Learning to Drive by Watching YouTube Videos: Action-Conditioned Contrastive Policy Pretraining Supplementary Material

Qihang Zhang[1], Zhenghao Peng[1], and Bolei Zhou[2]

[1] The Chinese University of Hong Kong, Hong Kong SAR, China
[2] University of California, Los Angeles, USA

A YouTube Driving Dataset

Our YouTube Driving Dataset contains a massive amount of real-world driving frames with various conditions, from different weather, different regions, to diverse scene types. We collect 134 videos with a total length of over 120 hours, covering up to 68 cities. Figure 7 illustrates the videos' lengths distribution. For pretraining, we sample one frame every one second, resulting to a dataset of 1.30 million frames in total.

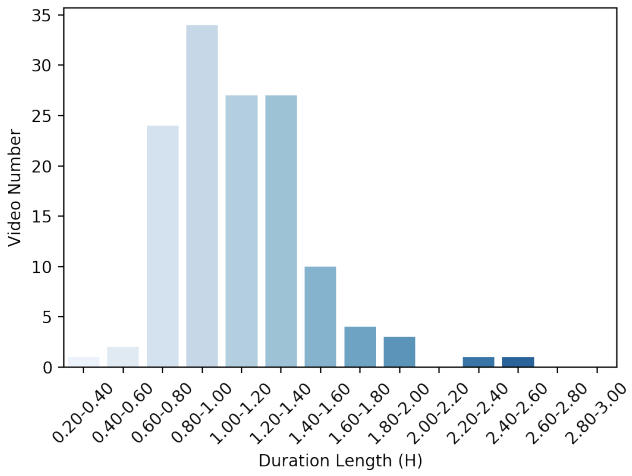

Fig. 7. Video length distribution

Table 5. Success Rate of Imitation Learning with frozen backbone

Methods	IL Demonstration Size ($\times 40K$)			
	10%	20%	40%	100%
Random-f	0.0±0.0	0.0±0.0	2.0±0.0	4.0±0.0
ImageNet-f	4.0±0.0	4.0±2.8	5.3±0.9	6.7±0.9
MoCo-f	10.7±2.5	30.7±4.1	38.7±1.9	54.7±6.6
ACO-f	**24.0**±1.6	**40.7**±3.4	**47.3**±0.9	**61.3**±3.8

Table 6. Effects of Batch Normalization in Imitation Learning. "bn" stands for Batch Normalization

Methods	IL Demonstration Size ($\times 40K$)			
	10%	20%	40%	100%
MoCo w/o bn	17.3±1.9	55.3±16.8	59.3±5.7	80.0±1.6
MoCo w/ bn	**19.3**±3.4	**60.3**±8.2	**76.0**±9.1	**80.7**±2.5
ACO w/o bn	27.3±3.8	34.0±4.3	72.0±4.3	78.0±4.3
ACO w/ bn	**30.7**±3.4	**66.0**±5.7	**82.0**±5.0	**96.0**±3.3

B Imitation Learning with frozen backbone

As shown in Table 5, we conduct imitation learning with frozen backbone. With backbone frozen, ACO demonstrates highest success rate. However, all methods suffer from severe performance drop compared to finetuned IL. For ImageNet and random pretrain, they drop by over 80% in full dataset size.

In contrast, RL methods exhibit less performance drop with frozen backbone. As shown in Fig. 5 in main paper, the biggest drop by frozen RL compared to finetuned RL is less than 50%. This phenomenon shows that freezing the backbone exacerbates the need for a good pretrain for IL. Policy trained with bad pretrain weight will give suboptimal actions, for example, driving the car to the road edge. IL, notoriously suffering from *distribution shift*, has no means to recover this error since the collected demonstrations do not contain any recovery action. Nevertheless, RL frequently visits dangerous states during exploration, and recovery strategies can hence be learned.

C Effects of Batch Normalization on ACO

As suggested in [3], Batch Normalization could help alleviate the problem caused by different distributions of contrastive-learning-based parameters and supervised-learning-based parameters. In this section, we compare the imitation learning performance on ACO and MoCo with and without batch normalization in downstream IL's fine-tuning.

As shown in Table 6, Batch Normalization is essential for both MoCo and ACO. Without adjusting parameters' distribution by batch normalization, these two contrastive learning methods show declining performance.

