Improving Autonomous Driving Policy Generalization via Neural Network Over-Parameterization*

Hemanth Manjunatha [1] **Andrey Pak** [1] **Panagiotis Tsiotras** [1]

Abstract

With the development of deep representation learning, reinforcement learning (RL) has become a powerful framework for automated driving tasks capable of learning complex policies in high-dimensional environments. However, one of the most critical criteria to deploy learned policies in real-world tasks is generalizing to unseen situations at deployment time while avoiding overfitting to the training environments. Studying this is vital if we use the RL algorithms for decision-making in real-world scenarios, where the environment will be diverse, dynamic, and unpredictable, like autonomous driving. In this work, we propose a novel architecture that combines different information about the driving conditions and the environment to inform the RL agent in the form of latent vectors. Although the use of latent representations for RL is quite common, the use of different latent representations and selectively combining them using self-attention, as proposed in this work, is novel. The basic premise here is that different latent representations provide parallel, complementary pathways that parameterize the actor/critic RL network. In essence, while the value (critic) network tries to estimate $Q_\theta(x, a)$ where (x, a) is the state-action pair, our network will try to estimate $Q_\theta(\ell, \eta, a)$ where ℓ and η are additional latent variables representing different aspects of driving. Preliminary results show the efficacy of using different latent vectors and combining them in a structured manner.

1. Motivation and Objective

Reinforcement Learning (RL) has been widely used for decision-making with great success over the past 20 years. Recently, it is making its way to many real-world applications. The domain of autonomous vehicles is one such application where RL has been implemented with promising results. State-of-the-art algorithms such as Deep Deterministic Policy Gradient (DDPG) and Soft Actor-Critic (SAC) are being applied to complex tasks such as robot control and autonomous driving (Arulkumaran et al., 2017). The greatest success of RL algorithms has been mainly in game environments where the failure to make good decisions does not usually incur a high cost. However, the same cannot be said for autonomous vehicles, where the decision's generalizability and robustness are paramount even in anomalous situations. By the latter, we mean the ability of the system/agent to perform reasonably well even in cases of failures or unforeseeable situations that have never been encountered before. In this vein, a significant body of work has accumulated (Novak et al., 2018; Neyshabur et al., 2018) in the domain of supervised learning (e.g., MNIST, CIFAR10) studying generalization in deep learning networks. However, the generalizability is less explored (Packer et al., 2018; Cobbe et al., 2019) in reinforcement learning, particularly for autonomous driving. In this regard, this work explores a popular empirical observation that suggests that overparameterized neural networks improve both optimization and generalization (Allen-Zhu et al., 2019), in the context of autonomous driving.

Deep learning (DL) has seen tremendous success in computer vision, speech recognition, natural language processing, audio recognition, and bioinformatics (Dong et al., 2021). Surprisingly, even though the number of parameters in Deep Learning (DL) models is significantly more than the number of training examples, the models exhibit good generalization and lower error rate in test cases (Zhang et al., 2021; Neyshabur et al., 2014). This empirical observation appears to contradict the traditional learning theory (Arora et al., 2018). Consequently, a significant amount of effort has been devoted to investigating the theoretical properties of deep learning models (Arora et al., 2018; Nagarajan, 2021). Recently, Casper et al. (Casper et al., 2019)

*Work in progress [1]School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-1050, USA. Correspondence to: Hemanth Manjunatha <hmanjunatha6@gatech.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

explored the emergence of prunable and redundant units in relation to the generalization ability of deep neural networks. The authors observed that the prunable and redundant units grow at a rate outpacing the model size.

Inspired by these works that relate overparameterization and deep learning, we explore whether overparameterization and redundancy can also improve the robustness in a reinforcement learning setting. This is particularly interesting because most of the analysis on generalization is explored in the supervised domain. At the same time, there is no direct way to convert an RL problem into a supervised problem. Our work differs crucially from previous works (Zhang et al., 2021; Arora et al., 2018; Casper et al., 2019), which primarily focus on the overparameterization by increasing the size of the learnable parameters within the network; here, the overparameterization arises, instead, in the form of redundancy in a structured manner. In addition, we also propose a self-attention mechanism to combine different information effectively.

2. Method

To study the effect of overparameterization in a structured way, we introduce a novel architecture with self-attention that explicitly combines redundant information, as shown in Fig. 1, leading to overparameterization. The architecture consists of five sub-networks: *base network, future latent vector prediction network, action prediction network*, and *value network*. The implementation details of each network are discussed in the subsequent text. The future latent vector prediction network and the base network are trained offline, and the weights are frozen for RL training. All the sub-networks receive a stack of 4 images ($x_{t-3} \dots x_t$) at time instance t. The base network compresses the input images to an embedding η_t through convolution operations. The future latent vector prediction network predicts the embedding ℓ_{t+1}, which signifies the future information. The structure of the future latent vector prediction network is inspired by the World Models (WM) architecture (Ha & Schmidhuber, 2018). The rationale for incorporating future latent vectors is to inform the policy network about the influence of present action and serve as a predictive model. A self-attention mechanism is used to combine the two embeddings (ℓ_{t+1}, and η_t) together to form a feature vector which is used by the action network and value network. Note that in the proposed architecture, the sub-networks can learn similar tasks providing redundant information for decision making.

