hat{l}_1, \ldots, \hat{l}_K \rangle$ are the predicted probabilities that each of the K LEDs is turned on. Using the model prediction, we recover the robot pose by back-projecting its image location using the camera intrinsic parameters, selecting the point at the estimated distance from the optical center, and combining it with the rotation $\hat{\psi}$. We optimize the model parameters $\boldsymbol{\theta}$ through gradient descent with a Binary Cross Entropy (BCE) loss defined on the LED states; details on the neural network architecture and training hyper-parameters can be found in the appendix.

Our model architecture is a Fully Convolutional Network (FCN) [40] composed solely of convolution and pooling layers. It outputs a set of two-dimensional feature maps (maps for short) composed of cells. Specifically, we exploit that each output cell of a FCN attends only to a local area of the input image, represented by the Receptive Field (RF). Given a monocular image, our model produces an LED state map $\hat{L}^k$ for each LED, a localization map $\hat{P}$, and a relative bearing map $\hat{\Lambda}$. All these maps have the same $H' \times W'$ shape. Each cell of the $\hat{L}^k$ map takes values in the $[0, 1]$ range, indicating the confidence that the k-th LED is turned on or off (represented by 1 and 0, respectively). If the LED is not visible inside the RF, the cell will have a value of 0.5 to indicate uncertainty; this is the case for most cells whose RF captures background areas of the image. To this end, we define a multi-label binary classification task on the LED states, with the loss $\mathcal{L}^k_{\text{led}} = \text{BCE}(\hat{l}^k, l^k)$ computed

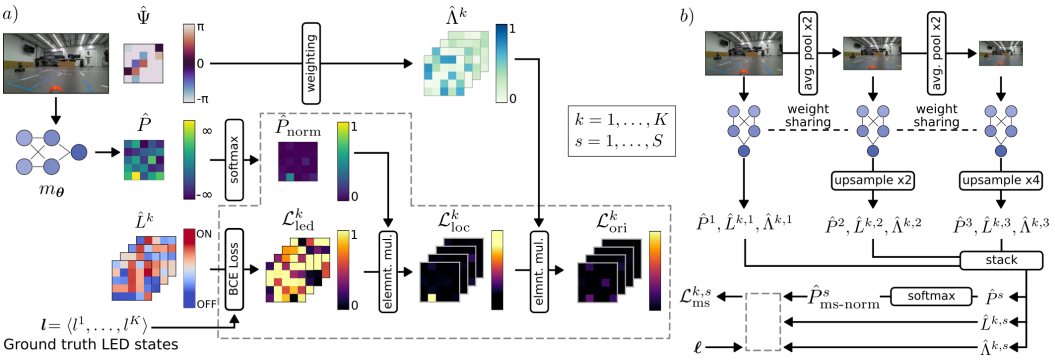


Figure 2: Overview of the approach: (a) given an input image, our approach predicts the robot's location in the image and its bearing relative to the camera. (b) We apply this mechanism over multiple rescaled versions of the input image to infer the robot's distance to the camera.

between each cell $\hat{l}^k$ of the LED state map $\hat{L}^k$ and the ground truth state l^k of the k-th LED; as such, the loss $\mathcal{L}^k_{\text{led}}$ is itself composed of maps divided into cells, having one map for each of the K LEDs.

Robot Localization. Most cells in $\hat{L}^k$ cannot predict the correct LED state because their limited RF does not capture the robot, leading to high values in the $\mathcal{L}^k_{\text{led}}$ loss maps. An intuitive way to lower the loss is to give less weight to errors corresponding to areas not depicting a robot and give more weight to the cells that see the robot (i.e., have the robot inside the RF) as they can predict the LED states. Thus, we allow the model to spatially weight the $\mathcal{L}^k_{\text{led}}$ maps through the $\hat{P}$ map; each cell in $\hat{P}$ takes values in the $[0, 1]$ range, and indicates the belief about the robot's presence inside its RF. We normalize this map with the softmax function, denoted as $\hat{P}_{\text{norm}}$, and define the localization loss $\mathcal{L}^k_{\text{loc}} = \mathcal{L}^k_{\text{led}} \odot \hat{P}_{\text{norm}}$, where $\odot$ indicates the element-wise product; the softmax prevents the model from trivially setting the loss to zero. With this formulation, the model is driven to produce high values in the cells of $\hat{P}$ whose RF contains the robot, as these will generally correspond to the low loss cells of $\mathcal{L}^k_{\text{led}}$. This weighting mechanism can be seen as spatial attention in CNNs [41], with the difference being that we apply it directly to a loss function instead of raw model features.

