IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 August 2023, accepted 24 October 2023, date of publication 8 November 2023, date of current version 20 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331678

RESEARCH ARTICLE

Visual Place Recognition Pre-Training for End-to-End Trained Autonomous Driving Agent

SHUBHAM JUNEJA[1], POVILAS DANIUŠIS[2,3], AND VIRGINIJUS MARCINKEVIČIUS[1]
[1]Institute of Data Science and Digital Technologies, Vilnius University, 08412 Vilnius, Lithuania
[2]Neurotechnology, 06118 Vilnius, Lithuania
[3]Research Institute of Natural and Technological Sciences, Vytautas Magnus University, 53361 Kaunas, Lithuania

Corresponding author: Shubham Juneja (shubham.juneja@mif.stud.vu.lt)

ABSTRACT End-to-end autonomous driving often relies on the concept of learning to imitate from expert demonstrations. Since those demonstrations cannot cover all possible variations in data, there always are situations where the trained agents encounter unseen conditions, which results in a shift in the data distribution. One of the most common causes of this shift is changes in weather and lighting conditions. In this study, we suggest using a pre-training based on the visual place recognition (VPR) method, in order to mitigate this effect. We compare the corresponding navigation agent to a baseline agent which relies on the commonly used ImageNet pre-training by evaluating as per the Leaderboard driving benchmark in CARLA environment. According to our experiments, pre-training on the VPR task shows higher resistance to unseen weather conditions. The findings calculated in our study are evaluated over multiple seeds to show statistical consistency. The accompanying open-source code repository can be accessed via https://github.com/Shubhamcl/vpr_pretrained_agent/.

INDEX TERMS Imitation learning, autonomous driving, agents, self-driving cars, deep learning, pre-training.

I. INTRODUCTION

Autonomous driving research is being approached in two different forms of methods. One being the modular method, where the functionality of driving is broken down into multiple sub-tasks, and each sub-task is carried out by a designated module [1]. While the second method uses end-to-end learning where driving is learned as a skill directly from expert demonstrations. As the former method requires high amount of engineering effort, research on end-to-end learning is progressively gaining traction [2], [3].

End-to-end learning methods learn behaviours out of data, and this learning relies upon the concept of imitation learning [4]. While imitation learning promises capabilities of learning behaviours, it suffers from the problem of shifts in data distribution between training data and testing data. This problem is also commonly known as co-variate shift [2]. Several methods [5], [6], [7] attempt to mitigate co-variate shift by integrating corrective demonstrations to the training data corpus, yet in the unseen environments the resulting agents fail to perform as correctly as they perform in the seen ones.

Depending on the area of application of imitation learning methods, the causes of co-variate shifts can vary. One of the causes of the shift in the data distribution in autonomous driving is when the trained agent encounters unseen weather and lighting conditions. As these changes can have a drastic impact on the inputs provided to the agent, the agent's decisions can also be affected, making its performance frail and unpredictable.

The concept of pre-training has revealed to be remarkably promising in the most recent works [8], [9]. The overall idea of pre-training refers to first training a neural network on a large and diverse dataset aimed to solve some task (e.g., classification [10]), expecting that the trained network will be able to produce a sufficient amount of general features useful for other tasks of interest (e.g., autonomous driving). This concept aims to take advantage of available datasets, which may not be directly related to the main task to be solved, but hold some common properties. Pre-training may be very

The associate editor coordinating the review of this manuscript and approving it for publication was Chih-Yu Hsu.

useful for problems that are not sample efficient and there may not be enough data to use. The trend of pre-training is very prominent in language models which are pre-trained over enormous text data corpus and in visual perception models for tasks such as recognition and detection where pre-training is commonly carried out over data from the ImageNet [11] dataset. Various authors [9], [12] have pointed out that pre-training on a task related to the target task rather than the ImageNet classification task can be beneficial for learning.

In this study, we hypothesise that autonomous driving may be highly dependent on specific visual features, which may not be detected by the neural network during commonly used ImageNet-based pre-training, as image classification is highly unrelated to the task of our interest. We investigate this hypothesis by analysing how pre-training an encoder of an end-to-end driving agent over the task of visual place recognition (VPR) can help adapt better to unseen weather and lighting conditions.

The main contributions of this paper are:

1) Using the offline Leaderboard benchmark [13], [14] on CARLA 0.9.11 we provide an empirical demonstration that the SegVPR-pretrained image encoder [15] is more efficient, compared to its ImageNet-pretrained counterpart, in an end-to-end trained driving agent.
2) We also demonstrate that our findings are (statistically) consistent across multiple evaluations, both under seen and unseen simulator environments.

The rest of the paper is organised in the following way. Section II covers related work on imitation learning for autonomous driving, pre-training in imitation learning. Section III describes the approach proposed in the paper, section IV elaborates how the experiments are carried out to show that results conform to the hypothesis. Finally, section V concludes the paper.

