Understanding Transformer Architecture in Learning Agile and Universal Dynamics Model

Group 10

Hang Liu Haoyu Ma Nithish Kumar

***Abstract*—Since 2017, transformer structure has not only largely replaced recurrent neural network in solving vanishing/exploding gradient and accelerating parallel computation, attention patterns has also shown strong potential in reinforcement learning tasks associated with robot perception and control. In this work, we have selected "AnyCar to Anywhere: Learning Universal Dynamics Model for Agile and Adaptive Mobility", aims to answer the question if transformers, with their inductive bias on pairwise attention between sequence inputs, can achieve efficient training, cross-enhancement, and low-level control along with their simulation and experimental work. For the purpose of summarizing key knowledge of the reference paper and making our a legit proposal, this study has proposed a universal model "AnyCar" with dynamic transformer training method to predict the trajectory of various cars across different settings and context adaptations. The model involves both simulation pretraining and real-world fine-tuning, and deployed with a sampling-based MPC. The goal of this study is to refine the AnyCar model by implementing and assessing the impact of various transformer architectures to improve generalization and more efficient vehicle control in diverse environments.**

***Index Terms*—DRL, transformer, robotics**

I. Introduction

In the field of vehicle dynamics modeling, the need for robust, adaptable, and cost-efficient solutions has become increasingly prominent. Traditional specialty dynamics models, while useful, may require substantial tuning and tweaking to accommodate specific situations and vehicle types. This technique is labor-intensive, but it also limits the models' adaptability across different contexts, resulting in an increasing interest in exploring and developing universal dynamics models. The goal of developing such models is to overcome the limits of specialized approaches and to enable agile and adaptive control over a wide range of vehicles and operating situations.

In tackling these issues, the release of the "AnyCar" transformer-based model represents a major breakthrough. The AnyCar model provides strong performance in both simulation and real-world applications, drawing on large datasets supplied by many modeling platforms. The authors illustrate the model's capacity to address sim-to-real differences efficiently by using a two-phase training approach that begins with synthetic datasets for pretraining and then progresses to real-world fine-tuning. Furthermore, implementing a sampling-based Model Predictive Control system enables for real-time trajectory monitoring, which improves the model's responsiveness and versatility in a variety of driving scenarios.

While previous research has demonstrated the AnyCar framework's usefulness, it has primarily focused on comparisons to recurrent neural network architectures such as LSTM, GRU, etc. However, the potential of other transformer topologies is largely unknown. Frameworks such as Transformer-XL may provide improved capabilities for managing long-horizon sequences, which are crucial for vehicle dynamics simulation. This project seeks to conduct comparative tests on various transformer topologies to evaluate their usefulness in enhancing performance across generative modeling and training efficiency, thereby contributing to the continued evolution of universal dynamics modeling.

II. Related works

A. Model-Based Control for vehicle

Vehicle trajectory tracking control is designed to follow desired paths across a variety of driving scenarios. To address these challenges, a range of control strategies has been developed, including PID control, linear quadratic regulator (LQR), feed-forward and feedback control, robust control, sliding mode control (SMC), model predictive control (MPC), and learning-based approaches [SYC+15], [DLZ17], [LKS+23], [ZZW+20], [LHD+23]. These methods aim to handle difficult conditions, such as driving on low-friction surfaces with significant steering demands. While model-based control algorithms like PID, LQR, SMC, and MPC have achieved significant advancements, challenges remain. However, the gap between real dynamics and simplified mathematical model still exists.

B. Dynamics Learning

For learning dynamics of vehicles, such as those mentioned in [SBK+19], [WWG+17], utilize historical vehicle state and control input data to train deep neural networks (DNNs) for understanding vehicle dynamics. For example, [HTBL20] introduced a recurrent neural network with a Gated Recurrent Unit (GRU) to achieve similar objectives. While these approaches perform effectively on data within their training distribution, they struggle to generalize to scenarios outside of that distribution. Furthermore, the complexity and non-linear nature of DNNs make them less practical for model-based predictive control. These models, although skilled at capturing intricate patterns in data, do not inherently comply

