Semantic Disagreement for Embodied Active Perception

Gianluca Scarpellini[1,2], Stefano Rosa[1], Pietro Morerio[1], Lorenzo Natale[1], Alessio Del Bue[1]
[1]Italian Institute of Technology, Genoa, Italy
[2]University of Genoa, Genoa, Italy
{name.surname}@iit.it

***Abstract*—Object detection for robotics is challenging, as detectors often fail to generalize to novel scenes or points of view. Previous works attempted to address this issue through self-training and hinted that curiosity-driven exploration can help to collect useful samples for fine-tuning. This work proposal is two-fold: (i) *Look Around*, a novel method to explore the environment and mine useful hard samples for fine-tuning the object detector, and (ii) *Disagreement Reconciliation*, a novel fine-tuning strategy for improving the object detector without relying on any human annotations. We train a reinforcement learning agent in simulation, and we reward it based on the uncertainty on the predicted classes of the same object instances across different views. Next, we deploy the policy on a real humanoid robot to explore an unseen scenario and fine-tune the detector. Experiments show that our approach outperforms state-of-the-art baselines on simulated environments and can transfer to the real world. Our findings demonstrate that proper data collection alone is sufficient to bridge the gap with fully supervised settings, eliminating the need for ground-truth annotations.**

I. Introduction

Inspired by the interplay of action and perception in human cognition and recent advances in active learning [30, 4, 5], we propose that the self-supervised fine-tuning of object detectors can significantly benefit from active exploration of the environment (Figure 1). In our setting, the agent explores an unknown environment using an off-the-shelf object detector. By actively seeking diverse views of the same object instances, the agent can improve its object detector without the need for external supervision. Our approach consists of an *action-perception* pipeline that includes learning both the *action* (referred to as *Look Around*) and *perception* (referred to as *Disagreement Reconciliation*) stages in a fully self-supervised manner.

Look Around (action stage): We train a RL policy for collecting examples for the object-detector. As the agent explores, it progressively builds a 3D semantic voxel map containing *all* gathered predictions, which are inherently noisy due to imperfections in the detector and variations in the new scenario's images. Notably, the same object instance may be predicted as belonging to different classes when seen from different viewpoints, or not detected at all in some images. We hypothesize that pseudo-labels for objects are more valuable if they are predicted as belonging to different classes across different viewpoints. Building upon this intuition, we propose measuring detector disagreement as the entropy of its predictions and rewarding the policy to maximize this disagreement. This novel approach extends existing literature on environment exploration, which primarily employs reinforcement learning to maximize the explored area [3], novelty of observations [20], or success in reaching predefined goals [31, 6].

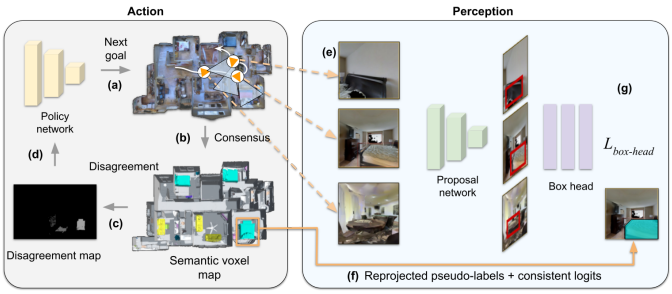

Fig. 1: *Action:* **(a)** Our policy predicts a long-term goal for the agent. **(b)** The agent moves toward the predicted goal and collects observations. At each step, we project the object detections into 3D space and build a semantically consistent voxel map. **(c)** We build a *disagreement map* by projecting the voxel map onto a top-down view and computing the disagreement score for every voxel of the map. **(d)** The disagreement map is the input of our policy network. During training, we reward the policy to maximize disagreement. *Perception*: **(e)** we deploy our policy in an environment to collect a dataset. **(f)** we build the pseudo-labels for the self-training scheme. **(g)** Finally, we fine-tune the object detector by relying only on the pseudo-labels.

Disagreement Reconciliation (perception stage): The predictions obtained during exploration serve as *pseudo-labels* to fine-tune the detector in a three-fold manner. First, we enforce the consistency of predictions across multiple views using a consensus mechanism, and the refined pseudo-labels are assigned to corresponding images. This consensus acts as feedback to fine-tune the detector. Second, we encourage the model backbone to map different views of the same object close in the feature space while pushing views of different instances apart, following a contrastive learning approach. Third, we leverage the soft targets obtained by averaging the multiple predictions acquired for each instance as additional feedback, fully exploiting the multiple acquired views. While self-supervised learning with multiple "views" of the same image has gained popularity, these views are typically obtained through limited augmentations ([12, 9, 7]), unable to fully capture the complexity of a 3D object seen from different viewpoints.

