Generative Adversarial Models for Vehicular Dynamics Prediction in V2X Networks

(Article begins on next page)

Generative Adversarial Models for Vehicular Dynamics Prediction in V2X Networks

Giuseppe Perrone, Claudio E. Casetti, Marco Rapelli
Department of Control and Computer Engineering
Politecnico di Torino
Turin, Italy
{giuseppe.perrone, claudio.casetti, marco.rapelli}@polito.it

***Abstract*—Real-time trajectory prediction is critical for safe and efficient operation of connected and autonomous vehicles (CAVs). Yet existing deterministic models struggle to capture the multi-modal, uncertain nature of traffic evolution. This paper addresses this gap by proposing a framework powered by Generative Adversarial Networks (GANs) that uses V2X (Vehicle To Everything) data to create a digital twin capable of simulating plausible future scenarios and of supporting planning and decision-making in CAVs. The Generative framework is designed to generate multiple trajectory predictions conditioned on received CAM (Cooperative Awareness Message) data, using a Transformer-based architecture with temporal consistency regularization. We evaluate it in a SUMO-simulated environment, demonstrating improved stability and realism over baseline GAN training.**

***Index Terms*—Generative AI, Vehicular Network, GANs, SUMO, Connected Vehicles**

I. Introduction

As the urban population increases, the infrastructure in the built-up areas becomes increasingly stressed by the constantly increasing number of vehicles, which causes congestion, inefficiencies, and potential safety hazards. Addressing these challenges requires transportation systems that can perceive, reason, and act in real time, adapting to dynamic traffic conditions before the situation worsens. Emerging Vehicle-to-Everything (C-V2X) technologies enable vehicles to share information about their current status, sensed environment, and surrounding traffic. This creates the foundation for a network-level situational awareness layer that can be exploited by advanced Machine Learning (ML)-based algorithms to anticipate traffic evolution, optimize flows, and enhance safety for all road users.

In this context, we propose a Generative AI-powered framework that, given an initial condition and contextual information derived from the network, generates multiple plausible future scenarios. The system creates a context representation from the received data to condition a Generative Adversarial Network (GAN), trained to generate diverse and realistic, plausible trajectories that the input could lead to.

The proposed framework is thought to be integrated into services such as [1], which collect data from multiple vehicular nodes in a central server. The integration of a generative model would allow the system to simulate diverse plausible scenario evolutions based on the stored data, enabling the development of algorithms capable of reasoning about the potential consequences of a given initial condition.

II. Related Works

Generative models are a class of machine learning algorithms designed to learn and approximate the underlying process that generates a real datum. By capturing statistical regularities and latent structures in the data, these models can produce plausible instances that mirror the characteristics of the original data distribution.

This work can be seen as an extension of the study presented in [2], which explored the use of a generative model to create critical driving scenarios. In this work, the authors trained a model to identify the characteristics of the initial conditions that lead to a vehicular collision and to generate new examples with comparable characteristics. Building on [2], we propose a model that includes the generation of social and temporal dynamics. The proposed architecture, conditioned on an initial condition inferred by V2X data, can generate a set of diverse future trajectories that the initial context could lead to.

This approach differs from traditional ML-based trajectory prediction models. Deterministic models struggle in capturing the chaotic and multi-modal nature of traffic [3]. While in real-life scenarios, the same initial conditions can unfold in many different ways, a deterministic model from the same input can produce only one single path prediction. In contrast, generative models can have a more flexible understanding of how traffic can evolve. They can produce diverse sets of realistic possible trajectories, sampled from a distribution similar to that which governs the evolution of real vehicular traffic; a practical property for safety evaluation, planning, and simulation of vehicular networks.

III. Methodology

We aim to develop a generative model that generates simulations of the future based on information coming from the vehicular network. The proposed framework will take input data from the vehicular network, i.e., the positions, speeds, and headings of the neighboring vehicles from Cooperative Awareness Messages (CAMs), and output a possible course of future events. In this first implementation, we used a simulated environment to verify whether the designed model was well-suited to this task; the features utilized are all types of data that could be transmitted through vehicular messages.