Further, to support goal-oriented actions, we introduce a gating unit that selects different branches of the action network depending on the global routing commands (Liang et al., 2018). The sub-networks in the action network are regular multi-layer networks whose output are continuous throttle, steer, and brake commands. Similarly, the value network receives the augmented feature vector, action, and the global planner command to provide the value approximation.

2.1. Base Network

The base network consists of a general convolution neural network architecture without any fully-connected layers. Given the expert driving action h_t at the state x_t and the routing command c_t, we can pre-train the policy network shown in Figure 1 via basic imitation learning (Liang et al., 2018) to mimic an expert driver. Note that, for imitation learning, only the *base network* in Figure 1 will be trained. The future latent vector prediction network will be trained separately and augmented with the base network for reinforcement learning. The control command c_t is introduced to handle goal-oriented behavior starting from an initial location to reach the final destination (similar to (Liang et al., 2018)). The command c_t is a categorical variable that controls the selective branch activation via the gating function $G(c_t)$, where c_t can be one of four different commands, i.e., follow the lane (Follow), drive straight at the next intersection (Straight), turn left at the next intersection (TurnLeft), and turn right at the next intersection (TurnRight). Four policy branches are specifically learned to encode the distinct hidden knowledge for each case and thus selectively used for action prediction. For loss function, we can directly use mean squared error between predicted and expert actions given as:

$$\mathcal{L}(\hat{h}_t, h_t) = \|\hat{s}_t - s_t\|^2 + \|\hat{a}_t - a_t\|^2 + \|\hat{b}_t - b_t\|^2, \quad (1)$$

where the loss function $\mathcal{L}$ is the weighted (here for simplicity taken as 1) summations of L2 losses for three predicted actions $\hat{h}_t$. The base network serves two purposes: a) a good initial policy (compared to random initialization) for warm starting learning; and b) the notion of re-training the pre-trained model in a reinforcement learning setting can be considered analogous to residual policy learning (Silver et al., 2018). Here, we start with a fixed policy and then learn a residual policy to modify the fixed policy for a more complex situation.

2.2. Future Latent Vector Prediction Network

We propose **C**ombined dyn**A**mic autoencode**R** **Net**work (CARNet), a novel DNN network, to learn the environment dynamics/world model from expert demonstrations (Figure 2). By "environment" here, we mean a succinct description of the interaction between the autonomous (host) vehicle and the surrounding vehicles in traffic. The proposed combined architecture for future latent vector prediction is shown in Figure 2. Given a history of states $x_0, \dots, x_n$, their latent representations $\ell_0, \dots, \ell_n$, and the corresponding sensor data $s_0, \dots, s_n$ (e.g., radar or Lidar scans, images from on-board cameras, IMU measure-

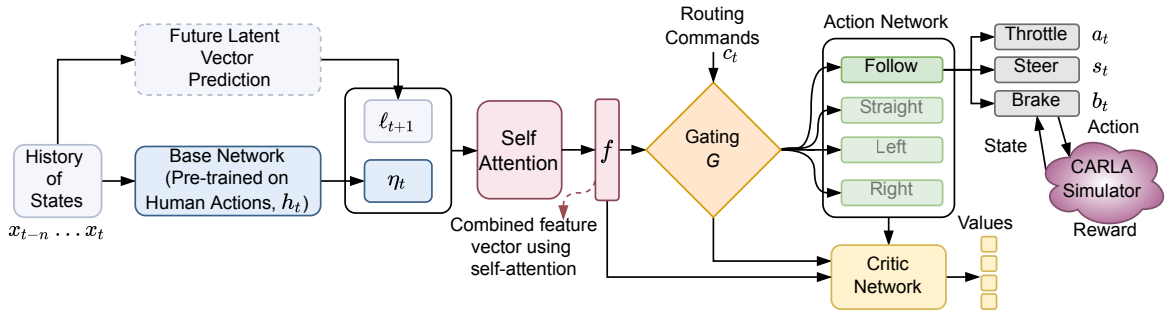

Figure 1. The architecture consists of four sub-networks: base network, future latent vector prediction network, action prediction network, and value network. The embeddings from the base network and future latent vector prediction network are augmented together with self-attention to form the input for the action prediction network. The network uses the routing commands from a global path planner to select a sub-network in action prediction. The output of these sub-networks is the throttle, steer, and brake.

ments, etc.), we wish to predict x_{n+1}, ℓ_{n+1} and s_{n+1}. In other words, the network learns the probability distribution $\mathbb{P}\left(x_{n+1}, \ell_{n+1}, s_{n+1} \mid x_{0:n}, \ell_{0:n}, s_{0:n}\right)$. Here, instead of the sensor readings $s_{0:n}$, we can also use the human-driver actions $a_{0:n}$.