II. RELATED WORK

ALVINN [16] was one of the earliest methods which showed the ability to learn the skill of driving in an end-to-end manner using imitation learning. With the use of modern forms of neural networks, PilotNet [17] improved upon what was previously done with much simpler learning methods. A line of research also came into existence pointing out the co-variate shift problem in imitation learning and approaching to solve it with gradually advancing data aggregation (DAgger) methods [5], [6], [7]. Meanwhile, the practice of applying DAgger (or one of its newer variants) has remained prominent in most newly proposed methods. Another method that has become a standard in more recent research is conditional imitation learning. Conditional imitation learning with ResNet (CILRS) [18], [19] conditions the neural network by the high-level action and uses a deeper neural network architecture. This has become a framework architecture for many of the research works that have followed.

Recent works explore various aspects of the problem of learning to drive. One area of exploration is leveraging temporal information. The spatio-temporal perception, prediction, and planning (ST-P3) [20] method shows how glancing at multiple input time steps can enable better planning and prediction by preserving geometric information and taking into account past motion variations. Meanwhile, trajectory-guided control prediction [21] is another method that shows predicting multiple action steps and a trajectory, given input state, shows improved action predictions. Another area of exploration has been the use of multiple modalities. Transfuser [22] is a method that integrates image and LiDAR representations. It uses a self-attention mechanism which leverages the use of transformer neural networks with some modifications. Model-based imitation learning [14], another recent method, explores learning the world model and a policy simultaneously. Whereas, planning-oriented autonomous driving [23] integrates the process of planning along with it's sub-tasks into the same neural network. Another important aspect is explored by Roach [13], which investigates how the quality of demonstrations used in training can be enhanced by using a reinforcement learning agent as a coach to generate better demonstrations for training, and shows a clear improvement in driving performance compared to the default autopilot. Most of the above-mentioned works use ImageNet initialised pre-trained weights, or they use randomly initialised weights in the selected neural network architecture of the driving agent.

ImageNet pre-trained architectures have been used to transfer learning to various vision tasks (object detection [24], image segmentation [25], etc.) with different datasets. This has been done under an implicit assumption that, architectures pre-trained with ImageNet transfer better which has been later studied [9], and the findings say that this assumption may not always hold true. The study indicates that ImageNet learned features do not transfer well to fine-grained tasks and such pre-training would provide up to minimal benefits. Another study that goes into the taxonomy of the transfer learning task [12] shows good ability of learning between tasks that are related and are in a similar domain. The ImageNet dataset may have many similarities to object detection datasets such as the COCO dataset [26] but is very distant from driving datasets. This brings up the need for specialised pre-training for specific tasks.

Pre-training of models for imitation-learning-based autonomous driving methods is relatively a new subarea of research. Video pre-training (VPT) [27] is a method that performs pre-training for behaviour cloning on a video game environment. It uses in-the-wild available data. As this data is unlabelled, prior to pre-training, pseudo-labels are formed using an inverse dynamics model (IDM) that is trained on a labelled dataset. Another method that uses an IDM to its advantage is action-conditioned contrastive pre-training (ACO) [28], which pre-trains on real-world driving videos from Youtube. Instead of pre-training with behaviour cloning, ACO chooses the task of contrastive representation

learning. While IDM can easily incorporate noisy data into the pre-training data, self-supervised learning can be a better alternative to IDM. Policy pre-training via geometric modelling (PPGeo) [29] uses self-supervised learning to pre-train on tasks of pose, depth, and future ego-motion prediction, followed by training on the task of interest, that is driving.

ACO points out that the agent should remain weather and lighting invariant because such factors can interfere with the process of driving, where the agent which may be seen as biased, will perceive unseen conditions as noisy inputs. The same is valid from the perspective of co-variate shift, as it is highly unlikely that the agent will only be operating in previously seen conditions. To reduce the dependency on many iterations of data aggregation techniques, the trained agent needs to hold a good amount of invariance against changes in data distribution. To achieve such invariance, we look at the VPR task which learns retrieval of place images while incorporating exposure to various weather and lighting conditions. Prior to learning the task of driving, we perform pre-training over the task of VPR.

III. METHOD

Generally, to train an end-to-end driving agent with imitation learning, training is performed over a dataset of demonstrations and then data aggregation is performed to refine the dataset further and prepare an agent to improve upon in the next iteration of training. For research and better reproducibility, the dataset is formed utilising the CARLA simulator [30]. Most methods incorporate an encoder for vision into the neural network, which is pre-trained on the ImageNet [11] dataset over the image recognition task. We propose pre-training an image encoder over the VPR task and further training the same encoder for learning the task of driving with imitation learning. The explanation of how we pre-train is described in section III-A and how is the pre-trained network used in the driving agent is described in section III-B.