This section describes the proposed generative model, the dataset and the loss functions used for training.

A. Architecture of the Models

The generation of the trajectories is performed through a Generative Adversarial Network (GAN), a generative model composed of two different neural networks trained together [4]. The two networks, referred to as Generator and Discriminator, evolve over training by competing against each other through an adversarial process: the Generator attempts to mislead the Discriminator, creating samples as realistic as possible, and the Discriminator tries to distinguish the generated samples from the real ones. The adversarial training is the primary cause of instability in GANs; they are notoriously difficult to train and are exposed to many problems like mode collapse and non-convergent behavior. However, they were chosen over other generative models like Variational Autoencoders (VAEs) for their superior ability to generate highly realistic samples. Moreover, while Diffusion Models, which are another class of Generative Models, are generally easier to train and often produce samples of comparable or even higher quality, they require significantly more time for sampling. Given the dynamic nature of vehicular applications, where rapid generation is essential, GANs were preferred for their lower inference time.

The core of both the Generator and the Discriminator, in our design, is a Transformer Encoder, first introduced in [5]. This design was chosen for the ability of Transformers to model relationships and dependencies across all elements of the input sequence through their self-attention mechanism. This feature is particularly beneficial in our setting, where capturing long-range and global interactions among vehicles is critical.

The Generator takes a sequence of latent vectors as input, sampled from a standard Gaussian distribution. The Encoder progressively maps latent vectors into a structured embedding space that encodes information about the temporal evolution and the interaction dynamics of the generated sequence. A Cross-Attention layer integrates external context data into the representation, guiding the mapping process toward more context-aware embeddings. The sequence of generated embeddings is converted into a meaningful trajectory through an Output Layer built of three identical blocks. Each block applies a one-dimensional convolution followed by a GELU activation, allowing each timestep to incorporate information from the neighboring embeddings in the temporal dimension. Residual connections between different blocks facilitate the flow of gradients throughout the Output Layer.

The Discriminator architecture mirrors that of the Generator. It takes as input a sequence, real or generated, and processes it through its Encoder to produce an embedding for each timestep. As in Generator, a Multi-Headed Cross-Attention Layer injects context data into the embeddings, guiding the mapping process. Finally, a Fully Connected Neural Network (FCNN) composed of 5 layers maps the embeddings into a scalar value. In our design, the Discriminator is not a binary classifier as in the original GAN formulation. Here, the Discriminator outputs a continuous score that can be interpreted as a measure of the realism of the input sequence. Accordingly, we refer to this module as the Critic rather than Discriminator [6].

The architectures of the two models are shown in Figure 1.

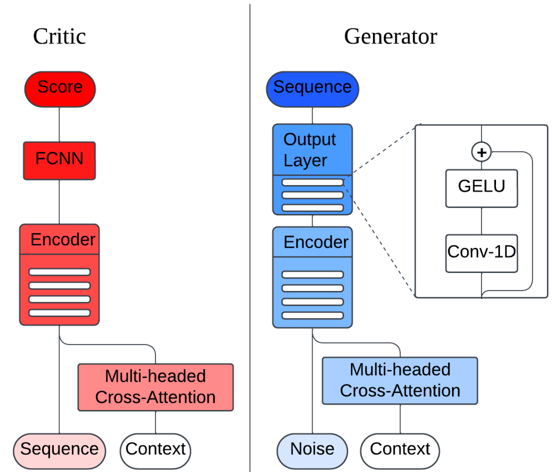

Fig. 1. Architectures of the Generator and Critic. The Critic, on the left, takes as input a sequence and a description of the context to output a measure of realism of the input sequence. The Generator, on the right, produces a possible future trajectory that uses context information to be as realistic as possible.