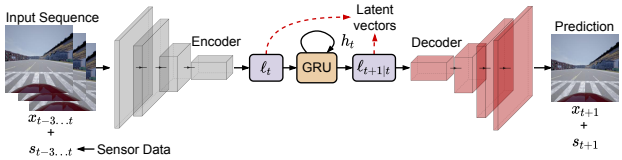

Figure 2. Future latent vector prediction network. A sequence of traffic states is first encoded into latent space, propagated through the recurrent network, and outputs the estimate of the latent space at time step t_{n+1}.

As shown in Figure 2, the state x_i is encoded into the latent space $\ell_i = E(x_i)$, where E is the encoder. Note that the encoder/decoder can be any architecture of a typical convolutional neural network or a recurrent neural network. The encoded vector ℓ_i is used as an input for the gated recurrent unit (GRU) block, whose output is the next time step latent space vector $\ell_{i+1|i}$. The predicted latent state $\ell_{i+1|i}$ is used as a hidden state for the next time step, as well as an input for reconstructing the next time step state $x_{i+1} = D(\ell_{i+1|i})$, where D is a decoder. Thus, the predicted latent vector ℓ_{i+1} at time step $i+1$ not only depends on ℓ_i, but also on ℓ_{i-1}. The overall loss function for the single-step prediction can be broken down into three terms given by equation (2) below

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_2(x_{0:n+1}, \hat{x}_{0:n+1})}_{\text{State Prediction}} + \underbrace{\mathcal{L}_2(\ell_{n+1|n}, \ell_n)}_{\text{Latent prediction}} + \underbrace{\mathcal{L}_2(s_{n+1|n}, s_n)}_{\text{Sensor prediction}}, \tag{2}$$

where ℓ_n, $\ell_{n+1|n}$ denote the encoded latent variable at step n and the conditioned latent prediction one step further, x_n, $\hat{x}_n$ are the source and reconstructed states, respectively, and s_n, $s_{n+1|n}$ are the current and predicted sensor measurements. Note that if the states include images, we can use image-relevant loss functions (Wang et al., 2003). Additionally, sharing reduces the number of trainable parameters, thus facilitating faster training times while using less computational resources.

3. Preliminary Results and Discussion

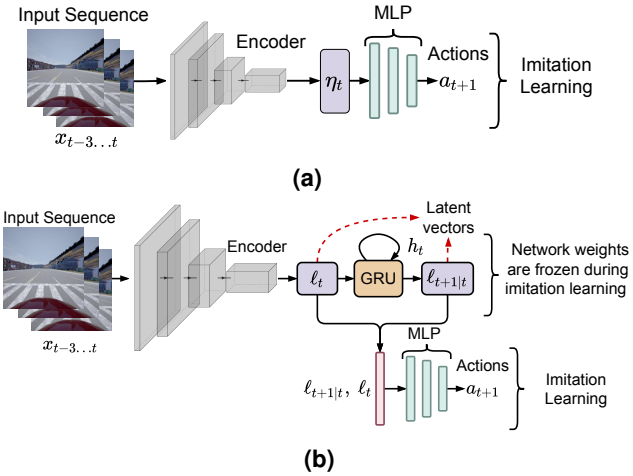

Figure 3. (a) Imitation learning using the base. (b) Imitation learning using the latent vector predicted by CARNet. Note for imitation learning using CARNet the weights are frozen.

This section provides preliminary results of two sub-networks: CARNet and *base network*. We validate the efficacy of the proposed models through imitation learning. In the experiments, we use the CARLA (Dosovitskiy et al., 2017) simulation environment. For imitation learning, given N expert driving sequences $ED_i, i \in (1, \dots, N)$ with corresponding observation frames $I_{i,t}$, we learn a deterministic policy using a network parameterized by θ to mimic the expert actions. The action space $A_{i,t}$ contains two continuous actions: steering angle and acceleration.

However, we discretized the continuous values into three levels for imitation learning. The steering angle assumes the values $[-0.2, 0.0, 0.2]$ rad and acceleration assumes the values $[-3, 0.0, 3]$ $\mathrm{m/s^2}$ values. Imitation learning using future latent vectors is set up as a classification problem with 9 classes resulting from different combinations of the previous discrete steering and acceleration values. Discretization of the action space is justifiable from a practical perspective since in autonomous driving applications the high-level decision policy derived by imitation learning, reinforcement learning, game theory, or some other decision-making techniques is combined with low-level trajectory planning and motion control algorithms (Nageshrao et al., 2019). For training, we have used mean cross-entropy loss between the predicted class and the autopilot class. The training was performed on a single machine equipped with a GeForce RTX3090 GPU, Ryzen 5950x CPU, and 16GB of RAM.