A. VISUAL PLACE RECOGNITION PRE-TRAINING

The classical problem of VPR which aims to enable the retrieval of places, largely suffers from mislabelling places when weather and lighting conditions change. In order to overcome this, the datasets for VPR contain data points from each location with varying conditions, resulting in approaching weather and lighting invariance. To inherit this ability, we utilise SegVPR [15], which is a neural network architecture consisting of a ResNet [10] encoder and a separate decoder with multiple modules.

SegVPR aims to take away attention from varying weather and lighting conditions in order to focus on other aspects of scene for place recognition. To achieve this, it incorporates multiple branches with interlinking in its decoder module, which follows an ImageNet pre-trained encoder. By branching in the decoder, SegVPR blends in a semantic segmentation decoder for an auxiliary task that is guided by multi-scale attention. The overall loss function is a sum of VPR loss and semantic segmentation loss [15]:

$$\mathcal{L}_{VPR-SemSeg} = \mathcal{L}_{VPR} + \alpha \cdot \mathcal{L}_{SemSeg}, \tag{1}$$

where $\alpha > 0$ is a scalar weight for semantic segmentation loss. The VPR loss $\mathcal{L}_{VPR}$ is a weakly supervised triplet margin loss given by,

$$\mathcal{L}_{VPR} = h(d(F_q, F_p) + m - d(F_q, F_n)), \tag{2}$$

where h is the hinge loss $h(x) = max(x, 0)$, d is the Euclidean distance, $m > 0$ is a fixed margin, and F_q, F_p, and F_n represent query, positive, and negative samples respectively. The semantic segmentation loss $\mathcal{L}_{SemSeg}$ is given by formula,

$$\mathcal{L}_{SemSeg} = -\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} y_i \cdot \log p_i^{y_i}((M^i \cdot f_d^i)), \tag{3}$$

which is equal to a cross-entropy loss that is computed for each class y_i at pixel i from the image $\mathcal{I}$, where M_i is an attention map related to the feature f_d^i coming from the segmentation decoder module and p_i denotes the probability of class y_i.

The dataset used for training SegVPR contains VPR data that is captured in the CARLA simulator. It consists of 40,000 images collected from two different towns (Town 3 and Town 10), under weather conditions that are also present in the training dataset for policy learning, as per the offline Leaderboard benchmark [13]. Pre-training over this dataset is highly advantageous for our approach as it keeps the domain of the data common between the pre-training phase and the later phase where we train over the task of interest, as both phases use the CARLA simulator.

B. TRAINING AGENT WITH A PRE-TRAINED ENCODER

To benefit from the pre-training over varying conditions we extract the image encoder from the SegVPR architecture and use it as the image encoder in our agent's neural network architecture.

We base the rest of the agent upon CILRS [19], where we train by conditioning inputs as per the high-level commands along with the input data. The high-level commands are provided by a planner and are in accordance with selected destinations to reach. These commands are of discrete form, consisting of instructions such as turn left, follow lane, change lane, etc. For initial data collection of demonstrations, we use the work presented in Roach [13]. The reinforcement learning trained agent Roach operates from the bird's eye viewpoint and hence has better visibility of the world to make informed driving decisions. While the trained Roach agent operates the vehicle, data from the front view camera is collected for the initial dataset of demonstrations. This automates the data collection process rather than requiring human control of the vehicle, and also enables reliable and longer data collection. This step is then followed by the training of our proposed agent with the pre-trained weights. After an agent is trained on the initial dataset to perform the