B. Dataset

To validate our approach, we initially considered a simpler scenario involving a single vehicle traversing a reference network. In this controlled setting, it is possible to verify whether the model is capable of generating plausible motion trajectories while adhering to several constraints: remaining within the designated lane, avoiding lane invasions, ensuring smooth trajectories without abrupt oscillations, and respecting traffic rules. However, more complex scenarios involving multiple surrounding vehicles can be considered, an extension we are currently working on.

The reference map showcases two four-way intersections. The first is a priority-regulated junction, where two lower-priority roads intersect a main street with right-of-way. The second intersection is a four-way stop, where all incoming edges have the same priority.

The dataset was generated using SUMO (Simulator of Urban MObility) and comprises 300,000 vehicle trajectories over the described road network. Each sample is a 10-second trajectory and is assigned to one of three classes, depending on the executed maneuver, i.e., turning left, turning right, or going straight. Trajectories contain positions, i.e., x and y coordinates within the reference network, velocities, and headings of the vehicle, sampled at 10 Hz frequency. Features and sampling frequency have been selected in compliance with the ETSI standard, which specifies the transmission of position, velocity, and heading of the connected vehicle through CAMs with a maximum frequency of 10 Hz.

All trajectory features were independently normalized to the range $[-1, 1]$ based on their respective minimum and maximum values, in order to avoid distortions caused by differences in scale and to ensure stable training.

C. Training

Two alternative adversarial loss formulations were used to train the model: the first is based on the Wasserstein Distance (WD), introduced by [6] and adopted in [2]; the second is an objective function introduced by the Relativistic GAN (RGAN) framework [7], also used in the R3GAN framework, the best-performing GAN model for image generation [8].

WD measures the distance between two distributions. Referring to GAN, it measures the difference between the real distribution and the distribution approximated by the Generator P_G. Equation (1) shows the variant of the Wasserstein GAN objective used for training: the first term estimates Wasserstein Distance in distributions by computing the difference between the average Critic scores for real samples, $\mathbb{E}(C(x))$, and generated samples, $\mathbb{E}(C(G(z)))$. Thanks to this formulation, more informative gradients can flow from Critic to Generator. However, the so computed WD is accurate only if the Critic is Lipschitz-continuous; a condition imposed by the second term of (1), the Gradient Penalty (GP), introduced by [9]. This term further stabilizes the training dynamics by constraining the norm of the Critic's gradient to be close to one.

$$\min_{g} \max_{c} \mathbb{E}(C(x)) - \mathbb{E}(C(G(z))) + \lambda \mathbb{E}(||\nabla C(\hat{x})||_2 - 1)^2 \quad (1)$$

Together, these two factors, i.e., a more informative gradient and more controlled Critic's outputs, make WD-based GANs more stable in training and less prone to mode collapse.

The second objective is the Relativistic GAN loss, which defines a relative notion of realism for a sample: generated samples are not evaluated in isolation, but are directly compared to real counterparts. The differences between the Critic's scores are used to encourage the Generator to produce samples that appear more realistic than their real counterparts. Specifically, the Generator is trained to decrease the probability that a generated sample is rated as less realistic than a real one, while Critic aims to increase this probability, establishing an adversarial dynamic described by (2). Here, f corresponds to the log-sigmoid function, $f(u) = \log(\sigma(u))$, while $C(x)$ and $C(G(x))$ are the scores assigned to real and generated samples.

$$\min_{G} \max_{C} \mathbb{E}_{x \sim p_D,\, z \sim p_z} \left[f\left(C(G(z)) - C(x) \right) \right] \qquad (2)$$

The training stability of Relativistic GAN Loss is improved by introducing R_1 and R_2 regularizations, as done in [8], whose expressions are shown in (3). R_1 and R_2 regularizations jointly promote local flatness of the Critic's score around both real and fake data points by penalizing the norm of the Critic's gradient with respect to input.

