International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 15, Number 2, pages 138–152, July 2025

Enhancing Dimensionality Reduction in Driving Behavior Learning: Integrating SENet with VAE

Yuta Uehara
Graduate School of Science and Engineering
Saga University, Saga, Japan

Susumu Matsumae
Graduate School of Science and Engineering
Saga University, Saga, Japan

Received: February 15, 2025
Accepted: April 5, 2025
Communicated by Takashi Yokota

Abstract

This study addresses a common limitation of conventional Variational Autoencoder (VAE)-based methods in dimensionality reduction for state representation learning, especially in autonomous driving, by integrating Squeeze-and-Excitation Networks (SENet) into the VAE framework. While traditional VAE approaches effectively handle high-dimensional data with reduced computational costs, they often struggle to adequately capture complex features in certain tasks. To overcome this challenge, we propose the SENet-VAE model, which incorporates SENet into the VAE architecture, and evaluate its performance in driving behavior learning using deep reinforcement learning. Our experiments compare three setups: raw image data, conventional VAE, and SENet-VAE. Furthermore, we examine how the placement and number of SE-Blocks affect performance. The results demonstrate that SENet-VAE surpasses the limitations of conventional VAE and achieves superior accuracy in learning. This work highlights the potential of SENet-VAE as a robust dimensionality reduction solution for state representation learning.

Keywords: Squeeze-and-excitation network, Variational autoencoder, Dimensionality reduction, Autonomous driving, Deep reinforcement learning

1 Introduction

Driving behavior learning refers to the process by which an autonomous vehicle acquires the ability to execute driving maneuvers using machine learning techniques. This process relies on the vehicle's interaction with its environment and its decision-making abilities. By analyzing a wide array of data collected from sensors, cameras, and other onboard devices, the vehicle learns to navigate safely and efficiently toward its destination. Among various approaches, Deep Reinforcement Learning (DRL) has emerged as one of the most effective methods for learning driving behaviors [1,2,3]. However, DRL's effectiveness comes at the cost of substantial training time and computational resources. To address these challenges, researchers have increasingly adopted dimensionality reduction techniques, which reduce computational costs while preserving essential information. This approach, commonly referred to as state representation learning, plays a crucial role in enhancing the efficiency of driving behavior learning.

One widely adopted approach for state representation learning is the Variational Autoencoder (VAE) [4,5,6,7]. The study by [7] explores the application of VAE for dimensionality reduction in the context of driving behavior learning, utilizing the Proximal Policy Optimization (PPO) algorithm [8] within a deep reinforcement learning framework. Experimental evaluations conducted in a simulated driving environment demonstrated that VAE-based dimensionality reduction achieved performance that was comparable to, or in certain cases superior to, the use of raw image data. However, there were instances where the performance of the VAE-based approach closely matched that of directly processing raw images. These observations highlight potential limitations in the representational capacity of the latent space generated by conventional VAE models, suggesting that further enhancements to the model architecture may be required to fully capture complex features.

To enhance the performance of Variational Autoencoder (VAE), we proposed SENet-VAE, which integrates Squeeze-and-Excitation Networks (SENet) [9] into the VAE framework [10]. SENet introduces an attention mechanism through SE-Block, dynamically emphasizing important features and thereby improving the representational capacity of the model. Our research demonstrated that incorporating SE-Block into the encoder of the VAE leads to improved dimensionality reduction performance in certain cases within the autonomous driving learning process. In this paper, we investigate the conditions under which SENet-VAE is effective, as well as the impact of SE-Block placement and the addition of multiple SE-Blocks on performance improvement.

2 Related Works

The utilization of Variational Autoencoders (VAEs) for state representation in driving behavior learning was first proposed in [5]. By employing VAE-based dimensionality reduction, the learning process proved more efficient, enabling the model to converge within fewer training episodes compared to non-VAE approaches. These findings underscore the efficacy of VAE in mitigating the complexity of high-dimensional input data, thereby offering substantial computational advantages.

In [7], the degree to which image dimensionality could be reduced while preserving the ability to learn driving behaviors was examined. This study compared training duration and memory utilization between models employing raw image inputs and those incorporating VAE-based dimensionality reduction. The experiments were conducted on a looped course within a deep reinforcement learning (DRL) framework, utilizing the PPO algorithm and the Donkey Simulator [11]. The findings indicate that VAE-based dimensionality reduction yields outcomes comparable to or exceeding those achieved with raw image inputs, with no significant performance differences observed when varying the latent space dimensionality among 32, 64, and 128. Moreover, the study reports a reduction in training time by up to 50% and a decrease in memory usage by up to 30%. While these results underscore the substantial efficiency gains afforded by VAE, they also highlight inherent limitations in its representational capacity for state learning, particularly when addressing complex input data.

In our work [10], we considered enhancing performance by incorporating SENet [9], which introduces an attention mechanism to weight the feature maps output by conventional CNN convolutional layers, into our VAE framework. SENet has gained significant attention, winning the 2017 ILSVRC with a top 5 error rate of 2.25 and being widely adopted in image recognition studies.

Several studies have achieved performance improvements in VAEs by incorporating SENet. In [12], a proposal was made to replace traditional hand-crafted features with VAE for loop closure detection. SENet was incorporated into VAE to improve its performance, and modifications to the loss function, as well as the introduction of hyperparameters to the KL divergence term of the objective function, were made. The proposed method maintained higher accuracy compared to conventional methods. In [13], VAE was applied to 3D reconstruction of porous media. Traditional physical experiment methods and numerical reconstruction methods faced various issues, and a VAE model based on SENet and fixed-point normalization was proposed. The proposed method demonstrated effectiveness and practicality compared to traditional methods. Building upon our work [7,10], we have employed a SENet-VAE, which integrates SENet into a VAE, to perform dimensionality reduction for state representation learning in autonomous driving. In some instances, this approach proved effective [10]. However, in [10], experiments were conducted using only a single

SE-Block fixed in the encoder. As a result, no data are available regarding configurations with multiple SE-Blocks or alternative SE-Block placements. Accordingly, this paper aims to investigate which configuration is most effective.

