Learning Offline Driving Policy with Decision Transformer in Latent Space

Bernard Lange
Department of Aeronautics & Astronautics
Stanford University
blange@stanford.edu

1 Abstract

In this project, we explore the combination of offline reinforcement learning (RL) and learned world models, specifically applied to the domain of autonomous driving. We tackle this tasks within the RL paradigm that learns from a range of experiences contained in a fixed-size dataset, without the need for a high-quality simulator.

Our project involves the development of a model-based RL framework. Learned world models have shown significant success in creating simplified state representations, providing ample training signals and facilitating interactions in the latent space of the model, a method that is less resource-intensive than conventional approaches.

Contrary to imitation learning which requires expert only dataset samples, we rely on non-expert datasets. We leverage the Decision Transformer, which has demonstrated robust performance in offline RL settings. Our aim is to learn a world model and an autonomous driving policy offline, using a generic driving dataset that encompasses a wide range of experiences. We strive to merge the insights from Model-Based Imitation Learning (MILE) and Decision Transformer to extend their applications to non-expert datasets.

We collected the offline dataset by training online RL policies, such as PPO and PPO-LSTM, on the CarRacing-v2 gym environment. We then trained a VAEGAN, demonstrating that our encoder and decoder can capture all necessary information to tackle the problem. Subsequently, we trained a Decision Transformer and compared its performance against behavioral cloning and online policies within the CarRacing-v2 gym environment.

Our approach delivers comparable results to the online RL methods and outperforms the behavioral cloning setup (see Table below). We believe there is still a room for further hyperparameter tuning within the current framework. As a future direction, we consider integrating latent imagination, which promises to be an interesting addition to enhance the capabilities of our model.

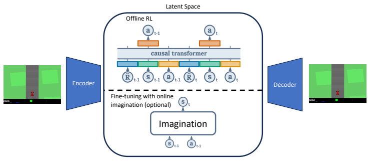

Figure 1: Decision Transformer in Latent Space.

Parameter	Episode Reward
PPO-LSTM 1 (online)	545.77 ± 127.44
PPO-LSTM 2 (online)	312.84 ± 73.85
PPO 1 (online)	257.77 ± 102.43
PPO 2 (online)	-93.73 ± 24.53
Behavioral Cloning (offline)	-52.69 ± 13.26
Latent Decision Transformer (offline)	432.97 ± 60.47

Figure 2: Evaluation on Car-Racing-v2

Preprint. Under review.

2 Introduction

The potential successful deployment of intelligent agents can revolutionize a multitude of application domains, including but not limited to transportation, warehouse management, and personal household aids. The development of these agents necessitates an accurate representation of the surrounding environment, a comprehension of how the state will evolve over time, and strategic long-term reasoning about future decisions that fulfill the desired objectives while adhering to predefined constraints. The latter is especially critical in safety-centric and high-consequence domains.

Reinforcement learning (RL) paradigm addresses these types of problems, where the agent interacts with the environment through receiving observations and potentially rewards, and responds with actions guided by its policy. RL frameworks can be classified based on their modeling assumptions and the accessibility of the simulation environment. Model-based RL enables explicit modeling of the environment, leveraging either expert knowledge or learning from experiences. Learned world models have seen considerable success, primarily because they create simplified and frequently disentangled state representations, generate more training signals compared to sparse and non-differentiable rewards, and facilitate interactions in the learned model's latent space, bypassing the need for computationally demanding and potentially unrealistic expert-designed simulators.

Nonetheless, in the majority of circumstances, we lack access to a high-quality simulator and must rely on a fixed-sized dataset of accumulated experiences. In imitation learning, a dataset consisting of expert trajectories is utilized to learn policies in a supervised manner. In contrast, offline RL derives policies from a fixed dataset with varying degrees of quality in the collected experiences. This setting, by nature, is considerably less restrictive and could facilitate the deployment of RL agents in a broader array of applications [21].