$$R = R_1 + R_2 = \frac{\gamma}{2} \left(\|\nabla_x C(x)\|^2 + \|\nabla_{\tilde{x}} C(\tilde{x})\|^2 \right), \qquad (3)$$

Additionally, we introduced further regularization in the Generator's loss function to enforce temporal coherence among consecutive time steps. This Temporal Consistency regularization, defined in (4), penalizes discrepancies between the temporal transitions of real and generated trajectories.

$$\mathcal{L} = \mathbb{E}_{t,i} \left[\left\| (\mathbf{x}_{t+1}^{\text{real},(i)} - \mathbf{x}_t^{\text{real},(i)}) - (\mathbf{x}_{t+1}^{\text{gen},(i)} - \mathbf{x}_t^{\text{gen},(i)}) \right\|_1 \right] \quad (4)$$

Specifically, for each sequence i and timestep t, we compare the difference between features of consecutive frames $(\mathbf{x}_{t+1} - \mathbf{x}_t)$ in the real and generated data using the L_1 norm, encouraging the Generator to produce sequences with step-to-step dynamics similar to the ones observed in real data.

IV. Results

The models were trained using the following hyperparameters: the noise vector dimension was set to $d_{\text{latent}} = 256$, while the embedding dimension was $d_{\text{embedding}} = 64$. The feed-forward layers had a dimension of $d_{\text{ff}} = 256$.

Our focus is on analyzing the training dynamics and the generation capability of our architecture, while verifying which adversarial loss formulation can lay the foundations for future developments. For this, we trained the Critic and Generator three times, each with a different adversarial loss formulation. First, we analyzed whether the temporal consistency regularization was beneficial for our application. We trained our architecture twice, minimizing the WGAN-GP objective, defined in (1); with temporal consistency regularization applied in one of the two runs. We will refer to these models as Regularized-WGAN and WGAN, respectively. The so-trained Generators were tasked with producing a sample for a turning right maneuver, and the results are shown in Figure 2. We take the samples as they are given by the Generator, in normalized form, to better understand the model's generation

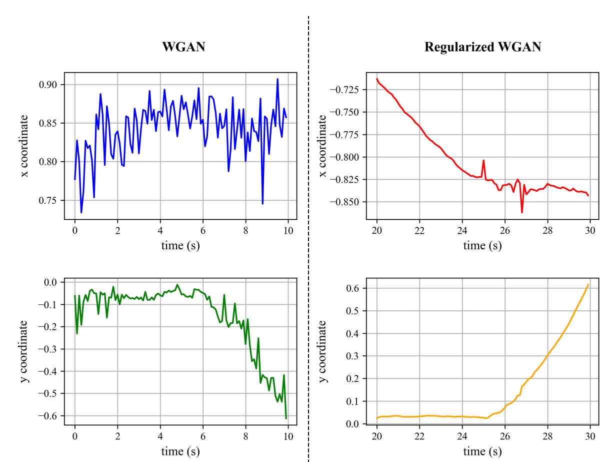

Fig. 2. Comparison of the x and y coordinates over time, generated by WGAN, on the left, and Regularized WGAN, on the right.

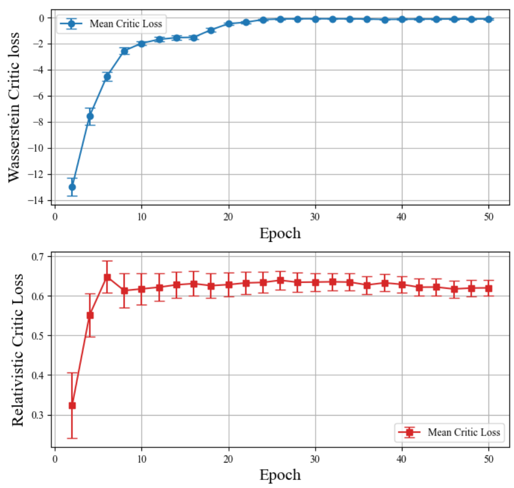

Fig. 3. Evolution of the Critic Loss for Regularized WGAN and Relativistic GAN during training