3 Proposed Method

This section outlines the proposed method, beginning with an explanation of the underlying concepts of VAE and SENet, followed by a detailed description of the SENet-VAE model proposed in this paper.

3.1 Variational Autoencoder (VAE)

Variational Autoencoder (VAE), proposed by Kingma et al. [4], is a deep learning framework combined with a probabilistic generative model, capable of encoding high-dimensional data into a compact latent representation. The VAE consists of three key components:

- Encoder: Maps the input data to the latent space and calculates the mean and standard deviation parameters of the Gaussian distribution associated with each data point in the latent space.
- Latent Space: Represents the potential information of each data point, obtained by random sampling based on the encoder's output values.
- Decoder: Takes the sampled points from the latent space and projects them back into the original input data space, aiming to faithfully reconstruct the original input data.

Figure 1 illustrates the composition of VAE.

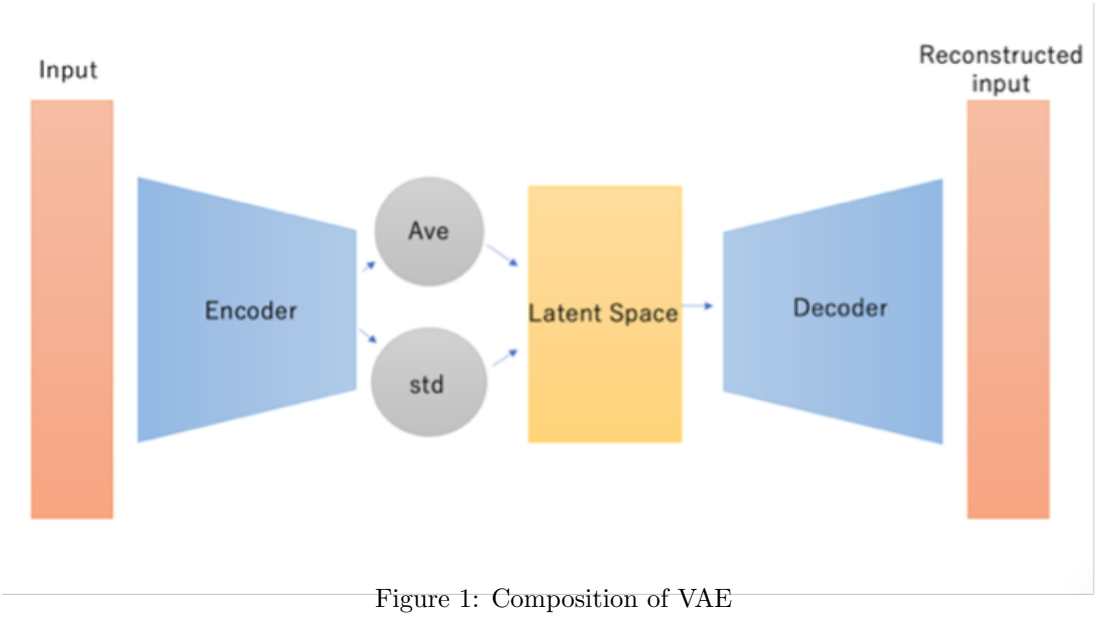


Figure 1: Composition of VAE

Variational Autoencoder (VAE) is a generative model framework designed to learn a probabilistic latent representation of input data. They consist of two main components: an encoder and a decoder. The encoder maps high-dimensional input data (such as images) into a latent space defined by a probability distribution, typically a Gaussian distribution learned from the data. The decoder, in turn, takes samples drawn from the latent distribution and attempts to reconstruct the original input.

The training process involves optimizing a joint objective function that balances two terms. The first is the reconstruction loss between the original input and the decoder's reconstruction. This term encourages the latent representations to capture the essential features of the input so that the decoder can reliably reconstruct it. The second term is the Kullback-Leibler (KL) divergence between the latent distribution inferred by the encoder and a predefined prior distribution, commonly a standard Gaussian. By minimizing the KL divergence, the model is incentivized to produce latent variables that are close to this prior, facilitating smoother interpolation within the latent space and promoting better generalization.

Through backpropagation and iterative gradient-based optimization, the VAE adjusts its encoder and decoder parameters, thereby refining the distribution of latent variables and improving the quality of the reconstructed data. This generative modeling approach enables the VAE to not only compress complex, high-dimensional data into a more manageable latent representation but also to generate new, similar data points by sampling directly from the learned latent space distribution.

3.2 Squeeze-and-Excitation Networks (SENet)

Squeeze-and-Excitation Network (SENet) [9] introduces a novel architectural unit known as the SE-Block. This unit specifically targets the channel-wise relationships within the feature maps of Convolutional Neural Network (CNN). Unlike traditional CNN, which typically process each feature channel independently, SENet explicitly models the interactions between channels. By learning how channels correlate and contribute to the overall representation, SENet dynamically recalibrates channel-wise feature responses. This approach emphasizes informative channels and suppresses those that are less useful, thereby enhancing the model's representational power. The main components of SENet are as follows:

- Squeeze (Global Information Embedding): The feature maps output by a convolutional layer are transformed into a set of channel-wise statistics. Specifically, global average pooling is applied across spatial dimensions for each channel, reducing each feature map to a single scalar. This operation captures a global context for each channel.
- Excitation (Adaptive Recalibration): The aggregated channel descriptors are passed through a small two-layer fully connected (FC) network.
- Recalibration (Channel Reweighting): The learned weights are then applied channel-wise to the original feature maps. Channels deemed important are emphasized ("excited"), while less relevant ones are suppressed, effectively attending to the most salient features.