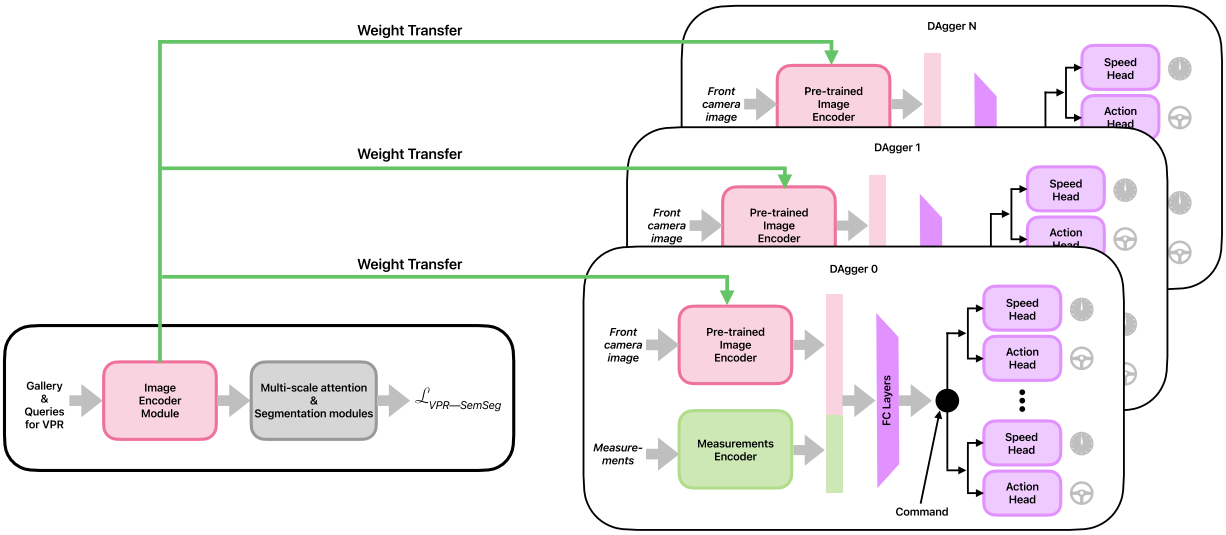


FIGURE 1. The figure illustrates the overall block diagram of the visual place recognition (VPR) pre-trained method, where at first, an image encoder is pre-trained on the VPR task (left) followed by weight transfer to train for the task of end-to-end driving (right).

DAgger process, the trained agent is let to drive; meanwhile, the Roach agent is supervising by proposing actions. When there is no consensus between the Roach agent and our trained agent, the data in such cases is aggregated in order to make our agent learn from the action proposals of the Roach agent. In our work, the way corrective demonstration data is aggregated into the initial dataset of demonstration is based on the methodology of the original DAgger [5] algorithm.

The agent's architecture (as in Fig. 1) consists of a measurements encoder that accepts current speed and high-level command in the form of one hot encoding. Parallel to this, the SegVPR encoder module encodes image input. Both encodings are concatenated and downsized using a join module which consists of fully connected layers. The joint encoding is then fed to the action branches module, where each branch is responsible for each discrete high-level command (as used in CILRS and Roach). Given the high-level command, the corresponding branch is chosen for the low-level driving command. During training, non-corresponding branches are zeroed out.

Let $X \in \mathbb{R}^{224\times224\times3}$ be an input image from the front camera sensor. The agent maps X onto an action in $\mathbb{R}^2$ and is represented by the following equation:

$$\widehat{\mathbf{a}}(X, u|\theta, \xi, \phi, \psi) := \sum_{i=0}^{n} c_i \phi_i(X, u|\theta, \xi, \phi, \psi), \qquad (4)$$

where $\phi_i(X, u|\theta, \xi, \phi, \psi)$ corresponds to the output of i^{th} action branch of $f_A(f_J(f_E(X|\theta), f_M(u|\xi)|\phi)|\psi)$. Here X is the input image, f_E is the image encoder with parameters θ pre-trained upon the VPR task (i.e., SegVPR encoder), u is a vector holding measurements (current speed and high-level command), f_M is the measurements encoder network with parameters ξ, f_J is another neural network module with parameters ϕ that concatenates the image and measurements encodings and downsizes it, f_A is the actions branches module with parameters ψ which calculates a low-level command for each high-level command, and c_i is the one-hot encoded command which is indexed with i that zero outs the non-command branches.

To simplify the comparison with a baseline, we use the loss function as the sum of action loss and a speed prediction regularisation,

$$\mathcal{L}_{Agent}(\theta, \xi, \phi, \psi) = \mathcal{L}_A(\theta, \xi, \phi, \psi) + \lambda_S \cdot \mathcal{L}_S, \qquad (5)$$

where the action loss $\mathcal{L}_A$ is equal to L1 loss between expert action $\hat{\mathbf{a}}$ and predicted action $\mathbf{a}$, given by

$$\mathcal{L}_A = \|\widehat{\mathbf{a}}(X, u|\theta, \xi, \phi, \psi) - \mathbf{a}\|_1, \qquad (6)$$

and the speed prediction regularisation $\mathcal{L}_S$ between measured speed $\hat{s}$ and predicted speed s is given by

$$\mathcal{L}_S = \left|\hat{s} - s\right|. \qquad (7)$$

The regularisation effect is regulated with a scalar value λ_s.

The entire process of training our agent for driving is also explained in Algorithm 1. The selection of conditions (towns and weather) for training, evaluating, and testing are described further in the following section IV, along with training details.

IV. EXPERIMENTS

A. BENCHMARK AND EXPERIMENT SETTINGS

For evaluating the effectiveness of the suggested VPR-based pre-training, we run our experiments using the offline Leaderboard benchmark [13], [14] on CARLA 0.9.11. The Leaderboard benchmark considers multiple towns with challenging traffic situations consisting of roundabouts, stop signs, freeways, and more. The benchmark also consists of a set of weather conditions and traffic densities.