Robot Relative Bearing Estimation. Whenever the robot is visible in the image, some of its LEDs are occluded by its own body. Since these LEDs are not visible, the model is unable to predict their state, contributing to high values in the $\mathcal{L}^k_{\text{loc}}$ maps. We introduce a weighing mechanism for the localization loss based on the predicted robot's bearing; it allows the model to downplay errors caused by occluded LEDs as long as it correctly predicts the robot's bearing. To accomplish this, we introduce the predicted robot's bearing map $\hat{\Psi}$, whose cells have values in the $[-\pi, \pi]$ range. Each cell represents the robot's bearing relative to the camera, i.e., which side of the robot is visible, encoded as an angle (see Figure 1). We use a differentiable function to map the predicted bearing $\hat{\Psi}$ to K visibility scores, one for each LED, resulting in the $\hat{\Lambda}^k$ maps. Each element of $\hat{\Lambda}^k$ is defined as $\hat{\lambda}^k = \cos(\hat{\psi} + \frac{2\pi}{K}(k-1))$, where $\hat{\psi}$ is an element of $\hat{\Psi}$; we designed $\hat{\Lambda}^k$ for K equidistant LEDs mounted on the robot. This visibility function reasonably approximates each LED's visibility from different viewing directions, though it is not precise nor the result of calibration. We normalize the map values such that, given an orientation, all K coefficients are non-negative and sum to one (see Figure 3). The $\hat{\Lambda}^k$ maps are then used to compute the orientation loss $\mathcal{L}^k_{\text{ori}} = \mathcal{L}^k_{\text{loc}} \odot \hat{\Lambda}^k$.

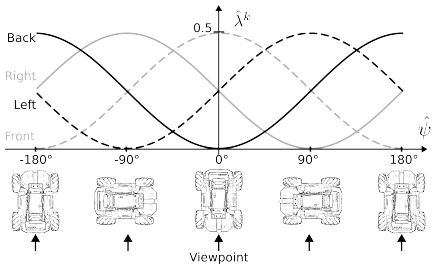


Figure 3: Visibility function for a robot with four LEDs at the cardinal directions. $\hat{\psi}$ is a cell of the predicted bearing map $\hat{\Psi}$, and $\hat{\lambda}^k$ is the visibility weight for each of the K LEDs.

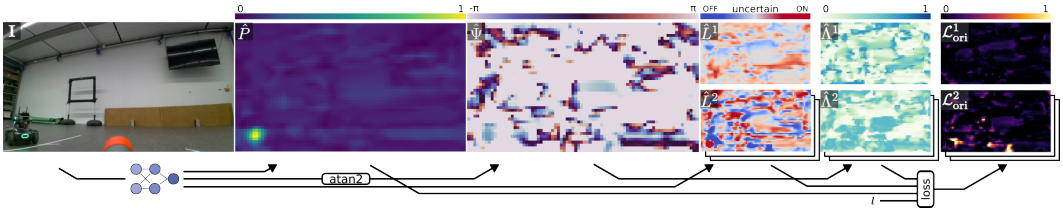


Figure 4: Model's output maps (a single scale was selected for visualization purposes). The $\hat{P}$ map represents the robot's presence, $\hat{\Psi}$ its bearing, and $\hat{L}^k$ the LED states (1 for back, 2 for front).

Robot Distance Estimation. Given an image, when the robot's apparent size and that of the RF do not match well, it interferes with correctly predicting the LED states: if the robot appears much larger than the RF, the cell will miss contextual information about the robot's structure; if it is much smaller, the RF will contain unrelated background information as the robot will be represented by few pixels. We address this problem and exploit its solution to estimate the relative robot's distance, as the apparent image size of an object is directly proportional to its distance from the camera: we pass the same image at different scales $\{s_1, \ldots, s_S\}$ to the model and consider the output maps $\hat{L}^{k,s}$, $\hat{P}^s$, $\hat{\Psi}^s$ from each forward pass, where k is an LED and s an image scale, as shown in Figure 2. The output maps from the different passes are upscaled to have matching spatial dimensions. Then, we compute the normalized multi-scale localization map $\hat{P}_{\text{ms-norm}} = \text{softmax}(\hat{P}^{s_1,\ldots,s_S})$. We combine the previous loss with the multi-scale formulation and define the loss $\mathcal{L}^{k,s}_{\text{ms}} = \text{BCE}(\hat{L}^{k,s}, l^k) \odot \hat{P}^s_{\text{ms-norm}} \odot \hat{\Lambda}^{k,s}$. By normalizing over all image scales, we enable the model to make a convex combination of scales, ensuring the robot size best fits into the combined multi-scale RFs. The coefficients of the combinations are used to compute the robot's distance from the camera.

Finally, we define the complete loss function in Equation (1) to obtain a scalar loss value. It computes the average over the spatial, LED, and scale dimensions of the multi-scale loss, where $X[i,j]$ is the indexing operator accessing the value of the cell at row i and column j of a generic map X. Note that this loss accounts for all aspects captured by $\mathcal{L}_{ori}$, reformulated in multi-scale fashion.