The advantages of SENet can be broadly categorized into three main areas. First, simplicity and flexibility are key features: SENet introduces minimal architectural changes, and the SE-Block can be seamlessly integrated into existing CNNs without substantial modification. Second, SENet demonstrates notable computational efficiency, adding less than one percent overhead in parameters and computational cost, thereby maintaining a high level of efficiency. Third, SENet delivers considerable performance gains, as its focus on channel-wise importance enhances the representational capacity of networks, leading to marked improvements in accuracy across diverse vision tasks.

In terms of practical impact, SENet's emphasis on channel-wise rather than spatial attention allows it to circumvent the complexities associated with spatial feature processing. Consequently, SENet can be effortlessly incorporated into various CNN architectures, offering improvements in image classification, object detection, segmentation, and generative modeling. Its ease of integration and proven effectiveness have resulted in its widespread adoption in state-of-the-art models.

3.3 SENet-VAE

This paper explores incorporating the SE-Block into VAE to enhance its performance. The VAE integrated with SE-Block is referred to as SENet-VAE. An example of the proposed SENet-VAE configuration is shown in Figure 2.

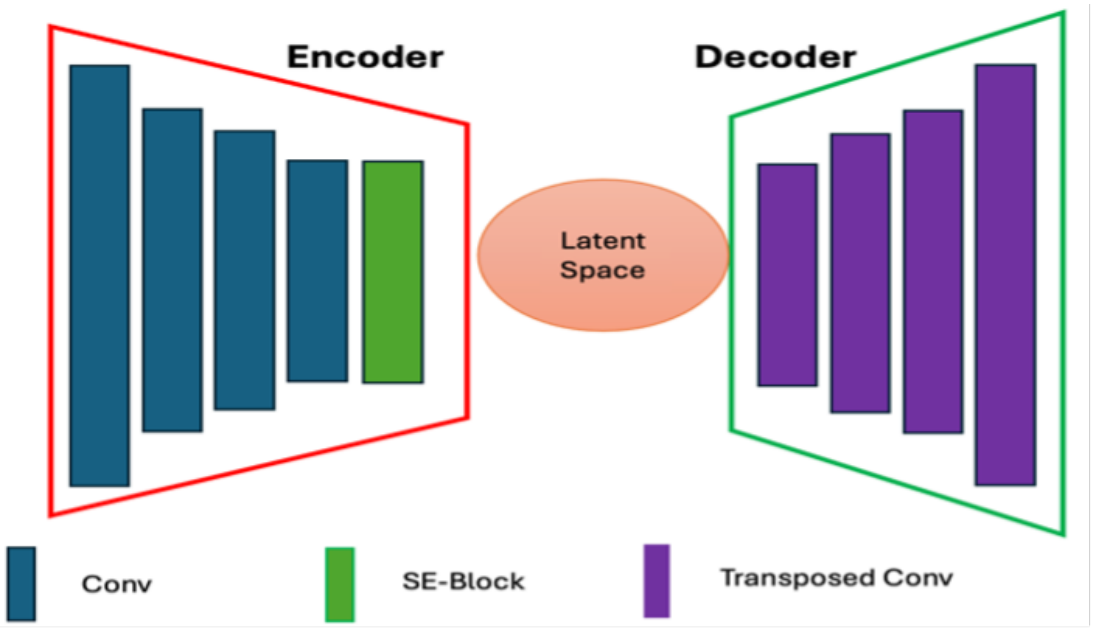


Figure 2: Composition of SENet-VAE

The encoder consists of four convolutional layers, and the decoder is composed of four transposed convolutional layers. In the configuration depicted in Figure 2, the SE-Block is applied to the output of the fourth layer of the encoder. By capturing inter-channel dependencies, the SE-Block is expected to reduce redundant information, thereby enabling more efficient feature representations and ultimately improving the model's performance. In this study, we investigate how variations in the placement and number of SE-Blocks influence the effectiveness of the VAE model, aiming to identify the most beneficial configuration.

4 Experiment

We conducted computer simulation experiments to examine how the learning process differs between raw image data, VAE, and SENet-VAE. In addition, we investigated the effects on learning accuracy when varying the placement of SE-Blocks and increasing their number.

4.1 Experimental Setup

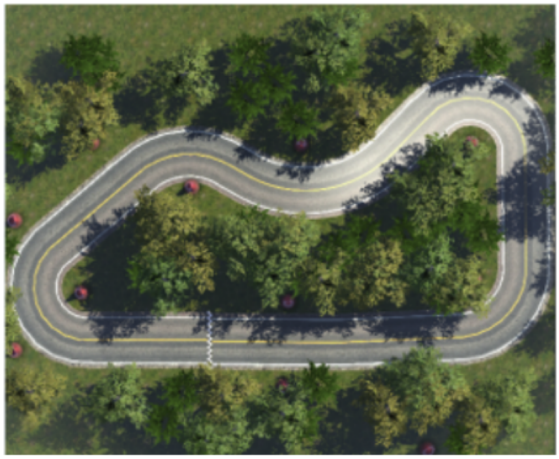
The reinforcement learning environment used in this study is primarily defined by gym-donkeycar [14], created by Kramer based on OpenAI's gym platform. The reward function provides a negative reward to the agent when the vehicle collides with an object or deviates beyond a certain distance from the center line of the track. Conversely, when the vehicle is moving forward, the agent receives a positive reward proportional to the vehicle's speed and its proximity to the center line. The optimal reward is achieved when the vehicle travels at high speed near the center of the lane.