Our experiments demonstrate that actively seeking challenging examples for the detector leads to more informative pseudo-labels and results in more accurate object detectors in simulation and on a real humanoid robot.

II. Approach

We equip a simulated agent with an RGB-D sensor, a position sensor, and an object detector. The agent explores a set of simulated environments [27] in the Habitat simulator [25] and collects samples without the use of ground-truth annotations. As in previous literature [4, 5], we adopt the object-detector MaskRCNN [11] with a Resnet50 backbone [10] as implemented in [26] and pre-trained on the COCO dataset [15]. To guide the agent's movements, we use a policy that sets long-term goals and allows the agent to select actions from a set of options: moving forward, turning left, and turning right. During the exploration, the agent builds a *disagreement map* that reflects the uncertainty of the object-detector in different parts of environment. We reward the agent to maximize the total disagreement in the map. It follows that the agent learns to collect hard samples for the object-detector, e.g., by seeing objects from unusual points of views. Intuitively, hard samples are the most beneficial for fine-tuning the detector. Next, we aim to improve the agent's object detection performance by fine-tuning its detector using only the samples collected autonomously during exploration. In the following section, we describe our methodology in more detail.

A. Action: Look Around

Disagreement map. At each time step t, we update a *disagreement map* H_t as follows. The semantic masks of objects detected by MaskRCNN are first projected into a point cloud in the global frame of reference using the agent's RGB-D and position sensors. The point clouds are aggregated into a *semantic voxel-map* by voxelization of the 3D points. Each voxel is assigned the class with the highest probability among the ones predicted for the points contributing to the voxel. We accumulate the novel predicted logits for each voxel to the previous ones. We use the assigned classes to extract a set of unique 3D object instances U as clusters of connected voxels. Then, for each unique 3D object instance $u \in U$, a *disagreement score* $s_{\text{disag}}(u)$ is computed using a disagreement measure. Finally, the disagreement scores are down-projected into a 2D disagreement map by projecting them into a top-down view H_t. Each map cell (i, j) contains the sum of the disagreement scores of all the objects that project to position (i, j).

Disagreement scores. We propose to measure the disagreement of the object detector as viable information for the policy. As we gather the object detections into a semantic voxel map, we preserve all the object detector's predictions–normalized logits and class predictions–for each unique 3D instance object u. Next, we compute the disagreement score as $s_{\text{disag}}(u) = \mathbb{E}[-\log p(u)]$, where $p(u)$ is the average normalized logits for the unique instance object u. This formulation is inspired by the literature on epistemic uncertainty estimation [8, 1].

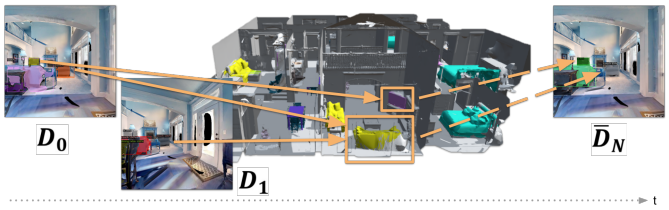

Fig. 2: Semantic voxel map creation and projection of detections onto 2D frames. First, we aggregate detections $D_0, \ldots, D_N$ into semantic voxel-map. We solve the inconsistency of the voxel-map by assigning to each voxel the class with maximum score among the predictions of the voxel. Next, we project the semantic voxel-map back onto each observation, obtaining the consistent pseudo-label for observation. Each pseudo-label $\overline{D}_N$ is associated to an object instance and contains a consistent logits vector, bounding boxes, and class.

Reward We take advantage of the disagreement map and define a dense reward as the accumulated disagreement over the whole environment as $r_t = \sum_{i,j} H_t[i, j]$, where $H_t[i, j]$ is the value of the (i, j) cell of the disagreement map at t.

B. Perception: Disagreement Reconciliation

During the exploration phase, we utilize the agent's depth and positional sensor data to generate a semantic point cloud by projecting the agent's detector predictions (comprising bounding boxes, instance segmentation masks, class labels, and logits) into 3D coordinates. This point cloud is subsequently voxelized, resulting in a voxel map where each non-empty voxel contains a set of logits vectors. However, this initial map may exhibit inconsistencies, with voxels potentially associated with multiple classes. To resolve this, we introduce a process to compute a consistent hard-label for each voxel, selecting the class with the highest score among all predictions linked to that voxel.