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} \sum_{s=1}^{S} \sum_{i=1}^{H'} \sum_{j=1}^{W'} \mathcal{L}^{k,s}_{\text{ms}}[i,j] \tag{1}$$

Model Inference. We compute the predicted robot location in the image $(\hat{u}, \hat{v})$ as the barycenter of the $\hat{P}^s_{\text{ms-norm}}$ maps. The predicted relative bearing is the weighted average of $\hat{\Psi}^s$ by $\hat{P}^s_{\text{ms-norm}}$ and is directly mapped to the robot's orientation. Lastly, we estimate the robot's distance by measuring its apparent size in the original input image, which is proportional to its physical distance from the camera. We identify the robot's apparent size in the image by summing each $\hat{P}^s_{\text{ms-norm}}$ map over its spatial dimensions, obtaining a vector of S elements that sum to one. Using this vector to apply a linear combination of scale factors, we obtain a scalar representing the robot's size in the original input image. Similarly to Depth Anything [9, 10], the predicted distance requires a calibration: we multiply it with the calibration factor d_c derived from a single image with a known robot distance. Note that we do not explicitly handle external occlusions to the LEDs. However, our experiments show the model is resilient to partial and total occlusions caused by the camera's FOV. Following the inference procedure, we report an average running time of 6.5ms per image (153 Hz) on a NVIDIA GeForce RTX 4080. More details on model inference can be found in the appendix.

4 Experimental setup

We apply our proposed approach to the DJI Robomaster S1 robot, a ground rover equipped with a monocular RGB camera with a resolution of 640×360 pixels mounted on top of a pan and tilt turret. The robot features six multi-color LEDs, of which we consider four for our experiments: the two on

Figure 5: Random training samples from the datasets: laboratory (top row), gym (1-3 on bottom), classroom (4-6 on bottom), and break room (7-9 on bottom). On the right, the LED state (blue for off, red for on; F for front, B for back, L for left, R for right) is reported for each LED; white circles mark the robot. Only 23% of collected samples feature a visible robot in the laboratory training set.

the turret and the front and back ones on the robot's base (see Figure 1); the left and right LEDs of the base are always turned off and ignored during training and evaluation.

Data Collection. We let the two robots randomly move in different environments and independently randomize the state of each of the four LEDs every five seconds. Each robot broadcasts its LED states which are used as labels for the images acquired by the other one. In total, the robots collected 131K samples in a laboratory environment, which are sequentially split into the the training $\mathcal{T}_{\text{lab}}$ (116K samples), validation $\mathcal{V}_{\text{lab}}$ (10K samples), and testing $\mathcal{Q}_{\text{lab}}$ (5K samples) sets. Having no access to pose labels while randomly exploring, no measures are taken to ensure that the robots are in each other's FOV; consequently, 77% of the training-set images depict an empty background with no robots, as shown in the top row of Figure 5. To validate the model and assess its performance, the pose of both robots is collected using a motion capture system. Our quantitative evaluation is carried out on the subset of testing set samples with a visible robot, amounting to 1K samples, denoted as $\mathcal{Q}^{\nu}_{\text{lab}}$. We stress that pose information is not made available to any of our models. However, we consider a supervised upperbound to measure the maximum performance achievable with our setup.

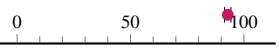
Additionally, data is collected in less unstructured environments, having no external tracking system and, as a consequence, no ground truth poses. In detail, the robots collected 34K samples in a classroom, 48K samples in a gym, and 45k samples in a break room. We combine samples from the three environments to create the Out of Domain (OOD) dataset, where we split data into a training set $\mathcal{T}_{\text{ood}}$ (120K samples), validation set $\mathcal{V}_{\text{ood}}$ (5K samples), and testing set $\mathcal{Q}_{\text{ood}}$ (2K samples).

Baselines. We compare our model against *Mean Predictor*, *CNOS* [6], and *Upperbound* models: *Mean Predictor* always returns the mean relative pose of the robot in the laboratory training set. *Upperbound* is a version of our architecture trained in a fully supervised fashion, i.e., it represents the assumption that pose labels are available for every image and trains with pose labels in the laboratory training set, denoted as $\mathcal{T}^{*}_{\text{lab}}$. *CNOS* is a SoA novel object pose estimation approach based on the CAD model of the object of interest; it segments the input image using SAM [18], matches the segmentations to rendered templates using the features from DINOv2 [19], and returns the known pose of the matched template as its prediction. Specifically, we generate 400 templates by rendering the robot's textured CAD model at 4 distance settings (0.5m, 1m, 2m, 4m) across 100 different orientations. We also consider MegaPose[17] as a baseline; similarly to *CNOS*, it matches the input image with templates of the robot's CAD model, and uses the recovered pose as an initial pose guess refined with a render & compare strategy. However, its quantitative performance is worse than all other baselines considered and takes more than a minute to infer the pose from an image.