We also define the end conditions of an episode as follows:

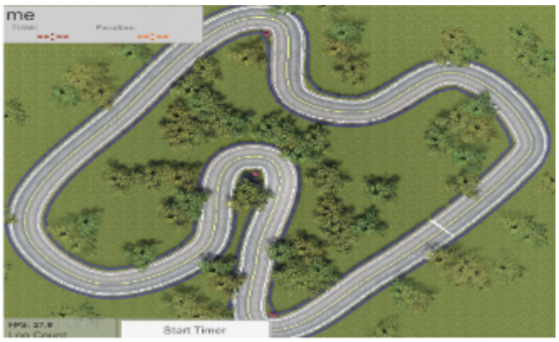
- Exceeding a certain distance from the center line
- Colliding with an obstacle
- Falling below a specified speed threshold for a defined duration

For our study, we incorporate five distinct courses, each serving as an experimental setting. The five courses are shown in Figure 3. Among these five courses, only course (c) is of a walled perimeter.

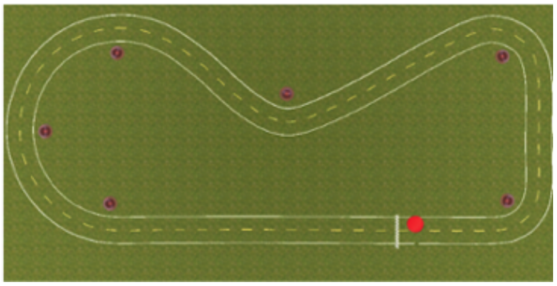
(a) Waveshare



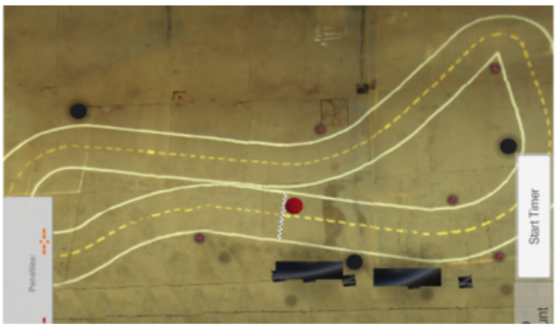
(b) Generated_track



(c) Mini_monaco



(d) Warren_track



(e) House

Figure 3: Five courses used in the simulation

Course (a) entails a training duration of 30,000 timesteps, while courses (b) to (e) undergo training for 50,000 timesteps. We employ the PPO for DRL. Learning is conducted using the total timesteps parameter, which is defined separately for each course. Within this setup, the n_step parameter is set to 512, determining that the agent updates its policy every 512 timesteps.

A pre-trained model is employed for dimensionality reduction using a VAE. For training the VAE, images captured by the vehicle's camera, originally of size 120px × 160px × 3 channels as shown in Figure 4, are resized to 80px × 160px × 3 channels as illustrated in Figure 5. These resized images are used as input data for the VAE. A dataset of 20,000 images per course is prepared, and the VAE is trained for 50 epochs using mini-batch learning.

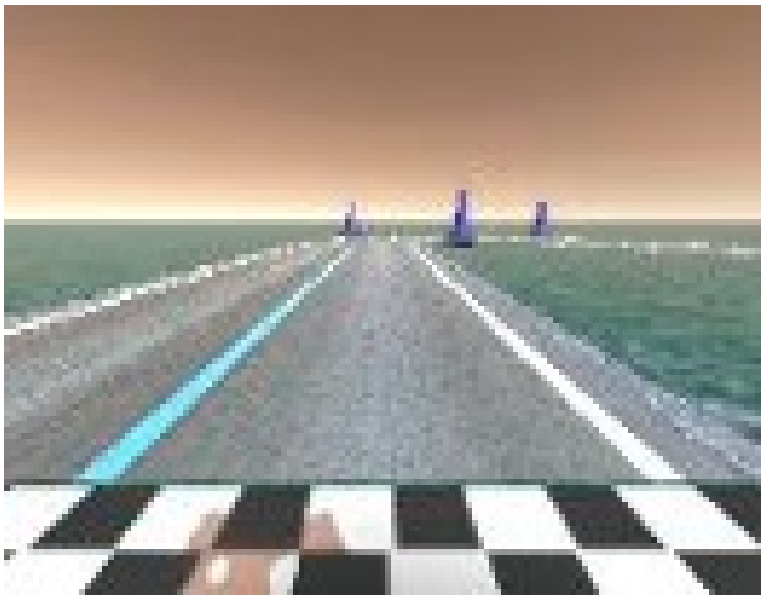


Figure 4: An original image captured by the camera

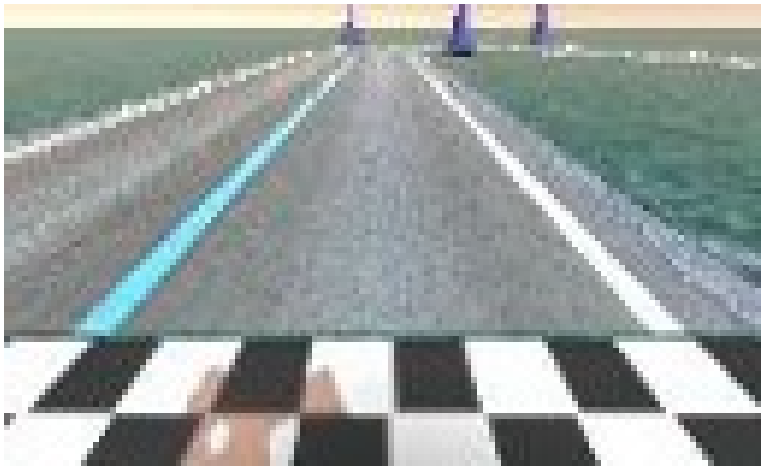


Figure 5: A resized image

The pre-trained model is then utilized to perform real-time resizing on images obtained from the vehicle, reducing the dimensionality of images with 38,400 pixels to 32 pixels using the VAE

encoder. Through this process, the input images are transformed into the latent space, as shown in Figure 6, and are used as input data for learning driving behaviors.