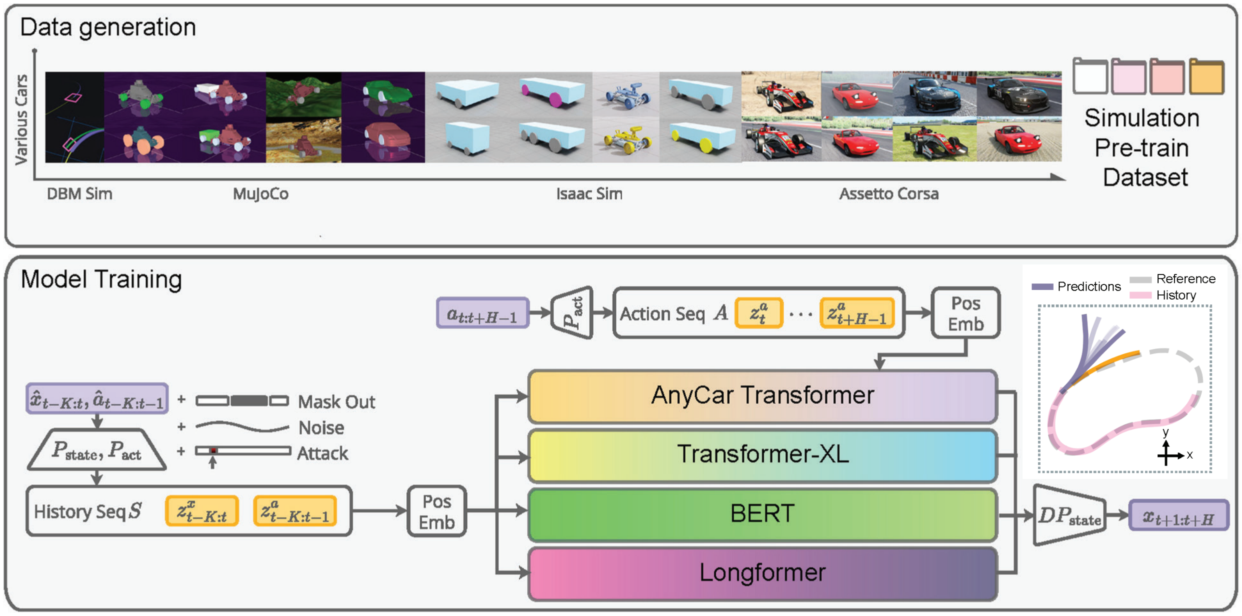

Fig. 1. Training pipelines which include first stage of synthetic data generation and the second stage of model training with selections of transformers.

with physical laws, which can lead to predictions that are physically infeasible for a vehicle to achieve. However, there is no previous work trying to use transformer to learn cross-embodiment vehicles' dynamics model like what 'AnyCar' did [XXT+24]. But in Anycar, they do not elaborate architecture of transformer, which might affect performance.

III. Methods

A. Pre-Training with Massive Simulated Data

1) Scene Generation: To collect a diverse dataset, we leverage the low-cost nature of simulation and generate a large amount of simulated data. Our data generation has three sources of diversity: 1) dynamics, 2) scenario, and 3) controller. we synthesize trajectories of various cars and terrains via MuJoCo.

2) Curriculum Model Training: To ensure data distribution coverage, we diversify between on-policy and off-policy data via a two-stage curriculum learning method. In stage one, we collect high-volume off-policy data as warm-up, by building a hybrid controller where we use a pure pursuit controller for steering δ and a PD controller for throttle T, to track randomly synthesized reference tracks. After collecting 200M timesteps that are usable for training a general model for non-agile tasks (in which the target velocity is smaller than the physical limit), we switch to stage two, where we deploy an on-policy NN-MPPI controller (described in Section V-A) to track agile trajectories, and periodically update the network with the collected on-policy trajectory.

B. Robust In-Context Adaptive Dynamics Model

1) Model Structure: the historical states in Equation (1) are linearly projected into a 64-dimensional latent space with an encoder layer

$$P_{\text{state}}(x) : \mathbb{R}^6 \to \mathbb{R}^{64}.$$

The historical actions pass through a different encoder

$$P_{\text{act}}(a) : \mathbb{R}^2 \to \mathbb{R}^{64}$$

of similar size. We then interleave and stack them to make a complete history token sequence

$$S_{2K-1\times 64},$$

which is supplied to the transformer as the context. The future actions are tokenized with $P_{\text{act}}(a)$ and stacked as a sequence $A_{H\times 64}$. Both S and A sequences are then summed with two learnable positional encoders. After passing the context S and input A to the transformer decoder, we obtain

$$S'_{H\times 64},$$

which is then downprojected into the state-space with

$$DP_{\text{state}}(x) : \mathbb{R}^{64} \to \mathbb{R}^6.$$

This becomes the final state sequence prediction.

2) Robust Training: To learn a robust model f_θ for Equation (1) under various disturbances, we propose to add three techniques in pre-training, i.e., mask out, add noise, and attack. We implement mask out by applying randomized cross-attention masks to the transformer, then add noise $\epsilon \sim \mathcal{N}(0, \epsilon_{\max})$. We also add an attack: unreasonably large or small values to random dimensions in history to simulate state estimation errors in the real-world.