Evaluation Metrics. All metrics are computed on $\mathcal{Q}^{\nu}_{\text{lab}}$: for localization, we measure the median distance of the robot's center pixel location, called E_{uv}; for the orientation, we compute the median circular error [42], called E_{ψ}; for the distance, we measure the mean absolute percentage error of the distance –which is not influenced by the distance distribution in the dataset compared to the absolute error–, called E_d, and defined as $|d - \hat{d}|/d$; for the LED classification, we measure the AUC averaged over the LEDs that are visible according to the ground truth robot pose. Similarly to SO-Pose [43], we measure the overall goodness of our approach with the pose accuracy metric

Table 1: Performance metrics computed on the laboratory testing set $\mathcal{Q}^{\nu}_{\text{lab}}$, three replicas per row.

Model	Supervision	E_{uv} [px] ↓	E_{ψ} [deg] ↓	E_d [%] ↓	$\Gamma^{45^\circ}_{1\text{m}}$ [%] ↑	AUC [%] ↑	Point plot for $\Gamma^{45^\circ}_{1\text{m}}$ [%] → Error bars mark 95% CI
Mean Predictor	$\mathcal{T}_{\text{lab}}$	141	86	34	10	50	
Ours	$\mathcal{T}_{\text{lab}}$	17	17	24	70	98	
Ours (OOD)	$\mathcal{T}_{\text{ood}}$	27	55	60	29	89	
CNOS [6]	CAD model	13	72	35	25	N/A	
Upperbound	$\mathcal{T}^{*}_{\text{lab}}$	18	14	11	93	99	



represented as $\Gamma^{45^\circ}_{1\text{m}}$ and defined as the percentage of predictions with a position error of less than 1 meters and orientation error of less than 45° from the ground truth pose. The threshold values are heuristically set such that *Upperbound* has a score greater than 90% in the $\Gamma^{45^\circ}_{1\text{m}}$ metric.

5 Results

We evaluate the pose estimation performance of our approach by training our model on the laboratory training dataset and evaluating it on the testing set collected in the same environment. For evaluating the model, we ignore empty scenes in the testing set. We report the results in Table 1, where we observe that our self-supervised model significantly better than the baselines despite requiring to be trained only on a dataset of images labeled with the binary state of four LEDs, needing no auxiliary segmentation model or rendered CAD templates.

Our model's performance largely surpasses the *Mean Predictor* and closely follows the *Upperbound*. The performance gap with the latter is in the distance estimation; this is explained by the fully-supervised approach regressing the robot's distance as a continuous value, as opposed to our method relying on a discrete number of scales for the prediction, as depicted in Figure 6. In preliminary experiments, we verified that the issue can be mitigated by using more scales during inference. The *CNOS* approach has the lowest E_{uv} error, thanks to the performance of SAM [18] and access to the robot's CAD model. However, it struggles the most in predicting the distance and heading. Upon inspecting the problem, we discovered that DINOv2's embeddings [19] have similar values for a rendered robot template and its horizontal reflection, and for the same pose at different distances.

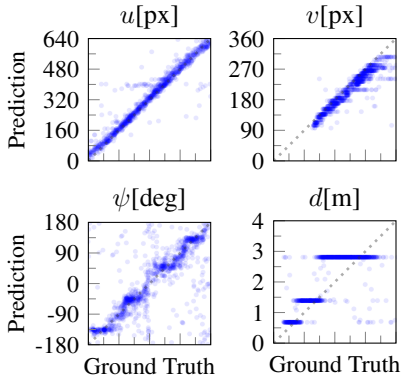


Figure 6: *Our* self-supervised model predictions vs ground truth on $\mathcal{Q}^{\nu}_{\text{lab}}$. Our approach discretizes distances into three bins, resulting in a coarse step function.

LEDs are not necessary at deployment time We test our model on the subset of $\mathcal{Q}^{\nu}_{\text{lab}}$ with images having all visible LEDs turned off, where it scores $E_{uv} = 19.6$px, $E_{\psi} = 19.23^\circ$, and $E_d = 25.7\%$. The small difference in performance compared to testing on the entire $\mathcal{Q}^{\nu}_{\text{lab}}$ demonstrates that the approach is robust and does not require a specific LED state for accurate pose estimation. Higher errors are attributed to the reduced visibility of the robot when LEDs are off.

