VANP: Self-Supervised Vision-Action Pretraining for Navigation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Self-supervised learning has revolutionized the fields of computer vision and natural language processing. Despite its potential, its application to robotic navigation tasks remains under-explored. This is due to the difficulty of defining effective self-supervision signals for robotics. Fortunately, with the recent development of many large-scale robotic navigation datasets with a variety of sensor and action data that can be used as self-supervision signals, self-supervised learning has become a viable approach for robotic navigation tasks. In this work, we propose a self-supervised method for learning visual features for end-to-end robotic navigation systems, using actions as the supervisory signal. This approach is motivated by the observation that humans tend to focus on specific regions of their frontal view in order to make navigation decisions and produce navigation actions. We reverse this procedure, using future actions to learn only the visual features that are important for navigation, as opposed to extracted features by conventional computer vision models that tend to extract every detail of the environment that can be misleading to a downstream navigation controller. Our results show that this approach enables small convolutional neural network-based visual encoders to achieve performance comparable to large vision foundation models trained on billions of images. This demonstrates the scalability and effectiveness of our self-supervised learning method for robotic navigation.

Keywords: Self-supervised, Navigation, Learning

1 Introduction

Recent advances in computer vision and deep learning rely on the development of increasingly large and complex Deep Neural Networks (DNNs) [1, 2, 3, 4]. However, training these DNNs from scratch can be computationally expensive and requires a large amount of computing resources [5, 6, 7, 8, 9]. Self-Supervised Learning (SSL) [10, 11, 12, 13] is a machine learning paradigm that can mitigate the need for annotated data by enabling DNNs to first pre-train the model from unlabeled data and then quickly fine-tune to adapt to specific tasks, avoiding the need to re-train everything from scratch, i.e., SSL trains DNNs to complete a pretext task that does not require labels. For example, DNNs might be trained to predict the rotation of an image [14] or to reconstruct an image from its corrupted/obstructed version [15]. By completing these pretext tasks, DNNs learn to extract meaningful features from the data, which can then be used to solve downstream tasks such as image classification and object detection [16].

Despite the effectiveness of SSL in a variety of computer vision tasks, challenges still remain that need to be addressed before SSL can be widely adopted in robotics applications. For example, SSL models typically require a large amount of data to train, which can be difficult to obtain in robotics settings. Additionally, SSL models trained on computer vision datasets such as ImageNet ILSVRC-2012 [17] and COCO [18] may not generalize well to robotic navigation tasks, which contain a significant amount of dynamic scenes of moving agents that can affect a robot's trajectory.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

Figure 1: An illustrative example of the difference between attentive regions from computer vision models (left) and navigation models (right). With the assumption that everyone respects social etiquette, some regions become redundant when making navigational decisions.

As shown in Figure 1 left, complex DNN models trained on vision tasks may extract all the existing features in the scene that may confuse downstream navigation controllers based on Neural Networks (NNs) by providing too much information and leading to improper actions. Although these features may be useful for other downstream tasks, e.g., computer vision or mobile manipulation [19], such complex features may not be necessary for navigation tasks, e.g., when navigating human-occupied spaces where everyone is respecting social etiquette [20]. Another argument for this limitation of conventional computer vision models in real-world navigation is that humans only pay attention to what is right in front of them to make decisions when navigating an environment. This efficiently limits the observation region, as shown in Figure 1 right, which illustrates the difference between features extracted by conventional computer vision models and those necessary to enable navigation tasks.

Considering both the success of SSL on a variety of computer vision tasks and the oftentimes redundant and confusing features provided by generic SSL models for navigation tasks, we present a Vision-Action Navigation Pretraining (VANP) approach that completely relies on a pretext task to train the visual encoder. The key observation behind VANP is when humans navigate crowded spaces, we do not need to pay attention to all the people and objects in the scene, but only the ones that affect our navigation trajectory *i.e.* observations that cause our actions. In this work, we reverse this causality by learning only relevant visual features with the help of future actions. To this end, we leverage Barlow Twins' redundancy-reduction principle [11] to train a visual encoder that discards redundant features of an image for navigation using an action latent space (see Figure 2). VANP focuses on extracting informative features from images that are aligned with the actions of a navigating robot. Our experimental results suggest that VANP-extracted features are more informative for a downstream controller. We show that even a simple Convolutional Neural Network (CNN) with 8 million parameters trained on only one million images can be as expressive for visual navigation as a vision foundation model with 21 million parameters pre-trained on 142 million curated images out of 1.2 billion source images.