In this project, we explore the merger of offline RL and learned world models within the realm of autonomous driving. Hu et al. [13] demonstrated the feasibility of attaining a high-quality policy by fusing a model-based approach with imitation learning. We relax the prerequisite for expert datasets using Decision Transformers, which have proven to perform well in an offline RL setting [6].

3 Related Work

Latent Imagination: Ha and Schmidhuber [9] drew similarities from our cognitive system, i.e. how humans make decisions based on their internal model. In the World Model, they used a variational autoencoder [17] to compress an image observation and provided the learned latent representation to the recurrent predictive model. Information from both modules was used to maximize the expected cumulative reward on the desired task. Kim et al. [16] applied the World Model setup to neural network simulation for autonomous driving. Similarly, latent spaces have been used in a plethora of other planning and control approaches to learn latent dynamics from pixels [31, 4, 11, 8, 33], generate fully imagined trajectories [10], model multi-agent interactions [32], learn competitive policies through self-play [28], imagine goals in goal-conditioned policies [19, 1], meta-reinforcement learning [36], and offline reinforcement learning [37].

Offline-RL: In offline RL, agents are trained with a collected dataset of varying quality of experiences without any further interactions with a simulator or the environment. This leads to potential challenges such as lack of exploration, counterfactual reasoning, distribution shift and value overestimation [21]. Some of the issues are tackled by restricting the action space [7, 18] or incorporating value [7] or dynamics pessimism [34]. Interestingly, Decision Transformer frames an offline RL problem as a conditional sequence modelling that is capable of capturing high quality policies with appropriate reward querying and outperforms state-of-the-art offline RL baselines on numerous popular tasks [6].

Imitation Learning: Behavior cloning has been widely explored in many areas of research [12, 25, 14], including autonomous driving [22, 26, 3, 2]. Its core assumption, which is one of the main differentiating factors from offline RL, is the expert quality of the dataset used for supervised learning of the policy. Imitation learning policies often suffer from compounding error due to distribution differences between the training set and deployment. This issue has been tackled with DAgger [25], experience perturbation [2], and adversarial training [12]. In autonomous driving, behavior cloning has been used to acquire the policy for self-driving vehicles and to predict future trajectories of other agents [23, 5, 24, 29].

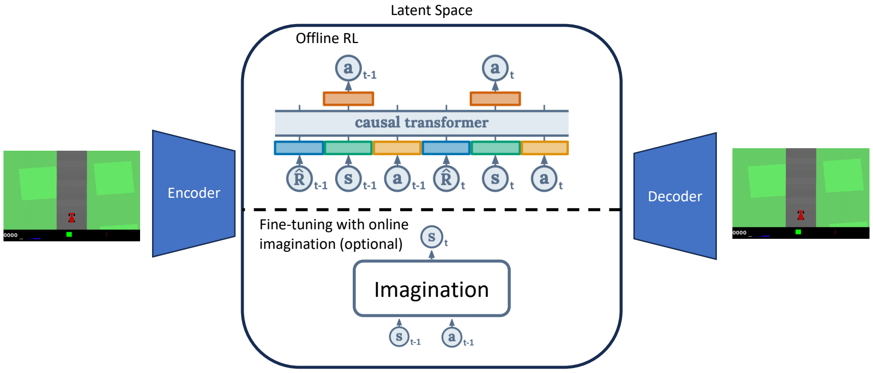

Figure 3: Decision Transformer in Latent Space.

4 Method:

In this project, we aim to learn a world model and a policy for autonomous driving in an offline manner, using a generic driving dataset that encompasses a range of experiences. Impressive results have been achieved in prior work with expert datasets using a Model-Based Imitation Learning (MILE) [13]. We endeavor to merge insights from MILE [13] and the Decision Transformer [6] frameworks to enable generalization to non-expert datasets. The Decision Transformer will model sequences in the latent space of the world model we've learned from the environment. We employ readily available Transformer networks [30] for sequence modeling and a VAEGAN [20] for representation learning. The offline dataset is created using a mixture of expertise-level policies trained in an online manner.