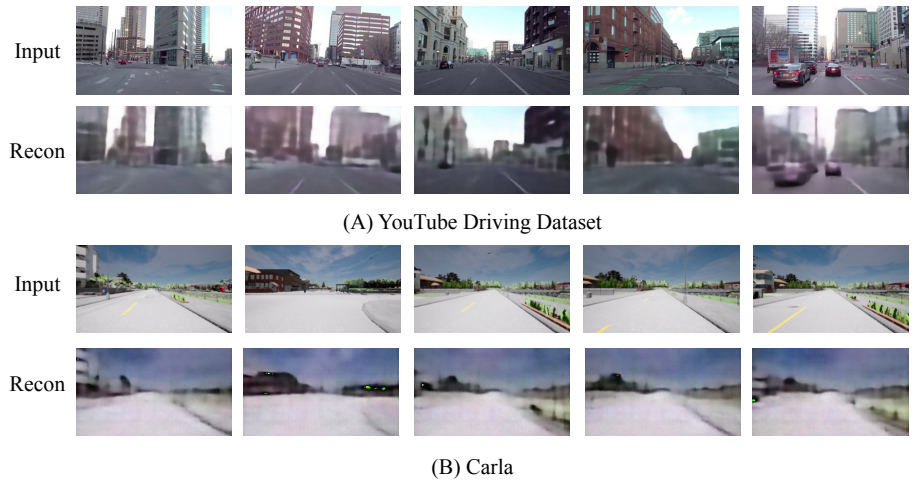

Fig. 8. Reconstructed scenes by AutoEncoder in YouTube and Carla domain

D Visualization of Auto-Encoder Baseline

To understand the poor performance of auto-encoder in all downstream tasks, we visualize the input and reconstructed image in both YouTube (training) and Carla (testing) domain in Figure 8.

It is obvious that much attention is paid to elements that are not relevant to driving decision-making in reconstructed images, such as buildings and the sky. On the contrary, the policy-relevant cues such as the lane lines and road edges are oversimplified. This means that useless information is learned in auto-encoder's feature and thus damages the performance of downstream tasks.

E Implementation Details and Hyper Parameters

We list the hyper-parameters used in ACO pretraining in Table 7, inverse dynamics model training in Table 8, imitation learning in Table 9, and reinforcement learning in Table 10.

For imitation learning, following CILRS [1], the training target is to optimize L1loss between predicted value and ground truth on steering, throttle, brake, and velocity:

$$\mathcal{L}_{il} = \lambda_s \mathcal{L}_s + \lambda_t \mathcal{L}_t + \lambda_b \mathcal{L}_b + \lambda_v \mathcal{L}_v, \tag{1}$$

where $\mathcal{L}_s$, $\mathcal{L}_t$, $\mathcal{L}_b$, and $\mathcal{L}_v$ stands for L1loss on steering, throttle, brake, and velocity respectively. The value of λ_s, λ_t, λ_b, and λ_v we used are listed in Table 9.

For reinforcement learning, we adopt the training pipeline in DI-drive [2][1].

Table 7. ACO

Hyper-parameter	Value
Initial learning rate	0.03
Weight decay	0.0001
Train epoch number	100
Train batch size	256
Key dictionary size	40960
λ_{ins}	1
λ_{act}	1
α for momentum update.	0.999
Hidden size of first layer in Projector	128
Hidden size of second layer in Projector	128
ϵ for action threshold	0.05

Table 8. Inverse dynamics model

Hyper-parameter	Value
Learning rate	0.0003
Weight decay	0.0001
Train epoch number	50
Train batch size	256

F Sampled frames of YouTube Driving Datasets

We provide sampled frames of five cities in Figure 9, 10, 11, 12, 13. Note that in captions we describe corresponding weather with blue text, region with red text, and scene style with green text. The columns are organized by different predicted steering values.

[1] DI-drive [2] is an open-source auto-driving platform, supporting reinforcement learning in Carla. The reward scheme can be found at: `https://github.com/opendilab/DI-drive/blob/main/core/envs/simple_carla_env.py`.

Fig. 9. Early morning drive at the center of Amsterdam, empty traffic on straight roads, with urban buildings surrounded

Fig. 10. Afternoon drive along the Italian Riviera, along narrow roads with frequent turnings

Fig. 11. Afternoon drive around the traditional Asakusa neighborhood, through rich traditional streets. Left-hand riding with middle density traffic