The contributions of this work can be summarized as follows:

- We propose an SSL framework to train a visual encoder for robotic navigation tasks.
- We provide a concrete pre-trained SSL model for deployment in an end-to-end navigation pipeline in social environments.

2 Related Work

Our work is motivated by several recent advances in Natural Language Processing (NLP) and computer vision, driven by the Self-Supervised Learning (SSL) paradigm. In this section, we categorize this pretraining paradigm into two groups for robotics and review related works pertaining to each.

Pretraining for better representation: General purpose models, also known as foundation models, pre-trained on pretext tasks can contribute to learning a rich representation that can help the model generalize to different downstream tasks in a zero/few-shot manner [16]. Foundation models for robot manipulation have been extensively studied in the literature [21, 22, 23, 24, 25, 26, 27]. For example, R3M [28] trained a general visual encoder for manipulation tasks on the Ego4D dataset [29], while CLIPort [30] leveraged the CLIP model [31] to enable language instructions for manipulation. Dadashi et al. [21] proposed AQuaDem, a framework to learn quantized actions from demonstrations in continuous action spaces, while VANP is doing the opposite by learning visual features from continuous action spaces. Luo et al. [22] improved AQuaDem by using VQ-VAE [32] for offline reinforcement learning. Huang et al. [23] proposed Skill Transformer to learn long-horizon robotic tasks with the help of transformers [33].

In autonomous driving Nazeri and Bohlouli [34] proposed two parallel networks, one encodes features from the past, and the other encodes plausible features from the future to expand the observation window so the model can make well-informed decisions. Codevilla et al. [35] showed that a deeper model can play an important role in training better policies. It is apparent that most of the works in AVs use pre-trained computer vision models that are trained on ImageNet [36, 35, 37, 38, 34, 39, 40, 41, 42]. However, VANP shows that a visual encoder specific to navigation tasks can help in learning better policies compared to pretraining with ImageNet.

Pretraining for better policies: Foundation models can not only help in learning a rich representation but also be used as a policy to generalize to multiple robotic tasks. SayCan [43] used Large Language Models (LLMs) to learn robotics skills by grounding LLMs and value functions in the physical world. Li et al. [44] and Reid et al. [45] used pre-trained LLMs as the policy backbone. VPT [46] pseudo-labeled Minecraft YouTube videos to learn a behavior cloning policy that can craft diamonds. VPT learns the inverse dynamics while VANP uses dynamics to learn visual features. GNM [47] learned a general policy to drive any robot by combining multiple datasets of different robot types. ViNT [48] further improved GNM by replacing the policy network with a transformer [33]. To the best of our knowledge, no prior work has used actions as the pretext training signal to learn visual features for visual navigation.

Large-Scale Datasets: Large-scale datasets are the primary driver of recent advances in SSL. Data collection in computer vision is relatively straightforward compared to interaction-rich robotics navigation. One reason for this is that collecting a large-scale navigation dataset through human teleoperation is expensive. Additionally, collecting interaction-rich datasets can be potentially dangerous due to the risk of collisions between humans and robots. Despite these challenges, the robotics community has made tremendous efforts to collect interaction-rich datasets in the real world in recent years [49, 50, 51, 52]. SCAND [49] was one of the first efforts to collect navigation data in social environments at large by using teleoperated Spot and Jackal robots. MuSoHu [50] is another effort to collect 20 hours of human interactions in crowded spaces by using a human wearing a helmet equipped with different sensors. SANPO [52] also used humans to collect both real and synthetic datasets of nearly 15 hours of annotated videos for both vision tasks and robotics navigation. In this work, we combine both SCAND and MuSoHu of both robot and human navigation demonstrations respectively to create a dataset of nearly 1 million visual navigation samples with real-world human-robot interactions.

3 Methodology

This section formally defines the end-to-end visual navigation task and describes the Vision-Action Navigation Pretraining (VANP) procedure for the visual encoder.