4.1 Data Collection for Offline RL

First, we must create our offline dataset, denoted as $\mathcal{D}$. To do this, we train several online policies, including PPO and PPO-LSTM [27], each with different initial seeds. After training, we sample a fixed number of rollouts from these trained policies to generate our offline dataset, $\mathcal{D}$.

4.2 Representation Learning

To reduce the dimensionality of our scene observations, we employ a convolutional autoencoder architecture denoted as $\mathcal{E}$ and $\mathcal{D}$. This architecture is trained using a combination of perceptual loss [35], Kullback-Leibler (KL) regularization [17], and patch-based adversarial losses [15], as shown in eq. (1). The encoder, $\mathcal{E}$, takes an input $x_t \in \mathbb{R}^{H \times W \times C}$ at timestep t and compresses it into a lower-dimensional representation $z_t \in \mathbb{R}^{h \times w \times c}$. Subsequently, the decoder, $\mathcal{D}$, reconstructs the compressed representation back into the original dimensional space, resulting in $\tilde{x}_t = \mathcal{D}(\mathcal{E}(x_t))$.

$$L_{\text{VAEGAN}} = \min_{\mathcal{E},\mathcal{D}} \max_{\psi} \left(L_{\text{rec}}(x, \mathcal{D}(\mathcal{E}(x))) - L_{\text{adv}}(\mathcal{D}(\mathcal{E}(x))) + \log D_{\psi}(x) + L_{\text{KL}}(x; \mathcal{E}, \mathcal{D}) \right) \quad (1)$$

Next, we convert all of the collected samples into latent space representations and store them as our latent offline dataset, $\mathcal{D}_{latent}$. We selected VAEGAN for this task due to its convenience. However, in future work, we should also consider utilizing VQVAE, a discrete adaptation of VAEGAN, as it has demonstrated superior performance when used in conjunction with the Transformer.

4.3 Decision Transformers in the Latent Space

Once we have obtained our latent dataset $\mathcal{D}_{latent}$, we proceed to train a Decision Transformer [6] that autoregressively models the trajectories in our dataset, as illustrated in fig. 3. A trajectory is represented as follows:

$$\tau = \left(\widehat{R}_1, s_1, a_1, \widehat{R}_2, s_2, a_2, \ldots, \widehat{R}_T, s_T, a_T \right) \quad (2)$$

where s signifies a state, a represents an action, and $\widehat{R}$ denotes returns to go, given by $\widehat{R}_t = \sum_{t'=t}^{T} r_{t'}$, with subscript t corresponding to the timestep. Aside from the RL-motivated representation, the remaining framework is aligned with a typical Transformer Decoder setup. The architecture we used incorporates a Transformer Decoder with causal attention layers to prevent the model from looking into the future during training. During the training phase, we sample minibatches of sequence length K. For each state, the network predicts the actions by optimizing the cross-entropy loss for discrete actions, and the mean-squared error for continuous ones. The states and 'rewards to go' are not predicted. At the evaluation phase, the 'reward to go' is set based on the desired performance, which decreases as the rollout progresses until it ends. The complete pseudocode for this process can be found in [6]. In the future, we could further explore the avenue of rollout in imagination or even finetuning in imagination.

5 Experiments

5.1 Experiment Setup

All experiments were conducted using an Nvidia TITAN RTX 24 GB, leveraging PyTorch and PyTorch-Lightning. Each of the online policies were trained until convergence (for 1 to 1.5 hours) using the Stable Baselines 3 framework. In total, we collected 303 episodes from all policies, each of length 1000 timesteps. The VAEGAN was trained for one day with a batch size of 64, and the Transformer was trained for a period of 4 hours (approx. 10 epochs). Our Decision Transformer in the latent space is compared with online RL policies and a behavioral cloning policy. We evaluated our framework on the CarRacing-V2 environment. This is a top-down racing environment with a randomly generated track. The observation is a 96x96 RGB image, with actions corresponding to steering, acceleration, and braking. These actions can be either continuous or discrete. The reward scheme is as follows: -0.1 is assigned at every timestep and +1000/N is given for each track tile visited. For the purpose of this task, we compressed the observations to 64x64 grayscale images.