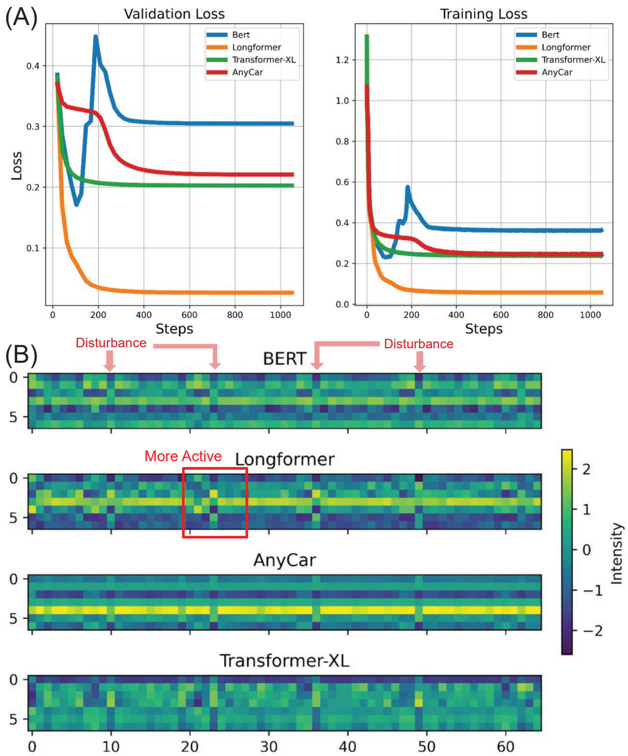

Fig. 2. (A) Training losses for four transformer architectures in learning carts' perturbed agile dynamics. (B) Heat maps of network's latent layers showing neuron activities.

C. Comparative Experiment

We will first follow their work to explore the development of a universal dynamics model for various cars, enabling agile and adaptive control. The motivation behind this work stems from the limitations of specialist models, which require extensive tuning for specific environments and vehicle types, making them costly and less adaptable. The authors propose the "AnyCar" transformer-based model, which leverages a large dataset from multiple simulators to achieve robust performance, demonstrating superior agility and adaptability compared to specialist models.

The core method described involves a two-phase training approach. In the first phase, the model is pre-trained on a synthetic dataset generated from four different simulation platforms, incorporating a variety of vehicle dynamics and environmental conditions. The model used is the GPT-2 Transformer for understanding the long horizon sequence information.

Although the authors implement many experiments about the framework and its effectiveness, they do not compare different transformer architectures. They only compare their method with LSTM, GRU, etc. However, different transformer frameworks could make a significant difference. For example, Transformer-XL might perform better in handling long horizon sequence information.

Therefore, as a milestone, we will conduct comparative experiments of implementing new transformer structures and aim for better performance, either through a more general embodiment or a more efficient training process.

Performance out of training distribution				
Architecture	Transformer (Anycar)	TransformerXL	Bert	Longformer
Validation Loss(MSE)	0.2833	0.2721	0.4066	0.1269

Fig. 3. Direct numerical comparison of validation losses showing Longformer outperforming other transformers in learning long-horizon control tasks of wheeled robots.

- GPT-2 [RWC+19]: A unidirectional, autoregressive Transformer optimized for text generation, where each token prediction is based only on previous tokens. It's highly effective for generating coherent, contextually accurate text but is limited in capturing long-range dependencies due to a fixed context window.
- Transformer-XL [DYY+19]: An improved version of the original Transformer, it introduces a recurrent memory mechanism that extends context length beyond the standard limits by passing hidden states between segments. This enables it to model much longer sequences effectively and capture long-term dependencies, making it well-suited for long text generation and language modeling tasks.
- BERT [DCLT19]: A bidirectional Transformer that excels at understanding the context of text by jointly considering both past and future tokens. It's pre-trained with a masked language model objective, which makes it powerful for various NLP tasks such as classification, question answering, and named entity recognition. However, its bidirectional nature limits its use for autoregressive tasks like text generation.
- Longformer [BPC20]: Designed for handling very long documents, Longformer uses a combination of local and global attention mechanisms to reduce computational complexity, making it feasible to process sequences with thousands of tokens.

IV. Discussion

Fig.2(a) are plotted trainings losses of four networks when learning a perturbed dynamic. Longformer achieves the lowest loss and a slightly slower convergence rate than the Transformer-XL among the four, denoting a better performance than AnyCar to learn long-horizon tasks. Moreover, in Fig.2(b), we examine the heat map of the latent layer within each network with horizontally time stamp and vertically each channel. We may observe Longformer has the strongest neuron response to the disturbance, which implies a greater potential to withstand perturbations posed in environment.