Fine Tuning to Novel Environments. We consider a model pre-trained using our approach on $\mathcal{T}_{\text{ood}}$ and fine-tuned (using our approach) on an increasing number of samples from the laboratory (from 5K to 115K). The third row of Table 1 reports the performance of the model pre-trained on OOD data and later tested on $\mathcal{Q}^{\nu}_{\text{lab}}$ without fine-tuning; the performance of the model is better than the mean predictor, while it struggles with distance estimation.

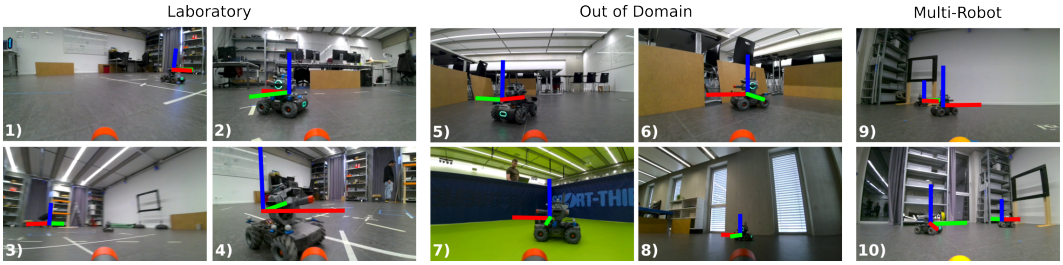


Figure 8: Predicted robot poses (x-axis in red, y-axis in green, z-axis in blue) by models trained with our approach: a model trained on $\mathcal{T}_{\text{lab}}$ applied to $\mathcal{Q}^{\nu}_{\text{lab}}$ (1-4); a model trained on $\mathcal{T}_{\text{ood}}$ and applied to the testing set in classroom (5, 6), gym (7), and break room (8); a model trained on $\mathcal{T}_{\text{lab}}$ applied to images with multiple robots (9, 10). Large errors occur when the images are blurred or robots are far from the camera (3), while smaller errors occur when robot and background blend together (4).

We further fine-tune the model on the target domain, represented by the *Fine-tuning* model, and compare with another trained from scratch on the same data, called *From Scratch*, in Figure 7. We observe that training from scratch with less than 30K fine-tuning samples achieves a worse performance than using a pre-trained model in different environments, highlighting the model's generalization ability. We note that the amount of data needed for the from-scratch model to achieve similar performance to the fine-tuned one depends on the task: localizing the robot is easier than estimating its heading which, in turn, is easier than estimating the robot's distance.

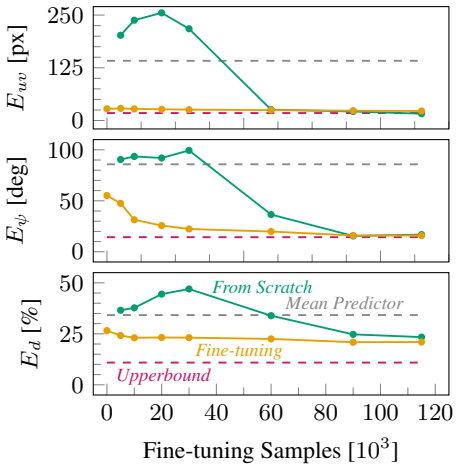


Figure 7: Metrics of a model trained from scratch on $\mathcal{T}_{\text{lab}}$ (in green) and a model pre-trained on $\mathcal{T}_{\text{ood}}$ and fine-tuned on $\mathcal{T}_{\text{lab}}$ (in yellow). The performance of *Mean Predictor* (in gray) and *Upperbound* (in magenta) are reported as dashed lines for comparison.

The Model is Capable of Multi-robot Pose Estimation. We show model predictions on unseen images featuring multiple robots in Figure 8. Despite being trained on images with at most one visible robot, the approach correctly works with images with multiple ones. In this experiment, each robot corresponds to a different local maximum of $\hat{P}$. Details on adapting the model inference for multi-robot pose estimation can be found in the appendix.

6 Conclusions

We presented a self-supervised pose estimation approach that does not require pose labels; instead, supervision is obtained from classifying the state of multiple, independent LEDs on the robot's body. A pair of robots collect images and the ground truth LED state of the peer autonomously, without external hardware, lending the approach to online learning and domain adaptation. Results indicate that our approach trains a competitive pose predictor, whose performance is not degraded by the lack of pose labels despite the complexity of the task.

Our approach only assumes that changes in the target variable (e.g., state of the LEDs) affect the appearance of the robot such that the sensor (e.g., monocular camera) can observe and predict this change of state. As such, the approach can be applied to different sensors, and other actuators (e.g. the position of an arm), as long as it affects the robot's appearance in the sensed data.