A total of 500K time steps were generated using random roll-outs of the internal CARLA vehicle autopilot module. Along with simulated data, we also explored the performance of CARNet in real-world data. For the real-world evaluation, we have used the Udacity dataset (Hou et al., 2019).

Finally, to show the efficacy of the ensemble model, we explored a linear combination of predictions from *base network* and the CARNet as shown in Fig. 4. For the combination, the predictions probabilities are weighted equally. Note that this step is only for validation of the performance of the ensemble model when compared with the sub-networks. However, the overall architecture involves training of the network with self-attention, as shown in Fig. 1.

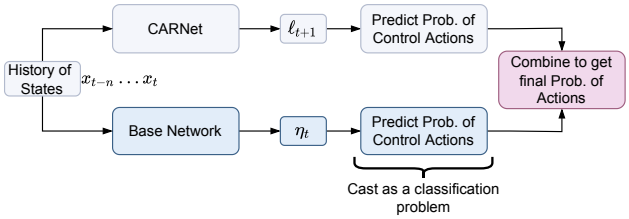

Figure 4. Experiment setup to validate the efficacy of ensemble model.

The imitation classification results with CARNet and the baseline model (World Models (WM) (Ha & Schmidhuber, 2018)) are shown in Table 1. The results show that the proposed model performs better than the baseline model while having a significantly less number of trainable parameters.

The proposed CARNet showed (Table 2) an even better performance margin on real-world data (Udacity dataset). While the overall performance accuracies were less than the ones for the simulated dataset. A performance drop on the real dataset is expected since real data is more diverse, e.g., texture, shadows, camera over/under-exposure, etc. Additionally, since the minimum possible configuration of the network was explored, the autoencoder might lack the capacity to reconstruct real images compared to simulation quality data.

Table 1. Imitation learning results using predicted future latent vectors.

Model	Classification Accuracy%
World Models	71.37 ± 0.52
CARNet	$\mathbf{77.08 \pm 0.51}$

Table 2. Imitation learning results comparison (Udacity Dataset).

Model	Classification Accuracy%
World Models	45.92 ± 1.66
CARNet	$\mathbf{59.80 \pm 1.10}$

Table 3. Imitation learning results using predicted future latent vectors.

Model	Classification Accuracy%
CARNet	77.08
Base Network	80.55
Combined Model	**81.68**

Table 3 shows the preliminary results of the combined model. Clearly, the ensemble model performs better than sub-networks.

4. Conclusion and Future Work

We propose a neural network architecture to study the effect of the generalization of over-parameterized neural networks in autonomous driving tasks. The reasoning behind selecting the proposed architecture is to introduce over-parameterization in a structured way rather than follow a "black-box" end-to-end learning approach. We achieve this by augmenting three different sub-networks which capture distinct driving characteristics. We highlight the efficacy of the proposed idea. Namely, it is shown that the combined neural network provides better performance than the individual sub-networks and the use of latent vectors captures well different aspects of driving. A naive combination of sub-networks as ensemble classifiers results in better agent action classification accuracy when compared with the individual sub-networks. More sophisticated combinations are currently under investigation and are expected to result in even better performance. In addition, more ablation experiments are needed to further validate the effectiveness of the proposed network.

References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 2019.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pp. 254–263. PMLR, 2018.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

Casper, S., Boix, X., D'Amario, V., Guo, L., Schrimpf, M., Vinken, K., and Kreiman, G. Frivolous units: Wider networks are not really that wide. *arXiv preprint arXiv:1912.04783*, 2019.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Dong, S., Wang, P., and Abbas, K. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16. PMLR, 2017.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hou, Y., Ma, Z., Liu, C., and Loy, C. C. Learning to Steer by Mimicking Features from Heterogeneous Auxiliary Networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33018433. URL https://doi.org/10.1609/aaai.v33i01.33018433.

Liang, X., Wang, T., Yang, L., and Xing, E. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 584–599, 2018.

Nagarajan, V. Explaining generalization in deep learning: progress and fundamental limits. *arXiv preprint arXiv:2110.08922*, 2021.

Nageshrao, S., Tseng, H. E., and Filev, D. Autonomous highway driving using deep reinforcement learning. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 2326–2331. IEEE, 2019.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., and Song, D. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Silver, T., Allen, K., Tenenbaum, J., and Kaelbling, L. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

Wang, Z., Simoncelli, E. P., and Bovik, A. C. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402. Ieee, 2003.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.