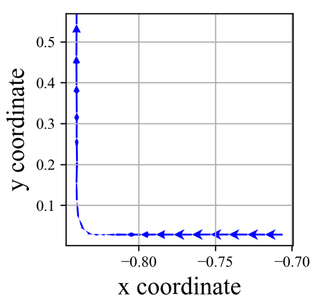

Fig. 4. Point-to-point trajectory in the x-y plane generated by the Relativistic GAN. Arrows indicate the vehicle's speed at each time step.

capability. The figure shows how the generated vehicle's position, expressed in terms of its coordinates, evolves. In particular, the coordinates produced by the WGAN show a markedly unstable behavior, characterized by pronounced oscillations in position. The regularized model reveals a more stable temporal dynamics, although some oscillations emerge when the maneuver begins, approximately at $25s$.

Concerning stability analysis, we compared the regularized WGAN with the model obtained through the minimization of the Relativistic GAN objective, from now on referred to as Relativistic GAN. Figure 3 shows the evolution of the critic loss over the training for both configurations. Both of them successfully converged; however, during the initial training phase, Relativistic GAN exhibited more unstable loss dynamics, characterized by higher fluctuations than its counterpart. Nevertheless, this is the model that produced the trajectory with the most realistic behavior, represented in Figure 4.

The figure represents the point-to-point trajectory of the vehicle in the x-y plane. In this case, the trajectory reveals a behavior consistent with the expectation: the vehicle proceeds along its lane without changing direction; it slows down at the turning point and accelerates again once it completes the turning maneuver.

V. Conclusion

In this work, we proposed a generative framework for modeling the evolution of vehicular scenarios over time using Generative Adversarial Networks (GANs). Our approach is capable of generating multiple plausible future trajectories conditioned on contextual information derived from vehicular networks.

We observed that WD-based training may lead to temporally inconsistent outputs, a problem mitigated using appropriate regularizations and alternative adversarial losses, such as Relativistic GAN loss. Experimental results demonstrate that GANs can effectively learn to generate realistic and diverse motion trajectories in a simplified traffic scenario, representing a promising tool for simulating future vehicular states, with potential applications in proactive decision-making and digital twin creation for connected and autonomous vehicles.

A. *Questions and Area of Improvement*

In the current implementation, temporal consistency is enforced primarily as a constraint between consecutive time steps. However, the ideal would be to introduce a regularization that forces the Generator to learn the causality over the whole sequence. How can we design regularization strategies that capture long-term temporal dependencies beyond pairwise step-to-step consistency?

We are actively working on extending the approach to support multiple interacting nodes. However, in its current form, without architectural modifications, the number of nodes the GAN can generate simultaneously is heavily constrained by hardware limitations. What architectural modifications to the Generator would enable scalable modeling of interactions among multiple agents while preserving causality across the sequence?

References

[1] Raviglione, Francesco, et al. "S-LDM: Server local dynamic map for vehicular enhanced collective perception." 2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring). IEEE, 2022.
[2] S. A. Angelini, D. Gasco and C. Casetti, "GenSUMO: GenAI-Creation of Critical Scenarios for Autonomous Vehicle Testing," 2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2025, pp. 1-2
[3] Gupta, Agrim, et al. "Social gan: Socially acceptable trajectories with generative adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[4] Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144.
[5] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
[6] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." International conference on machine learning. PMLR, 2017.
[7] Jolicoeur-Martineau, A. "The Relativistic Discriminator: A Key Element Missing from Standard GAN." arXiv preprint arXiv:1807.00734 (2018).
[8] Huang, Nick, et al. "The gan is dead; long live the gan! a modern gan baseline." Advances in Neural Information Processing Systems 37 (2024): 44177-44215.
[9] Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." Advances in neural information processing systems 30 (2017).