Similar to the VAE, the SENet-VAE undergoes pre-training, and dimensionality reduction is performed using the same process.

Figure 6: A latent representation (32 pixels)

Table 1 shows the configuration of the computer on which the experiments in this study were conducted.

Table 1: Computer specifications

OS	Ubuntu 22.04 LTS
CPU	Intel Core i7-13700
GPU	NVIDIA RTX 4000
Memory	32.00 [GB]

4.2 Performance Evaluation of Three Methods

Driving behavior learning was performed using three methods: (1) using image data as input, (2) applying dimensionality reduction with a VAE, and (3) applying dimensionality reduction with SENet-VAE. The experiments were conducted on all courses shown in Figure 3 (a) through (e). The SENet-VAE model used in these experiments is illustrated in Figure 2. The results are summarized in Figure 9. In each figure, the vertical axis indicates the average reward obtained from reinforcement learning, while the horizontal axis indicates the timesteps.

4.3 Performance Evaluation of SE-Block Insertion Points

In this experiment, we examined how performance changes with varying the insertion positions of SE-Blocks. The insertion positions were varied across four locations, as shown in Figure 7, and performance was measured. The courses used for the experiment are illustrated in Figure 3 (a) through (c). The results are summarized in Figure 10. In each figure, the vertical axis represents the average reward from reinforcement learning, and the horizontal axis represents timesteps.

4.4 Performance Evaluation with Varying SE-Block Numbers

In this experiment, we investigated how performance changes with increasing the number of SE-Blocks. The patterns for increasing SE-Blocks were varied across four locations, as shown in Figure 8, and performance was measured. The courses used for the experiment are illustrated in Figure 3 (a) through (c). The results are summarized in Figure 11. In each figure, the vertical axis represents the average reward from reinforcement learning, and the horizontal axis represents timesteps.