5.2 Offline Dataset

We trained several online policies, such as PPO and PPO LSTM, with different initializations using the Stable Baselines 3 framework. The hyperparameters of the network are provided in table 1, and the performance of each policy is presented in table 2. The policies are then picked randomly and used to collect the offline dataset with stochastic action sampling and different seeds.

Table 1: Hyperparameters of PPO-LSTM.

Parameter	Value
batch_size	128
clip_range	0.2
ent_coef	0.0
obs_transform	obs_shape: 64, gray_scale_observation
frame_stack	2
gae_lambda	0.95
gamma	0.99
learning_rate	1e-4
max_grad_norm	0.5
n_envs	8
n_epochs	10
n_steps	512
n_timesteps	4000000.0
policy	CnnLstmPolicy
policy_kwargs	log_std_init=-2, ortho_init=False, enable_critic_lstm=False, activation_fn=nn.GELU, lstm_hidden_size=128
sde_sample_freq	4
use_sde	True
vf_coef	0.5
normalize_kwargs	norm_obs: False, norm_reward: False

5.3 Representation Learning

We illustrate examples of reconstructed observations and samples from prior in fig. 4 and fig. 5, respectively. As can be observed, the network captures all the essential details necessary to address the task.

Table 2: Hyperparameters of the policy used for offline dataset collection with different initialization or stopping criteria. We added a single badly tuned policy (PPO 2).

Parameter	Episode Reward
PPO-LSTM 1	545.77 ± 127.44
PPO-LSTM 2	312.84 ± 73.85
PPO 1	257.77 ± 102.43
PPO 2	-93.73 ± 24.53

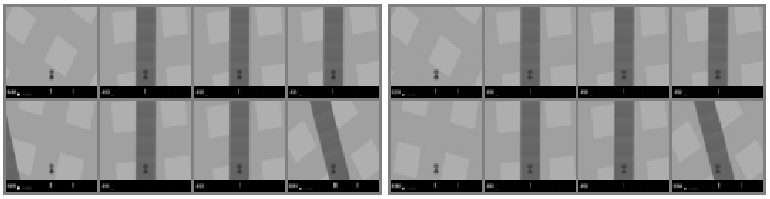

Figure 4: VAEGAN on CarRacing-v2. Left: Ground truth. Right: Reconstructions.

5.4 Decision Transformer in the Latent Space

Subsequently, we evaluate the Decision Transformer in table 3 in the latent space of the VAEGAN and compare it with online policies trained for the dataset collection and behavioral cloning. Our approach gets close to the performance of online methods and it outperforms the behavioral cloning. The latter should be expected, as for the behavior cloning policy to work, it would need to have a expert only dataset. In our case, some offline samples can be very suboptimal.

Table 3: Evaluation on Car-Racing-v2.

Parameter	Episode Reward
PPO-LSTM 1 (online)	545.77 ± 127.44
PPO-LSTM 2 (online)	312.84 ± 73.85
PPO 1 (online)	257.77 ± 102.43
PPO 2 (online)	-93.73 ± 24.53
Behavioral Cloning (offline)	-52.69 ± 13.26
Latent Decision Transformer (offline)	432.97 ± 60.47

6 Conclusion

In this project, we investigated the integration of offline RL and learned world models in the context of autonomous driving, evaluating the performance on a task with a non-expert dataset. We initially generated our offline dataset using online policies with a diverse range of capabilities. Subsequently, we trained a VAEGAN, demonstrating that our encoder and decoder can effectively capture all the necessary information to tackle the problem.