7 Limitations

Our model and experiments focus on 2D pose estimation for ground robots. Extension to 3D would be trivial for the position component of the pose; for the rotation component, the LED visibility function would be defined on the space of unit quaternions (representing 3D rotations). In order to disambiguate robot pitch, additional LEDs would be required that face in directions with positive and negative pitches. Reconstructing the roll would require considering the detected locations of each specific LED, which are accessible in the raw LED maps.

The laboratory testing set used in our quantitative experiments, despite being cluttered and full of distractors on shelves, does not contain images where the target robot is occluded by other objects; as such, the efficacy of the approach on partial occlusions needs further testing.

Our current formulation relies on rescaling an input image multiple times (three in our experiments), running a forward pass on each scaled version, and combining the predictions to get the estimated distance. The accuracy of the resulting distance estimate is limited by the coarse discretization of the image scales used to estimate the distance. At inference time, this can be mitigated by considering more scaling factors at the expense of additional computation. In future work, we plan to explore different multi-scale architectures, such as U-Net models [44], to address this limitation and improve the pose estimation performance. Further, our approach considers each training frame individually, disregarding valuable temporal information such as the robot's odometry or the image optical flow. The approach can benefit from incorporating this information in the training process, e.g., by employing an auxiliary consistency loss [45] between pairs of frames.

The training approach assumes to have at most one robot inside the FOV, limiting its application to a pair of robots collecting data. We plan to extend the approach to handle large groups of collaborative robots, dramatically improving data collection efficiency and increasing the frames featuring visible robots (23% in our training set): the amount of useful collected data over a given time interval scales quadratically with the number of robots simultaneously deployed in the environment.

The main limitation of our approach compared to CAD-based ones is the need to retrain from scratch for every new robot whose pose is to be estimated. This drawback is mitigated by the ease of collecting data and the advantage of training directly on real-world images, eliminating the sim-to-real gap.

References

[1] M. Dorigo, G. Theraulaz, and V. Trianni. Swarm robotics: Past, present, and future. *IEEE Point of View*, 109(7):1152–1165, 2021.

[2] G. Wang, F. Manhardt, F. Tombari, and X. Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.

[3] X. Liu, R. Zhang, C. Zhang, G. Wang, J. Tang, Z. Li, and X. Ji. Gdrnpp: A geometry-guided and fully learning-based object pose estimator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[4] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[5] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9: 153171–153187, 2021.

[6] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 2134–2140, 2023.

[7] P. Ausserlechner, D. Haberger, S. Thalhammer, J.-B. Weibel, and M. Vincze. Zs6d: Zero-shot 6d object pose estimation using vision transformers. *IEEE International Conference on Robotics and Automation*, 2023.

[8] S. Moon, H. Son, D. Hur, and S. Kim. Genflow: Generalizable recurrent flow for 6d pose refinement of novel objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10039–10049, 2024.

[9] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024.

[10] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[11] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems*, 2018.

[12] H. X. Pham, A. Sarabakha, M. Odnoshyvkin, and E. Kayacan. Pencilnet: Zero-shot sim-to-real transfer learning for robust gate perception in autonomous drone racing. *IEEE Robotics and Automation Letters*, 7(4):11847–11854, 2022.

[13] S. Li, C. De Wagter, and G. C. De Croon. Self-supervised monocular multi-robot relative localization with efficient deep neural networks. In *IEEE International Conference on Robotics and Automation*, pages 9689–9695, 2022.

[14] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox. Self-supervised 6d object pose estimation for robot manipulation. In *IEEE International Conference on Robotics and Automation*, pages 3665–3671, 2020.

[15] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield. Camera-to-robot pose estimation from a single image. In *IEEE International Conference on Robotics and Automation*, pages 9426–9432, 2020.

[16] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2022.

[17] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *PMLR Conference on Robot Learning*, pages 715–725, 2023.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[19] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[20] J. Lin, L. Liu, D. Lu, and K. Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27906–27916, 2024.

[21] V. N. Nguyen, T. Groueix, M. Salzmann, and V. Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9903–9913, 2024.

[22] E. P. Örnek, Y. Labbé, B. Tekin, L. Ma, C. Keskin, C. Forster, and T. Hodan. Foundpose: Unseen object pose estimation with foundation features. In *European Conference on Computer Vision*, pages 163–182, 2024.

[23] 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 *European Conference on Computer Vision*, 2020.

[24] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[25] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[26] M. Bortolon, T. Tsesmelis, S. James, F. Poiesi, and A. Del Bue. Iffnerf: Initialisation free and fast 6dof pose estimation from a single image and a nerf model. *arXiv preprint arXiv:2403.12682*, 2024.

[27] B. Matteo, T. Tsesmelis, S. James, F. Poiesi, and A. Del Bue. 6dgs: 6d pose estimation from a single image and a 3d gaussian splatting model. In *European Conference on Computer Vision*, pages 420–436, 2024.