Further, we aggregate connected voxels sharing the same class using a 26-connect-components algorithm [24], effectively grouping them as distinct object instances, each assigned a unique identifier and sharing a common class. This step ensures a consistent voxel map, where each object instance has an unambiguous class label. We further compute a consistent logits vector for each instance by averaging the logits vectors assigned to that instance. We adopt this vector as a soft target for fine-tuning the detector.

Any 3D object instance present in the semantic map is projected onto each observation using intrinsic and extrinsic camera matrices, resulting in per-instance semantic masks overlaid on each observation (as in Figure 2). This projection also includes the object classes, consistent logits, and instance identifiers. For each projection, we compute the minimum bounding box that contains the projected masks. Importantly, this projection phase resolves both of the agent's prediction issues: (i) it ensures that pseudo-labels for an object instance remain consistent across different views, and (ii) it provides

(a) (b) (c) (d)

Fig. 3: **(a)** The agent observes the environment and detects objects with a RGB-D camera and **(b)** builds a map of the explored area. Detections are accumulated into a 3D semantic map, that is down-projected to 2D to build the disagreement map **(c)**. By exploring the environment, the agent maximizes the disagreement in the map **(d)**.

consistency even for frames initially lacking any detections. Our training dataset, denoted as $\mathcal{D}$, consists of the resulting observations and pseudo-labels, encompassing images, object instance identifiers, consistent logits, masks, bounding boxes, and class labels.

Fine-tuning We compute losses for masks, bounding-boxes, and classes on the consistent pseudo-labels obtained in the previous step. To fully exploit the information contained in our semantic voxel map, we adopt two additional losses in the self-training strategy: instance matching and knowledge distillation.

Instance matching: We adopt a triplet-loss that encourages feature vectors belonging to the same object to be close in a feature space while pushing feature vectors of different objects apart. This aids the model's feature extractor, particularly for objects viewed from diverse and challenging perspectives.

Soft-distillation loss: To further enhance the model's performance, we introduce a knowledge distillation loss that leverages the consistent logits as soft targets. This loss guides the prediction towards a smooth average of the logits obtained from different views, improving the model's robustness.

III. Experimental setup

For each RL policy, we let an agent move in the environment and collect 7,500 frames from the Gibson dataset [27]. Next, we adopt this dataset for fine-tuning the object-detector with different perception methods. We opt for MaskRCNN [11] as the object detector as in previous approaches, although our proposed approach is model-agnostic. We adopted PyTorch [19] for the deep learning components of our proposal, and Habitat [21] for simulating the agent's exploration. We adopt the same environments as in [5, 4] for training and testing. We evaluate the object-detector on 6 commonly used classes: couch, plant, bed, toilet, tv, and table. We compare our method with Random goals, frontier exploration [29], NeuralSLAM [3], SEAL [5], Semantic Curiosity [4] and [13].

Training details We train the policy with PPO [23]. We adopt Adam optimizer [14] with learning rate $2.5e-5$, a discount factor $\gamma = 0.99$, an entropy coefficient 0.001, value loss coefficient 0.5, and replanning steps $N_{\text{replanning}} = 20$, as proposed in previous works [5, 4]. Episode length is fixed to $T = 300$ steps for comparison with [5]. We trained policies for $250,000$ steps, which took 72 hours of training on 2 V100 GPUs. For fine-tuning the object detector, we use SGD with learning rate $\text{lr} = 1e-4$, epochs $= 10$, weight-decay $= 1e{-}5$, momentum $= 0.9$ and batch size $= 16$. For *Disagreement Reconciliation*, we adopt the default parameters: soft-distillation weight $\alpha = 0.7$ for soft-distillation loss $\mathcal{L}_{\text{distill}}$ and margin $= 0.3$ for the triplet-loss [22].

Evaluation We test our approach on 4,000 samples from 4 unseen scenes. We adopt the mean Average Precision with IOU threshold on bounding boxes at 0.5 [15] to evaluate the object detector performance. Section IV compares our policy across different self-training approaches with the baselines and supervised labels. For the latter, we use the annotations provided by Habitat and originally proposed in Armani et al. [2].

Action baselines In this section, we present a comprehensive comparison of our *action module* with a diverse set of classic and learned policies designed for exploration tasks. The following baselines are considered:

Random Goals: The agent employs a simple strategy where it randomly samples goals in freespace following a uniform distribution and follows a global path planner to navigate to the selected point.

Frontier Exploration [29]: Based on the classical frontier-based active exploration method [29], this baseline maintains an internal representation of the explored map. It identifies goal points of interest at the frontiers of the explored map, selects the next goal greedily, and moves towards it using a path planner.