Following this, we trained a Decision Transformer and compared its performance with that of behavioral cloning and online policies on the Car-Racing-v2 gym environment. Our framework delivered comparable results, outperforming the behavioral cloning setup. However, we believe that the current framework could be further refined to achieve superior performance. As a future consideration, the integration of latent imagination could prove interesting.

References

[1] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm. Unsupervised state representation learning in atari. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[2] M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

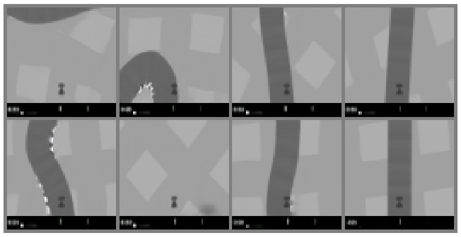

Figure 5: VAEGAN on CarRacing-v2. Samples from prior.

[3] E. Bronstein, M. Palatucci, D. Notz, B. White, A. Kuefler, Y. Lu, S. Paul, P. Nikdel, P. Mougin, H. Chen, et al. Hierarchical model-based imitation learning for planning in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8652–8659. IEEE, 2022.

[4] L. Buesing, T. Weber, S. Racaniere, S. Eslami, D. Rezende, D. P. Reichert, F. Viola, F. Besse, K. Gregor, D. Hassabis, and D. Wierstra. Learning and querying fast generative models for reinforcement learning. *arXiv*, 2018.

[5] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Conference on Robot Learning (CoRL)*, 2020.

[6] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

[7] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[8] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning (ICML)*, pages 2170–2179. PMLR, 2019.

[9] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[10] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2019.

[11] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019.

[12] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[13] 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. *Advances in Neural Information Processing Systems*, 35:20703–20716, 2022.

[14] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[16] S. W. Kim, J. Philion, A. Torralba, and S. Fidler. DriveGAN: Towards a controllable high-quality neural simulation. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5820–5829, 2021.

[17] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[18] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[19] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel. Learning plannable representations with causal InfoGAN. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[20] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning (ICML)*, pages 1558–1566. PMLR, 2016.

[21] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[22] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, B. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. *arXiv preprint arXiv:2212.11419*, 2022.

[23] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. *arXiv preprint arXiv:2207.05844*, 2022.

[24] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. J. Weiss, B. Sapp, Z. Chen, and J. Shlens. Scene Transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations (ICLR)*, 2021.

[25] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[26] O. Scheel, L. Bergamini, M. Wolczyk, B. Osiński, and P. Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Conference on Robot Learning*, pages 718–728. PMLR, 2022.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[28] W. Schwarting, T. Seyde, I. Gilitschenski, L. Liebenwein, R. Sander, S. Karaman, and D. Rus. Deep latent competition: Learning to race using visual control policies in latent space. In *Conference on Robot Learning (CoRL)*. PMLR, 2021.

[29] S. Shi, L. Jiang, D. Dai, and B. Schiele. Motion transformer with global intention localization and local movement refinement. 2022.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[31] N. Wahlström, T. B. Schön, and M. P. Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv*, 2015.

[32] A. Xie, D. P. Losey, R. Tolsma, C. Finn, and D. Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on Robot Learning (CoRL)*. PMLR, 2021.

[33] Z. Xu, Z. He, J. Wu, and S. Song. Learning 3D dynamic scene representations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2020.

[34] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142, 2020.

[35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[36] T. Z. Zhao, A. Nagabandi, K. Rakelly, C. Finn, and S. Levine. MELD: Meta-reinforcement learning from images via latent state models. In *Conference on Robot Learning (CoRL)*. PMLR, 2021.

[37] W. Zhou, S. Bajracharya, and D. Held. PLAS: Latent action space for offline reinforcement learning. In *Conference on Robot Learning (CoRL)*. PMLR, 2021.