Algorithm 1 The VPR Pre-Trained Navigation Agent

Input: Initial dataset D collected using the Roach agent, trained SegVPR encoder f_E.
Output: trained *agent*

1: **for** dagger iteration `i = 0 to 5` **do**
2: Initialise agent $agent_i$.
3: Initialise $agent_i$'s image encoder with f_E.
4: Train $agent_i$ on D.
5: Collect dataset $D_i = (X, u, \pi^*(X, u))$, where X and u are input image and measurements (speed and high-level command), correspondingly, and π^* is the supervising Roach agent's output, measured in situations when there is a disagreement between the predictions of Roach and $agent_i$.
6: Aggregate dataset: $D \leftarrow D \cup D_i$.
7: **end for**
8: Return best $agent_{i^*}$ as per scores on the Leaderboard benchmark.

All experiments are run under the busy traffic density as simpler traffic density conditions have previously been shown to be obsolete [13].

Along with the VPR pre-trained method, a baseline method is also trained. The only difference between the two is that the baseline method contains an ImageNet-trained ResNet encoder. As per the Leaderboard benchmark's specification, both methods are trained on data from train towns and train weather conditions and evaluated on the subset of training settings (evaluation town & weather conditions) as well as on test town and test weather conditions. The distribution of training, evaluation, and test conditions is described in tables 1 and 2. The example images from different weather conditions are depicted in Fig. 2. Both methods are trained for 5 iterations of DAgger and evaluated as per the benchmark settings after every iteration. The reported results in the latter section are obtained by running over 3 evaluation seeds.

Both models share the same architecture with a ResNet image encoder f_E, however with different initial weights: VPR pre-trained model uses weights from the place recognition task [15], and the baseline model relies on the initialisation from ImageNet classification task [11]. The measurement encoder f_M is a stack of 2 fully connected layers with output dimension set to 128 at each layer. The join module f_J consists of 3 fully connected layers with the output dimension set to 512, 512 and 256. Whereas each of the action branches f_A hold 3 fully connected layers with the output dimensions set to 256, 256 and 2, respectively. All modules consisting of fully connected layers use a rectified linear unit activation, except the last layers in action branches.

B. IMPLEMENTATION DETAILS

All the conducted experiments are implemented using PyTorch [31]. For evaluation, we use the already implemented code provided by Roach study [13].

TABLE 1. Distribution of weather conditions for training, evaluation and testing.

Training weathers	Evaluation weathers	Testing weathers
Wet noon	Wet noon	Wet sunset
Clear sunset	Clear sunset	Soft rain sunset
Clear noon		
Hard rain noon		

TABLE 2. Distribution of towns for training, evaluation and testing.

Training towns	Evaluation towns	Testing towns
Town 1	Town 1	Town 2
Town 3	Town 3	Town 5
Town 4 - train routes	Town 4 - train routes	Town - 4 test routes
Town 6	Town 6	

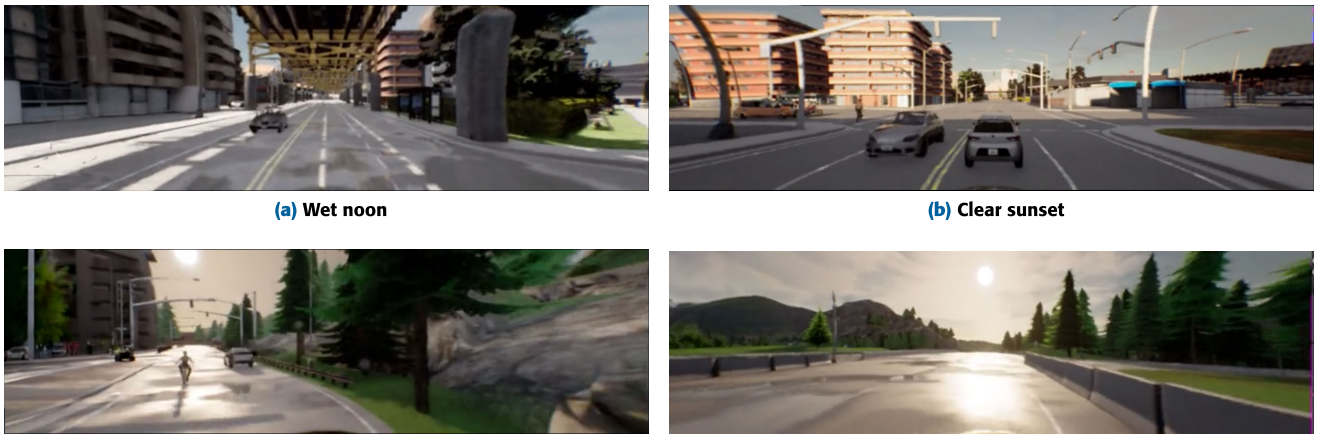
At first, agents of both methods (VPR pre-trained and baseline) are trained on the same initial dataset collected with the Roach agent, and then each of the two agents is trained on the initial set of demonstration data and the DAgger data generated by that particular agent. During each iteration, the agents are trained with increased dataset size, but the rest of the settings remain constant. In comparison to previously published works [13], we train our agents on images with a resolution down-scaled to 224 × 224 pixels instead of the roughly $\frac{256\times900}{224\times224} \approx 4.6$ times higher resolution [13]. This step allows us to train faster with the limited amount of computational resources available, as our resolution of choice operates with ≈ 4.6 times less memory. However, this restricts us from comparing our results with other methods, according to their reported metrics.