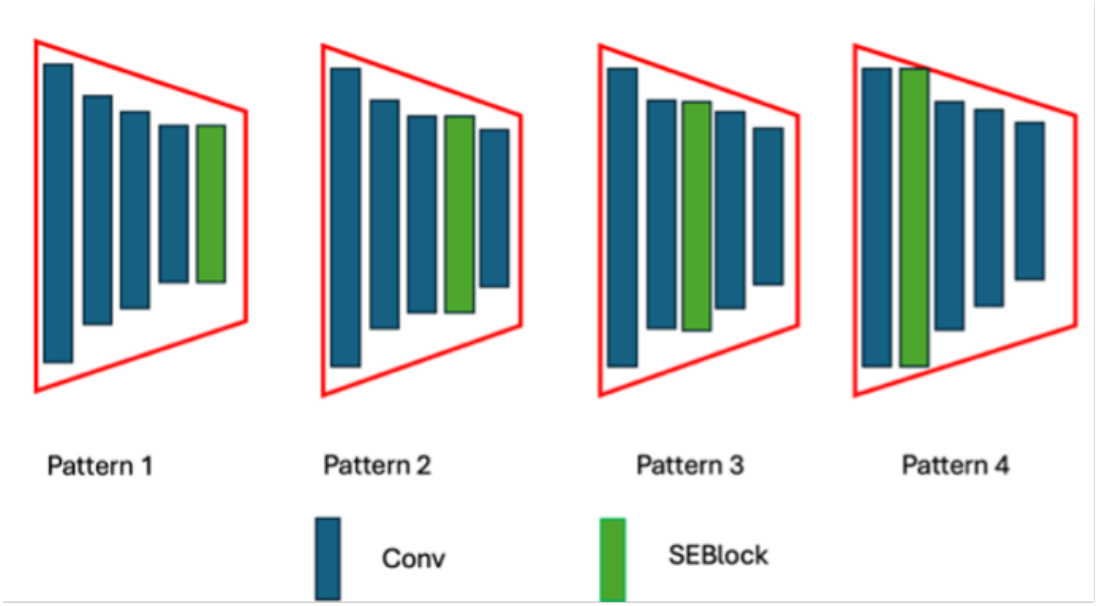


Figure 7: Four patterns of SE-Block insertion points

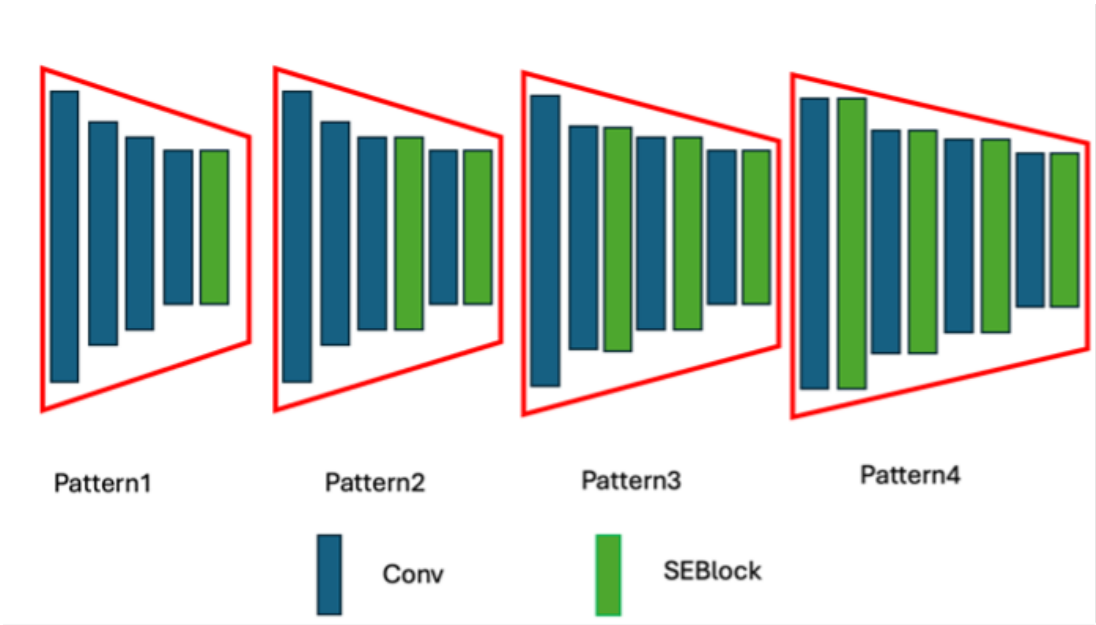
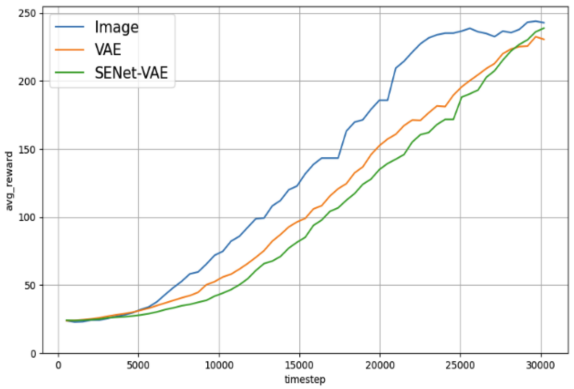
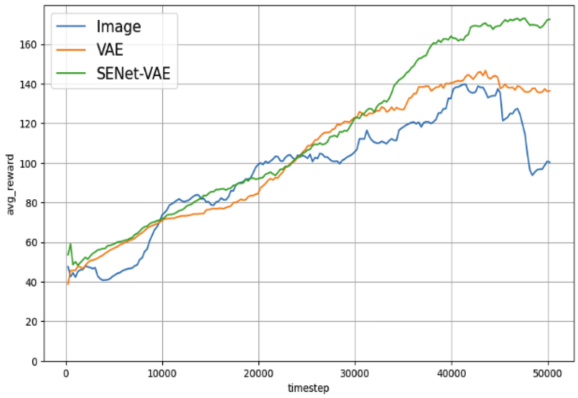


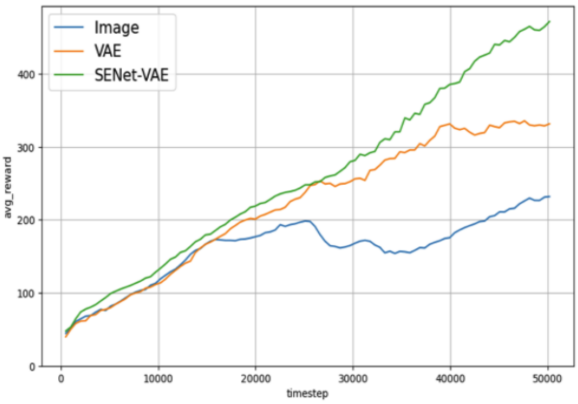
Figure 8: Four patterns for increasing SE-Blocks