In this study, we examined the performance of four transformer architectures in training a dynamics model that aims to achieve cross-embodiment and long-horizon control tasks.

The original proposed transformer AnyCar(GPT2) being autoregressive and unidirectional, is better suited for sequential generation tasks rather than tasks requiring the model to learn long-term dependencies or control. However, because it generates each token based on the previous tokens, it can perform well in tasks that involve predicting sequential steps over relatively short to medium horizons. Such network may struggle with extremely long-horizon control tasks. However, Longformer can handle long-horizon control tasks better than traditional Transformer models due to its ability to scale to long sequences while maintaining an efficient attention mechanism. In tasks involving long-term memory, where the model needs to remember key information over time, Longformer can perform well as it can consider more of the past context during decision-making.

However, although our work has not demonstrated the performance of transformer selections in cross-embodiment, GPT2 is supposed to be more limited in cross-embodiment tasks due to its autoregressive, single-agent design than that of Longformer. Because, like Transformer-XL, Longformer is equipped with memory retention features and therefore can handle long sequences across multiple agents or environments with modifications.

The results of our study show that transformer architectures vary in their effectiveness for vehicle dynamics modeling, depending on the task's horizon and complexity. The GPT-2-based AnyCar model performs well for short to medium-horizon tasks but struggles with long-term dependencies and cross-embodiment applications. In contrast, Transformer-XL, with its memory retention capabilities, performs better in long-horizon tasks, though its potential in cross-embodiment contexts needs further exploration. Longformer excels in handling long sequences efficiently, making it particularly suitable for real-time, long-horizon control tasks in dynamic environments. These findings highlight the importance of selecting the appropriate transformer architecture based on specific task requirements, with Transformer-XL and Longformer offering clear advantages for complex vehicle dynamics modeling.

V. Conclusion

In conclusion, transformer-based models like Transformer-XL and Longformer show clear advantages over GPT-2 for long-horizon control and cross-embodiment tasks in vehicle dynamics modeling. While GPT-2 excels in short-term predictions, its limitations in handling long-term dependencies are overcome by Transformer-XL and Longformer, which offer better memory retention and efficiency, and Longtransformer could achieve best performance.

VI. Future Works

Future work will focus on deploying these models in real-world scenarios using controllers like Model Predictive Control (MPC) or Model Predictive Path Integral (MPPI) to assess their performance and bridge the sim-to-real gap. With continued refinement, transformer-based models can drive advancements in more agile, adaptive, and efficient vehicle control systems.

References

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[DLZ17] Shihong Ding, Lu Liu, and Wei Xing Zheng. Sliding mode direct yaw-moment control design for in-wheel electric vehicles. *IEEE Transactions on Industrial Electronics*, 64(8):6752–6762, 2017.

[DYY+19] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

[HTBL20] Leonhard Hermansdorfer, Rainer Trauth, Johannes Betz, and Markus Lienkamp. End-to-end neural network for vehicle dynamics modeling. In *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pages 407–412, 2020.

[LHD+23] Wei Liu, Min Hua, Zhiyun Deng, Zonglin Meng, Yanjun Huang, Chuan Hu, Shunhui Song, Letian Gao, Changsheng Liu, Bin Shuai, Amir Khajepour, Lu Xiong, and Xin Xia. A systematic survey of control techniques and applications in connected and automated vehicles, 2023.

[LKS+23] Yukun Lu, Amir Khajepour, Amir Soltani, Ruilong Li, Ran Zhen, Yegang Liu, and Minghui Wang. Gain-adaptive skyhook-lqr: a coordinated controller for improving truck cabin dynamics. *Control Engineering Practice*, 130:105365, 2023.

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[SBK+19] Nathan A. Spielberg, Matthew Brown, Nitin R. Kapania, John C. Kegelman, and J. Christian Gerdes. Neural network vehicle models for high-performance automated driving. *Science Robotics*, 4(28):eaaw1975, 2019.

[SYC+15] Yunxiao Shan, Wei Yang, Cheng Chen, Jian Zhou, Ling Zheng, and Bijun Li. Cf-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. *International Journal of Advanced Robotic Systems*, 12(9):134, 2015.

[WWG+17] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721, 2017.

[XXT+24] Wenli Xiao, Haoru Xue, Tony Tao, Dvij Kalaria, John M. Dolan, and Guanya Shi. Anycar to anywhere: Learning universal dynamics model for agile and adaptive mobility, 2024.

[ZZW+20] Jian Zhou, Hongyu Zheng, Junmin Wang, Yulei Wang, Bing Zhang, and Qian Shao. Multiobjective optimization of lane-changing strategy for intelligent vehicles in complex driving environments. *IEEE Transactions on Vehicular Technology*, 69(2):1291–1308, 2020.