C. TRAINING DETAILS

For every DAgger iteration of training for both methods, the entire neural network architectures with all the parameters are tuned against the loss function in equation (5). We carry out training for 20 epochs with learning rate of $1e-4$ and weight decay of $1e-5$. The learning rate is stepped down to $1/10^{th}$ from epoch 15. We train in batches of 256 samples per batch. The training is run on a RTX 3090 with the data stored on a Gen4 NVME solid state drives for faster reading of data. Training a single iteration takes around 20 to 35 hours. The variability in time is mostly caused by the increments in the size of the full dataset with every DAgger iteration as previously mentioned in section IV-B.

D. EVALUATION METRICS

The performance in the evaluation is validated over two metrics, namely route completion and distance completion. Route completion represents the percentage of routes completed by the agent under combinations of town and weather conditions from a given starting point to an endpoint. Distance completion is the percentage of distance travelled



(a) Wet noon (b) Clear sunset

(c) Soft rain sunset (d) Wet sunset

FIGURE 2. The figure shows two weather conditions used for evaluation 2a and 2b, that are used as a part of evaluation set to test in known conditions, followed by weather conditions 2c and 2d, that are unseen by the agent and used in testing.

TABLE 3. Route completion (%) of driving agents on training and new (testing) conditions. Highest of all DAgger iterations reported.

Pre-training Method	Train town & weather	New town & weather
ImageNet	64.67 ± 2	49.35 ± 9
VPR	81.33 ± 4	60.25 ± 2

TABLE 4. Distance completion (%) of driving agents on training and new (testing) conditions. Highest of all DAgger iterations reported.

Pre-training Method	Train town & weather	New town & weather
ImageNet	79.02 ± 2	75.75 ± 7
VPR	91.97 ± 3	86.01 ± 0

with respect to the actual route distance. Route completion answers the question: What percentage of routes is the agent able to complete on average? Whereas distance completion answers the question: at what percentage of distance completed does the agent fail on average?

E. RESULTS

We make a comparison between the effect of using a pre-trained encoder which is trained to possess invariance against changes in weather conditions, and an ImageNet pre-trained encoder for the task of autonomous driving. We run evaluations on the baseline and the VPR pre-trained method where for each method we have 6 trained agents (each representing one DAgger iteration). Each agent is tested in familiar towns and weather conditions (train town and weather), followed by unfamiliar towns and weather conditions (test town and weather). Since the evaluations are carried out in a simulator that contains multiple actors that are placed randomly (e.g. pedestrians, other vehicles, etc.), we perform the evaluation three times for every agent, reporting its average performance.

Table 3 provides empirical confirmation of our hypothesis, that VPR pre-training may be more efficient compared to the general ImageNet-based pre-training (our baseline), as we see that using VPR pre-training method results in an increase of route completion by $\approx 17\%$.

When exposed to unseen town and weather conditions, VPR pre-training shows $\approx$ 11% higher score of completed routes than the baseline method. The use of a VPR pre-trained method not only shows better route completion but also shows faster convergence towards higher performance over DAgger iterations than the baseline, as in Fig. 3. In addition to successfully completing routes, we also compare distance completion as in Fig. 4 and table 4. Our VPR pre-trained method consistently drives longer distances than the baseline. The baseline shows a substantial downfall in distances travelled when exposed to unseen environmental conditions, while our VPR pre-trained method shows relatively higher resistance to changes in town and weather conditions. This indicates that the VPR pre-trained method potentially holds higher resistance to co-variate shift [5] than the baseline as the performance is less affected by the change in data distribution.

We hypothesise that the main reason for these effects may be that the baseline method is pre-trained upon the task of image classification, which has no direct relation with the driving agent's data distribution. Whereas our method is pre-trained over a dataset which holds data from a potentially similar distribution (i.e., coming from the same simulated environments).