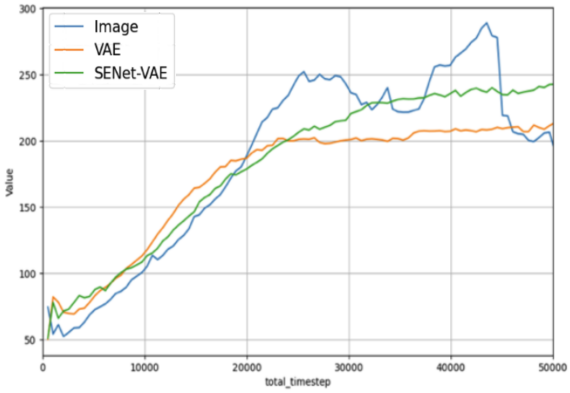
(a) Waveshare



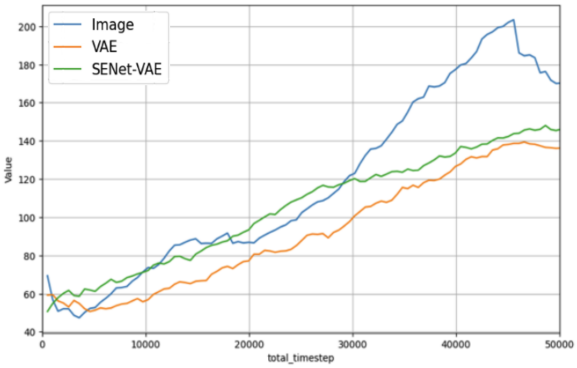
(b) Generated_track



(c) Mini_monaco

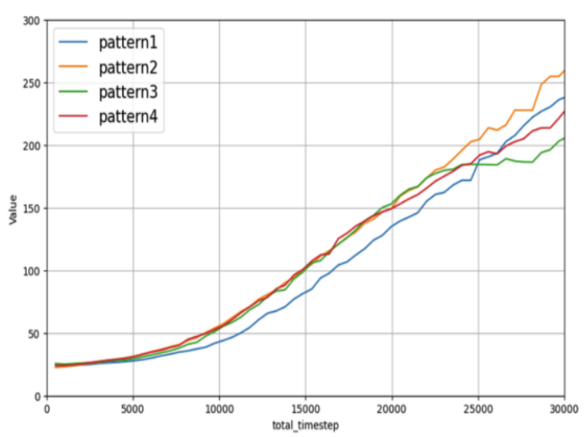


(d) Warren_track

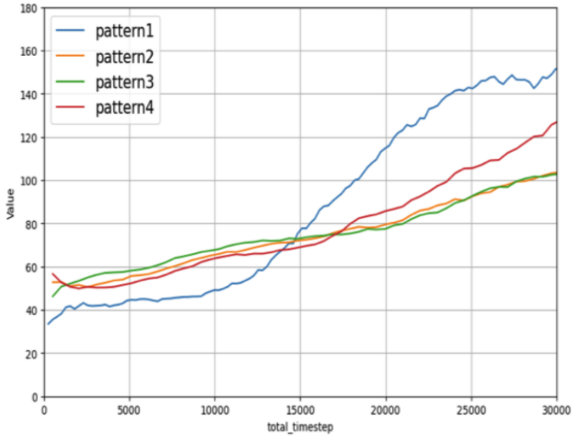


(e) House

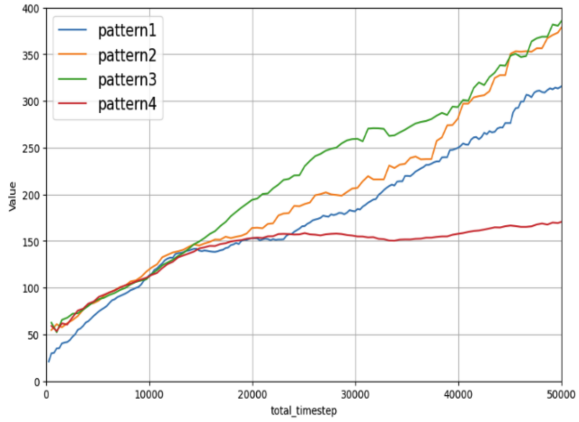
Figure 9: Average reward progression for Image, VAE, and SENet-VAE



(a) Waveshare

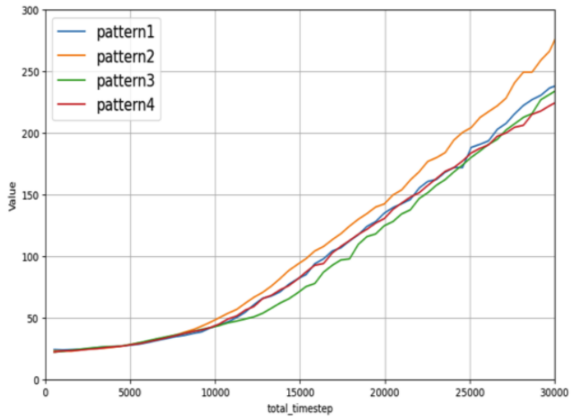


(b) Generated_track

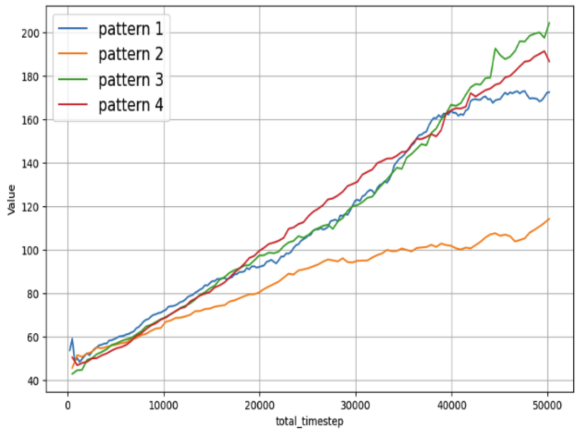


(c) Mini_monaco

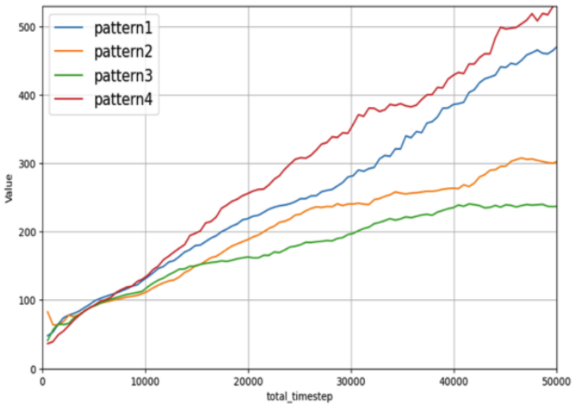
Figure 10: Average reward progression for different SE-Block insertion positions



(a) Waveshare



(b) Generated_track



(c) Mini_monaco

Figure 11: Average reward progression with increasing number of SE-Blocks

4.5 Discussion of Experimental Results

Regarding learning efficiency, the following observations can be made based on Figure 9. In course (a), it was confirmed that the learning accuracy of the SENet-VAE was comparable to that of the standard VAE. This suggests that the data characteristics and task difficulty were relatively simple, and the feature enhancement effects of the SENet were not significantly manifested. On the other hand, in courses (b) and (c), the SENet-VAE demonstrated higher learning accuracy than the standard VAE. The improvement was particularly notable in course (c), which can be attributed to the effectiveness of the SENet under conditions of greater data complexity and noise influence, emphasizing important features and suppressing unnecessary information. In courses (d) and (e), using raw image data as input resulted in the highest accuracy. However, when dimensionality reduction was performed using SENet-VAE, it was confirmed that the accuracy was higher compared to when dimensionality reduction was carried out using a conventional VAE.