NeuralSLAM [3]: In this approach, the agent predicts both long-term and short-term goals to maximize map coverage. A global policy predicts the next long-term goal, while a local policy predicts the sequence of short-term steps to reach the global goal. The policy's reward is determined by the percentage of exploration achieved in the environment.

Semantic Curiosity [4]: The agent predicts local steps to collect a dataset for fine-tuning the object detector. The reward is based on the number of inconsistent predictions for the same object. A 2D map of these inconsistencies is projected onto the ground plane and fed to the policy.

Informative Trajectories [13]: This RL policy is trained to maximize the KL divergence between current observations and a 3D semantic voxel map of accumulated predictions. Additionally, it looks for objects for which the detector exhibits high uncertainty between two classes.

SEAL [5]: In this approach, the agent predicts the next long-term goal and follows a local path planner to reach it. SEAL builds a voxel map by projecting the detections into 3D, and this voxel map is fed to the policy to predict the next long-term goal. The reward is based on the number of voxels assigned to any possible class with a score above 0.9.

Perception baselines We compare our action policies with two different perception modules:

Self-training [28] – it uses the off-the-shelf object detector to produce pseudo-labels for training itself without any further processing. It should be noted that self-training often requires millions of images to provide substantial benefits.

SEAL perception [5]– SEAL adopts the off-the-shelf object-detector to provide noisy detections and aggregate them into

TABLE I: We compare mAP of our action and perception phases with the baselines. Our action-perception loop outperforms the baselines with a mAP@50 of 46.60. As a reference, off-the-shelf MaskRCNN reaches mAP@50 40.33.

Policy	Self-training [28]	SEAL perc. [5]	**Disag. reconc.**	*Ground truth*
Random goals	39.67	41.19	41.88	*47.20*
Frontier [29]	40.18	41.98	43.09	*45.06*
NeuralSLAM [3]	39.98	39.56	40.32	*44.86*
Sem. Curiosity [4]	40.23	41.06	41.37	44.67
SEAL policy [5]	39.33	43.01	42.38	*44.57*
Inform. Traj.[13]	**40.25**	44.15	43.70	45.49
Look Around	38.66	**45.90**	**46.60**	***48.01***

a semantic voxel map. Each voxel is assigned the class with the highest score. The semantic voxel map is projected onto each example to produce consistent pseudo-labels. The new dataset is adopted for fine-tuning.

IV. Quantitative results in simulation

We evaluate our policy against different baselines and for different perception methods. Table I reports our results, while Figure 3 shows an exploration example. As a reference, the off-the-shelf MaskRCNN [11] reaches mAP@50 40.33 on the test set. Our policy explicitly maximizes disagreement and therefore is well suited for *Disagreement Reconciliation* and SEAL. Among the baselines, SEAL policy and perception module [5] achieves mAP 43.01%. On the other hand, our policy with SEAL's perception achieves mAP 45.90%. We generally notice higher results when adopting *Disagreement Reconciliation* as the perception module. Among the baseline policies, the classical frontier exploration achieves the highest mAP 43.09% in combination with it. When *Disagreement Reconciliation* is combined with our policy, we achieve the best mAP of 46.60%. These results prove the soundness of our policy design. Unsurprisingly, all policies fail to improve the object detector when fine-tuning with self-training [28]. In particular, our policy collects data that maximize the inconsistency of the detections provided by the object detector. If we use those detections directly as pseudo-labels–without solving the disagreement, the detector receives contrasting training examples, and its performance degrades. In the ideal scenario where *ground-truth* is available, *Look Around* outperforms the other actions. Indeed, these results further prove that our policy collects a dataset of useful examples for fine-tuning.

V. Quantitative results in the real world

We finally experiment with the zero-shot transfer of our approach on a real robotic platform. We deploy a policy that was trained only in the simulator on a humanoid wheeled robot and collect samples by exploring two different environments. The robot is the wheeled humanoid R1 [18], equipped with a Realsense D455 RGB-D camera mounted on its head, at an height of 1.2m. The robot is controlled via YARP [17] and the path planner used to follow goals and avoid obstacles is part of the ROS2 library [16]. We compare *Look Around* with two baselines: random goals and human guidance. In the latter, an operator manually sends a series of goals to the robot, with the aim of seeing as many objects as possible, from many different views. In this setting, we fine-tune pre-trained MaskRCNN with *Disagreement Reconciliation*, self-training [28], and SEAL perception [5]. We test the object-detector on a set of 20 annotated images, of which 50% in set and 50% out of set, from the two environments.