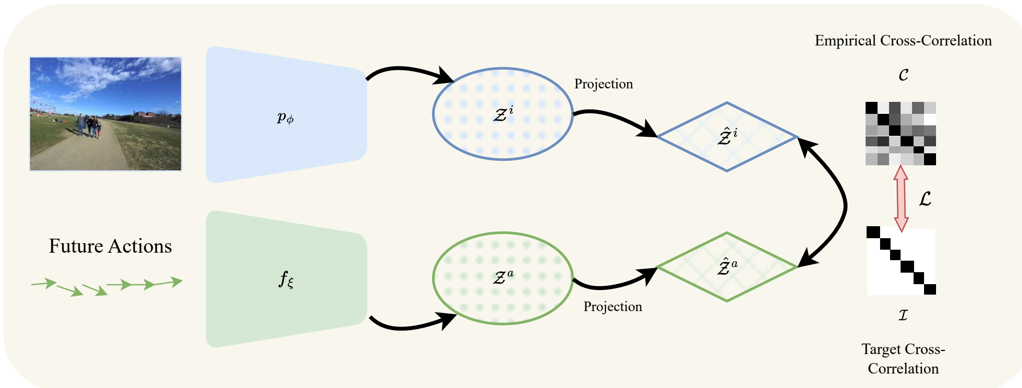


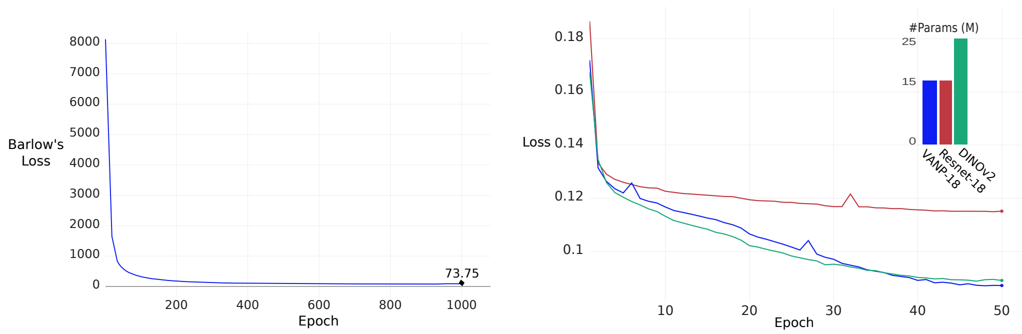
Figure 2: VANP first maps the sequence of actions to a sparser, higher-dimensional space Z^a (green). Then, leveraging Barlow Twins' redundancy reduction principle, VANP induces sparsity on Z^i (blue), the visual embedding, by minimizing the mutual information between Z^i and Z^a.

Problem Definition: Visual navigation is the task of navigating an environment with only RGB camera input. Unlike conventional geometric navigation tasks using, e.g., LiDARs or depth images, this task is challenging due to the lack of explicit geometric information. The visual navigation problem can be formalized as follows. **Input:** The robot is given a sequence of past and current images from its front-facing camera, $o_t = [I_{t-\tau_p}, I_{t-\tau_p+1}, \dots, I_t] \in \mathcal{O}$, where t is the current time step, τ_P is the number of past frames, and $\mathcal{O}$ is the space of all possible image sequences. The robot is also given its current goal e.g. GPS coordinates, pose, image, or next local coordinate in 2D space, $g \in \mathcal{G}$, which determines the direction it should move in the next time step. **Output:** The robot must select an action, $a_t \in \mathcal{A}$ consists of continuous linear and angular velocities, where $\mathcal{A} \in [-1, 1]^2$ is the action space, where $[-1, 1]$ maps to the minimal and maximal linear and angular velocity of the robot. **Visual Navigation:** The goal is to learn a policy, $\pi_{\mathbf{\Theta}} : \mathcal{O} \times \mathcal{G} \rightarrow \mathcal{A}$, where $\mathbf{\Theta}$ represents the policy's parameters, to determine which action to take at each time step to reach its goal destination efficiently while avoiding collisions with other agents and observing underlying social norms.

End-To-End Model: In end-to-end or holistic models we define the policy $\pi_{\mathbf{\Theta}}$ as follow: $\mathbf{a} = \pi_{\mathbf{\Theta}}(\mathbf{o}, g) = \sigma_\zeta(p_\phi(\mathbf{o}) \oplus q_\psi(g))$, where σ is the controller policy parametrized by ζ, p is the image encoder parameterized by ϕ, q is the goal encoder parameterized by ψ, and $\oplus$ is the concatenation of two output vectors. To learn these parameters, two common approaches are (1) to learn all of them together in an end-to-end manner which makes the training difficult and time-consuming or (2) to train the image encoder separately and only fine-tune the goal encoder along with the controller to reduce training time.