[28] D. Zhang, J. Han, G. Cheng, and M.-H. Yang. Weakly supervised object localization and detection: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5866–5885, 2021.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *cvpr*, pages 2921–2929, 2016.

[30] H. G. Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *IEEE/CVF winter conference on applications of computer vision*, pages 983–991, 2020.

[31] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

[32] W. Lu, X. Jia, W. Xie, L. Shen, Y. Zhou, and J. Duan. Geometry constrained weakly supervised object localization. In *European Conference on Computer Vision*, pages 481–496, 2020.

[33] J. Xie, J. Xiang, J. Chen, X. Hou, X. Zhao, and L. Shen. C2am: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–998, 2022.

[34] E. Jang, C. Devin, V. Vanhoucke, and S. Levine. Grasp2vec: Learning object representations from self-supervised grasping. In *PMLR Conference on Robot Learning*, pages 99–112, 2018.

[35] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *PMLR Conference on Robot Learning*, pages 416–426, 2023.

[36] L. Antsfeld and B. Chidlovskii. Self-supervised pretraining and finetuning for monocular depth and visual odometry. In *IEEE International Conference on Robotics and Automation*, pages 14669–14676, 2024.

[37] S. Qian, K. Mo, V. Blukis, D. F. Fouhey, D. Fox, and A. Goyal. 3d-mvp: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024.

[38] M. Nava, N. Carlotti, L. Crupi, D. Palossi, and A. Giusti. Self-supervised learning of visual robot localization using led state prediction as a pretext task. *IEEE Robotics and Automation Letters*, 9(4):3363–3370, 2024.

[39] N. Carlotti, M. Nava, and A. Giusti. Learning to estimate the pose of a peer robot in a camera image by predicting the states of its leds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2763–2769, 2024.

[40] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[41] M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng, and S.-M. Hu. Attention mechanisms in computer vision: A survey. *Springer Computational visual media*, 8(3):331–368, 2022.

[42] K. V. Mardia and P. E. Jupp. *Directional statistics*. John Wiley & Sons, 2009.

[43] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari. So-pose: Exploiting self-occlusion for direct 6d pose estimation. In *IEEE/CVF International Conference on Computer Vision*, pages 12396–12405, 2021.

[44] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Springer Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[45] M. Nava, L. M. Gambardella, and A. Giusti. State-consistency loss for learning spatial perception tasks from partial labels. *IEEE Robotics and Automation Letters*, 6(2):1112–1119, 2021.

[46] D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[47] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Appendix

A Neural Network Training

Throughout this document, we adopt the same FCN architecture, receiving as input an image with an original resolution of 640×360 pixels and producing maps of 80×45 cells, with a RF of 70×70 pixels. In detail, we designed a lightweight FCN with 6 blocks interleaved by 2x max-pooling and totaling 179K parameters; each block consists of a 2D convolution, batch normalization, and ReLU non-linearity. Our approach uses the scaling factors $(1, \frac{1}{2}, \frac{1}{4})$ to rescale the original input image with average pooling and, for each one, does a forward pass to produce the output maps. After the three forward passes, we upscale the smaller maps to match the 80×45 size of the largest map (which corresponds to the largest scale) using bilinear interpolation. Given these maps, we optimize the loss function defined in Equation (1) using Adam [46] with an initial learning rate $\eta_{\text{initial}} = 1e^{-3}$ that smoothly decreases to $\eta_{\text{final}} = 1e^{-4}$ with cosine interpolation [47] over 100 training epochs. During training, we apply image augmentation using multiplicative simplex noise and color jittering. The best set of parameters $\boldsymbol{\theta}$ for the model is chosen as the one leading to the smallest validation loss, usually occurring within the first 60 epochs of training.

B Model Inference

The target robot location in the image $\hat{u}$, $\hat{v}$ is computed from $\hat{P}_{\text{ms-norm}}$: at first, we sum the map cells over the scales S to have a two-dimensional map whose cells indicate the presence of robots across all scales; secondly, we compute the barycenter of the map by multiplying the values of each cell by its integer coordinates and summing all cells to obtain $\hat{u}'$, $\hat{v}'$; finally, we localize the robot by multiplying $\hat{u}'$, $\hat{v}'$ by the integer factor relating the output maps' resolution to the original input image resolution. Formally, the procedure can be written as

$$\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} W/W' \\ H/H' \end{pmatrix} \sum_{s=1}^{S} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (\hat{P}^s_{\text{ms-norm}} \odot M)[i,j] \tag{2}$$

where $\odot$ is the element-wise product, and M is the coordinate matrix consisting of cells $m_{ij} = (i\ j)^T$.