In experiments where the placement of SE-Blocks was varied, Figure 10 suggests the following interpretation. It was observed that all patterns exhibited equivalent performance in course (a). In course (b), Pattern 1 achieved the highest performance, whereas in course (c), Patterns 2 and 3 demonstrated the best performance, followed by Pattern 1. Overall, Pattern 1 consistently yielded superior results compared to the other patterns across all courses. This can be attributed to SENet's mechanism of weighting important channels, which likely leads to enhanced performance when the number of channels is high, as is the case with Pattern 1.

When the number of SE-Blocks incorporated was increased, as shown in Figure 11, no significant changes were observed across the different patterns. In some instances, an increase in the number of SE-Blocks even resulted in a decline in performance. The lack of performance improvements upon adding SE-Blocks is attributed to the increased complexity of the VAE model structure with each additional SE-Block, leading to a rise in the number of parameters.

5 Conclusion

The primary objective of this study is to enhance the state representation in Variational Autoencoders (VAEs) and improve the performance of driving behavior learning by incorporating SENet into the VAE. As a result, the proposed approach achieved improved learning accuracy without a significant increase in training time compared to using a conventional VAE for dimensionality reduction. This improvement can be attributed to the integration of SENet, which emphasizes essential information and enhances the state representation capabilities of the VAE.

Furthermore, when varying the placement of SE-Blocks, it was confirmed that placing them closest to the latent space resulted in the most stable learning. Additionally, increasing the number of SE-Blocks did not generally have a positive impact on the learning outcomes. These findings indicate that the most effective configuration for enhancing state representation is to place a single SE-Block near the latent space.

Future work involves enhancing the performance of Variational Autoencoders (VAEs) through alternative methodologies. Although the application of SENet to VAEs demonstrated performance improvements in several instances, it was observed that in certain training courses, the performance remained comparable to conventional methods. Consequently, future efforts will focus on improving VAE performance through various approaches.

Firstly, architectural enhancements can be pursued. Introducing U-Net-like skip connections and residual blocks can stabilize the learning process while minimizing information loss. Additionally, incorporating self-attention mechanisms facilitates the capture of complex dependencies within the data. Furthermore, employing hierarchical latent variable models, such as Ladder VAEs and Hierarchical VAEs, can enhance representational capacity, although these approaches introduce challenges related to model complexity and learning stability.

Secondly, alternative models that complement or substitute VAEs should be considered, including Normalizing Flow-based models, Diffusion models, and Vector Quantized VAEs (VQ-VAEs). Normalizing Flow models (e.g., Glow, RealNVP) achieve high expressiveness through reversible transformations but tend to incur higher computational costs during training. Diffusion models

(e.g., DDPM, Stable Diffusion) enable high-fidelity generation but require substantial time and resources for both training and inference. VQ-VAEs discretize the latent space, and approaches such as VQ-VAE-2 have been developed to enhance the generation performance of high-resolution images.

Thirdly, implementation techniques should not be overlooked. The appropriate integration of Batch Normalization and Layer Normalization can stabilize gradient computations even in deep architectures. In addition to KL annealing, methods such as progressive growing, which incrementally increase the model size during training, can be employed to enhance learning. Data augmentation has the potential to improve generative performance; however, it is essential to carefully consider the impact of the augmented data distribution on the latent space.

References

[1] Q. Zhang, T. Du, and C. Tian. Self-driving scale car trained by deep reinforcement learning. *arXiv preprint arXiv:1909.03467*, 2019.

[2] S. Wang, D. Jia, and X. Weng. Deep reinforcement learning for autonomous driving. *arXiv preprint arXiv:1811.11329*, 2018.

[3] B. Balaji, S. Mallya, S. Genc, S. Gupta, L. Dirac, *et al.* DeepRacer: Autonomous racing platform for experimentation with Sim2Real reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2746–2754, Paris, France, 2020. doi: 10.1109/ICRA40945.2020.9197465.

[4] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254, 2019.

[6] A. Gupta, A. S. Khwaja, A. Anpalagan, L. Guan, and B. Venkatesh. Policy-gradient and actor-critic based state representation learning for safe driving of autonomous vehicles. *Sensors*, 20(21):5991, 2020.

[7] Y. Uehara and S. Matsumae. Dimensionality reduction methods using VAE for deep reinforcement learning of autonomous driving. In *International Workshop on Advances in Networking and Computing (WANC), CANDARW*, 2023.

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

[9] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Y. Uehara and S. Matsumae. Effect of integrating variational autoencoder with SENet on dimensionality reduction in driving behavior learning. In *International Workshop on Advances in Networking and Computing (WANC), CANDARW*, 2024.

[11] Donkey® Car – Home. `https://www.donkeycar.com/`.

[12] S. Song, F. Yu, X. Jiang, J. Zhu, W. Cheng, and X. Fang. Loop closure detection of visual SLAM based on variational autoencoder. *Frontiers in Neurorobotics*, 17, 2024. doi: 10.3389/fnbot.2023.1301785.

[13] T. Zhang, Y. Yang, and A. Zhang. 3D reconstruction of porous media using a batch normalized variational auto-encoder. *Computational Geosciences*, 26:1261–1278, 2022. doi: 10.1007/s10596-022-10159-1.

[14] T. Kramer. OpenAI Gym environments for Donkey Car [source code]. `https://github.com/tawnkramer/gym-donkeycar.git`, 2018.

[15] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, 22(1):12348–12355, 2021.