The VPR pre-trained method shows higher performance without any reliance on noisy unlabelled data by learning tasks that can be regarded as alternate tasks while the actual goal is to drive. Other methods [27], [28], [29] which use pre-training for the same goal, either directly or indirectly involve training on the task of interest during pre-training. We do not compare our results experimentally to the aforementioned methods as they are not directly being applied to autonomous driving or are tested over a benchmark that does not cover sufficient variability in towns and

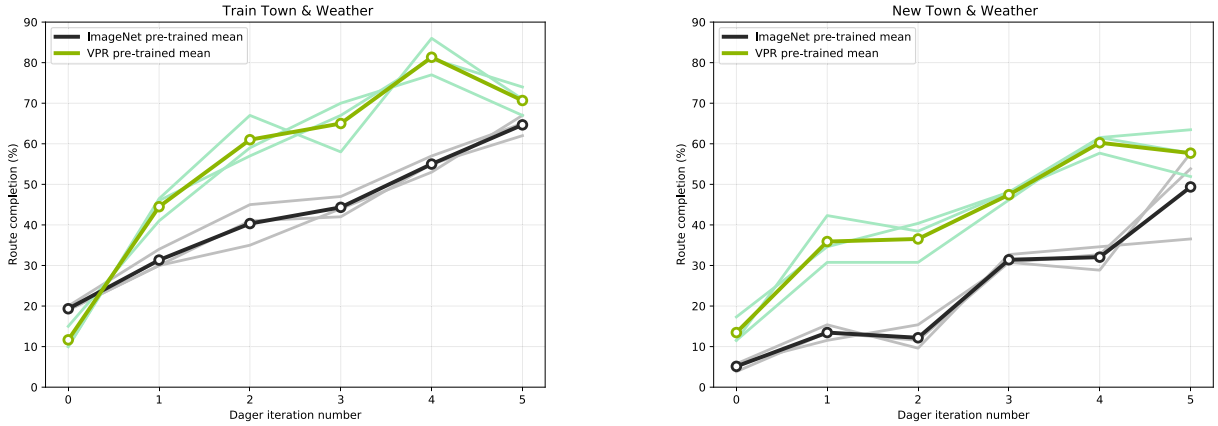


FIGURE 3. Route completion (%) of agents over the offline Leaderboard benchmark on training conditions (left) and testing conditions (right), evaluated three times over different seeds and plotted along with the average of performance.

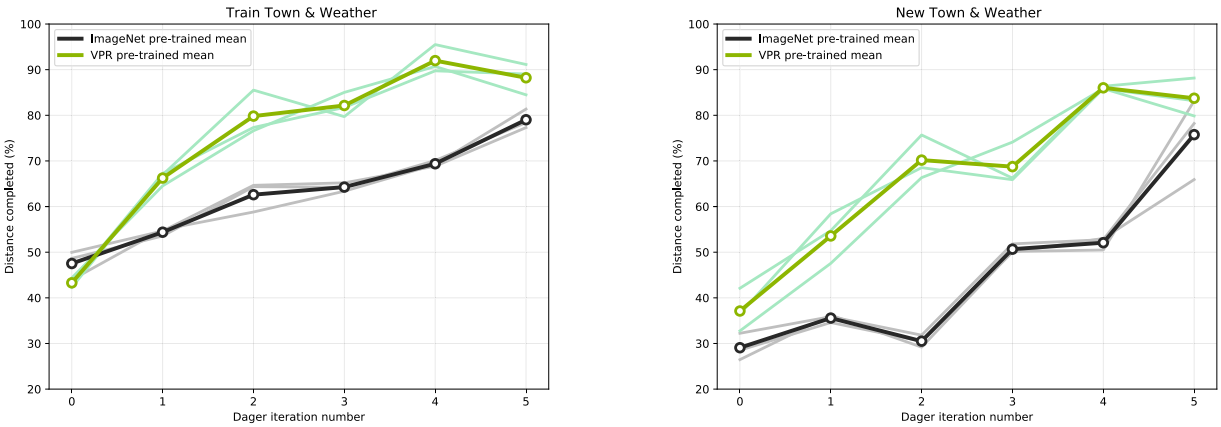


FIGURE 4. Distance completion (%) of agents over the offline Leaderboard benchmark on training conditions (left) and testing conditions (right), evaluated three times over different seeds and plotted along with the average of performance.

weather conditions. As the aforementioned results portray that learning the alternate task of VPR can be beneficial to the agent in learning to drive, it makes the potentially promising idea of combining our proposed pre-training approach with other recent works that leverage pre-training in different contexts.

V. CONCLUSION

We propose VPR-based pre-training step for autonomous driving agents trained with imitation learning, to overcome co-variate shift in terms of weather and resulting lighting changes. Our experiments show how pre-training with VPR helps in upholding the performance when the trained agent is exposed to unseen weather conditions in the Leaderboard benchmark. Such performance is seen in the form of a trend over multiple consecutive iterations of data aggregation and reruns with different seeds.