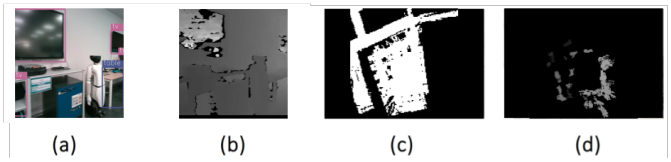

Fig. 4: **(a)** RGB-D camera with detections superimposed; **(b)** depth image; **(c)** map of the environment; **(d)** disagreement.

TABLE II: We deploy our policy on a humanoid robot and fine-tune MaskRCNN with our action-perception approach. As reference, off-the-shelf detector reaches mAP@50 55.83.

Policy	Self-training	SEAL perception	**Disag. Reconc.**
Random goals	48.82	54.49	56.65
Human guidance	46.06	51.50	56.00
Look Around	**53.47**	**54.78**	**58.41**

Results are shown in Table II and Figure 4. As reference, off-the-shelf MaskRCNN achieves mAP@50 55.83. *Look Around* with *Disagreement Reconciliation* achieves an improvement of 2.6% in mAP. We think that this encouraging transfer ability is due to the policy being fairly agnostic to the input domain as well as the dynamics of the agent. In particular, the policy inputs sit at an higher level of abstraction compared to the agent's camera, being top-down grid maps; the policy outputs are also abstracted from the dynamics of the robot, being goals in 2D space. On the other hand, manually collecting examples (human guidance) leads to worse results after fine-tuning. Humans struggle to solve tasks that are not clearly defined, e.g., collecting useful hard samples for fine-tuning. *Disagreement Reconciliation* greatly outperforms the other perception methods, demonstrating its ability to improve the object-detector without relying on any human annotations.

VI. Conclusion

In this paper, we presented an exploration policy, refereed to as *Look Around*, and a fine-tuning approach, referred to as *Disagreement Reconciliation*, to improve object detection through active data collection. We evaluated our action-perception loop extensively in simulation and in the real world.

Overall, our experiments demonstrate the effectiveness of *Look Around* in improving object detection through active data collection. Our approach's ability to outperform existing methods in both simulated and real-world scenarios suggests its potential for practical applications in robotics and computer vision. Future work will focus on further refining *Look Around* for real-world data and extending this approach for broader use cases.

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5664–5673, 2019.

[3] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklXn1BKDH.

[4] Devendra Singh Chaplot, Helen Jiang, Saurabh Gupta, and Abhinav Gupta. Semantic curiosity for active visual learning. In *European Conference on Computer Vision*, pages 309–326. Springer, 2020.

[5] Devendra Singh Chaplot, Murtaza Dalal, Saurabh Gupta, Jitendra Malik, and Ruslan Salakhutdinov. SEAL: Self-supervised embodied active learning using exploration and 3d consistency. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=guHXB1dcD3l.

[6] Annie S Chen, HyunJi Nam, Suraj Nair, and Chelsea Finn. Batch exploration with examples for scalable robotic reinforcement learning. *IEEE Robotics and Automation Letters*, 6(3):4401–4408, 2021.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[13] Ya Jing and Tao Kong. Learning to explore informative trajectories and samples for embodied perception. *arXiv preprint arXiv:2303.10936*, 2023.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[16] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66):eabm6074, 2022. doi: 10.1126/scirobotics.abm6074. URL https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[17] Metta, Giorgio et al. Yarp - yet another robot platform. URL www.yarp.it.

[18] Alberto Parmiggiani, Luca Fiorio, Alessandro Scalzo, Anand Vazhapilli Sureshbabu, Marco Randazzo, Marco Maggiali, Ugo Pattacini, Hagen Lehmann, Vadim Tikhanoff, Daniele Domenichelli, et al. The design and validation of the r1 personal humanoid. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 674–680. IEEE, 2017.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[20] Santhosh K. Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. An exploration of embodied visual exploration, 2020.

[21] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, page nil, 10 2019. doi: 10.1109/iccv.2019.00943. URL https://doi.org/10.1109/iccv.2019.00943.

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. doi: 10.1109/cvpr.2015.7298682. URL http://dx.doi.org/10.1109/CVPR.2015.7298682.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization

algorithms, 2017.

[24] Will Silver Smith. connected-components-3d. *GitHub. Note: https://pypi.org/project/connected-components-3d/*, 2022.

[25] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat, 2021.

[26] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[27] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[28] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[29] Brian Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53, 1998.

[30] Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2040–2050, 2019.

[31] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.