The target robot orientation $\hat{\psi}$ is computed as the weighted average of $\hat{\Psi}^s$ by $\hat{P}^s_{\text{ms-norm}}$, defined as

$$\hat{\psi} = \sum_{s=1}^{S} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (\hat{P}^s_{\text{ms-norm}} \odot \hat{\Psi}^s)[i,j] \tag{3}$$

Finally, the target robot distance is computed from $\hat{P}^s_{\text{ms-norm}}$ as follows:

$$\hat{d} = d_c \cdot \hat{d}'\,; \qquad \hat{d}' = \sum_{s=1}^{S} f_s \cdot \sum_{i=1}^{H'} \sum_{j=1}^{W'} (\hat{P}^s_{\text{ms-norm}})[i,j] \tag{4}$$

In this formula, we first sum the map cells over their width and height to have a one-dimensional vector whose elements indicate the degree of compatibility between the robot's image size and the model's RF at different scales – the higher the value, the better the robot size matches the RF. Recalling that the size of the robot's bounding box in the image is inversely proportional to its distance, we recover the distance $\hat{d}'$ as the average of the inverse of scale factors weighted by the vector defined above. To get a metric prediction from $\hat{d}'$, we employ a simple calibration procedure: the robot is placed in front of the camera, and images of it are taken while adjusting its distance; we pick d_c as the distance at which the robot appears with a size of $r \times r$ pixels (i.e., our model's RF, as introduced in Section 3) in the captured images.

The target robot LED states $\hat{l}^k$ are computed as the average of $\hat{L}^{ks}$ weighted by $\hat{P}^s_{\text{ms-norm}}$, defined as

$$\hat{l}^k = \sum_{s=1}^{S} \sum_{i=1}^{H'} \sum_{j=1}^{W'} (\hat{P}^s_{\text{ms-norm}} \circ \hat{L}^{ks})[i,j] \tag{5}$$

C Multi-robot Inference

To allow the model to predict the pose of multiple robots, we modified the way the $\hat{P}$ map is obtained. As described in Section 3, the cells in $\hat{P}$ are bound to the $[0, 1]$ value range. This is achieved by applying a softmax function to the raw activation values of the map. When multiple robots are present in the input image, the raw map contains multiple peak values at different spatial locations. Because the softmax function suppresses all non-maxima peaks when the absolute difference between peaks is high enough, the post-softmax map generally presents only one peak. Hence, our solution is first to linearly rescale the values into the $[0, 1]$ and then to apply the softmax function to further suppress noise.

D Detecting Robot Presence

Taking inspiration from Weakly Supervised Learning, we explore how to use our learned model for robot detection. We consider the problem of classifying whether a robot is visible anywhere in an image; we take the maximum of $\hat{P}_{\text{ms-norm}}$ as the belief about the presence of a robot. When testing this binary classification approach over $\mathcal{Q}_{\text{lab}}$, we obtain an AUC of 83.4%. Additionally, we consider a method that estimates robot presence based on the model's confidence in its LED predictions based the entropy formula:

$$\frac{1}{4}\sum_{k=1}^{4} 1 - \left(-\hat{l}^k \cdot \log(\hat{l}^k) - (1-\hat{l}^k) \cdot \log(1-\hat{l}^k)\right) \tag{6}$$

The resulting scalar measures the model's confidence in its LED state prediction; the closer the values are to zero (off) or one (on), the higher the confidence. Assuming that the model predicts the LED state with high confidence only when the robot is visible in the image, we use this value to detect the robot's presence. Using this alternative method, we report a robot detection AUC of 97.2%. We believe the difference between the two methods to be caused by the softmax operator applied to $\hat{P}_{\text{ms-norm}}$: when no robot is visible, the softmax drastically accentuates the noise in the localization map, leading to many false positives. In this situation, our model produces LED maps with values close to 0.5, resulting in low confidence and, thus, stronger robot detection.

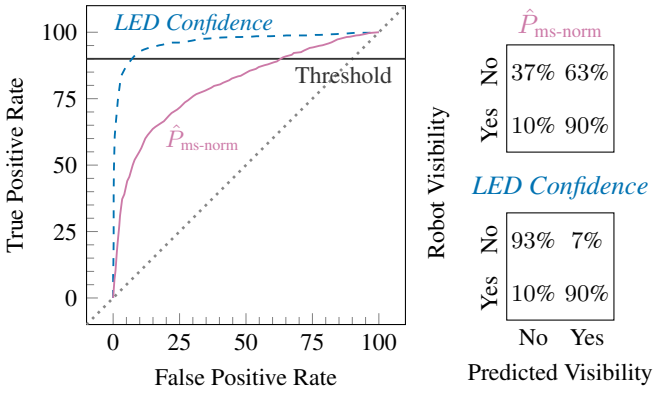


Figure 9: Receiver Operator Characteristic (ROC) curves for the robot detection methods presented in Section D on the laboratory testing set.