3.1 Vision-Action Model

VANP is inspired by the redundancy reduction principle of Barlow's Twins to train the image encoder p. However, unlike vision self-supervised learning (SSL) models that work on the joint embedding of augmented images [53, 54], VANP correlates the action space $\mathcal{A}$ with the pixel latent space $\mathcal{O}'$. Under the assumption that every dynamic object or person in the environment will adhere to social norms, we define VANP pretraining as follows: We sample a batch of $(I^i, a^i_{t:t+\tau_F})$ from dataset D where i is the sample number, I^i is a single image at time t and $a^i_{t:t+\tau_F}$ is a sequence of actions starting from t and ending in $t + \tau_F$, where τ_F is the number of frames in the future. We then feed I^i to p_ϕ and $a^i_{t:t+\tau_F}$ to f_ξ, typically a multilayer perceptron (MLP), to learn image Z^i and action Z^a embeddings, respectively. Finally, we use Barlow Twin's objective function to learn ϕ and ξ:



(a) Barlow's loss smoothly decreases which leads to correlations between actions and visual features.

(b) Downstream navigation training loss with Resnet-18, VANP-18, and DINOv2 as visual encoder.

Figure 3: Vision-Action Navigation Pretraining and Downstream Navigation Fine-Tuning.

$$\mathcal{L}_{\mathcal{BT}} = \underbrace{\sum_i (1-\mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j\neq i} \mathcal{C}_{ij}^2}_{\text{redundancy reduction term}} \tag{1}$$

where λ is the trade-off between the first and second terms of the loss, and C is the cross-correlation matrix computed between the outputs of the action and image embeddings along the batch dimension.

Leveraging Barlow's objective function provides the advantage of not requiring negative samples, which can be difficult to define in action space. For example, when a person is in front of us, there may be two correct actions: overtake from the left or overtake from the right. Therefore, simply negating the angular velocity does not give us a negative sample and may introduce ambiguity like the example as mentioned earlier. Using actions from another sequence may not provide useful information for visual navigation, as the actions are inherently conditioned on the observations. Furthermore, the sparse high dimensional action latent space acts as a soft-whitening constraint on the image latent space to reduce the redundancy in extracted features from the image [55, 11].

4 Preliminary Experimental Results

4.1 Implementation Details

We implement our method with PyTorch [56] and the training is performed on a single A100 GPU with 80 gigabytes of memory. You can find the code here.

Model architecture: Considering the limited computation resources onboard most mobile robots, we choose ResNet-18 [57] without the classification head as a lower latency image encoder and we call it VANP-18. We use a multilayer perceptron (MLP) with two hidden layers as the action encoder to produce the embeddings of $Z^i, Z^a \in \mathbb{R}^{512}$. Both encoders were followed by MLPs with three layers as the projection heads to generate the final $Z'^i, Z'^a \in \mathbb{R}^{8192}$, the same as Zbontar et al. [11]. The most important challenge here is that the two distinct networks for producing the embeddings have different modalities and therefore the output range significantly varies. Therefore, we initialize all the deep networks with the Kaiming Normal initialization [57] with mean zero and variance one to mitigate the discrepancy in the output of the networks. For the end-to-end model, we follow the model used by Nguyen et al. [50]. We freeze the image encoder and only train the goal encoder and controller during downstream task training for all the experiments.

Optimization: As proposed by Zbontar et al. [11], we use the LARS optimizer [58] and train the model for 1000 epochs with a batch size of 16384. For the other hyperparameters, we use a learning rate of 0.2 for the weights and 0.0048 for the biases. We use the first ten epochs as the warm-up phase and update the learning rate by a factor of 8 during these epochs. We observe that, as suggested by Zbontar et al. [11], using any other factor than 8 results in gradient explosion during training.

Dataset: We leverage two unique datasets: SCAND [49] and MuSoHu [50], both of which encapsulate robot and human navigation data from the egocentric perspective. Both large-scale real-world datasets are collected in a variety of natural crowded public spaces. MuSoHu comprises approximately 20 hours of data captured from human egocentric motion. The recordings capture human walking patterns in public spaces, providing insights for learning human-like, socially compliant navigation behaviors. SCAND is an autonomous robot navigation dataset that captures 8.7 hours of human-teleoperated robot navigation demonstrations in naturally crowded public spaces on a university campus. By combining these two, we create a dataset of over 1 million samples to train the image encoder on the pretext task. For pretext task training, we use a single image $I_t \in \mathbb{R}^{70\times70}$ along with a sequence of actions $a_{t:t+\tau_F} \in \mathbb{R}^{\tau_F\times2}$ parsed at 25 Hz, comprising of 4 seconds in the future. For the downstream task, we use a sequence of past observations $I_{t-\tau_P:t} \in \mathbb{R}^{t\times70\times70}$ along with the polar coordinates of the next local goal $g \in \mathbb{R}^2$ parsed at 4 Hz, containing 1.5 seconds history as the network input to produce the actions $\mathcal{A}_{t:t+\tau_F} \in \mathbb{R}^{\tau_F\times2}$ for two seconds in the future. In both stages, we use the augmentations proposed by Codevilla et al. [35].