In future work, we plan to combine such pre-training with existing pre-training methods in the form of multi-task learning. We also plan to conduct an investigation of goal-directed visual navigation, using pre-trained road-following controllers (e.g., [32]). As VPR helps to cope with the changes in weather conditions, there can be further uncertainties that can arise and shift data distribution differently, therefore trying out more ways of pre-training may be a valuable area of exploration.

ACKNOWLEDGMENT

The authors would like to thank Neurotechnology, Vilnius University, and Vytautas Magnus University for providing partial support for this research and the anonymous reviewers for their constructive feedback.

REFERENCES

[1] J. Ziegler et al., "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Summer 2014.

[2] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1364–1384, Apr. 2022.

[3] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," 2023, *arXiv:2306.16927*.

[4] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Found. Trends Robot.*, vol. 7, nos. 1–2, pp. 1–179, 2018.

[5] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," *J. Mach. Learn. Res.*, vol. 15, pp. 627–635, Nov. 2010.
[6] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end simulated driving," in *Proc. 31st AAAI Conf. Artif. Intell. (AAAI)*. Palo Alto, CA, USA: AAAI Press, 2017, pp. 2891–2897.
[7] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger, "Exploring data aggregation in policy learning for vision-based urban autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11763–11773.
[8] T. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 1877–1901.
[9] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2661–2671.
[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[12] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3712–3722.
[13] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15222–15232.
[14] 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," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 20703–20716.
[15] V. Paolicelli, A. Tavera, C. Masone, G. Berton, and B. Caputo, "Learning semantics for visual place recognition through multi-scale attention," in *Image Analysis and Processing—ICIAP 2022*. Lecce, Italy: Springer, May 2022, pp. 454–466.
[16] D. A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Proc. 1st Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 1988, pp. 305–313.
[17] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Müller, J. Zhang, X. Zhang, and J. Zhao, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, pp. 1–6, Apr. 2016.
[18] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4693–4700.
[19] F. Codevilla, E. Santana, A. Lopez, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9329–9338.
[20] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "ST-P3: End-to-end vision-based autonomous driving via spatial–temporal feature learning," in *Computer Vision—ECCV 2022*. Tel Aviv, Israel: Springer, Oct. 2022, pp. 533–549.
[21] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," in *Proc. NeurIPS*, 2022, pp. 6119–6132.
[22] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "TransFuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12878–12895, Nov. 2023, doi: 10.1109/TPAMI.2022.3200245.
[23] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 17853–17862.
[24] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7310–7311.
[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[26] 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 *Computer Vision—ECCV 2014*. Zürich, Switzerland: Springer, Sep. 2014, pp. 740–755.
[27] 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," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 24639–24654.
[28] Q. Zhang, Z. Peng, and B. Zhou, "Learning to drive by watching YouTube videos: Action-conditioned contrastive policy pretraining," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 111–128.
[29] P. Wu, L. Chen, H. Li, X. Jia, J. Yan, and Y. Qiao, "Policy pre-training for autonomous driving via self-supervised geometric modeling," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 1–16.
[30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
[31] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.
[32] P. Daniušis, S. Juneja, L. Valatka, and L. Petkevičius, "Topological navigation graph framework," *Auto. Robots*, vol. 45, no. 5, pp. 633–646, May 2021.

SHUBHAM JUNEJA received the B.S. degree in computer engineering from Mumbai University, India, in 2014, and the M.S. degree in multimedia informatics from the Kaunas University of Technology, Lithuania, in 2016. He is currently pursuing the Ph.D. degree in autonomous navigation with Vilnius University.

Apart from researching autonomous navigation for robots and self-driving cars, he is also a Researcher with the Department of Neurotechnology, investigating AI-related methods for EEG-based brain–computer interfaces.

POVILAS DANIUŠIS received the Ph.D. degree in computer science from Vilnius University, in 2012. He is currently an Algorithm Engineer with the Department of Neurotechnology and a Senior Research Fellow with Vytautas Magnus University. His research interests include artificial general intelligence, data-driven robotics, causal inference, statistical dependence, and EEG-based brain–computer interfaces.

VIRGINIJUS MARCINKEVIČIUS received the B.S. degree in teacher of mathematics and informatics and the M.S. degree in mathematics from Vilnius University, Lithuania, in 2001 and 2003, respectively, and the Ph.D. degree in informatics from Vytautas Magnus University, in 2010.

Since 2000, he has been working in various research positions. Currently, he is a Senior Researcher and a Professor, the Head of the Intelligent Technologies Research Group, and the Head of the Artificial Intelligence Laboratory, Institute of Data Science and Digital Technologies, Vilnius University. His research interests include machine learning, information security, and natural language processing.