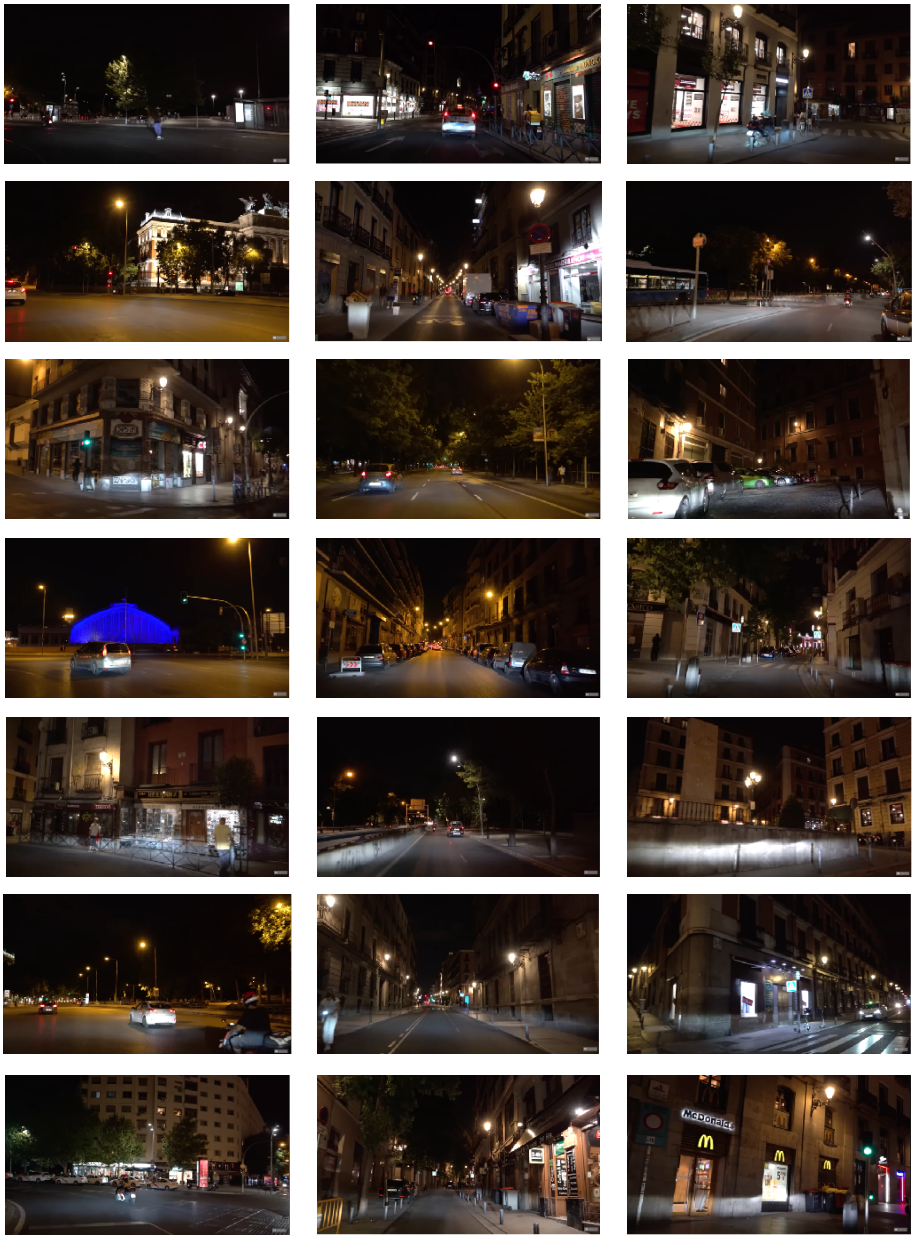

Fig. 12. Night drive in Madrid Spain, through Straight roads with middle density traffic

Fig. 13. Sunset drive in Lan Kwai Fong neighborhood, Hong Kong, middle traffic, modern building, left-hand, some one-way streets

Table 9. Imitation learning (CILRS [1])

Hyper-parameter	Value
Learning rate	0.0001
Weight decay	0.0001
Train epoch number	100
Train batch size	128
Velocity loss weight λ_v	0.05
Steering loss weight λ_s	0.5
Throttle loss weight λ_t	0.45
Brake loss weight λ_b	0.05

Table 10. Reinforcement learning (PPO [5])

Hyper-parameter	Value
Learning rate	0.0003
Weight decay	0.00001
SGD epoch number	6
λ for GAE [4]	0.95
Train batch size	1024
SGD mini batch size	256
Clip ratio	0.2

References

1. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9329–9338 (2019)
2. drive Contributors, D.: DI-drive: OpenDILab decision intelligence platform for autonomous driving simulation. `https://github.com/opendilab/DI-drive` (2021)
3. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
4. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438 (2015)
5. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)