4.2 Results Discussion

We report our preliminary results. To evaluate the effectiveness of VANP pretext training, we quantitatively compare it with DINOv2 [12], a self-supervised vision transformer model that learns universal features suitable for eight different visual tasks including depth estimation, semantic segmentation, instance retrieval, dense and sparse matching. DINOv2 models exhibit robust out-of-distribution performance, and the learned features can be used directly without fine-tuning. We use DINOv2 as the upper performance bound and ResNet-18 trained on ImageNet ILSVRC-2012 as the performance baseline. To ensure a fair comparison, the architectures of all other components of the end-to-end model are kept fixed. During the downstream navigation task, we only train the goal encoder and controller, and the weights of the image encoder are frozen, regardless of the architecture used.

We use ResNet-18 as the architecture for VANP pretext training (VANP-18). The smoothly decreasing Barlow's loss in Figure 3a indicates that the learned visual features align with the actions. Figure 3b shows the training loss for the downstream visual navigation task. During end-to-end training, the weights of the visual encoder ϕ are frozen, and we only train the goal encoder q_ψ and the controller σ_ζ. As shown in the figure, VANP-18 outperforms ResNet-18 with the same architecture but a different training paradigm which shows the effectiveness of VANP's pretext training. VANP-18 also achieves slightly better performance than DINOv2, a vision transformer trained on billions of images. Figure 3b shows a comparison of parameters for the holistic model when we use different visual encoders.

We separate a few trajectories from the dataset and use them as unseen scenarios for qualitative evaluation. After qualitatively comparing the model outputs, we observe that VANP-18 performs human avoidance. Figure 4 shows a few examples from the evaluation set. Note that negative values for angular velocity denote turning right, and positive values mean turning left. VANP-18's decisions do not agree with human decisions in some scenarios, but it does not mean that they are completely wrong: for example, the disagreement in the second red border image is because the human predicts the group's future trajectory, and decides to move from the left while VANP-18 decides to go from the right.

5 Conclusions and Future Work

In this work, we propose a self-supervised training approach to train visual encoder models specifically designed for visual navigation. This approach is motivated by the observation that humans only pay attention to a small region of their frontal view to make navigation decisions. By reversing this observation, we use the decisions to extract only visual features that are relevant to the visual navigation task, unlike computer vision models that tend to extract every detail in the environment, which can lead to confusion of neural-based controllers.

Figure 4: VANP's qualitative performance on unseen scenarios. Green: VANP outputs align with the demonstrations; Red: VANP outputs do not align with the demonstrations.

We hope that this work will inspire new ideas in the field of visual navigation. In the future, we plan to conduct more real-world experiments and train deeper models, such as VAM-34 and VAM-50, based on ResNet-34 and ResNet-50, respectively. In this work, we only use datasets collected in social environments. Another future direction is to merge datasets from different environments, such as off-road, indoor, outdoor, and social environments, to evaluate the generalizability of the proposed approach.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

[2] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, and J. Gao. Vision-Language Pre-training: Basics, Recent Advances, and Future Trends.

[3] H. Cao, C. Tan, Z. Gao, G. Chen, P.-A. Heng, and S. Z. Li. A Survey on Generative Diffusion Model.

[4] W. Zhang, L. He, H. Wang, L. Yuan, and W. Xiao. Multiple Self-Supervised Auxiliary Tasks for Target-Driven Visual Navigation Using Deep Reinforcement Learning. 25(7):1007. ISSN 1099-4300. doi:10.3390/e25071007.

[5] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the Carbon Emissions of Machine Learning.

[6] E. Strubell, A. Ganesh, and A. McCallum. Energy and Policy Considerations for Deep Learning in NLP.

[7] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. Carbon Emissions and Large Neural Network Training. . doi:10.48550/ARXIV.2104.10350.

[8] D. Patterson, J. Gonzalez, U. Hölzle, Q. H. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink, .

[9] L. B. Heguerte, A. Bugeau, and L. Lannelongue. How to estimate carbon footprint when training deep learning models? A guide and review.

[10] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners.

[11] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction.

[12] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision.

[13] A. Bardes, J. Ponce, and Y. LeCun. MC-JEPA: A Joint-Embedding Predictive Architecture for Self-Supervised Learning of Motion and Content Features.

[14] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations.

[15] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

[16] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum. A Cookbook of Self-Supervised Learning.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. doi:10.1109/CVPR.2009.5206848.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8693, pages 740–755. Springer International Publishing. ISBN 978-3-319-10601-4 978-3-319-10602-1. doi:10.1007/978-3-319-10602-1_48.

[19] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-Driven Representation Learning for Robotics.

[20] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, volume 9912, pages 549–565. Springer International Publishing. ISBN 978-3-319-46483-1 978-3-319-46484-8. doi:10.1007/978-3-319-46484-8_33.

[21] R. Dadashi, L. Hussenot, D. Vincent, S. Girgin, A. Raichuk, M. Geist, and O. Pietquin. Continuous control with action quantization from demonstrations. In *International Conference on Machine Learning*, pages 4537–4557. PMLR.

[22] J. Luo, P. Dong, J. Wu, A. Kumar, X. Geng, and S. Levine. Action-quantized offline reinforcement learning for robotic skill learning. In *7th Annual Conference on Robot Learning*.

[23] X. Huang, D. Batra, A. Rai, and A. Szot. Skill Transformer: A Monolithic Policy for Mobile Manipulation.

[24] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale.

[25] N. Di Palo, A. Byravan, L. Hasenclever, M. Wulfmeier, N. Heess, and M. Riedmiller. Towards A Unified Agent with Foundation Models. doi:10.48550/ARXIV.2307.09668.

[26] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters. Motion Planning Diffusion: Learning and Planning of Robot Motions with Diffusion Models.

[27] A. Hiranaka, M. Hwang, S. Lee, C. Wang, L. Fei-Fei, J. Wu, and R. Zhang. Primitive Skill-based Robot Learning from Human Evaluative Feedback.

[28] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR.

[29] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li,

R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*.

[30] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR.

[31] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

[32] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. 30.

[34] M. H. Nazeri and M. Bohlouli. Exploring Reflective Limitation of Behavior Cloning in Autonomous Vehicles. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1252–1257. IEEE. ISBN 978-1-66542-398-4. doi:10.1109/ICDM51629.2021.00153.

[35] F. Codevilla, E. Santana, A. M. Lopez, and A. Gaidon. Exploring the Limitations of Behavior Cloning for Autonomous Driving. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE. doi:10.1109/ICCV.2019.00942.

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

[37] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 66–75. PMLR, 2020-11-30, 2020.

[38] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger. Learning Situational Driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. doi:10.1109/cvpr42600.2020.01131.

[39] D. Chen, V. Koltun, and P. Krähenbühl. Learning to drive from a world on rails. In *ICCV*.

[40] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving.

[41] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton. Model-Based Imitation Learning for Urban Driving.

[42] B. Jaeger, K. Chitta, and A. Geiger. Hidden biases of end-to-end driving models. In *International Conference on Computer Vision (ICCV)*.

[43] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances.

[44] S. Li, X. Puig, C. Paxton, Y. Du, C. Wang, L. Fan, T. Chen, D.-A. Huang, E. Akyürek, A. Anandkumar, et al. Pre-trained language models for interactive decision-making. 35: 31199–31212.

[45] M. Reid, Y. Yamada, and S. S. Gu. Can wikipedia help offline reinforcement learning?

[46] B. Baker, I. Akkaya, P. Zhokov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. 35:24639–24654.

[47] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. GNM: A General Navigation Model to Drive Any Robot, .

[48] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation, .

[49] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone. Socially Compliant Navigation Dataset (SCAND): A Large-Scale Dataset of Demonstrations for Social Navigation.

[50] D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao. Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset.

[51] N. Hirose, D. Shah, A. Sridhar, and S. Levine. SACSoN: Scalable Autonomous Data Collection for Social Navigation.

[52] S. M. Waghmare, K. Wilber, D. Hawkey, X. Yang, M. Wilson, S. Debats, C. Nuengsigkapian, A. Sharma, L. Pandikow, H. Wang, H. Adam, and M. Sirotenko. SANPO: A Scene Understanding, Accessibility, Navigation, Pathfinding, Obstacle Avoidance Dataset.

[53] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR.

[54] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

[55] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. Whitening for self-supervised representation learning. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3015–3024. PMLR.

[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[57] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778. doi:10.1109/cvpr.2016.90.

[58] Y. You, I. Gitman, and B. Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.